Spatial Attention Deep Net with Partial PSO for Hierarchical Hybrid Hand Pose Estimation

Qi Ye, Shanxin Yuan(✉), and Tae-Kyun Kim

Department of Electrical and Electronic Engineering,
Imperial College London, London, UK
{q.ye14,s.yuan14,tk.kim}@imperial.ac.uk

Abstract. Discriminative methods often generate hand poses kinematically implausible, then generative methods are used to correct (or verify) these results in a hybrid method. Estimating 3D hand pose in a hierarchy, where the high-dimensional output space is decomposed into smaller ones, has been shown effective. Existing hierarchical methods mainly focus on the decomposition of the output space while the input space remains almost the same along the hierarchy. In this paper, a hybrid hand pose estimation method is proposed by applying the kinematic hierarchy strategy to the input space (as well as the output space) of the discriminative method by a spatial attention mechanism and to the optimization of the generative method by hierarchical Particle Swarm Optimization (PSO). The spatial attention mechanism integrates cascaded and hierarchical regression into a CNN framework by transforming both the input (and feature space) and the output space, which greatly reduces the viewpoint and articulation variations. Between the levels in the hierarchy, the hierarchical PSO forces the kinematic constraints to the results of the CNNs. The experimental results show that our method significantly outperforms four state-of-the-art methods and three baselines on three public benchmarks.

Keywords: Hierarchical hand pose estimation · Particle Swarm Optimization · Convolutional neural network · Iterative refinement · Spatial attention · Hybrid method · Kinematic constraints

1 Introduction

The problem of 3D hand pose estimation can be formulated as the configuration of the variables representing a hand model given depth images. The problem is challenging with complicated variations caused by high Degree of Freedom (DoF)

Q. Ye and S. Yuan are equally contributed.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46484-8_21) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 346–361, 2016.
DOI: 10.1007/978-3-319-46484-8_21

articulations, multiple viewpoints, self-similar parts, severe self-occlusions, different shapes and sizes. With these variations, configurations of the hand variables given a depth image lie in a high-dimensional space.

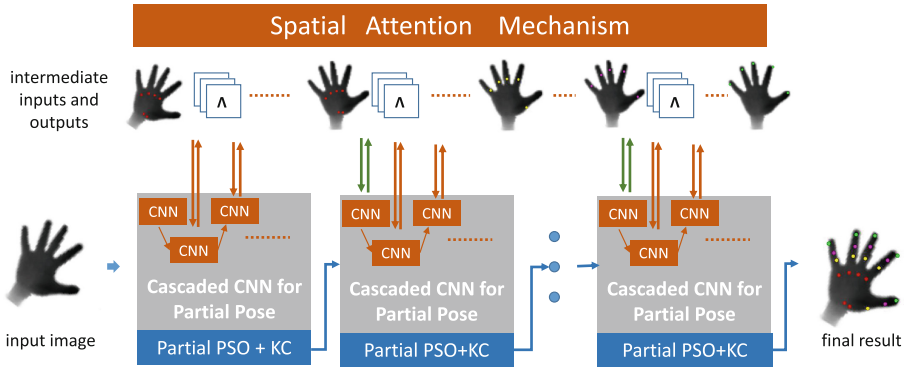

Fig. 1. Structure of the proposed method. The Spatial Attention Mechanism integrates the cascaded and hierarchical hand pose estimation into one framework. The hand pose is estimated layer by layer in the order of the articulation complexity, with the spatial attention module to transform the input/feature and output space. Within each layer, the partial pose is iteratively refined both in viewpoint and location with the spatial attention module, which leads both the feature and output space to a canonical one. After the refinement, the partial PSO is applied to select estimations within the hand kinematic constraints (short as KC in the figure) among the results of the cascaded estimation. Λ denotes the CNN feature maps.

Many prior works have achieved good performance by different methods [1–16]. Among the discriminative methods that learn the mapping from the depth images to the hand pose configurations, Sun *et al.* [17] refine the hand pose by two levels of a hierarchy (palm, and fingers) in a cascaded manner by viewpoint-invariant pixel difference features in random forest. Oberweger *et al.* [18] apply the cascaded method to CNN for iteratively refining partial poses, initialized by the full hand pose estimation.

The discriminative and generative methods are combined in a hierarchy in the Hierarchical Sampling Optimization (HSO) [19]. In each layer, random forests are first used to regress partial poses and a partial joint energy function is introduced to evaluate the results and select the best one to the next layer. The hierarchical optimization with refinement that estimates the hand pose in the order of articulation complexity of the hand is a promising framework as the searching space is decomposed into smaller parts and the refinement leads to more accurate results.

However, the method in [17] and the discriminative part of HSO [19] only focus on breaking down the complexity in the output space hierarchically, i.e., decomposing the hand variables; in other words, the hierarchical strategy is

carried out in the output space while the input space or the feature space stays the same along the hierarchy. For the cascaded refinement [17,18], the input or feature space is only partially updated with results from previous stages, either by cropping or rotating, and the features [17] are computed on the original whole images in each iteration. In addition, the optimization of the energy function is performed in a brute force way in [19].

In this paper, we propose a hybrid method with iterative (cascade) refinement for hand pose estimation, illustrated in Fig. 1, which not only applies the hierarchical strategy to the output space but also the feature space of the discriminative part and the optimization of the energy function of the generative part.

