Cascaded Hand Pose Regression

Xiao Sun
Chinese University of Hong Kong
sx014@ie.cuhk.edu.hk

Yichen Wei
Microsoft Research
yichenw@microsoft.com

Shuang Liang*
Tongji University
shuangliang@tongji.edu.cn

Xiaoou Tang
Chinese University of Hong Kong
xtang@ie.cuhk.edu.hk

Jian Sun
Microsoft Research
jiansun@microsoft.com

Abstract

We extends the previous 2D cascaded object pose regression work [9] in two aspects so that it works better for 3D articulated objects. Our first contribution is 3D pose-indexed features that generalize the previous 2D parameterized features and achieve better invariance to 3D transformations. Our second contribution is a principled hierarchical regression that is adapted to the articulated object structure. It is therefore more accurate and faster. Comprehensive experiments verify the state-of-the-art accuracy and efficiency of the proposed approach on the challenging 3D hand pose estimation problem, on a public dataset and our new dataset.

1. Introduction

The problem of pose estimation of 3D articulated objects such as human body and hand has been studied for decades. Recent years have seen rapid progress and significant success of human body pose estimation [18, 29, 1, 19, 36] using consumer depth sensors. The state-of-the-art learning approaches [18, 29] classify depth pixels into body parts and then infer the body pose from the pixel classification result. This paradigm has been applied for hand pose estimation [6, 39, 10, 22] but is less successful than for body pose. This is because body is mostly near-frontal and there is less occlusion between limbs. However, hand motion exhibits much larger variations in both camera viewpoints and finger articulations. This generates more complex depth images and makes the pixel classification much more difficult. Furthermore, the pixel classification approaches do not capture the structural constraints in the hand pose.

Regression based approaches directly estimate the hand pose from the depth image, using latent regression forest [34] or deep convolutional neutral networks [35]. Such methods are more principled since their learning is directly guided by the task. Nevertheless, only one regression model is learnt in such works, which may have insufficient capacity to model the complex image variations, especially under large viewpoints and hand motions.

We present a cascaded regression approach that is more robust under large viewpoints and complex hand poses. It is directly motivated by the cascaded pose regression framework [9], where the object pose is estimated progressively via a sequence of weak regressors and each weak regressor uses features that depend on the estimated pose from the previous stage. Such *pose indexed features* provide better geometric invariance and simplify the learning tasks. This framework has been successfully applied to several 2D pose estimation tasks [9, 5]. Yet, it is unclear how to use it for 3D objects with complex articulated structure like human hand.

We extend the framework for 3D articulated objects. Our first contribution is *3D pose-indexed features*. While we use the similar pixel difference features as in cascaded pose regression [9], face [5], human body [18, 29] and hand [6, 39, 10, 20, 22], we show that the pixel parameterization is the key to achieve certain geometrical invariance. We analyze the invariance properties of previous parameterization methods, explain our rationale and propose a new 3D parameterization that generalizes the previous methods and achieves better invariance to 3D transformations.

Our second contribution is a principled *hierarchical* approach that is adapted for the structure of articulated objects. Our key observation is that different object parts typically exhibit different amount of variations and degrees of freedom due to the articulated structure. Thus, regressing all parts together is unnecessarily difficult and causes slow convergence and degraded accuracy. Our hierarchical approach *regresses the pose of different parts sequentially in the order of their articulation complexity*. It firstly estimates the pose of the easier root part (such as palm). Estimation

*Corresponding author.

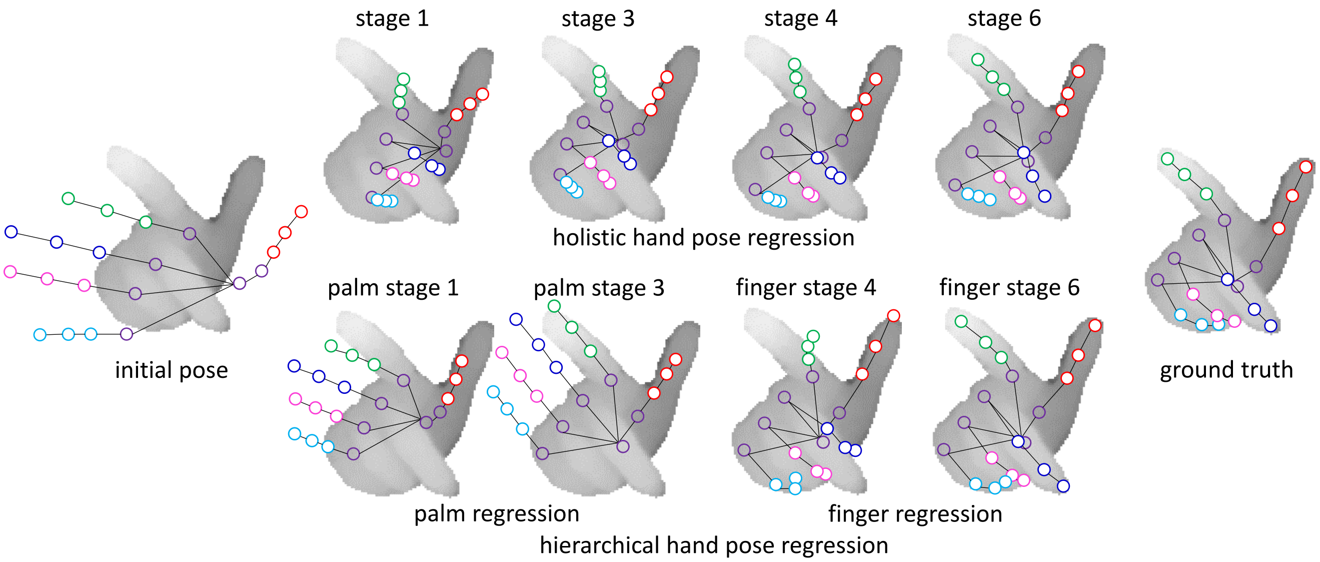

Figure 1. Illustration of cascaded hand pose regression on a real example. Starting from the depth image and a rough initial hand pose (see Section 3.5 about initialization), the hand pose is iteratively updated through six stages and approaching the ground truth. The top row is the holistic hand regression (Section 3.3). The bottom row is the hierarchical regression (Section 3.4), *i.e.*, palm is updated in the first three stages with fingers fixed (relatively to the palm) and fingers are updated in the last three stages with palm fixed.

of more difficult sub-parts (such as fingers) are then conditioned on the pose of the root part and thus easier. The hierarchical approach does not only converge faster but is also more accurate.

