# Resolving Ambiguous Hand Pose Predictions by Exploiting Part Correlations

Hui Liang, Junsong Yuan, *Senior Member, IEEE,* and Daniel Thalmann

***Abstract*—The positions of the hand joints are important high-level features for hand-based human-computer interaction. We present a novel method to predict the 3D joint positions from the depth images and the parsed hand parts obtained with a pre-trained classifier. The hand parts are utilized as the additional cue to resolve the multi-modal predictions produced by the previous regression-based method without increasing the computational cost significantly. In addition, we further enforce the hand motion constraints to fuse the per-pixel prediction results. The posterior distribution of the joints is formulated as a weighted Product of Experts model based on the individual pixel predictions, which is maximized via the Expectation-Maximization algorithm on a learned low dimensional space of the hand joint parameters. The experimental results show the proposed method improves the prediction accuracy considerably compared to the rivals that also regress for the joint locations from the depth images. Especially, we show that the regressor learned on synthesized dataset also gives accurate prediction on real-world depth images by enforcing the hand part correlations despite their discrepancies.**

***Index Terms*—Hand Joint Prediction, Multimodal Prediction Fusion, Random Regression Forest.**

## I. Introduction

HAND pose estimation is an important research topic in human-computer interaction (HCI) which has various applications, such as gesture recognition and animation synthesis. Previously the specialized hardware, *e.g.* the optical sensors [1] and the data-gloves [2], are commonly used to accomplish this task. Although they provide accurate measurements and achieve real-time performance, such devices are cumbersome to use and expensive. Thus the vision-based methods have been the mainstream in this field, which are cheaper and provide more natural interaction experiences. However, due to the high flexibility and self-occlusion of the hand, it remains a challenging task to capture the articulated hand motions from the visual inputs.

Most conventional vision-based methods utilize certain global characteristics, *e.g.* the contour and silhouette, to infer the hand pose either by template matching [3-5] or model-based fitting [6-9, 35]. In template matching, the hand pose is obtained by nearest-neighbor search in a vast set of templates, each of which contains the descriptor for matching and the associated pose parameters. As the global descriptors are generally incapable of encoding all the information needed to infer the complete pose parameters, these methods suffer from ambiguous pose predictions. In model-based fitting, the hand pose is estimated by fitting an adjustable hand model to the hand image to minimize the matching error between the model and image. Such methods are still popular as they work well in the constrained environments, *e.g.* the hand is precisely extracted and the model is consistent with the input hand. However, they lack the robustness against imperfect inputs and are relatively slow due to the high computation cost involved in optimizing the high dimensional pose model. To address these issues, the recent trend in pose estimation has been fusing the individual estimations obtained by many weak pose estimators [10-13, 27, 31] to reconstruct the complete pose, and each weak estimator infers a subset of the pose parameters based on part of the input, *e.g.* an image patch or a set of pixels. These methods prove more efficient and robust compared to those that rely on the global image features.

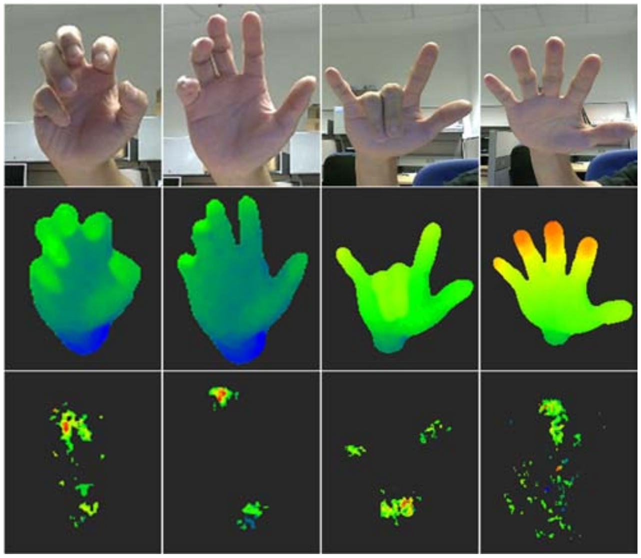


Fig. 1. Illustration of the per-pixel vote distributions of the middle fingertip obtained by regression using [13] (Lower Row). The upper and middle rows show the input color and depth images.

In this paper we follow the idea of fusing weak pose estimators to predict the hand pose from single depth images, and define the pose as the 3D positions of the hand joints. The proposed method is motivated by the human pose regression framework in [13], in which the regression forest is used for per-pixel joint prediction. Despite its high accuracy and capa-

H. Liang and D. Thalmann are with BeingThere Centre, Institute of Media Innovation, Nanyang Technological University, 50 Nanyang Drive, Singapore 637553. (e-mail: hliang1@e.ntu.edu.sg, danielthalmann@ntu.edu.sg).

J. Yuan is with School of Electrical & Electronics Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. (e-mail: jsyuan@ntu.edu.sg ).

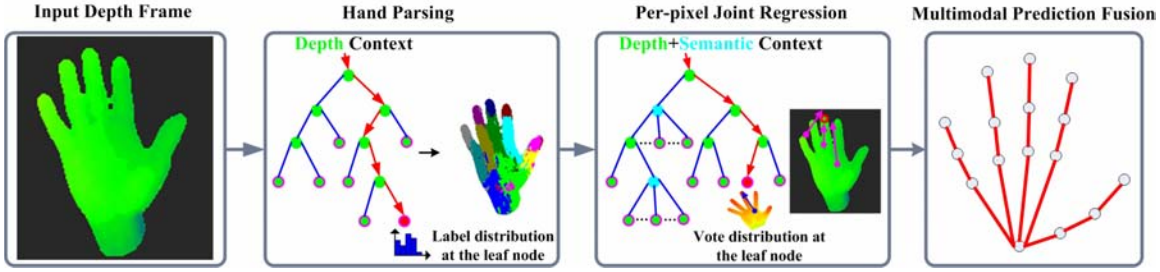

Input Depth Frame
Hand Parsing
Depth Context
Label distribution at the leaf node
Per-pixel Joint Regression
Depth+Semantic Context
Vote distribution at the leaf node
Multimodal Prediction Fusion

Fig. 2. The pipeline of the proposed hand pose estimation scheme.

bility to handle partial occlusion, it relies on the independent per-pixel votes and the correlations among the hand parts are not exploited. Given the inconsistency between the real-world inputs and the synthesized training data, many pixels will be misclassified and the vote distribution can be multi-modal. Fig. 1 shows the vote distributions for the middle fingertip obtained by accumulating the per-pixel predictions with the regression forest in [13], in which the warm colors indicate high confidences, and vice versa. The vote distributions obtained on the real-world images using the regressor trained on synthesized images are quite scattered. Besides, the different joint positions are estimated independently of each other by aggregating the votes from all the pixels via the Mean-Shift algorithm in [13]. However, such independent estimation scheme can easily lead to infeasible poses without exploiting the joint correlations, considering the severe self-occlusion of the hand and large viewpoint variations compared to the body.

Therefore, we propose to enforce the hand part correlations to resolve the ambiguous per-pixel predictions obtained using the regression forest [13]. The hand part correlations are exploited from two different aspects. First, the co-occurrence pattern between the hand parts, *e.g.* some parts are more likely to be adjacent than others [26], is useful to refine the per-pixel votes in a discriminative way. As the hand parsing results have encoded such co-occurrence patterns, we propose to extract the semantic context descriptor from the discrete hand part labels to complement the depth context descriptor for pose regression. The combination of the depth and semantic contexts proves more effective than regression using the depth context alone. Besides, the hand motion is highly constrained and the joint parameters are embedded in a low dimensional space. To handle the multi-modal per-pixel predictions obtained by the regression forest, the pose can be refined by maximizing the joint posterior of the pose parameters in the low dimensional space instead of the original space. In this paper the posterior distribution is modeled as a weighted Products of Experts [28] based on the per-pixel joint predictions. The low dimensional space of the joint parameters is learned by Principal Component Analysis (PCA) of the training samples. Based on this formulation we show that the posterior distribution can be efficiently maximized by the Expectation-Maximization (EM) algorithm [29].

The pipeline to process one frame during the testing stage is shown in Fig. 2. The depth image is first parsed by per-pixel classification using a pre-trained Random Decision Forest (RDF) classifier to obtain the hand parts. The parsing results are then combined with the depth image to be encoded into the depth and semantic context descriptors. Each pixel casts its votes for the joint locations with the Regression Forest, which is trained using both descriptors. To illustrate this we show some exemplary per-pixel votes for the middle fingertip in Fig. 2. The final joint locations are obtained by fusing the multimodal per-pixel predictions. By utilizing the hand part correlations in this way, the prediction accuracy is substantially improved compared to [13] and [33] on both a synthesized dataset and a real-world dataset. Especially, on the challenging real-world dataset which consists of four different subjects, the proposed method improves the prediction accuracy by $14.44\%$ compared to the baseline method [13], and many of the results are visually very close to the ground truth joint positions.