For the discriminative part, a spatial attention mechanism is introduced to integrate cascaded (with multiple stages) and hierarchical (with multiple layers) regression into a CNN framework by transforming both the input (and feature space) and the output space. In the transformed space, the viewpoint and articulation variations of the feature space and the output space is largely reduced, which greatly simplifies the estimation. Along the hierarchy, with the spatial attention mechanism, the features for the initial stage of each layer are spatially transformed from input images based on the estimation results of the last stage of the previous layer. Within each layer, the features are iteratively updated by the spatial attention mechanism. By this dynamic spatial attention mechanism, not only the most relevant features for the hand variable estimation are selected but also the features are transformed to a canonical, expected viewpoint gradually, which simplifies the estimations in the following stages and layers. As such, discriminative features are extracted for each partial pose estimation in each iteration. In this way, we learn a deep net with spatial transformation tailored towards our hand pose estimation problem.

In the generative part, the optimization organized in the hierarchy prevents error accumulation from previous layers. Between the levels of the hierarchy, partial PSO with a new energy function is incorporated to enforce hand kinematic constraints. It generates samples under the Gaussian distribution centered on the results of the discriminative method, and selects estimations within the hand kinematic constraints. The search space of the generative method is largely reduced by estimating partial poses.

To evaluate our method, extensive experiments have been conducted on three public benchmarks. The experimental results show that our method significantly outperforms state-of-the-art methods on these datasets.

2 Related Work

Feature Selection with Attention. Learning or selecting transformation-invariant representations or features by neural networks has been studied in many prior works and among them, attention mechanism has gained much attention in object recognition and localization recently. Girshick *et al.* [20] produce region proposals as representations for CNN to focus its localization capacity on

these regions instead of a whole image. DRAW [21] integrates a spatial attention mechanism mimicking that of human eye into a generative model to generate image samples in different transformations. Sermanet*et al.* [22] use an attention model to direct a high resolution input to the most discriminative regions to do fine-grained categorization. An end-to-end spatial transformation neural network is proposed in [23].

The attention mechanism is tailored to our highly articulated problem by breaking down to the viewpoint and articulation complexity in a hierarchy and refining estimation results in a cascade. The hierarchical structure with cascade refinement enables us to use a spatial transformation to not only select most relevant features as in prior works aforementioned and also transform the feature and the output space into a new one which leads to our expected, canonical space.

Cascaded and Hierarchical Estimation. The cascaded regression strategy has shown good performances in the face analyses [24,25], human body estimation [26,27] and hand pose estimation [17,18] and in most of these works, the features are hand-crafted, such as pixel difference features [17,26], landmark distance features [24], SIFT [28]. Oberweger *et al.* [18] use CNN to learn features automatically but with only partial spatial transformations by cropping input images and in another work [4], they use the images generated by CNN as the feedback to refine the estimation. Sun *et al.* [17] refine the hand pose using pixel difference features updated for viewpoints of whole images in each iteration. The features in both works are partially transformed either by cropping patches from the input images or rotating features calculated from the whole image. On the other hand, a hierarchical strategy that estimates hand poses in the order of hand articulation complexity achieves good performance [17,19]. HSO [19] estimates partial poses separately in the kinematic hierarchy while the input space remains unchanged. Sun *et al.* [17] estimate partial poses holistically in two layers of a hierarchy by calculating rotation invariant pixel difference features from the whole image.

Our proposed method fully transforms the feature space and the output space together in both cascaded and hierarchical manner. For each iteration of the cascade, no new features are learned as features are obtained by a spatial transformation applied to the feature maps of an initial stage. For the hierarchy, only a small region which has been transformed to a canonical view is fed into CNN. In this way, the hierarchical and cascaded strategy is not only applied to the output space as in prior work but also the transformed input and feature space.

Hybrid Methods. A standard way of combining the discriminative methods and the generative methods is first providing candidate results by the discriminative methods, then using them as the initial state of the generative methods to optimize full hand poses [3,16,29,30], and it has demonstrated good performances. As discussed in the above, searching the full hand pose space has a high complexity. We adopt a partial pose optimization to reduce the complexity of

each estimation, which is integrated into our hierarchical structure. HSO [19] also has partial pose evaluations between the levels of a hierarchy but the evaluations are carried out in a brute-force way, while we propose a new kinematic energy function which is optimized by the partial PSO.

3 Method Overview

Hand pose estimation is to estimate the 3-D locations of the hand's 21 key joints S given depth image I, which is normalized by the size of the depth image. The ground truth of S is denoted as S^*. In our approach we divide the 21 joints into four layers $\{S_l\}_{l=0}^{3}$ where the value of l is also the order of our hierarchical estimation, see Fig. 2. For each layer l, j is used to denote a single joint on one finger, in the order from thumb to pinky with the number starting from 1 to 5 (for the wrist joint in the first layer, j is 0). With all the definitions, the hand variables to be estimated are expressed as $\{\{S_{lj}\}_{j=0}^{5}\}_{l=0} \cup \{\{S_{lj}\}_{j=1}^{5}\}_{l=1}^{3}$.

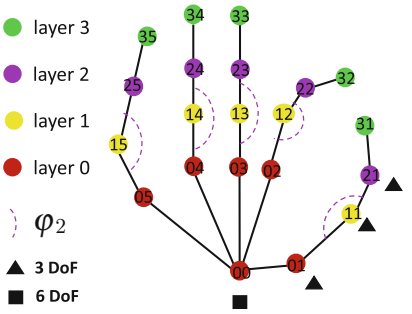

Fig. 2. Hand model. 21 joints are divided into four layers, each joint overlaid with index number. φ_2 is the bone rotations for five joints in Layer 2.