The proposed approach works on general 3D articulated objects. It is applied for hand pose estimation in this work, as exemplified in Fig. 1. Comprehensive experiments show that it significantly outperforms the state-of-the-art, on both public data and a large challenging dataset collected by us. In addition to the high accuracy, our regression is also very fast ($>$ 300 FPS on CPU, single thread) and would be influential for real applications.

2. Related Work

The literature of (articulated) object pose estimation is comprehensive. We briefly review the previous work from a few viewpoints that are of interests and related to our work.

Viewpoint estimation Many previous works model the camera viewpoint distribution as a hidden variable and learn its distribution first. The object pose distribution is then conditioned on it and learnt afterwards. This reduces the variations caused by viewpoint and simplifies the pose learning problem. This framework has been applied to facial landmark localization [8], human body pose estimation [33] and hand pose estimation [6, 10]. However, for hand this approach is less successful, because the viewpoint variations for hand are far more complex than that in body and face, which usually have only small viewpoint variations in yaw. Previous techniques (pre-clustering of hand pose in [6] and using an augmented cost function with a viewpoint classification term in [10]) are simple and can only perform coarse viewpoint estimation. In this work, we parameterize the viewpoint into the hand pose and estimate it iteratively in a boosted regression framework. This is more robust and accurate.

Cascaded pose regression and pose-indexed features The framework [9] progressively updates the object pose via a sequence of weak regressors. It extends boosted regression [12] by exploiting the *pose-indexed* features, *i.e.*, regressor learning in the current stage uses features that are defined on the estimated pose from the previous stage. This achieves better geometry invariance and makes each stage's learning easier. It has been successfully applied for 2D object pose estimation problem [9], especially for facial landmark localization [5, 27]. In this work, for the first time we extend the framework for 3D object pose estimation and show how to define pose indexed features in 3D.

Per-joint estimation vs. holistic regression Many methods [6, 39, 10, 22] estimate hand joints individually by following the per-pixel classification approaches for human body pose recognition [18, 29]. Such methods firstly classify all pixels into object parts and then convert them into semantic joints. This strategy has two drawbacks. First, evaluating many pixels is slow. Second and more importantly, estimating the joints individually may violate the inherent structural constraints. By contrast, our holistic approach evaluates the whole image just once and regresses all the joints simultaneously. This is not only faster but also better preserves the structural constraints, such as shown in [5]. We note that holistic regression is also used in [34, 35] for hand pose estimation, but not in a cascaded manner.

Hierarchical pose estimation The progressive human pose estimation approach [11] shares a similar idea with our hierarchical regression at a high abstract level, but differs a lot in details. We do not perform part detection but directly estimates the pose of object parts in the order of their articulation complexity. This turns out more effective than regressing all parts together.

Model based pose tracking There are a lot of model based approaches for hand tracking [30, 14, 15, 16, 32, 23, 37, 26, 25, 31] and human body tracking [1, 19, 36]. They are mostly based on slow but accurate local optimization and require good initializations to work well. Such methods are mostly complementary to learning based methods, which can provide fast and rough pose estimate as initialization.

Early work on hand Most early works [17, 38, 4, 2, 13, 24] use RGB images or videos. Due to the lack of 3D information, they usually work under near-frontal viewpoints and become less stable under large viewpoints.

3. Cascaded Hand Pose Regression

As illustrated in Fig. 2(a), the hand pose Θ is parameterized as 21 kinematic joints. We also represent Θ as six parts, the palm Θ^p (6 joints) and five fingers $\{\Theta^f\}$ (each 3 joints), where $f \in F = \{1, 2, 3, 4, 5\}$. The palm encodes 6 degrees of freedom of the global viewpoint. Each finger and its corresponding root point on the palm (4 joints in total) encode 4 degrees of freedom of finger articulation.

To estimate the hand pose, we start from a depth image I and an initial pose Θ^0. In each stage $t(= 1, ..., T)$, the current pose estimation is progressively updated by the stage regressor $\mathcal{R}^t$ as

$$\Theta^t = \Theta^{t-1} + \mathcal{R}^t(I, \Theta^{t-1}). \quad (1)$$

The final pose estimation is Θ^T.

During training, the stage regressor $\mathcal{R}^t$ is learnt to approximate the current pose residual $\delta\Theta_i$, which is the difference between the ground truth pose and the estimated pose from the previous stage Θ_i^{t-1}, over all training samples i. Note that the features for learning $\mathcal{R}^t$ depend on the previous pose estimation Θ_i^{t-1}. Such *pose-indexed features* provide better geometric invariance and have been shown effective in several 2D pose regression tasks [9].

This cascaded pose regression framework is general and introduced in [9]. Now we show how to extend it for 3D hand pose regression. We first describe the basic principles, *i.e.*, *3D pose normalization* and *3D pose-indexed features* in Section 3.1 and Section 3.2. Accordingly, we present a holistic regression algorithm in Section 3.3. We then motivate and develop a new *hierarchical* regression algorithm in Section 3.4. It exploits the characteristics of articulated object and improves the holistic algorithm. Implementation details are provided in Section 3.5.