The remainder of this paper is organized as follows: In Section II, we give a literature review of vision-based hand pose estimation techniques. In Section III, IV and V, we present a detailed description and analysis of the proposed hand pose estimation scheme. In Section VI we show the experimental results and performance comparison. In Section VII we give our concluding remarks and further work.

## II. Related Work

The problem of vision-based pose estimation for the human hand and the full body has been extensively studied in literature. However, as the hand is highly flexible and the hand parts often occlude each other, it is still difficult to restore the full degree-of-freedom hand motion from the color and depth inputs. Generally the pose can be defined as either the joint positions or the joint angles. As in Section I, we categorize the related techniques in pose estimation into template-matching or model-based fitting with the global descriptors, and fusion of multiple estimations obtained by weak pose estimators.

In template-matching based methods, a set of templates are usually required to contain the possible postures and indexed for fast nearest-neighbor search. The pose of the input can be obtained by looking for the templates that have the similar descriptor to the input. However, as the global descriptors alone are hardly capable to encode all the information needed to infer the complete pose parameters, the estimated poses can be ambiguous. This problem is more severe if using descriptors extracted in the color images, *e.g.* the contour, as the hand is chromatically homogeneous in color. In [14] the set of possible body poses are defined by a few clusters obtained from the training data, and a function is learned to map the low-level descriptors, *e.g.* the image moment, to each of the clusters. The pose of the input is inferred by fusing the multiple candidate poses based on the mapping confidence. This method can handle a very limited number of hand poses. In [15] the simple hand grasping motion is captured with a single color camera. The locality-sensitive Hashing (LSH) is utilized to retrieve multiple candidates from the database based on the HoG feature of the input image. The hand pose is estimated by applying the temporal constraints on the retrieved candidates to resolve ambiguity. In order to restore more hand poses from the color inputs, a color glove with specially designed pattern is adopted for hand pose tracking in [3], which provides sufficient discriminative power to estimate the natural hand rotation and articulation via template matching in a large database. The final hand pose is determined by blending the multiple nearest neighbors retrieved from the database.

The model-based fitting methods infer the pose of the input by adjusting the parameters of a pre-defined model to fit the input features. To this end, the model should resemble the hand or the body in terms of both the appearance and the feasible pose configurations, and the correspondences between the model and the inputs need to be built correctly to estimate the matching error. In [16] the human body is modeled as the deformable pictorial structure, in which the pair-wise correlations between the body parts are approximated as spring-like connections. The body pose is estimated by jointly minimizing the matching error between the model and inputs and the pair-wise energy between the body parts, which is efficiently solved by the generalized distance-transform technique. In [17], a 3D hand model with twenty seven pose parameters is used to fit to the 2D positions of a set of colored markers placed on the hand to estimate the pose. In order to reduce the complexity, the hand motion constraints are analyzed to reduce the twenty seven pose parameter to twelve.

In [6], the hand pose parameters are decoupled into the global motion and local finger articulations and estimated separately. The global motion is estimated by assuming the finger poses are fixed. The local finger motion is estimated by inverse kinematics using the fingertips as the end-effectors. The method is not robust as extraction of fingertips is difficult and sensitive to self-occlusion. In [18] the quadric surfaces are used to model the hand to generate the model contours efficiently, which are used to match to the image contour. A frame rate of 3 Hz is reported on a seven DOF hand motion sequence. In [19] the feasible hand configuration space is discretized and indexed with a KD-tree. The Nelder-Mead simplex algorithm is adopted to search for the hypothesized pose that best matches the input in terms of edge and silhouette similarities. However, no quantitative results are reported. In [7] the pixel depth and skin color are used to evaluate the fitting error, and its minimization is solved using a variant of particle swarm optimization (PSO) algorithm. In [9], the hand pose is restored by fitting an elaborate hand model to the inputs of eight high-resolution cameras. Several salient points on the fingers are detected by pre-trained classifiers, which are used with the edges and optical flow to build reliable correspondence between the inputs and the model. The subtle motion of the hand can be captured precisely, such as wearing the ring on the finger.

Template matching and model-based fitting can be combined to supplement each other. In [20] the geodesic extrema are extracted from the depth images, which are used to retrieve the candidate body pose by searching in the database of geodesic extrema templates. Another candidate pose is obtained by fitting a mesh body model to the depth image, and the fusion of both produces the final estimation. A similar framework is adopted in [21], in which the fingertips are detected with the SVM classifier and HoG descriptor in the depth images, and used to retrieve the finger poses from the database. The retrieved poses are fused with the fitted poses obtained by fitting the hand model to multiple color input images. The results show the detected fingertips largely reduce the estimation error compared to model fitting alone.

The performances of these two methods largely rely on the quality of the extracted global descriptor, and thus are sensitive to imperfect inputs. Therefore, it would be more favorable to restore the complete pose by fusing the partial estimations obtained by many weak pose estimators. In [12], the whole parameter set of the hand pose is decomposed into many overlapping subsets. LSH-based nearest neighbor search is used to get the partial estimation for each subset, and the results are further integrated by a simulated annealing EM algorithm to estimate the global pose. In [10, 22], the RDF classifier is trained for per-pixel classification of the depth images and the joint locations are obtained by mean-shift mode seeking based on the labeled results. In [30] the RDF classifier is adopted for contour pixel labeling for hand pose recognition based on a rotation-invariant depth feature. [23] presents a human body pose tracking framework based on 3D model fitting. While the input body size can vary a lot, the RDF classifier provides rough body parsing for fitting the size of the 3D model to the real inputs as well as for initialization and recovering from tracking failure. In [13] the regression forest is utilized to prediction the body joint positions independently, but the motion constraints of the joints are not fully exploited. Therefore, in [33] the authors propose to improve [13] by first finding a set of candidate locations for each joint through mode-seeking, and then applying the bone length constraints to obtain the optimal combination of the different joint locations via Dynamic Programming (DP). However, the bone lengths alone are still insufficient to describe the feasible hand pose space, *e.g.* the motion constraints between multiple fingers. Besides, this method can only work for a fixed hand size, and does not generalize well to different users.

## III. The Proposed Scheme

We propose to infer the 3D positions of the hand joints from single depth images, and aim to address the inconsistency between the real-world inputs and the synthesized training datasets by enforcing the hand part correlations. The sixteen objective joint positions are shown in Fig. 3(a), which consist of the wrist center, the five fingertips, the inter-phalangeal and metacapophalangeal joints of the thumb, and the proximal inter-phalangeal and metacapophalangeal joints of all the other four fingers.

Similar to [13], the regression forest is utilized for per-pixel voting for the individual joints separately. However, we

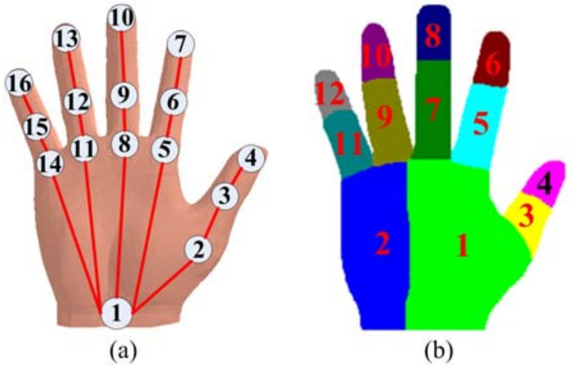


Fig. 3. (a) The objective positions of the sixteen hand joints, denoted as the circles. (b) The hand partition scheme for hand parsing.

enforce the hand part correlations to improve the per-pixel regression results by (1) improving the discriminative power of the regression forest and (2) fusing the per-pixel votes for all the joints simultaneously. First, by incorporating high-level features to model the hand part interdependence, we design a more effective regression forest to predict the joint locations. Compared with the raw depth image, the parsed hand parts [26] can be taken as the semantic context to encode the co-occurrence pattern in the neighborhood of a given pixel, which is helpful to produce more compact per-pixel predictions. Second, as the hand motion is highly constrained, the parameters of the sixteen joints are essentially embedded in a much lower dimensional space. The posterior distribution of the entire joint set can be modeled via the Product of Experts model to fuse the independent per-pixel predictions, and the optimized joint parameters can be obtained by maximizing the posterior in the low dimensional space.