Our method estimates (and trains) 4 layers sequentially with the spatial attention mechanism(see Sect. 4.1) linking the layers by transforming the input (and feature) and output space interactively and partial PSO enforcing kinematic constraints to the CNN prediction, which is shown in Fig. 1. In each layer l, the estimation is refined iteratively by learning the residual of the ground truth S_{lj}^* to the results S_{lj}^{k-1} of the previous stage, where k denotes the k^{th}cascaded stage (for details, see Sect. 4.2). The spatial transformation modules are applied to the feature maps from the initial stage of the cascade and the outputs of stage $k-1$ to get aligned attention features and output space for the learning of residual of stage k.

The result $S_{lj}^{K_l}$ of the final stage K_l is fed into the post-optimization process using PSO for initialization. The partial PSO (see Sect. 5) is introduced to enforce kinematic constraints to the results from the cascaded estimation and refine the partial pose. Along with PSO, we adopt the hand bone model (Fig. 2), which has 51 DoFs: layer 0 has 6 DoFs, denoting the global orientation (represented by a 4-D unit quaternion) and global location (3 DoFs); each of layer 1, 2, 3, has 15 DoFs, denoting the five bone rotations. Our hand model fixes the six joints on the palm and keeps the bone lengths of the fingers.

The optimal of the PSO is passed to layer $l+1$. Before the estimation of the next layer, the spatial attention mechanism is applied on input images and estimation results of current layer (and the ground truth for next layer during training).

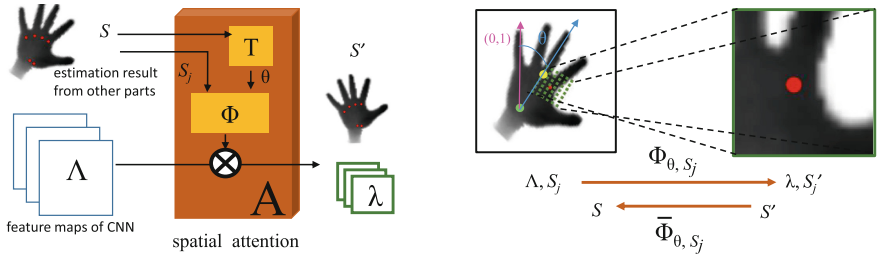

Fig. 3. Spatial attention mechanism. Left: the spatial attention module is split into the calculation of rotation T and the spatial transformation Φ. Right: the mapping between input feature maps and output features maps. For clarity, we use hand images to represent the feature maps. Both the feature maps, estimation results (and ground truth in training) are transformed to a new space by Φ_{θ,S_j}. The locations can be transformed back by the inverse function $\bar{\Phi}_{\theta,S_j}$.

4 Partial Pose Estimation by Spatial Attention Deep Net

4.1 Spatial Attention Mechanism for Hand Space

Before the elaboration of the hand pose estimation, the mechanism of spatial attention is explained. For notational simplicity, we skip the layer index l and the stage index k as the mechanism is applied to all layers and all stages similarly. The inputs of the spatial attention module are the estimation result of S_j , where j denotes j_{th} joint in the layer, and the features maps of CNN (and input images), denoted by $\Lambda \in \mathbb{R}^{W \times H}$.

The spatial module A, illustrated in the left figure of Fig. 3, can be split into two parts: the calculation of rotation T and the pixel mapping Φ. The global in-plane rotation θ (see the right figure of Fig. 3) is the angle between the vector of the wrist joint (joint 00 in Fig. 2) to the root joint of middle finger (joint 03 in Fig. 2) in Layer 0 and the vector $(0, 1)$ representing the upright hand pose and can be expressed as $\theta = T(S_3, S_0)$. For the other layers l $(l > 0)$, the rotation is obtained from Layer 0.

For the pixel mapping, displayed in the right figure of Fig. 3, in which *pixel* here means an element of the feature maps (and input images), we use (x^i, y^i) to denote a pixel on the input feature map Λ and (x^o, y^o) on the output feature maps $\lambda \in \mathbb{R}^{W' \times H'}$. For the deep features for joint j, the translation parameter is the xy coordinates of S_j on the feature map (Λ) coordinate system, i.e. (t_x, t_y). The mapping between (x^i, y^i) and (x^o, y^o) is

$$\begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix} = \begin{bmatrix} b \cdot \cos(\theta) & b \cdot \sin(\theta) & t_x \\ -b \cdot \sin(\theta) & b \cdot \cos(\theta) & t_y \end{bmatrix} \begin{bmatrix} x^o \\ y^o \\ 1 \end{bmatrix} \tag{1}$$

$$(\lambda, S') = \Phi_{\theta,S_j}(\Lambda, S) \tag{2}$$

where (x^i, y^i) and (x^o, y^o) are normalized by its corresponding width and height of the input and output feature maps. b is the ratio of the width of λ to the width of Λ (or the height as we keep the aspect ratio). If b is 1, the transformation is rotation and translation. When b is less than 1, the transformation allows cropping and the cropping size is the same as the size of the output feature maps λ.

Once we get the transformation parameters, the mapping between λ and Λ are established by interpolating the pixel values. We also apply the transformation to the estimation results S (and the ground truth S^* in training) by Eq. 1, only on their xy coordinates, the value of the z coordinate remains unchanged. All the inputs are in a new coordinate system, or a new space. We use Φ_{θ,S_j} in Eq. 2 to wrap the mapping function in Eq. 1 for all the pixels on the feature maps Λ and also symbolize the transformation for S. $\bar{\Phi}_{\theta,S_j}$, denoting the inverse function of Φ_{θ,S_j}, acquired by replacing θ by $-\theta$, (t_x, t_y) by $-(t_x, t_y)$ and b by $1/b$ in Eq. 1, transforms the output space of CNNs to the original one.