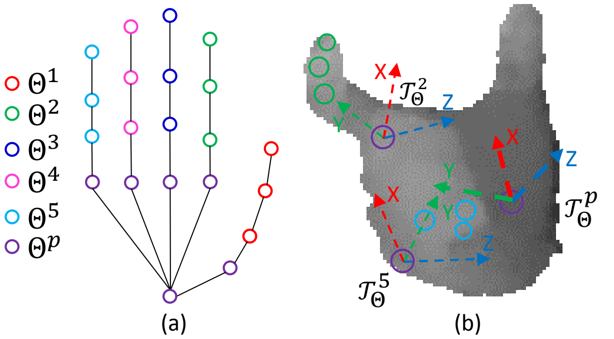

Figure 2. (a) 21-joint representation of a canonical hand pose. (b) Illustration of two finger coordinate frames and the palm coordinate frame on a real 3D hand pose, as well as the corresponding 3D transformations. See Section 3.1 for details.

3.1. 3D Pose Normalization

The hand pose update in Eq. (1) is defined in the 3D camera coordinate frame in the depth image. This is undesirable because it can be affected by non-essential transformations between such coordinate frames, *e.g.*, an in-plane image rotation generates different depth images for the same hand.

To achieve certain invariance, we should use a single *canonical coordinate frame* that is irrelevant to the individual camera coordinate frames. This is realized by applying a 3D transformation $\mathcal{T}_\Theta$ to normalize the pose Θ to the canonical coordinate frame. This normalization is done for all training samples so that they are roughly aligned in the canonical coordinate frame before training. During testing, the inverse transformation $\overline{\mathcal{T}_\Theta}$ is used to transform the poses back to the camera coordinate frame, where the actual pose update happens. In essence, all previous body/hand pose regression work implicitly uses this 3D pose normalization, which nevertheless degenerates to a 3D translation, *e.g.*, by aligning the depth image patch/3D point cloud in advance using a translation, or regressing the relative 3D joint offsets (in camera coordinate frame) to a depth pixel [29].

In this work, for the first time we use a full 3D rigid transformation so it has better invariance with respect to 3D rotation in addition to translation. We define two types of canonical coordinate frames. As illustrated in Fig. 2(b), the first is aligned with palm root and the second is aligned with finger root. The corresponding 3D normalization transformations are denoted as $\mathcal{T}_\Theta^p$ and $\mathcal{T}_\Theta^f$. Specifically, the palm coordinate frame has its origin at wrist, positive Y axis pointing to the middle finger root and positive Z axis pointing outwards of the palm plane. As it is defined only using the palm part, the transformation $\mathcal{T}_\Theta^p$ can also be written as $\mathcal{T}_{\Theta^p}^p$. The finger coordinate frame has its origin at finger root, positive Y axis pointing along the first phalange and X axis orthogonal to the finger plane [1]. Note that the finger coordinate frame has different 3D rotation from the palm coordinate frame, which reflects the degrees of freedom at the finger root. Therefore, for finger pose estimation $\mathcal{T}_\Theta^f$ is better than $\mathcal{T}_\Theta^p$ because fingers from different poses are better aligned in their local coordinate frame than the global palm coordinate frame.

3.2. 3D Pose Indexed Features

Similar to previous depth based learning methods for human body [18, 29] and hand [6, 39, 10, 34, 22], we use the pixel difference features, *i.e.*, a feature is the difference of two random pixels' depth value, $I(u_1) - I(u_2)$. The key to achieve certain geometric invariance is how to parameterize $u_i (i = 1, 2)$.

Following [18], almost all previous works parameterize u_i using random $2D$ offsets δu_i as

$$u_i = u + \frac{\delta u_i}{z(u)}, i = 1, 2, \qquad (2)$$

where u is the reference pixel, *i.e.*, the center of the depth patch [34] or any pixel under consideration [6, 10, 22], and $z(u)$ is its depth. It is shown in [18] that such parameterization is invariant to depth change of u.

The work in [39] extends Eq. (2) to achieve invariance up to a 2D in-plane-rotation as

$$u_i = u + \frac{Rot(\delta u_i; u, \alpha)}{z(u)}, i = 1, 2, \qquad (3)$$

where $Rot(\cdot)$ is a 2D transformation that rotates δu_i around pixel u by angle α. In [39], the in-plane-rotation angle α of the hand pose is estimated in an initial regression step and then fixed.

From 2D to 3D Both Eq. (2) and (3) index a pixel u_i using a 2D offset δu_i. We note that this 2D offset can also be viewed as a 3D offset Δu_i, lying on a 3D plane that is parallel to the image plane with depth $z(u)$. Its major role is to define a back-projection ray from the camera center to find a 3D surface point, which is then projected to the pixel to compute the depth feature.

From this viewpoint, we notice that the 3D offset Δu_i actually does not have to be on that 3D plane but could be arbitrarily defined. This provides us additional flexibility to achieve certain invariance. In our case, given a current pose Θ and a corresponding 3D transformation $\mathcal{T}_\Theta$ that normalizes it to some canonical coordinate frame, we parameterize the 3D offset in the canonical coordinate frame to make it invariant to $\mathcal{T}_\Theta$. Our approach works as follows:

Algorithm 1 Training algorithm for holistic cascaded hand pose regression.

```
1: input: image $I_i$, ground truth pose $\Theta_i$, and initial pose
   $\Theta_i^0$ for all training samples $i$
2: for $t = 1$ to $T$ do
3:     $\delta\Theta_i = \mathcal{T}^p_{\Theta_i^{p,t-1}}(\Theta_i) - \mathcal{T}^p_{\Theta_i^{p,t-1}}(\Theta_i^{t-1})$        ▷ residual
4:     learn $\mathcal{R}^t$ to approximate $\delta\Theta_i$
5:     $\Theta_i^t = \Theta_i^{t-1} + \overline{\mathcal{T}^p_{\Theta_i^{p,t-1}}}(\mathcal{R}^t(I_i, \Theta_i^{t-1}))$        ▷ update
6: end for
7: output: regressors $\{\mathcal{R}^t\}_{t=1}^T$
```

1. Generate random $3D$ offsets Δu_i. They represent 3D locations in the canonical coordinate frame.

2. Transform them using $\overline{\mathcal{T}_\Theta}$. Now they represent 3D locations in the camera coordinate frame and are ready for feature generation.