Let the sixteen objective positions be $\mathbf{\Phi} = \{\phi_k\}_{k=1}^{K}$ and $K = 16$. The regression forest determines a set of maximum $J$ relative votes $\{\mathbf{\Delta}_{ijk}, w_{ijk}\}_{j=1}^{J}$ for each $\phi_k$ for the pixel $i$ during testing, where $\mathbf{\Delta}_{ijk}$ is the 3D relative offset between the 3D position of the pixel and the objective; $w_{ijk}$ is the weight of the vote. Given the 3D position $\boldsymbol{v}_i$ of the pixel $i$, the relative votes can be converted to the absolute votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$, where $\boldsymbol{v}_{ijk} = \mathbf{\Delta}_{ijk} + \boldsymbol{v}_i$. Let the depth image and label image be $I_D$ and $I_L$ respectively, and the depth context and semantic context descriptors for the pixel be $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{S}}$ respectively. Let the low dimension joint space be $\Omega$. We present the proposed pose estimation method in Algorithm 1, with the details provided in the following sections. Overall, the goal to predict the joint locations can be achieved by two sub-tasks, each of which models different aspects of the hand part correlations:

**Per-pixel Joint Regression:** for each pixel $i$, retrieve the absolute votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$ reached by $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{S}}$ in the regression forest. This sub-task corresponds to Step 1-7 in Algorithm 1.

**Multimodal Prediction Fusion:** aggregate the individual votes from all the candidate pixels to produce the posterior distribution $P(\mathbf{\Phi}|I_D)$. Maximize $P(\mathbf{\Phi}|I_D)$ with the constraints $\mathbf{\Phi} \in \Omega$ to find the optimal joint locations. This sub-task corresponds to Step 8-9 in Algorithm 1.

**Algorithm 1** Hand Joint Position Prediction.

```
1: Parsing the depth image $I_D$ to get the label image $I_L$;
2: for all pixel $i$ in the input depth image $I_D$ do
3:     Retrieve the 3D position $\boldsymbol{v}_i$ of the pixel;
4:     Retrieve the relative votes $\{\mathbf{\Delta}_{ijk}, w_{ijk}\}_{j=1}^{J}$ from the regression forest based on both $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{S}}$;
5:     Calculate the absolute votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$ by setting $\boldsymbol{v}_{ijk} = \mathbf{\Delta}_{ijk} + \boldsymbol{v}_i$;
6: end for
7: Eliminated unreliable long range predictions by setting the vote weights to zero for those votes which satisfy $\|\mathbf{\Delta}_{ijk}\| > \lambda_k$;
8: Down-sample the pixels and fuse their votes to obtain the joint posterior distribution $P(\mathbf{\Phi}|I_D)$;
9: Estimate the optimal joint positions $\mathbf{\Phi}^*$ by maximizing $P(\mathbf{\Phi}|I_D)$ subject to the hand motion constraints $\Omega$;
```

## IV. Per-Pixel Joint Regression

We utilize the regression forest [13] to obtain the joint location predictions for each pixel in the depth images. The regression forest is an ensemble of multiple random regression trees, each of which consists of a number of split nodes and leaf nodes. Each split node contains one split function learned from the training data to branch to the child node based on the feature values of the descriptor of an input pixel $i$. Each leaf node contains the distributions over the 3D relative offsets to the objective positions, which are collected from the training samples.

Different from [13] which directly regresses for the relative votes from the depth image, we propose to use the hand parsing results as the supplemental cue to the depth image for regression. Similar idea proves effective in [32], in which the input image is first parsed to get the independent body part potentials at each pixel and the potentials are then used as extra multi-channel feature for pose regression. However, this is quite memory-inefficient for training as it requires extra 12-channel potential images in our problem compared to only 1-channel depth image in [13], especially considering the large amounts of training samples involved. We therefore use the 1-channel discrete hand part labels $I_L$ instead.

We illustrate this idea in Fig. 4. During training, the depth images are used to train a hand parser to classify the testing depth image into non-overlapping hand parts. By running the trained hand parser on the training depth images, their label images can be obtained. The depth and parsed label images are then combined to train the regressor. By training the regression forest with both the depth and semantic contexts, the training samples that reach each leaf node will be similar in terms of both the depth values and hand part co-occurrence pattern in the neighborhood, which is helpful for the leaf nodes to generate more consistent predictions of the joint locations. In the testing phase, the raw input depth image is first processed to generate $I_L$. Then both the depth image $I_D$ and label image $I_L$ are concatenated as the input of the regression forest to predict the objective joint positions.

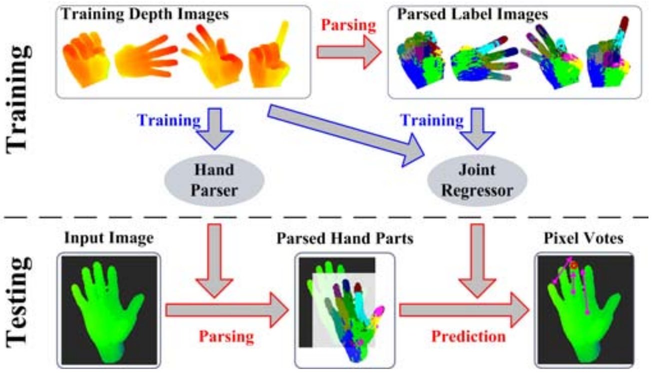


Fig. 4. The training and testing phases for per-pixel joint regression with the depth and semantic contexts. The exemplary votes for the middle fingertip are illustrated.

### A. Hand Parsing

The task of hand parsing is to assign a label $l \in L$ to each pixel in the depth image of the hand region. Fig. 3(b) shows our hand label partition scheme for classification, and the whole hand is divided into twelve non-overlapping parts. We adopt the depth-context feature [26] and the RDF classifier [24] to fulfill this task. For a pixel $i$, the depth context descriptor $\boldsymbol{\mathcal{D}}$ is defined as the depth differences between $i$ and a set of its neighboring points:

$$\mathcal{D} = I_D\left(\boldsymbol{p} + \frac{\boldsymbol{u}}{I_D(\boldsymbol{p})}\right) - I_D(\boldsymbol{p}), \tag{1}$$

where $\mathcal{D}$ is one dimension of $\boldsymbol{\mathcal{D}}$; $\boldsymbol{p}$ is the pixel coordinate of the pixel $i$; $\boldsymbol{u}$ is the offset of a neighboring point. Usually the depth context $\boldsymbol{\mathcal{D}}$ consists of hundreds of dimensions, each of which uses a different offset $\boldsymbol{u}$ to estimate $\mathcal{D}$. Based on the depth context, the RDF classifier is trained for the classification task.

During the test stage, an input pixel $i$ is first processed by each tree in the forest. For each tree, the posterior probability $P_t(l|\boldsymbol{\mathcal{D}})$ is obtained by starting at the root and recursively branching to the left or the right child based on the tree node test result until it finally reaches a leaf node. The final posterior probability $P(l|\boldsymbol{\mathcal{D}})$ is obtained by fusing the results of all the trees:

$$P(l|\boldsymbol{\mathcal{D}}) = \frac{1}{T_c}\sum_{t=1}^{T_c} P_t(l|\boldsymbol{\mathcal{D}}), \tag{2}$$

where $T_c$ is the number of trees in the forest. The label of the pixel can be directly determined by MAP estimation: $l^* = \arg\max_l P(l|\boldsymbol{\mathcal{D}})$. The label image $I_L$ is produced by finding the hand part labels for all the pixels in $I_D$.

### B. Prediction with Regression Forest

The regression forest is used to predict a set of up to $J$ votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$ for each input pixel $i$ and objective $\phi_k$ given the depth context and semantic context descriptors $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{S}}$. During the training phase, the depth images in the training dataset are first parsed to get the label images by the learned hand parser in Section IV.A. We then concatenate the depth context and semantic context to generate the new samples to train the regression forest for prediction. Thus, for each sample pixel $i$, the depth context $\boldsymbol{\mathcal{D}}$ is still defined with Formula (1). At the same time, the semantic context $\boldsymbol{\mathcal{S}}$ is defined with the hand part labels of a set of offsets $\boldsymbol{u}_l$ which are similar to those used in calculating $\boldsymbol{\mathcal{D}}$. The descriptor $\boldsymbol{\mathcal{S}}$ is obtained by:

$$\mathcal{S} = I_L\left(\boldsymbol{p} + \frac{\boldsymbol{u}_l}{I_D(\boldsymbol{p})}\right), \tag{3}$$

where $\mathcal{S}$ is one dimension of $\boldsymbol{\mathcal{S}}$. To train the regression forest, each sample pixel $i$ is also associated with the ground truth of the hand part label $l_i$ and the offsets between its 3D position and the sixteen objectives, *i.e.* $\boldsymbol{\Delta}_{ik}$, $k = 1, ..., K$.