4.2 Cascaded Regression Within Each Hierarchical Layer

Within each layer of the hierarchy, the joint locations $\{S_j\}$ are estimated in a cascaded manner, shown in Fig. 4. We leave out the layer subscript l as the cascaded regression is applied to all layers. At first, an initial CNN model $(\{f_j^0\})$ regresses the joint location $\{S_j\}$. It not only provides an initial state $\{S_j^0\}$ for the following iterative refinements but also deep feature maps Λ for other regressors. In the following stages, the joint locations are refined iteratively. Between the refinement stages, the spatial attention modules A transform the deep feature maps Λ to a new space based on the estimation result $\{S_j^{k-1}\}$ from the previous stage to achieve viewpoint-invariant and discriminative features for the following regressors.

For a certain joint j in stage k, the features λ_j^k is mapped from Λ by $\Phi_{\theta^{k-1},S_j^{k-1}}$, where the S_j^{k-1} is the result of the previous stage and θ^{k-1} is calculated by $T(S_0^{k-1}, S_3^{k-1})$ (the updating of θ happens only in Layer 0, and for

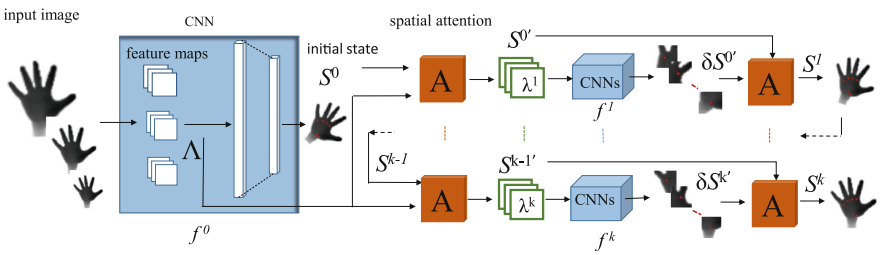

Fig. 4. Cascaded partial pose estimation with spatial attention modules for Layer 0. The feature maps Λ from the initial stage is transformed by spatial attention modules A with estimation result S^{k-1} form previous stage before feeding into the current stage k.

other layers the value of θ is fixed after Layer 0). At the same time, the estimation result S_j^{k-1} and the ground truth S_j^* are both transformed by the module, resulting $S_j^{k-1'}$ and $S_j^{*'}$. Therefore, all the inputs for the regressor f_j^k in stage k that estimates the residual $S_j^{*'} - S_j^{k-1'}$ of joint j are in a new space. After training or testing, the output of the regressor is then transformed back by $\bar{\Phi}_{\theta,S_j}$. For the joint j, the process is repeated until a satisfactory result is achieved (seen Sect. 6 for the choice of cascaded stages)and we use K_l to denote the final stage for Layer l. For other joints, the refinement is carried out in parallel with the same process.

The above refinements for a single joint can be mathematically expressed as

$$(\lambda_j^k, S_j^{k-1'}) = \Phi_{\theta^{k-1},S_j^{k-1}}(\Lambda, S_j^{k-1}) \tag{3}$$

$$S_j^k = \bar{\Phi}_{\theta^{k-1},S_j^{k-1}}(f_j^k(\lambda_j^k) + S_j^{k-1'}) \tag{4}$$

where Eq. 3 is the spatial attention mechanism which transforms all the inputs of stage k to a new space and Eq. 4 estimates the residual $\delta S_j^{k'}$ by $f_j^k(\lambda_j^k)$, updates the estimation by adding the residual estimated $\delta S_j^{k'}$ to the result from the previous stage $S_j^{k-1'}$, and transforms the added result back to the original space.

4.3 Hierarchical Regression

For the regression in layer 0, all the joints in the initial stage are learned together in order to keep the kinematic constraints among them as the values of these joints are highly correlated. The input of the initializor f_0^0 is multi-resolution images I, the original image and the images downsampled from the original one by the factor of 2 and 4, the output is the joint locations. The input and feature space of the regressors for different joints in the cascaded stages are updated separately by the spatial function $\Phi_{\theta^{k-1},S_{0j}^{k-1}}$. The output of the regressor in the stage k refines the estimation result S_{0j}^{k-1} in the previous stage $k-1$ in a new space and are transformed back by $\bar{\Phi}_{\theta^{k-1},S_{0j}^{k-1}}$. The cascaded regression stop in stage K_0. The whole refinement stages are the same as in Sect. 4.2.

For the hierarchical estimation in layer $l(l > 0)$, the inputs are multi-resolution input images I, the estimation result $\{S_{l-1,j}^{K_{l-1}}\}$ from the previous layer $l-1$ and the viewpoint estimation θ^{K_0} from layer 0. For notational simplicity, we denote θ^{K_0} as θ and skip the joint index j. θ is fixed for all layers $(l > 0)$ and the same process is applied to all the joints separately.

The input space for the initializor f_l^0 of layer l is transformed from multi-resolution images I by the spatial attention module. The mapping is

$$(I', S_{l-1}^{K'_{l-1}}, S_l^{*'}) = \Phi_{\theta,S_{l-1}^{K_{l-1}}}(I, S_{l-1}^{K_{l-1}}, S_l^*) \tag{5}$$

so the input for the initializor f_l^0 is patches I' cropped from multi-resolution input images I centred at $S_{l-1}^{K_{l-1}}$ and its corresponding coordinates in the downsampled images, and rotated by θ. The offset labels for training f_l^0 is $\Delta S_l^{*'} = S_l^{*'} - S_{l-1}^{K'_{l-1}}$, which is equivalent to the sum of the ground truth offset $S_l^{*'} - S_{l-1}^{*'}$ and the remaining residual of the previous layer $S_{l-1}^{*'} - S_{l-1}^{K'_{l-1}}$. This implies the initializor f_l^0 not only predicts the joint offsets of the current layer to the previous layer but also corrects the residual errors of the previous layer.