3. Project them to image and compute depth feature.

In summary, our proposed parametization is written as

$$u_i = CamProj(\overline{\mathcal{T}_\Theta}(\Delta u_i)), i = 1, 2. \qquad (4)$$

Discussions Our parameterization in Eq. (4) is a generalization of these in Eq. (2) and (3). If $\mathcal{T}_\Theta$ degenerates to a 3D translation, Eq. (4) degenerates to Eq. (2). If the rotation part of $\mathcal{T}_\Theta$ is constrained to contain only 2D in-image-plane rotation, Eq. (4) degenerates to Eq. (3).

We note that none of the three parameterization in Eq. (2), (3) and (4) achieve strict 3D invariance, because the features are defined in the 2D depth image and the depth image is an incomplete and distorted (due to occlusion and projection) observation of the 3D object surface. Nevertheless, they are still good approximation of such invariance.

Our 3D parameterization is potential to generate features that are fully invariant to the 3D transformation $\mathcal{T}_\Theta$. We note that steps (1) and (2) above are 3D invariant, *i.e.*, the same 3D offset Δu_i corresponds to the same relative 3D location for different poses Θs that only differ by their 3D transformation $\mathcal{T}_\Theta$s. Therefore, if we are able to replace the depth feature in step (3) with some real 3D feature, *e.g.*, the signed distance of the 3D location to the 3D object surface (if available, *e.g.*, captured by multi-view depth sensors), our approach is fully 3D invariant.

3.3. Holistic Regression

The holistic algorithm simply regresses the entire hand pose Θ. It uses the palm based transformation $\mathcal{T}_\Theta^p$ for pose normalization and feature parameterization. The training algorithm is given in Algorithm 1. For each sample the pose residual is normalized before training (line 3) and normalized back to update the pose (line 5) for the next stage.

[1] A finger plane is defined by four finger joints. When the four joints are collinear, the plane is degenerate and we simply use the X axis of palm coordinate frame instead.

Algorithm 2 Testing algorithm for hierarchical cascaded hand pose regression.

```
input: palm regressors $\{\mathcal{R}^{p,t}\}_{t=1}^{T_1}$
input: finger regressors $\{\mathcal{R}^{f,t}\}_{t=1}^{T_2}, f \in F$
input: image $I$ and initial palm pose $\Theta^{p,0}$
for $t = 1$ to $T_1$ do                                        ▷ update palm
    $\Theta^{p,t} = \Theta^{p,t-1} + \overline{\mathcal{T}^{p}_{\Theta^{p,t-1}}}(\mathcal{R}^{p,t}(I, \Theta^{p,t-1}))$
end for
initialize $\{\Theta^{f,0}\}_{f \in F}$ as canonical finger poses on $\Theta^{p,t}$
$\Theta^0 = \Theta^{p,T_1} \cup \{\Theta^{f,0}\}_{f \in F}$                ▷ initialize whole hand
for $t = 1$ to $T_2$ do
    for all fingers $f \in F$ do                               ▷ update fingers
        $\Theta^{f,t} = \Theta^{f,t-1} + \overline{\mathcal{T}^{f}_{\Theta^{t-1}}}(\mathcal{R}^{f,t}(I, \Theta^{t-1}))$
    end for
    $\Theta^t = \Theta^{p,T_1} \cup \{\Theta^{f,t}\}_{f \in F}$                ▷ update whole hand
end for
output: $\Theta^{T_2}$
```

Due to the normalization, the general pose update rule in Eq. (1) becomes

$$\Theta^t = \Theta^{t-1} + \overline{\mathcal{T}^{p}_{\Theta^{p,t-1}}}(\mathcal{R}^t(I, \Theta^{t-1})). \tag{5}$$

To learn $\mathcal{R}^t$ (line 4), we use the standard regression random forest [3, 7]. To train each split node in the trees, we sample a large number of random pixel difference features as described in Eq. (4) and pick the one that gives rise to maximum variance reduction over all dimensions of the pose residual. Each leaf node stores the average of all pose residuals falling into the leaf, as the prediction of this leaf. See details in Section 3.5.

3.4. Hierarchical Regression

The holistic regression algorithm above is general and can be applied to any 3D object pose estimation problem. It already works well, as shown in the experiments. For highly articulated object like hand, we propose further improvement by making the following observations.

1. The pose variations of different parts (palm and fingers) are significantly different. Palm is much more stable than fingers. Regressing all parts together is less effective and it slows down the convergence of boosted regression framework [12], because each weak regressor has relatively large errors.

2. The articulated structure of hand indicates that palm pose severely affects finger pose. In other words, a large amount of variations in the finger poses are caused by the changes in the palm pose, other than finger articulation.

3. The five finger poses are largely independent and they have additional degrees of freedom at the finger root, in addition to the global viewpoint transformation of the palm. Therefore, regressing all finger poses together and using palm based 3D pose normalization is suboptimal for individual fingers.