Let the regression forest consist of $T_r$ random regression trees, each of which contains a set of split and leaf nodes. As the feature values may be either continuous or discrete, *i.e.* $\mathcal{D}$ and $\mathcal{S}$, and the split functions are tested on a single dimension of the feature, the number of the child nodes and the split criteria are thus different for the split nodes. Below we explicitly write out the indices of the feature dimensions for clarity. For the split nodes that contain the continuous dimension $\mathcal{D}^m$, they have two child nodes and the split function takes the following form:

$$\mathcal{D}^m \leq \tau, \tag{4}$$

where $\mathcal{D}^m$ is the $m^{th}$ dimension of $\boldsymbol{\mathcal{D}}$, and $\tau$ is a threshold to determine the branch to one of the child nodes, *i.e.* the left child for $\mathcal{D}^m \leq \tau$ and right child otherwise. For the split nodes that contain the discrete dimension $\mathcal{S}^n$, they have a maximum of twelve child nodes, each of which corresponds to one hand part label. The split function of the node selects the branch to the child node by checking the label value of $\mathcal{S}^n$.

To learn the tree structures of the regression forest, a set of candidate split functions $\{\psi\} = \{\psi_D, \psi_S\}$ are first generated as the proposals. $\{\psi_D\}$ are associated with the continuous dimensions and generated by sampling $m$ and $\tau$. $\{\psi_S\}$ are associated with the discrete dimensions and generated simply by sampling $n$. Similar to [13], we choose the Shannon Entropy Gain for the hand part labels to select the split functions, which proves to be more effective than the other criteria such as the variances of the offsets. The Shannon Entropy Gain for $\psi_D$ is calculated by:

$$G(\psi_D) = H(A) - \sum_{b\in\{l,r\}} \frac{|A_b(\psi_D)|}{|A|} H(A_b(\psi_D)), \tag{5}$$

where $H$ is the entropy of the hand part label distributions in the sample set $A$ that reaches the split node, and $A_l$ and $A_r$ are the two subsets of $A$ split by the function $\psi_D$. The Shannon Entropy Gain for the split function $\psi_S$ is calculated by:

$$G(\psi_S) = H(A) - \sum_{b=1}^{12} \frac{|A_b(\psi_S)|}{|A|} H(A_b(\psi_S)), \tag{6}$$

where $A_b$, $b = 1, ..., 12$ are the twelve subsets of $A$, each of which contains the samples that have the hand part label $b$. Thus, the optimal split function is selected so that $\psi^* = \arg\max_\psi G(\psi)$ at each split node during training. With this criterion, the tree structure of the forest is learned with the procedure similar to that in [22].

The regression models for the relative votes at the leaf nodes are learned from the set of relative offsets $\{\boldsymbol{\Delta}_{ik}\}$ associated with the training samples reaching them. At each leaf node, we define the regression model as a single relative vote $(\boldsymbol{\Delta}_k, w_k)$ for each joint, where $\boldsymbol{\Delta}_k$ represents the possible prediction of the relative offset based on the relative offsets $\{\boldsymbol{\Delta}_{ik}\}$ from the training samples, and $w_k$ represents the confidence of the prediction. As in [13], we adopt the mean-shift algorithm [25] to obtain the modes of the relative offset using the following density estimator:

$$g_k(\boldsymbol{\Delta}) = \sum_{i=1}^{n_L} \exp\left(-\left\|\frac{\boldsymbol{\Delta} - \boldsymbol{\Delta}_{ik}}{b_k}\right\|^2\right), \tag{7}$$

where $n_L$ is the number of the training samples reaching the leaf node; $b_k$ is the bandwidth. Besides, the weight $w_k$ is estimated for each mode at the leaf node to reflect their significance in prediction. Following [13], it is defined as the sum of the depth-adjusted weights of the samples that reach each mode.

In the testing phase, an input pixel $i$ will recursively branch down the tree and reach one leaf node in each regression tree in the forest based on the descriptors $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{S}}$. In total the pixel reaches $T_r$ leaf nodes in the regression forest and thus retrieves at most $J = T_r$ votes from the forest for each objective, *i.e.* $\{\boldsymbol{\Delta}_{ijk}, w_{ijk}\}_{j=1}^{J}$. In addition, as shown in [13], the long range predictions are usually unreliable and could be eliminated to improve the prediction accuracy by threshold of $\boldsymbol{\Delta}_{ijk}$ with a constant $\lambda_k$. Therefore, the votes that do not satisfy $\|\boldsymbol{\Delta}_{ijk}\| \leq \lambda_k$ are taken away from the vote set of each pixel by setting their corresponding weights to zero. The threshold $\lambda_k$ takes different value for each objective and can be learned during training. With the 3D position $\boldsymbol{v}_i$ of the pixel $i$, we can finally obtain the absolute votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$ with the regression forest, where $\boldsymbol{v}_{ijk} = \boldsymbol{\Delta}_{ijk} + \boldsymbol{v}_i$ is the absolute vote for the $k^{th}$ joint.

Since we also learn classification when building the regression forest during the training stage, the regression forest can thus be used for hand parsing again with the input of the depth image and the parsing results from the RDF classifier in Section IV.A. Therefore, our proposed two-layered forest can be further extended to $N_L$ layers. During testing, the parsing results from the $n$ layer are reutilized with the depth image as the input for the $n+1$ layer, and finally we regress for the joint positions at the $N_L$ layer with the depth and parsing results from $N_L - 1$ layer. We have tested the performance of such extended multi-layered forest in Section VI.E.

### C. Discussion of the Impact of the Semantic Context

To make the regression forest accurate in estimating the joint locations, we expect the samples reaching the same leaf node to give consistent predictions. However, as only a subset of the dimensions of $\boldsymbol{\mathcal{D}}$ is tested for a testing sample going from the root to the leaf, the chances that the sample pixels at completely different positions of the hand can reach the same leaf node is still high during the testing stages. In addition, as the multiple regression trees of the forest are trained independently and the subsets of the dimensions of $\boldsymbol{\mathcal{D}}$ tested can thus be quite different for the same input sample going through different trees. As a result, the misclassified pixels give considerable false responses during testing, and the votes retrieved for the same pixel from the multiple trees are also multimodal. The problem becomes more prominent especially for the real-world test inputs that are inconsistent with the synthesized training data.

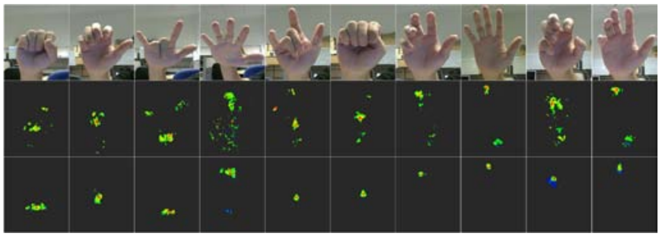


Fig. 5. Comparison of the vote distributions for the middle fingertip, in which the long range votes have been eliminated. **Middle Row:** The remaining votes obtained with [13]; **Lower Row:** the votes obtained with the proposed regression forest.

In contrast, the samples reaching one leaf node in the proposed regression forest will be similar in terms of the depth context as $\boldsymbol{\mathcal{D}}$ is utilized. Moreover, $\boldsymbol{\mathcal{S}}$ has encoded the semantic context of the pixel, *i.e.*, the co-occurrence pattern of the different hand parts. That is, some hand parts are more likely to be neighbors than others [26], *e.g.* the hand part 8 and 7 stay together more often than 8 and 3 in Fig. 3(b). By utilizing $\boldsymbol{\mathcal{S}}$ to build the regression forest, we enforce that the samples reaching the same leaf node will share similar semantic contexts in their neighborhood, and thus the predictions could be more consistent. In Fig. 5 we show several examples to compare the distributions of the prediction confidence for the middle fingertip obtained with [13] and the proposed regression forest, which are obtained by projecting the 3D votes of each pixel to the 2D image plane and accumulating the votes for all the pixels. Note here the long range predictions have been eliminated for both methods as in Section IV.B, and we can see the proposed regression forest produces much more compact prediction votes. In Section VI the experimental results also demonstrate that the prediction accuracy is improved considerably by incorporating the semantic context.