The initializor f_l^0 provides the initial offset state $\Delta S_l^{0'}$ and feature maps Λ for the refinement stages. For the refinement stages, the procedure is the same as discussed in Sect. 4.2. The only difference from Sect. 4.2 is that the viewpoint is static, whose value is the result of the final cascaded stage in Layer 0, and feature maps Λ has already been transformed by rotation in the initial stage; thus for the stage k, the feature space is transformed and updated by the function $\Phi_{S_l^{k-1}}$ (no rotation transformation) and the output space is transformed with $\Phi_{\theta,S_l^{k-1}}$ (rotation and translation transformation).

The parameter b in the spatial attention module needs to be set. For the initial stage (except the initial stage of layer 0), it is set according to the offset range. All the ground truth S_l^* and the estimation result $S_{l-1}^{K_{l-1}}$ of layer $l-1$ is first transformed by $\Phi_{\theta,S_{l-1}^{K_{l-1}}}$ with $b = 1$ to get means along the xy coordinates of the absolute value of the offsets for all the training data in the new space. b is set to be two times of the larger offset mean divided by original image width W. For the refinement stages, they are set according to the residual range of the estimation results in the initial stage. All the ground truth and the estimation results of the initial stage are first transformed by Φ_{θ^0,S_j^0} (for layer $l(l > 0)$, θ^0 is the final estimation result θ^{K_0} of layer 0) with $b = 1$ to get means along the xy coordinates of the absolute value of the residuals for all the training data in the new space. As the feature maps are filtered by kernels, max-pooled, and have different resolutions but the ground truth are normalized according to original image size, S_j^0 and the mean of residual should also be changed with the kernel size, the pool size and the downsampling ratio to set the value of b.

5 Partial PSO with Kinematic Constraints for Final Refinement

For each layer, based on our discriminative part's prediction, we do final refinement by explicitly introducing partial kinematic constraints with Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a stochastic optimization algorithm introduce by Kennedy and Eberhart [31] in 1995, originated in the social behaviors' studies of synchronous bird flocking and fish schooling. The original PSO algorithm has been modified by several researchers to improve its convergence properties and search capabilities. We adopt the variant of PSO with an inertia weight parameter [32].

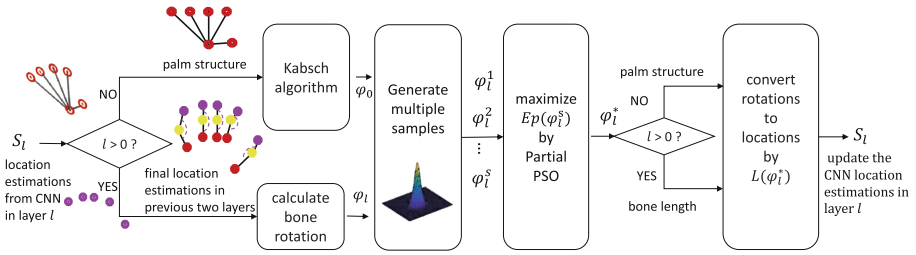

Fig. 5. Pose refinement with partial PSO enforcing kinematic constraint. Given palm spatial structure and layer 0's location estimation by CNN, we first inference the φ_0 using Kabsch algorithm [33], and then find φ_0^* maximizing the energy function by partial PSO. The rotation φ_0^* is converted to locations using the palm structure to update the CNN estimation result. For other partial pose $\varphi_l(l > 0)$, the optimization is the same with layer 0 while the inference for the initialization of the optimization is calculating the bone rotation and the conversion back to locations uses the bone length.

Our whole hand pose for PSO is defined as $\{\varphi_0, \varphi_1, \varphi_2, \varphi_3\}$, where $\varphi_0 \in \mathbb{R}^7$ and $\varphi_l(l = 1, 2, 3) \in \mathbb{R}^{15}$ are our partial poses. $\varphi_0 = \{q, x, y, z\}$, where q is a 4-D unit quaternion [1,3] representing the global rotation, $[x, y, z]$ is the global location of the whole hand. φ_l denotes five 3D Euler angles in layer l, each angle representing a bone rotation which is the angle between the bone connecting the joint in layer l and the corresponding joint in layer $l-1$, and the other bone connecting the joint in layer $l-1$ and the corresponding joint in layer $l-2$ (when $l-2<0$, the corresponding joint in layer $l-2$ is wrist). Figure 2 demonstrates φ_2, five angles in layer $l = 2$.

Energy Function. For each layer, PSO is used to estimate the final partial pose base on the inferred partial pose. We designed a new energy function that applied to partial pose and explicitly taking into account the kinematic constraints. Our energy function Ep for each layer is as follows:

$$Ep(\varphi_l^s) = P(\varphi_l^s)Q(\varphi_l^s), \tag{6}$$

where the first item, $P(\varphi_l^s) \propto N(\varphi_l^s; \varphi_l, \Sigma)$, is the prior probability of the s_{th} Gaussian sample from mean φ_l, Σ is a diagonal covariance matrix that is manually set to ensure that each parameter varies in valid ranges. $s = 1, 2, ..., N$ is the index of samples for each layer, we set $N = 100$ in our experiments. $P(\varphi_0^s)$ encodes the spatial structure of the six joints on the palm and $P(\varphi_l^s)(l = 1, 2, 3)$ keeps the bone length information.