Algorithm 3 Training algorithm for hierarchical cascaded hand pose regression.

```
input: image $I_i$, ground truth pose $\Theta_i$, and initial palm
    pose $\Theta_i^{p,0}$ for all training samples $i$
for $t = 1$ to $T_1$ do                                        ▷ learn palm
    $\delta\Theta_i^p = \mathcal{T}^{p}_{\Theta_i^{p,t-1}}(\Theta_i^p) - \mathcal{T}^{p}_{\Theta_i^{p,t-1}}(\Theta_i^{p,t-1})$        ▷ residual
    learn $\mathcal{R}^{p,t}$ to approximate $\delta\Theta_i^p$
    $\Theta_i^{p,t} = \Theta_i^{p,t-1} + \overline{\mathcal{T}^{p}_{\Theta_i^{p,t-1}}}(\mathcal{R}^{p,t}(I_i, \Theta_i^{p,t-1}))$ ▷ update
end for
initialize $\{\Theta_i^{f,0}\}_{f \in F}$ as canonical finger poses on $\Theta^{p,t}$
$\Theta_i^0 = \Theta_i^{p,T_1} \cup \{\Theta_i^{f,0}\}_{f \in F}$                ▷ initialize whole hand
for $t = 1$ to $T_2$ do                                        ▷ learn fingers
    for all fingers $f \in F$ do
        $\delta\Theta_i^f = \mathcal{T}^{f}_{\Theta_i^{t-1}}(\Theta_i^f) - \mathcal{T}^{f}_{\Theta_i^{t-1}}(\Theta_i^{f,t-1})$        ▷ residual
        learn $\mathcal{R}^{f,t}$ to approximate $\delta\Theta_i^f$
        $\Theta_i^{f,t} = \Theta_i^{f,t-1} + \overline{\mathcal{T}^{f}_{\Theta_i^{t-1}}}(\mathcal{R}^{f,t}(I_i, \Theta_i^{t-1}))$ ▷ update
    end for
    $\Theta_i^t = \Theta_i^{p,T_1} \cup \{\Theta_i^{f,t}\}_{f \in F}$            ▷ update whole hand
end for
output: palm regressors $\{\mathcal{R}^{p,t}\}_{t=1}^{T_1}$
output: finger regressors $\{\mathcal{R}^{f,t}\}_{t=1}^{T_2}, f \in F$
```

All these observations are intuitive. They naturally motivate us to develop a new *hierarchical* regression algorithm. It extends the holistic algorithm in two aspects. Firstly, regression is performed sequentially along the articulated chain. The root part is estimated at first. Because it has less variations, the learning is easier and converges faster, than learning all parts together. Secondly, a sub-part is normalized to its local coordinate frame that connects it to the root part, and is learnt with the root part fixed. Such local normalization can better align the sub-parts, reduce the variations and make the learning easier.

The hierarchical regression is general for any articulated objects. When applied to hand, we first estimate the palm pose, fix it and then estimate the finger poses accordingly. The palm pose learning is similar as in Section 3.3. The five finger poses are learnt separately using different regressors. Corresponding finger based 3D transformation is used for finger pose normalization and feature parameterization. The testing and training algorithms for hierarchical hand pose regression are given in Algorithm 2 and 3. We use almost exactly the same regression random forest. The only difference is that the regression target is now palm Θ^p or finger Θ^f instead of the whole hand Θ.

3.5. Implementation Details

Initialization Both training and testing require an initial pose Θ^0. It is computed heuristically. The global hand position (middle finger root) is initialized as the center of the 3D point cloud of the hand. The global rotation is initialized from the principle component analysis (PCA) of the point cloud: the Y axis is the 3D direction with the largest variation and the Z axis is the 3D direction with the smallest variation. All joints are then initialized at their canonical positions, relative to the global pose, as shown in Fig. 2(a). Unlike [9] that uses multiple random initial poses on the same image to enhance the robustness, we only use one initialization during training and testing, as describe above.

The heuristic initialization is quite rough and may causes errors when such initialization is unstable and inconsistent on different hand poses. Yet, the cascaded regression approach is usually strong enough to converge from such rough initialization. Fig. 1 and 7 show real examples.

Training parameters Each regression random forest (for whole hand, palm, or fingers) consists of 10 trees. Each split node is trained with 512 random features. The tree node splitting stops when the node contains less than 10 samples. The 3D offsets Δu_i in Eq. (4) are randomly sampled in a 3D volume of roughly the same size of a real 3D hand model. We note that the size of the 3D hand model is different for different persons. In our experiment (training and testing), such hand size parameters are assume known and manually defined. For holistic regression, we use 6 cascaded stages. For fair comparison, we use 3 stages for both palm and finger in hierarchical regression.

4. Experiments

Dataset There exists few public datasets for hand pose estimation. Most early works perform quantitative evaluation only on synthetic data. Several recent hand tracking works [21, 32, 26] provide a small number of real video sequences. They are insufficient for learning based methods.

The recent work [34] releases its training/testing data[2] and allows a fair comparison. It turns out that these data are relatively simple and our approach achieves very good results on them. Therefore, in this work we collect and release a large and more challenging dataset[3].

Evaluation metric We use two accuracy metrics. The first one is the per-joint error (in millimeters) averaged on all images. The second one is the success rate, *i.e.*, the percentage of good frames. A frame is considered good if its maximum joint error is smaller than a small threshold (like 20 mm). This strict measure has been firstly used for human pose estimation [19] and then used for hand [34].

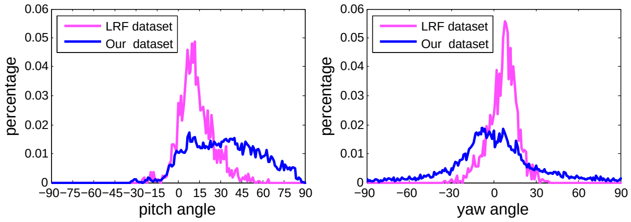

Figure 4. Viewpoint (pitch and yaw) distribution of testing dataset in [34] and our dataset.

Two baselines For comparison, we implement two baselines that can be considered as degenerated variants of our approach. Both use regression random forest to estimate the whole hand pose Θ with different pixel feature parameterization in Eq. (2) and (3). The reference pixel u in Eq. (2) and (3) is the center of the patch. The in-plane rotation angle α in Eq. (3) is estimated from our 3D PCA based initialization (see Section 3.5), *i.e.*, the 2D projection of the Y axis of the initial global rotation. The other training parameters in forest are exactly the same. We denote the two baselines as hand pose regression 2D (*HPR-2D*) and hand pose regression 2D+Rot (*HPR-2D+Rot*). Note that the second baseline is in a similar spirit as in [39], *i.e.*, both estimate the in-plane rotation first and then use the features as in Eq. (3).