## V. Multimodal Prediction Fusion

By per-pixel prediction with the regression forest in Section IV, each pixel $i$ casts maximum $J$ votes for the individual joints independently, *i.e.* $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$. As we have seen in Fig. 1 and 5, the per-pixel predictions can form multimodal votes for each joint even when the unreliable long range predictions have been eliminated. It is difficult to determine which mode corresponds to the real joint position if we perform mode-seeking for each joint separately as in [13]. However, as the hand motion is constrained, the 3D positions of the multiple joints are highly correlated. Therefore, a large portion of the false mode combinations of different joints can be easily eliminated if we seek the modes of the per-pixel predictions for multiple joints simultaneously subject to specific learned hand configuration constraints. In Fig. 6 we

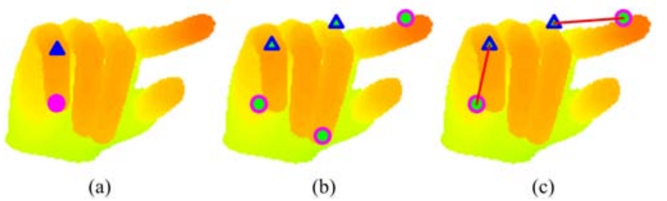


Fig. 6. Illustration of multimodal prediction fusion using the joint position constraints. (a) The two objective joints. (b) The multiple modes obtained by per-pixel regression. (c) The valid mode combination subject to the constraints.

illustrate this idea with a simple example in which we only constrain the relative positions between the proximal inter-phalangeal joint and the fingertip of the pinky. Assume two modes are found for the fingertip and three modes are found for the proximal inter-phalangeal joint according to per-pixel regression, which are denoted by the triangles and circles separately. In total there are six combinations of the modes for the two joints. Since the maximum distance between the two joints are limited, the number of candidate valid combinations is reduced from six to only two, as shown in Fig. 6 (c).

Based on the above observations, we propose to fuse the multimodal per-pixel votes through the maximization of the joint posterior distribution of all the joints in a learned low dimensional joint parameter space, which can be efficiently solved via an Expectation-Maximization (EM) framework. Due to the vast number of the pixels, we first randomly sample $N$ candidate pixels from the input depth image for further prediction fusion to strike a balance between computational cost and prediction accuracy. For each candidate pixel $i$, its retrieved votes $\{\boldsymbol{v}_{ijk}, w_{ijk}\}_{j=1}^{J}$ for the $k^{th}$ joint can be taken as a multimodal distribution $P(\phi_k|\boldsymbol{p}_i)$. We approximate this distribution with the Gaussian Mixture Model:

$$P(\phi_k|\boldsymbol{p}_i) = \sum_{j=1}^{J} \rho_{ijk} \exp\left(-\frac{\|\phi_k - \boldsymbol{v}_{ijk}\|^2}{\delta^2}\right), \tag{8}$$

where $\rho_{ijk} = w_{ijk}/\sum_j w_{ijk}$ is the weight of each mode. For simplicity we assume the same bandwith $\delta$ for the $J$ modes. Following the weighted Products of Experts model [28], the joint posterior distribution of the entire joint set given the depth image observation can be formulated as the weighted product of the individual predictions from all the candidate pixels:

$$\begin{aligned} P(\boldsymbol{\Phi}|I_D) &\propto \prod_i P(\boldsymbol{\Phi}|\boldsymbol{p}_i) = \prod_i \prod_k P(\phi_k|\boldsymbol{p}_i)^{w_{ik}} \\ &= \prod_i \prod_k \left[\sum_j \rho_{ijk} \exp\left(-\tfrac{\|\phi_k - \boldsymbol{v}_{ijk}\|^2}{\delta^2}\right)\right]^{w_{ik}}, \end{aligned} \tag{9}$$

where $w_{ik} = \sum_j w_{ijk}/\sum_{i,j} w_{ijk}$ is the normalized weight to indicate the total contribution of pixel $i$ to $\phi_k$, and $\sum_i w_{ik} = 1$.

The optimal joint locations can be obtained by maximizing $\log P(\boldsymbol{\Phi}|I_D)$ with respect to $\boldsymbol{\Phi}$, which is difficult to solve directly as the $\log \sum$ term in $\log P(\boldsymbol{\Phi}|I_D)$ cannot be further simplified. To this end, we further assume that the real joint location $\phi_k$ could be consistent with at most one mode among the maximum $J$ votes from each pixel $i$. Therefore, $\rho_{ijk}$ should be adjusted so that the inconsistent modes could be filtered out before estimating $\phi_k$. To this end, we use an EM algorithm to maximize $P(\boldsymbol{\Phi}|I_D)$ with respect to both the joint locations $\phi_k$ and the mode weights $\rho_{ijk}$ alternately. In addition, as the locations of the joints are highly correlated, such inter-dependence can be further utilized to resolve the ambiguous predictions. To be specific, we perform PCA analysis to the joint locations in the training data to learn a low dimensional representation $\Omega$ of the hand configuration. During maximization of the posterior, $\boldsymbol{\Phi}$ is constrained to take the linear form $\boldsymbol{\Phi} = \sum_m^M \alpha_m \boldsymbol{e}_m + \boldsymbol{\mu}$, $M \ll 3 \times K$, where $\{\boldsymbol{e}_m\}$ is the set of the principal components. The problem to find the optimal $\boldsymbol{\Phi}^*$ is thus formulated as follows:

$$\begin{aligned} \boldsymbol{\Phi}^*, \rho^* &= \arg\max_{\boldsymbol{\Phi},\rho} \log P(\boldsymbol{\Phi}|I_D) \\ &= \arg\max_{\boldsymbol{\Phi},\rho} \sum_{i,k} w_{ik} \log\left[\sum_j \rho_{ijk} \exp\left(-\tfrac{\|\phi_k - \boldsymbol{v}_{ijk}\|^2}{\delta^2}\right)\right] \\ s.t.\ \boldsymbol{\Phi} &= \textstyle\sum_m \alpha_m \boldsymbol{e}_m + \boldsymbol{\mu}, \sum_j \rho_{ijk} = 1 \end{aligned} \tag{10}$$

Note that the constraint on the mode weights is enforced only within the $J$ votes, and $\rho_{ijk}$ can thus be optimized separately in Formula (10) during the E step. By maximizing $P(\boldsymbol{\Phi}|I_D)$ with respect to $\rho_{ijk}$ we can get:

$$\rho^*_{ijk} = \begin{cases} 1 & if\ j = \arg\max_j \exp\left(-\frac{\|\phi_k - \boldsymbol{v}_{ijk}\|^2}{\delta^2}\right) \\ 0 & otherwise \end{cases} \tag{11}$$

This result conforms to our goal to filter out the inconsistent modes, *i.e.*, the optimized weights $\rho^*_{ijk}$ only keep the mode $\boldsymbol{v}^*_{ijk}$ that is most consistent with the current estimation $\phi_k$ among the maximum $J$ modes of the pixel $i$, and discard all other modes. Given $\rho^*_{ijk}$, $P(\phi_k|\boldsymbol{p}_i)$ is now simplified to a uni-modal distribution, and Formula (10) is thus easy to optimize in the M step. Also, the partitioned representation for each objective can be written as: $\phi_k = \sum_m \alpha_m \boldsymbol{e}_{m,k} + \mu_k$, where $\{\boldsymbol{e}_m\} = [\boldsymbol{e}^T_{m,1}, , \boldsymbol{e}^T_{m,K}]^T$ is a partition of the principal components for each joint $k$. The M step is thus equivalent to finding the optimal coefficients $\{\alpha^*_m\}$ to maximize the posterior distribution:

$$\begin{aligned} \boldsymbol{\Phi}^* &= \arg\max_{\boldsymbol{\Phi}} \sum_{i,k} w_{ik} \log\left[\exp\left(-\tfrac{\|\phi_k - \boldsymbol{v}^*_{ijk}\|^2}{\delta^2}\right)\right] \\ &= \arg\min_{\boldsymbol{\Phi}} \sum_{i,k} w_{ik} \tfrac{\|\phi_k - \boldsymbol{v}^*_{ijk}\|^2}{\delta^2} \\ &= \arg\min_{\boldsymbol{\Phi}} \sum_{i,k} w_{ik} \tfrac{\|\sum_m \alpha_m \boldsymbol{e}_{m,k} + \mu_k - \boldsymbol{v}^*_{ijk}\|^2}{\delta^2} \end{aligned} \tag{12}$$

Denote $\boldsymbol{\chi} = [\chi_1^T, ..., \chi_K^T]^T$, where $\chi_k = \sum_i w_{ik} \boldsymbol{v}^*_{ijk}$ and $\boldsymbol{\chi}$ is thus the weighted sum of the filtered votes $\boldsymbol{v}^*_{ijk}$ from all the candidate pixels. Without the low-dimensional assumption of the joint parameter space, the solution $\phi^*_k$ of Formula (12) actually equals to $\chi_k$. As shown in the Appendix, given the constraints $\boldsymbol{\Phi} = \sum_m \alpha_m \boldsymbol{e}_m + \boldsymbol{\mu}$, the optimal coefficients $\{\alpha^*_m\}$ are:

$$\alpha^*_m = \sum_k \boldsymbol{e}^T_{m,k} \left(\sum_i w_{ik} \boldsymbol{v}^*_{ijk} - \mu_k\right) \tag{13}$$

We can see the optimal solution of $\{\alpha^*_m\}$ with the constraints is the projection coefficients of $\boldsymbol{\chi}$ on the principal component subspace, *i.e.* $\alpha^*_m = \boldsymbol{e}^T_m(\boldsymbol{\chi} - \boldsymbol{\mu})$. The optimal joint locations

$\mathbf{\Phi}^*$ are then reconstructed by back projecting the coefficients $\{\alpha_m^*\}$ to the original space with the principal components.