To acquire the prior probability $P(\varphi_0^s)$, we first choose Kabsch algorithm [33] to find the optimal affine transformation matrix (global translation and rotation, i.e. φ_0) from our hand model for the six joints on the palm to the CNN results, as shown in the top pipeline of Fig. 5. The hand model for the palm joints can be seen as the palm joint locations of an upfront reference hand pose with

wrist located on the original coordinate. By generating samples from Gaussian distribution centred on φ_0 instead of S_0 from CNN which usually violates the kinematic constraint, we get the $P(\varphi_0^s)$ that keeps the spatial structure of palm joints.

For $P(\varphi_l^s)$ of other layers $l > 0$, we first get five bone rotations φ_l by calculating the angles which are demonstrated in the bottom pipeline of Fig. 5 with joint estimation locations of CNN in current layer l and the joint estimation locations from layer $l-1$ and $l-2$, and then sample from the Gaussian distribution centred on φ_l. When converting rotations to locations for evaluations of the second term, we enforce the constraint of the bone length.

The second item, $Q(\varphi_l^s) \propto \sum_{S_{lj}^s \in L(\varphi_l)} [B(S_{lj}^s) + D(S_{lj}^s)]$, denotes the likelihood of all joint $\{S_{lj}^s\}$ belongs to the hand, where $L(\varphi_l^s)$ converts rotations φ_l^s into locations $\{S_{lj}^s\}$. Similar to Tang *et al.* [19] silver function, the term $B(S_{lj}^s)$ forces each joint joint S_{lj}^s to lie inside the hand silhouette. The term $D(S_{lj}^s)$ makes sure joint S_{lj}^s lies inside the depth range of a major point cloud.

6 Experiment

The evaluation of our proposed method is conducted on three publicly datasets. **ICVL** [19] dataset is a real sequence captured by Intel RealSense with the range of view about 120 degrees consisting 1596 test frames and 16008 training frames. 16 bone centre locations are provided for each hand pose. **NYU** [30] dataset is a real sequence acquired by PrimeSense containing 8252 test-set and 72757 training-set frames with a full range of views. 36 joint locations are provided for each hand pose. **MSRC** [3] dataset is a challenging dataset that covers a full range of views and complex articulations with 100000 synthetic images in the training-set and 2000 synthetic images in the test set. 22 joint locations are provided for each hand pose. As the annotations of these datasets do not conform to each other, we use the annotation version in [19] that labels locations of the joints as demonstrated in Fig. 2.

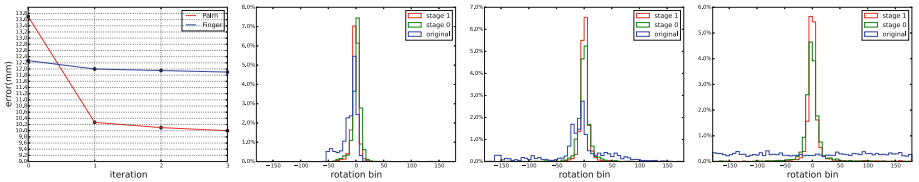

Fig. 6. First: Errors for a joint on the palm S_{00} and a joint on the middle finger S_{13} for 4 stages. Second, third and forth: In-plane viewpoint distribution of testing set for different stages on ICVL, NYU, MSRC respectively. The blue, green and red line corresponds to the in-plane rotation distribution of original ground truth, ground truth rotated after initial stage and the first stage. The rotation estimation error after the initial stage and the first stage is 5.9 and 4.4 for ICVL, 8.0 and 6.1 for NYU, 10.9 and 9.2 for MSRC in the unit of degree. (Color figure online)

We compare the results of different methods by the proportion of joints within a certain maximum error of the distance of the predicted results to the ground truth [3]. We set the number of iterations K on the observation of the error saturates after a certain stage in the cascaded stages, shown in the first figure of Fig. 6. We set K_0 for Layer 0 to 1 and $K_l, l > 0$ to 0, which gives us a good balance between the accuracy and the memory consumption. All the experiments are run with Intel i7, 24 GB RAM and NVIDIA GeForce GTX 750 Ti. The structures for our CNN models are implemented by Theano [34] and the details are provided in the supplementary material. For our partial PSO, we generates 100 samples for each layer, and iterate 5 generations.

6.1 Self-comparison

To evaluate our proposed method (Hybr_Hier_SA) and the discriminative part Hier_SA, we implement three baselines. The first baseline (Holi) estimates the whole hand pose with a single CNN. The second baseline (Holi_Derot) consists of two steps: one step predicting the in-plane rotation of the hand pose by a CNN and rotating the hand pose to upright view; the other step estimating the whole hand pose by another CNN. The third one (Holi_SA) is a holistic cascaded regression network without hierarchy, which initializes the whole hand pose with a CNN and refines the hand pose joint by joint via spatial attention mechanism by a set of CNNs. For fair comparison, we set the size of the parameters of the methods to be roughly the same: the parameters are stored in 32 bit float and the size of parameters is 130 MB.