Comparison with state-of-the-art Many hand pose estimation methods [6, 39, 10, 22] are based on the pixel classification paradigm and seldom report joint estimation accuracy, therefore not directly comparable to our approach. The recent Latent Regression Forest method (short for *LRF*) in [34] is similar to ours in that it also performs holistic (coarse to fine) joint regression and uses random forest. Therefore, we compare with LRF using the same training and testing data in [34]. Note that *LRF* [34] has been shown superior than the methods in [6, 23]. Therefore we also indirectly compare with [6, 23].

Results in Fig. 3 (left and middle) shows that: 1) cascaded regression using 3D pose-indexed features is effective since our two methods significantly outperform the baselines and LRF [34]; 2) hierarchical regression is better than holistic regression. Example results are shown in Fig. 6.

Fig. 3 (right) further analyzes the convergence properties of holistic and hierarchical regression over the 6 cascaded stages (3 for palm and 3 for finger in hierarchical regression). There are several interesting observations: 1) palm is much easier than fingers; 2) palm part in hierarchical regression is more accurate and converges faster than that in holistic regression; 3) finger parts are updated for 3 stages in hierarchical regression and they are more accurate than using 6 stages in the holistic method. These observations indicate that, when articulated object parts exhibit different amount of variations, the holistic regression is less effective. It is better to perform hierarchical regression of all parts in the order of their articulation complexity.

[2] In its ground truth annotation, each hand has only 16 joints, one less in each finger than ours. We interpolate the missing joint in our experiment.

[3] Available at http://research.microsoft.com/en-us/people/yichenw

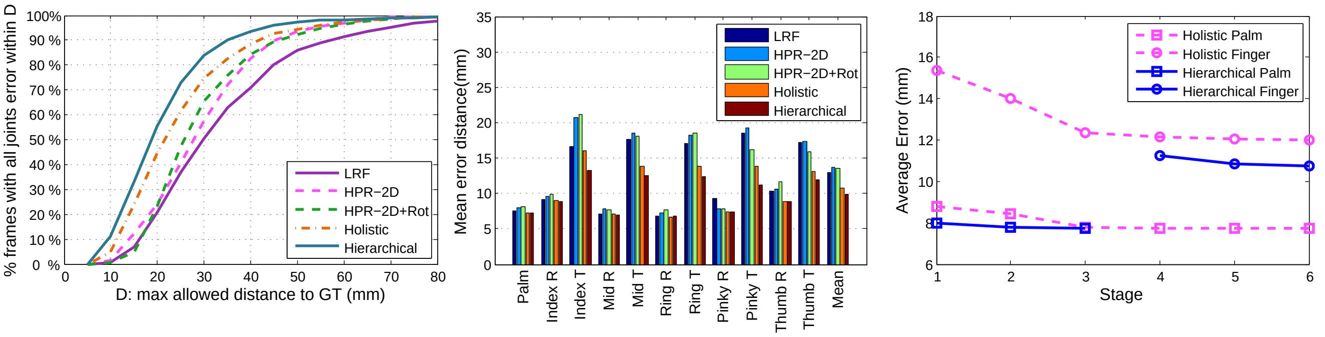

Figure 3. Comparison of different methods on the dataset in [34]. Left: success rates over different thresholds. Middle: per-joint average errors (R:root, T:tip). Right: average joint errors of palm and all fingers over 6 cascaded stages, for our holistic and hierarchical regression.

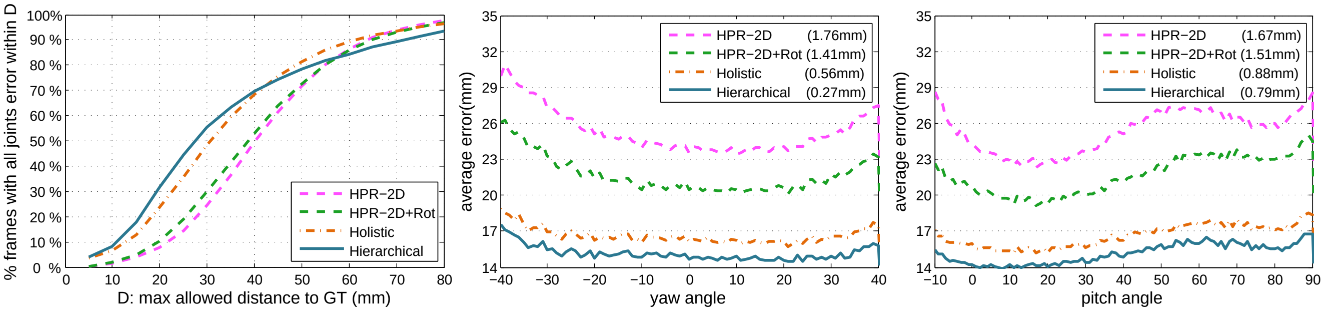

Figure 5. Comparison of different methods on our dataset. Left: success rates over different thresholds. Middle/Right: the average joint errors distributed over all yaw/pitch viewpoint angles. The standard deviations of the error distributions are shown in the legend titles.

Evaluation on our new dataset Our results on the data in [34] are surprisingly good. The mean joint error of hierarchical regression is about 10 mm, as shown in Fig. 3 (middle). Such high accuracy is even comparable to the state-of-the-art model based tracking approaches [15, 32, 23, 26] but may indicate that the dataset is too simple. Fig. 4 shows that the viewpoint variation of the testset is relatively small (yaw within $[-30, 30]$ degrees and pitch within $[-10, 45]$ degrees) and most images are at nearly frontal viewpoint.

To better reveal the challenges of hand pose estimation and further validate our approach, we collect a large scale and more challenging dataset. It consists of $76,500$ depth images captured from 9 subjects, using Intel's Creative Interactive Camera. The ground truth hand pose is annotated using the optimization method in [26] in a semi-automatic manner, *i.e*., the annotator runs optimization and manually corrects the hand pose iteratively until it is found correct.