To sum up, the proposed multimodal prediction fusion (MPF) algorithm consists of a series of iterations. During each iteration, the pixel modes that are inconsistent with the current estimation are first filtered out. The optimal joint locations are then obtained by maximization of the posterior with the remaining modes subject to the hand motion constraints $\Omega$, and the solution proves to be the reconstructed vector of the weighted sum of the filtered votes $\boldsymbol{\chi}$ on the space $\Omega$. To start the iterative procedure, we first calculate the weighted sum of the unfiltered votes from all the candidate pixels, *i.e.* $\phi_k(0) = \sum_{i,j} w_{ijk}\boldsymbol{v}_{ijk}$, and then choose $\mathbf{\Phi}^*(0)$ to be the reconstructed vector of $\{\phi_k(0)\}_{k=1}^K$ on the subspace spanned by $\mathbf{\Phi} = \sum_m \alpha_m \boldsymbol{e}_m + \boldsymbol{\mu}$ to initialize the EM steps. The E and M steps then iterate until a minimum increase of $P(\mathbf{\Phi}|I_D)$ or a maximum number of iterations are met. Also note that the value of $P(\mathbf{\Phi}|I_D)$ is monotonically increasing during both the E and M steps, and the optimization algorithm is thus guaranteed to converge.

## VI. Experimental Results

In this section we present the experimental results on both synthesized dataset and real-world dataset. The synthesized dataset is used for both forest training and quantitative evaluation of the prediction accuracy. The real-world dataset is used to test the generalization ability of the proposed method when the regression forest is trained on synthesized datasets. In addition, we also investigate the impact of various parameters on the system performance.

### A. Datasets and Evaluation Metrics

The synthesized dataset consists of $114.2k$ templates to quantitatively evaluate the performance of the methods for a large variety of hand configurations. Each template in the dataset includes the depth image, the ground truth of the hand part labels and the sixteen joint positions. Similar to [26], we use a CyberGlove II [2] to capture the hand articulation parameters for various hand motions, *e.g.* grasping, pinching, single and multiple finger bending, performing ASL gestures, etc. The captured local articulation parameters are clustered to approximately 400 templates. The range of global hand rotation is defined to be to $(-60°, 20°)$ for global rotation around the X axis and $(-80°, 80°)$ around Y axis, *i.e.* the axes parallel to the image plane of the camera, and $(-35°, 35°)$ around the Z axis, *i.e.* the axis perpendicular to the image plane. The global rotation parameters are discretized uniformly into near 300 sets within this range, and combined with the local articulation parameters to drive a 3D hand model to generate the synthesized datasets. This dataset is quite challenging for joint location prediction as a large portion of the joints are invisible in the templates, as shown in Fig. 7. In the experiments $80\%$ of the synthesized templates are used to train the regression forests and the rest $20\%$ for testing, *i.e.* $91.4k$ vs $22.8k$.

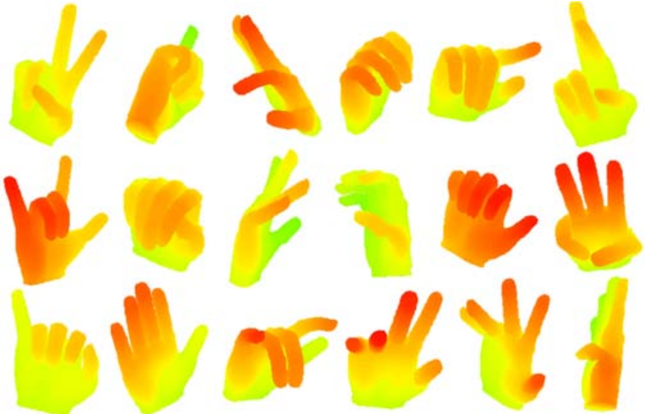


Fig. 7. Examples of the hand configurations with large viewpoint variations and different finger articulations in the synthesized dataset.

To demonstrate that the proposed depth+semantic contexts based regression forest and the MPF algorithm can well handle the discrepancy between the synthesized datasets and real-world inputs, we collect in total 1354 real depth images of four subjects using a SoftKinetic DS325 camera. In this dataset the hands of the subjects go through large viewpoint changes and various postures, and the hand sizes also vary considerably for the different subjects. In addition, the depth points are relatively noisy compared to the synthesized image; the wrist of the real hand inputs is not well segmented and part of the lower arm is still visible, as shown in Fig. 8. The resulting per-pixel votes can thus be multimodal, as in Fig. 5. This is disadvantageous for independent mode-seeking for each joint alone, since it can easily get trapped in local optima.

The ground truth joint positions in the real dataset are obtained by manual annotation. To ensure the annotation quality, we require the 3D hand skeleton to match both the 2D depth image and the 3D point cloud, as shown in Fig. 8. Each pixel in the depth image view is associated with a point in the 3D point cloud view. Starting from the initial skeleton template in Fig. 8 (a), the depth image is first used for fast annotation of the non-occluded joints by moving the projected joints to the correct pixel, as shown in Fig. 8 (b). The 3D position associated with the pixel is assigned to the joint. Since this position lies on the hand surface, an offset of $0.75cm$ is then added to the depth of the joint to compensate the surface-to-interior distance. As shown in Fig. 8 (b), the four occluded joints in the yellow rectangle cannot be annotated in the depth image. Therefore, the occluded joints are labeled by moving each of them in the 3D space and matching to the point cloud as in Fig. 8 (c). Besides, the inaccurate annotations in the previous step, *e.g.* the wrist, can also be corrected with the point cloud. Fig. 8 (d) illustrates some annotated examples in the dataset. This dataset and its ground-truth annotations can be found on our website [1].

For performance evaluation of the methods, we define the prediction accuracy for each joint as the percentage of the predictions that are within a distance of $D_T$ centimeters from the ground truth. The overall accuracy is obtained by the average of the prediction accuracies of the sixteen joint locations. In this paper we define $D_T = 1.5cm$, while the overall accuracies

[1] https://sites.google.com/site/seraphlh/projects

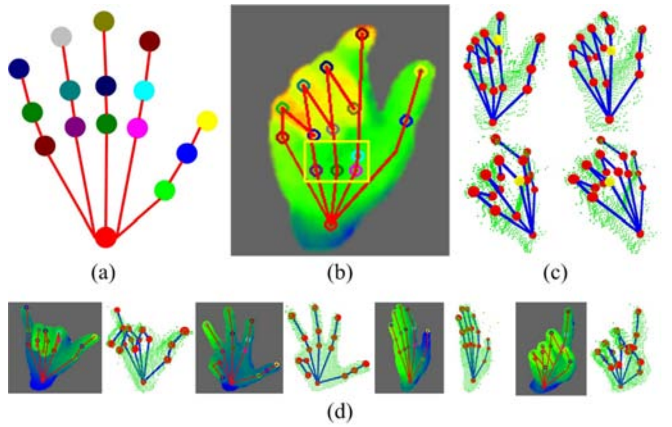


Fig. 8. Joint position annotation on the real dataset by manually fitting the hand skeleton to the depth image and the 3D point cloud.

with different values of $D_T$ *i.e.* $D_T \in [1.0, 4.0]$, are also provided to better illustrate the distribution of correct joint predictions. Note that the high accuracy corresponding to small values of $D_T$ should be more favorable, as the large values of $D_T$ indicate imprecise measurement.