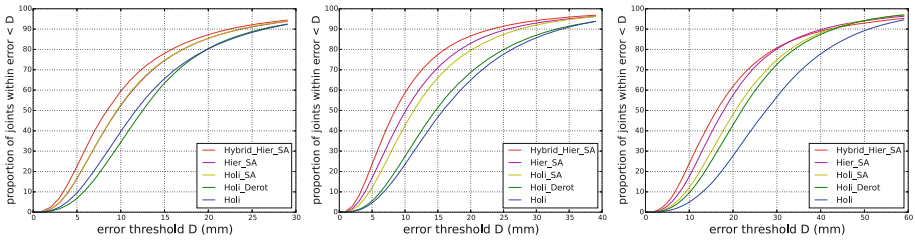

Fig. 7. Comparison of different methods on three datasets. Left:ICVL; Middle: NYU; Right: MSRC.

On all the datasets, Hier_SA outperforms the baselines significantly, see Fig. 7. The improvement margin is related to the range of viewpoints and the complexity of articulations. The in-plane rotation distributions of original data, the ones after the initial stage and the first stage are shown in Fig. 6. As MSRC dataset covers a full range of viewpoints and articulations, the improvement on this dataset from the baseline Holi is the largest. For example, the percentages of frames under 20 mm on ICVL, NYU and MSRC are improved by Hier_SA with margins of 5 %, 18 % and 30 % respectively, compared to that of Holi.

The curves of Hier_SA and Holi_SA on three datasets illustrate the efficacy of hierarchical strategy in conquering the articulations, while the curves of Holi_SA, Holi and Holi_Derot show that spatial attention mechanism is effect in reducing the viewpoint and articulation complexity. By refining viewpoints with stages and spatially transforming the feature space to focus on the most relevant area for a certain joint estimation, Holi_SA achieves better results than Holi and Holi_Derot. Note that the curve of Holi_Derot is under that of Holi on ICVL dataset, which implies that the error of estimating the rotation by a separate network may deteriorate the later estimation when the variations of the viewpoint in training set is small.

Hybr_Hier_SA further improves the result of Hier_SA by a large margin consistently on all the datasets, which verifies that the kinematic constraints by the partial PSO is effective.

6.2 Comparison with Prior Works

We compare our work with 4 state-of-the-arts methods: Hierarchical Sampling Optimization (HSO) [19], Sharp *et al.* [3], HandsDeep [18], FeedLoop [4] on three datasets, see Fig. 8. The former two are hybrid methods and the latter two are refinement method based on CNN. The results are obtained either from the authors for HSO [19] or from the reported accuracies [3,4,18]. The examples of the estimation results of HandsDeep, FeedLoop, HSO and our method are shown in Fig. 9.

On ICVL dataset, we compare HSO with parameters set as $N = 100, M = 150$. Our method is better by 26 % of joints within $D = 10$ mm. On NYU dataset, we compare our method with HandsDeep and FeedLoop which are all based on CNN. As the hand model of these methods are different, we evaluate the result by comparing the error of the subset of 11 joint locations (removing the palm joints except the root joint of thumb). Our estimation result is better than HandsDeep by a large margin, for example, an improvement of 10 % within $D = 30$ mm, and achieves roughly the same accuracy with FeedLoop.

We finally test our method with HSO and Sharp *et al.* on MSRC dataset. The dataset is more challenging than the above two as it covers a wider range of viewpoints and articulations. The curves demonstrate the superiority of our

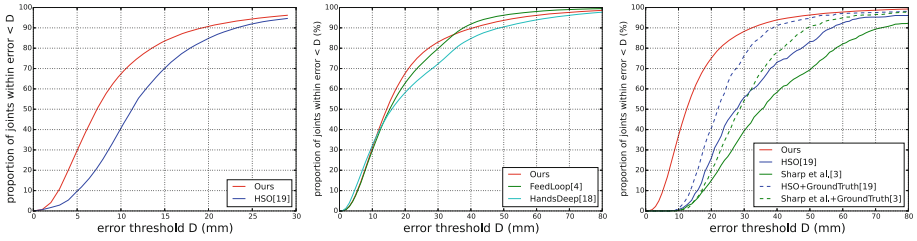

Fig. 8. Comparison of prior work on three datasets. Left:ICVL; Middle: NYU; Right: MSRC.