During our data capture, each subject is asked to follow one of the 17 hand gestures each time, move rapidly under large viewpoints and change the finger articulation moderately. For each gesture 500 frames are recorded. The 17 gestures are manually chosen by us and are mostly from American Sign Language, in order to span the space of finger articulation as much as possible. As shown in Fig. 4, our dataset has larger viewpoint variations (yaw nearly spans the full $[-90, 90]$ range and pitch within $[-10, 90]$ degrees). To our best knowledge, our dataset is of the largest magnitude and most difficult in the literature.

In the experiment, we train the regression model using 8 subjects and test it on the remaining one. This is repeated 9 times for all subjects and we report the average metrics. Results in Fig. 5 (left) show similar conclusions: our methods outperform the baselines and hierarchical regression is the best. Overall the results are clearly worse than that in Fig. 3, indicating that our dataset is much more difficult. In Fig. 5 (middle and right) we report the average joint errors distributed over all yaw and pitch viewpoint angles (according to ground truth). While all methods perform worse under larger viewpoints, our two methods are not only better but also more robust (with smaller standard deviations) than the baselines. Fig. 7 shows intermediate results of hierarchical regression on several challenging examples.

In Fig. 5 (left), hierarchical regression performs worse than holistic regression under larger error thresholds. This is different from Fig. 3 (left), where hierarchical regression is consistently better than holistic regression. After inspection, we found that this is because the pose initialization (see Section 3.5) is much more unstable on our new dataset than on the dataset in [34] since our dataset has larger viewpoint and gesture variations. A poor initialization usually causes large error in palm pose estimation in hierarchical regression and in turn leads to large error in fingers.

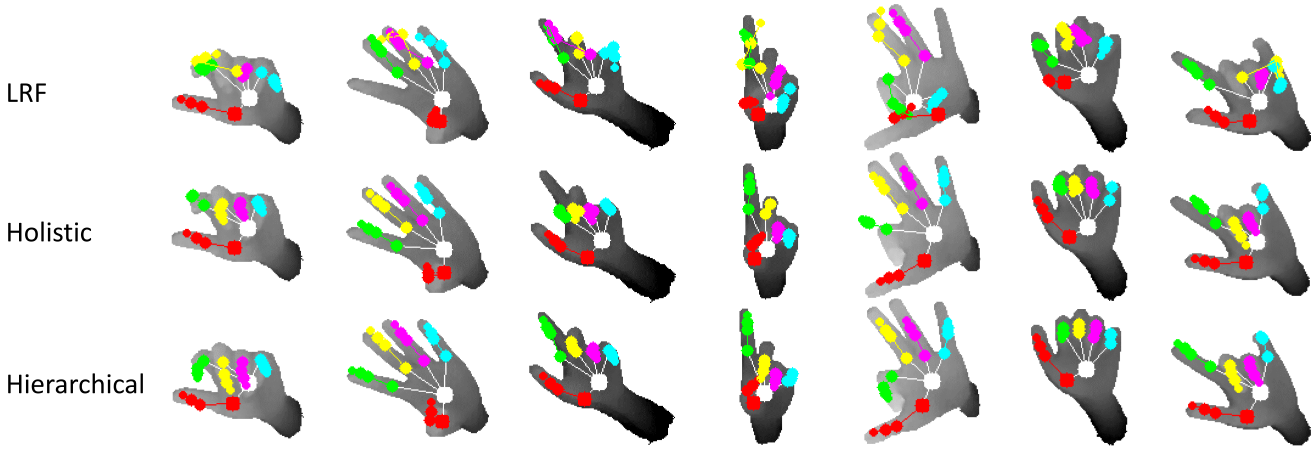

Figure 6. Example results in dataset of [34] of three methods: *LRF* [34], our holistic and hierarchical regression.