### B. Implementation Details

We implemented [13] and [33] with the depth context descriptor in Formula (1) for comparison. For [13], the $E^{cls}$ criterion is adopted to learn the tree structure, which minimizes the Shannon entropy of the hand part labels at the split nodes and is reported to have the best performances. The regression forest is then learned in the same way to Section IV. Following [13], the final joint locations are obtained by mode-seeking with the Mean-Shift algorithm based on the votes and weights produced by per-pixel regression. The density estimator for each objective location is given by:

$$g_k(\boldsymbol{v}) = \sum_{i=1}^{N}\sum_{j=1}^{J} w_{ijk} \exp\left(-\left\|\frac{\boldsymbol{v}-\boldsymbol{v}_{ijk}}{b_k}\right\|^2\right) \tag{14}$$

where $\boldsymbol{v}$ is the 3D point, $(\boldsymbol{v}_{ijk}, w_{ijk})$ is the pixel votes of the $i^{th}$ pixel, and $b_k$ is the bandwidth.

To implement [33], the same $E^{cls}$ criterion and density estimator is used, and we perform Mean-Shift mode-seeking to find maximum five modes for each joint. The Dynamic Programming (DP) algorithm is utilized to find their best combination given the bone length constraints, as in [33]. In addition to (A) $\boldsymbol{\mathcal{D}}$ and MS [13] and (B) $\boldsymbol{\mathcal{D}}$ and DP [33], we further tested other three methods: (C) Regression with $\boldsymbol{\mathcal{D}}$ and MPF fusion; (D) Regression with $\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and Mean-Shift (MS) mode-seeking; (E) Regression with $\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MPF fusion. In the experiments, the regression forests consist of 4 trees with a maximum depth of 20. To learn the tree structure of the regression forest, 10000 candidate split functions are selected to train each tree, and each leaf node contains at most three relative votes for each objective. During testing, 1000 pixels are randomly selected for per-pixel regression, and the EM iteration lasts for at most three times in the MPF algorithm. These parameters are used in the following tests unless explicitly specified. All the evaluated methods were coded in C++/OpenCV, and tested on a server with two Intel Xeon X5675 CPUs and 16G RAM. The resolution of the images in all the datasets is $320 \times 240$.

### C. Quantitative Evaluations on Synthesized Datasets

The results obtained with [13], [33] and our three methods on the synthesized dataset are summarized in Table I, which consist of the average prediction accuracies $\zeta_{pred}$ for $D_T = 1.5cm$, the time costs for hand parsing, per-pixel regression and joint prediction with MS, DP and MPF, denoted as $t_{parse}$, $t_{reg}$, $t_{MS}$, $t_{DP}$ and $t_{MPF}$, and the overall time cost $t_{total}$. Besides, note the regression forest can also classify the hand part labels with the label distribution in the leaf nodes, as in Formula (2), we also report the part classification accuracy $\zeta_{cls}$ over all the pixels in Table I for reference, while it is worth mentioning that only a small number of the pixels are needed for regression to predict the joint positions. Fig. 9 and Table II show the overall accuracies for $D_T \in [1.0, 4.0]$.

In this experiment the training and testing data are relatively consistent as they are synthesized by the same hand model, and therefore the per-pixel votes obtained with the regression forest are compact and the problem of multimodal joint predictions is not quite serious. Overall, the five tested methods produce good prediction accuracy, as shown in Table I, and all our three methods outperform the baseline methods [13] and [33]. Compared to [13], the prediction accuracy is improved by 4.33% via incorporating the semantic context in discriminative regression alone, and is improved by 3.20% via the MPF algorithm. Finally, the combination of the depth and semantic contexts and the MPF algorithm obtains the highest prediction accuracy, which is 6.42% enhancement compared to the baseline. In contrast, [33] obtains very little improvement upon [13] by utilizing the bone length constraints alone, as the motion constraints between the joints are still not modeled. Moreover, the results in Fig. 9 show that the predictions obtained with the proposed methods are more compact, as the distributions with different values of $D_T$ indicate that the joint predictions obtained by our methods are nearer to the ground truth joint locations on average.

### D. Quantitative Evaluations on Real-world Datasets

For this experiment we tested [13], [33] and our three methods with the regression forests trained on the synthesized datasets in Section VI.A. The Iisu Middleware SDK provided by SoftKinetic [34] is utilized to get relatively rough hand segmentation from the depth image. Due to the noisy depth image, inaccurate wrist segmentation and the various hand sizes of the different subjects, the performance of all the five methods degrades substantially on this dataset when the regression forests and the PCA space in the MPF algorithm are learned with the synthesized data. However, we show that the MPF is capable of compensating the gap between the training data and the real-world input to some extent by learning the PCA space from the real data.

We randomly pick up a small portion of the real dataset, *i.e.* 270 images, to learn the PCA space for the MPF algorithm and

TABLE I
COMPARISON OF [13], [33] AND OUR THREE METHODS ON THE SYNTHESIZED DATASET. SAA DENOTES SAME AS ABOVE.

| Method | $\zeta_{pred}$ | $\zeta_{cls}$ | $t_{parse}$ (ms) | $t_{reg}$ (ms) | $t_{MS}$ (ms) | $t_{DP}$ (ms) | $t_{MPF}$ (ms) | $t_{total}$ (ms) |
|---|---|---|---|---|---|---|---|---|
| (A) $\boldsymbol{\mathcal{D}}$ and MS [13] | 83.26% | 81.96% | - | 38.81 | 1.35 | - | - | 42.70 |
| (B) $\boldsymbol{\mathcal{D}}$ and DP [33] | 83.51% | SAA | - | 38.70 | - | 6.64 | - | 47.90 |
| (C) $\boldsymbol{\mathcal{D}}$ and MPF | 86.46% | SAA | - | 37.63 | - | - | 2.66 | 43.59 |
| (D) $\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MS | 87.59% | **86.18%** | 28.16 | 41.15 | 1.21 | - | - | 74.04 |
| (E) $\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MPF | **89.68%** | SAA | 28.23 | 40.88 | - | - | 2.28 | 75.83 |

TABLE II
THE AVERAGE PREDICTION ACCURACIES OF [13], [33] AND OUR METHODS ON THE SYNTHESIZED DATASET FOR DIFFERENT $D_T$ (CM).

| $D_T$ | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 | 2.50 | 2.75 | 3.00 | 3.25 | 3.50 | 3.75 | 4.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (A) | 70.16% | 78.34% | 83.26% | 86.39% | 88.57% | 90.23% | 91.48% | 92.41% | 93.14% | 93.71% | 94.23% | 94.70% | 95.09% |
| (B) | 70.30% | 78.57% | 83.51% | 86.68% | 88.87% | 90.55% | 91.80% | 92.73% | 93.45% | 94.05% | 94.57% | 95.04% | 95.43% |
| (C) | 71.72% | 81.00% | 86.46% | 89.92% | 92.16% | 93.68% | 94.71% | 95.45% | 96.02% | 96.42% | 96.78% | 97.05% | 97.27% |
| (D) | 74.18% | 82.31% | 87.59% | 91.04% | 93.33% | 94.82% | 95.82% | 96.49% | 96.98% | 97.35% | 97.66% | 97.91% | 98.13% |
| (E) | 75.89% | 84.53% | 89.68% | 92.89% | 94.99% | 96.27% | 97.12% | 97.69% | 98.09% | 98.41% | 98.63% | 98.84% | 99.01% |

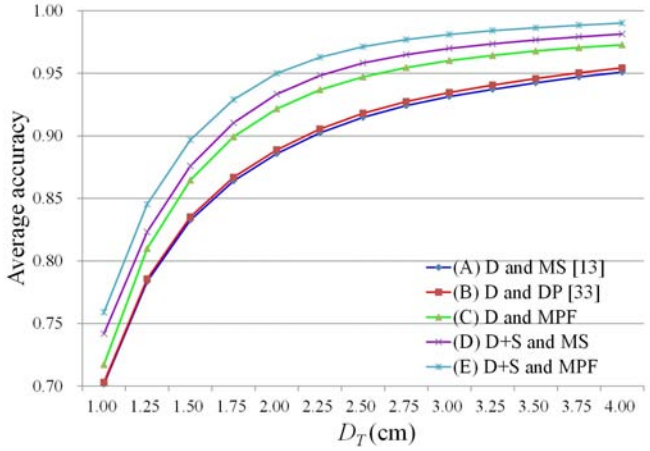


Fig. 9. Comparison of the average prediction accuracies on the synthesized dataset for different $D_T$.