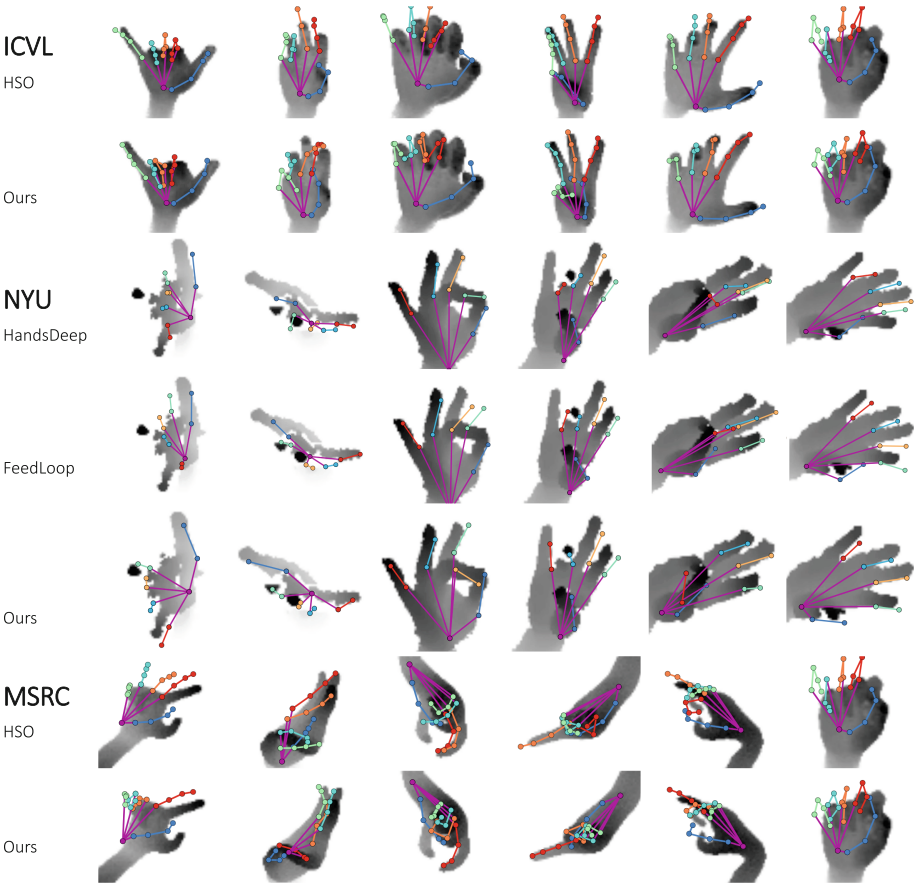

Fig. 9. Examples comparing to prior work on three datasets. The first two rows, the middle three rows, and the last two rows are examples from ICVL dataset, NYU dataset and MSRC dataset, respectively. Compared to other methods, our method has a good performance in discriminating the fingers and a better precision. For many challenging viewpoints, our method still has a good estimation.

method under large variations. For example, the proportion of joints (when $D = 30\,\text{mm}$) of our method is 35 % and 50 % more than those of HSO and Sharp *et al.* respectively. Note that our estimation is even better than the results of HSO and Sharp *et al.* using ground truth rotation [19].

7 Conclusion

To apply the hierarchy strategy to the input and feature space and enforce the hand kinematic constraints to the hand pose estimation, we present a hybrid method by applying the kinematic hierarchy to both the input and feature space

of the discriminative method and the optimization of the generative method. For the integration of hierarchical input and feature space of the discriminative, a spatial attention mechanism is introduced to spatially transform the input (and feature) and output space interactively, leading to new spaces with lesser viewpoint and articulation complexity and gradually refining th estimation results. In addition, the partial PSO is incorporated between the layers of the hierarchy to enforce the kinematic constraints to the estimation results of the discriminative part. This helps reduce the error from previous layer to accumulate. Our method demonstrates good performance on three datasets, especially on the dataset under large variations.

References

1. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3D tracking of hand articulations using kinect. In: BMVC (2011)
2. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: CVPR (2014)
3. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Leichter, D., Wei, A.V.Y., Krupka, D., Fitzgibbon, A., Izadi, S.: Accurate, robust, and flexible real-time hand tracking. In: CHI (2015)
4. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: ICCV (2015)
5. Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Hand segmentation with structured convolutional learning. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9005, pp. 687–702. Springer, Heidelberg (2015)
6. Tang, D., Yu, T.H., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV (2013)
7. Keskin, C., Kıraç, F., Kara, Y.E., Akarun, L.: Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 852–863. Springer, Heidelberg (2012)
8. Ionescu, C., Carreira, J., Sminchisescu, C.: Iterated second-order label sensitive pooling for 3D human pose estimation. In: CVPR (2014)
9. Liang, H., Yuan, J., Thalmann, D.: Parsing the hand in depth images. TMM **16**(5), 1241–1253 (2014)
10. 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 Workshop (2014)
11. Stenger, B., Thayananthan, A., Torr, P.H., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. TPAMI **28**(9), 1372–1384 (2006)
12. Ballan, L., Taneja, A., Gall, J., Van Gool, L., Pollefeys, M.: Motion capture of hands in action using discriminative salient points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 640–653. Springer, Heidelberg (2012)
13. Intel: Perceptual computing SDK (2013)
14. Supancic III., J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: methods, data, and challenges. arXiv preprint arXiv:1504.06378 (2015)

15. Taylor, J., Stebbing, R., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., Fitzgibbon, A.: User-specific hand modeling from monocular depth sequences. In: CVPR (2014)
16. Krejov, P., Gilbert, A., Bowden, R.: Combining discriminative and model based approaches for hand pose estimation. In: FG (2015)
17. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR (2015)
18. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. arXiv preprint arXiv:1502.06807 (2015)
19. 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)
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
21. Gregor, K., Danihelka, I., Graves, A., Wierstra, D.: Draw: a recurrent neural network for image generation. arXiv preprint arXiv:1502.04623 (2015)
22. Sermanet, P., Frome, A., Real, E.: Attention for fine-grained categorization. arXiv preprint arXiv:1412.7054 (2014)
23. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NIPS (2015)
24. Zhao, X., Kim, T.K., Luo, W.: Unified face analysis by iterative multi-output random forests. In: CVPR (2014)
25. Zhu, S., Li, C., Change Loy, C., Tang, X.: Face alignment by coarse-to-fine shape searching. In: CVPR (2015)
26. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: CVPR (2010)
27. Toshev, A., Szegedy, C.: Deeppose: human pose estimation via deep neural networks. In: CVPR (2014)
28. Xiong, X., Torre, F.: Supervised descent method and its applications to face alignment. In: CVPR (2013)
29. Sridhar, S., Mueller, F., Oulasvirta, A., Theobalt, C.: Fast and robust hand tracking using detection-guided optimization. In: CVPR (2014)
30. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. TOG **33**(5), 169 (2014)
31. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: International Conference on Neural Networks (1995)
32. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation (1998)
33. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography (1976)
34. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv.1605.02688, May 2016