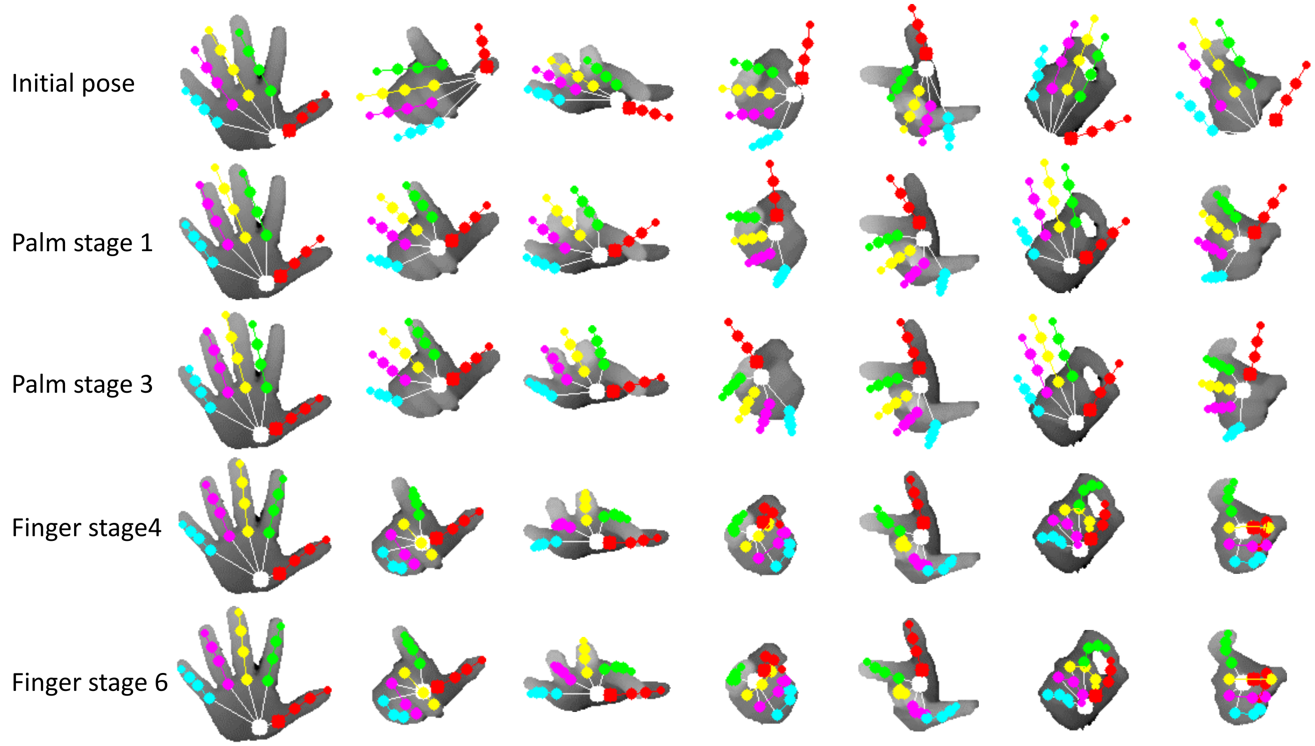

Figure 7. Example intermediate results of hierarchical regression in our dataset. Note how the initial rough poses are gradually refined.

Runtime Random forests with pixel difference features are fast to evaluate. Our hierarchical regression (including the PCA based initialization) runs in more than 300 FPS (Intel Xeon CPU 3.70GHz, single thread). Such high performance is critical for real applications. Live demo video can be found on the author's website.

For comparison, our method is much faster than previous learning based methods (8.6 FPS in [6], 12 FPS in [39], 25 FPS in [10], 62.5 FPS in [34], a few FPS in [22], 40 FPS in [35]) and model based methods (10 FPS in [32], 60 FPS in [23], 25 FPS in [26]).

5. Conclusion

We present a novel cascaded pose regression approach for 3D articulated objects. It provides a new feature parameterization for better 3D invariance and a hierarchical principle that better exploits the articulated structure. It achieves the state-of-the-art performance in both accuracy and efficiency for hand pose estimation. Future work includes automatic estimation of hand size and using stronger regression models such as refined random forest [28] or convolutional network [35] in a cascaded framework.

Acknowledgements This work is supported by The National Science Foundation of China (No.61305091), The Fundamental Research Funds for the Central Universities (No.2100219038), and Shanghai Pujiang Program (No.13PJ1408200).

References

[1] A.Baak, M.Muller, G.Bharaj, H.P.Seidel, and C.Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, 2011. 1, 3

[2] A.Erol, G.Bebis, M.Nicolescu, R.D.Boyle, and X.Twombly. Vision-based hand pose estimation: A review. *CVIU*, 2007. 3

[3] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 5

[4] B.Stenger, A.Thayananthan, P.H.S.Torr, and R.Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 2006. 3

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012. 1, 2

[6] C.Keskin, F.Kirac, Y.E.Kara, and L.Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*, 2012. 1, 2, 4, 6, 8

[7] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013. 5

[8] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012. 2

[9] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010. 1, 2, 3, 6

[10] D.Tang, T.Y, and T.K.Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013. 1, 2, 4, 6, 8

[11] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 3

[12] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 2001. 2, 5

[13] H.Hamer, K.Schindler, E.K.Meier, and L.V.Gool. Tracking a hand manipulating an object. In *ICCV*, 2009. 3

[14] I.Oikonomidis, N.Kyriazis, and A.A.Argyros. Markerless and efficient 26-dof hand pose recovery. In *ACCV*, 2010. 3

[15] I.Oikonomidis, N.Kyriazis, and A.A.Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, 2011. 3, 7

[16] I.Oikonomidis, N.Kyriazis, and A.A.Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 3

[17] J.M.Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, 1994. 3

[18] J.Shotton, A.Fitzgibbon, M.Cook, T.Sharp, M.Finocchio, R.Moore, A.Kipman, and A.Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011. 1, 2, 4

[19] J.Taylor, J.Shotton, T.Sharp, and A.Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*, 2012. 1, 3, 6

[20] E. Krupka, A. Vinnikov, B. Klein, A. B. Hillel, D. Freedman, S. Stachniak, and C. Keskin. Discriminative ferns ensemble for hand pose recognition. In *CVPR*, 2014. 1

[21] L.Ballan, A.Taneja, J.Gall, L.V.Gool, and M.Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012. 6

[22] H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *IEEE Trans. Multimedia*, 2014. 1, 2, 4, 6, 8

[23] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Interactive 3D Graphics and Games*, 2013. 3, 6, 7, 8

[24] M.L.Gorce, D.J.Fleet, and N.Paragios. Model-based 3d hand pose estimation from monocular video. *PAMI*, 2011. 3

[25] I. Oikonomidis, M. I. Lourakis, and A. A.Argyros. Evolutionary quasi-random search for hand articulations tracking. In *CVPR*, 2014. 3

[26] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *CVPR*, 2014. 3, 6, 7, 8

[27] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps by regressing local binary features. In *CVPR*, 2014. 2

[28] S. Ren, X. Cao, Y. Wei, and J. Sun. Global refinement of random forest. In *CVPR*, 2015. 8

[29] R.Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *ICCV*, 2011. 1, 2, 3, 4

[30] R.Y.Wang and J.Popovi. Real-time hand-tracking with a color glove. In *SIGGRAPH*, 2009. 3

[31] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible realtime hand tracking. In *CHI*, 2015. 3

[32] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*, 2013. 3, 6, 7, 8

[33] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *CVPR*, 2012. 2

[34] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*, 2014. 1, 2, 4, 6, 7, 8

[35] J. Tompson, M. Stein, Y. LeCun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 2014. 1, 2, 8

[36] V.Ganapathi, C.Plagemann, D.Koller, and S.Thrun. Real-time human pose tracking from range data. In *ECCV*, 2012. 1, 3

[37] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai. Video-based hand manipulation capture through composite motion control. In *ACM Trans. Graph*, 2013. 3

[38] Y. Wu, J. Y.Lin, and T. S.Huang. Capturing natural hand articulation. In *ICCV*, 2001. 3

[39] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *ICCV*, 2013. 1, 2, 4, 6, 8