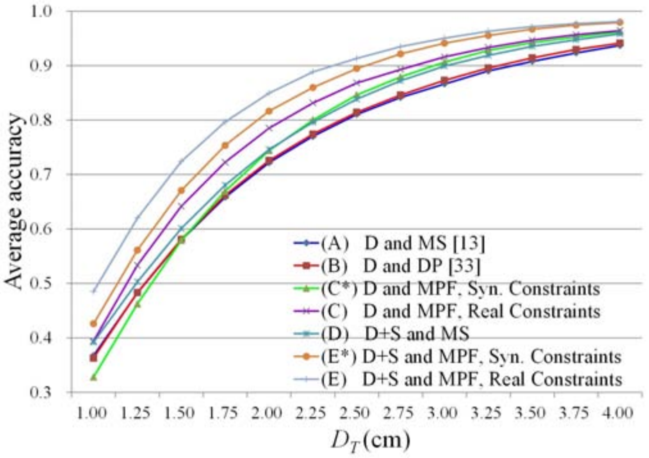


Fig. 10. Comparison of the average prediction accuracies on the real-world dataset for different $D_T$.

use the rest 1084 images for testing. Besides, the bone length constraints of the hand for [33] are obtained by calculating the average bone lengths in the 270 images. The performance of the MPF algorithm is also evaluated with the PCA space learned from the synthesized training data for comparison, which is denoted as (C*) and (E*). Fig. 10 and Table III show the comparison of the overall accuracies for $D_T \in [1.0, 4.0]$. As illustrated, the improvement of [33] over [13] is still trivial. Besides, MPF alone does not perform quite well with the PCA space learned from the synthesized data, while it still produces $9.09\%$ enhancement when combined with the semantic context compared to [13]. In contrast, with the constraints learned from only a small portion of the dataset, MPF achieves $64.18\%$ prediction accuracy with the depth context and $72.40\%$ with the depth and semantic contexts, which improves [13] by $6.22\%$ and $14.44\%$ respectively. As it is generally difficult to model both the pose and size variations of the hand to train the regression forest due to the huge amount of training data, these results give us an insight into another way to generalize well to different users. Fig. 11 and Fig. 15 show some exemplary results obtained with [13], (C) and (E). The 3D joint positions are projected onto the image plane to get their 2D positions and overlaid on the depth images for better illustration.

It is interesting to note that the semantic context and MPF appear to utilize the hand part correlations from different aspects and complement each other. That is, both methods can improve the prediction accuracy separately, while their combination produces further improvement. By utilizing the parsed hand parts, (D) improves [13] by $2.13\%$. On the other hand, with MPF, (C) improves [13] by $6.22\%$, while the best performance, *i.e.* $14.44\%$ improvement, is obtained by the combination of the two. A similar conclusion can also be drawn from the results on the synthesized dataset.

### E. Extension to Multi-Layered Forest

The results on the two datasets demonstrate that the parsed hand parts can effectively improve the joint prediction accuracy at the extra cost to classify all the pixels first. As discussed in Section IV.B, our two-layered forest can be easily extended

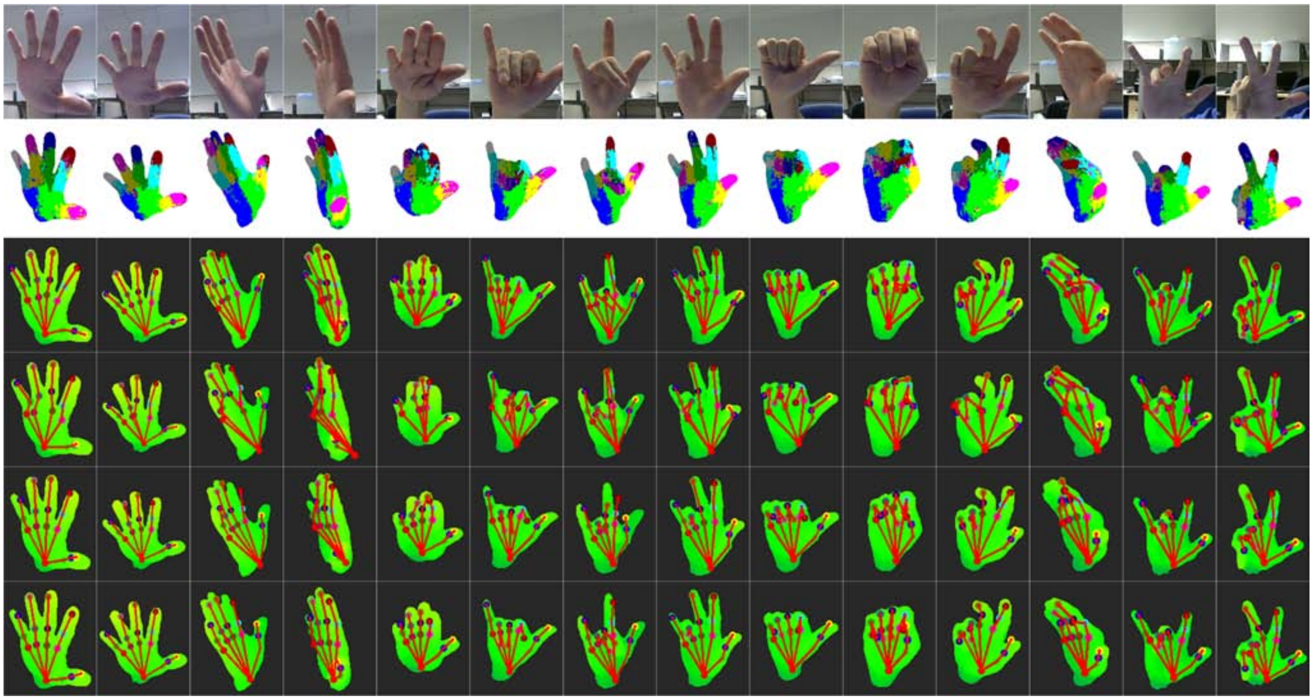


Fig. 11. Comparison of joint position predictions with [13], (C) $\boldsymbol{\mathcal{D}}$ and MPF and (E) $\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MPF on the real dataset. **Second row**: parsed hand parts with the RDF. **Third row**: ground truth joint annotations. **Fourth row**: predictions obtained with [13]. **Fifth row**: predictions obtained with (C). **Sixth row**: predictions obtained with (E).

TABLE III
THE AVERAGE PREDICTION ACCURACIES OF [13], [33] AND OUR METHODS ON THE REAL-WORLD DATASET FOR DIFFERENT $D_T$ (CM).

| $D_T$ | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 | 2.50 | 2.75 | 3.00 | 3.25 | 3.50 | 3.75 | 4.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (A) | 36.73% | 48.26% | 57.96% | 65.84% | 72.17% | 77.00% | 81.04% | 84.16% | 86.63% | 89.02% | 90.78% | 92.32% | 93.70% |
| (B) | 36.29% | 48.33% | 58.09% | 66.19% | 72.56% | 77.43% | 81.43% | 84.59% | 87.29% | 89.56% | 91.42% | 92.95% | 94.13% |
| (C*) | 32.79% | 46.23% | 57.89% | 66.93% | 74.39% | 80.01% | 84.60% | 87.92% | 90.64% | 92.79% | 94.21% | 95.25% | 96.21% |
| (C) | 39.42% | 53.34% | 64.18% | 72.25% | 78.52% | 83.14% | 86.82% | 89.28% | 91.60% | 93.32% | 94.67% | 95.66% | 96.43% |
| (D) | 39.20% | 50.28% | 60.09% | 68.03% | 74.58% | 79.61% | 83.85% | 87.23% | 89.92% | 91.87% | 93.53% | 94.74% | 95.83% |
| (E*) | 42.58% | 56.11% | 67.05% | 75.35% | 81.64% | 85.98% | 89.46% | 92.15% | 94.11% | 95.53% | 96.69% | 97.44% | 97.93% |
| (E) | 48.51% | 61.99% | 72.40% | 79.66% | 84.94% | 88.84% | 91.32% | 93.47% | 95.03% | 96.32% | 97.17% | 97.71% | 98.12% |

to $N_L$ layers, where $N_L = 1$ means no parsed label inputs from the previous layer, *i.e.* the regression forest in [13], and $N_L = 2$ corresponds exactly to our regression forest in Section IV.B, *etc.*. To implement an $N_L$-layered forest, $N_L$ separate forests must be trained sequentially. The first layer forest is trained with the training depth images in the synthesized dataset. Given that the $n$-layer forest is trained, it is then used to parse the training depth images to get the label images, and the depth and newly parsed label images are used to train the $n+1$-layer forest. This procedure continues until all $N_L$ forests are trained.

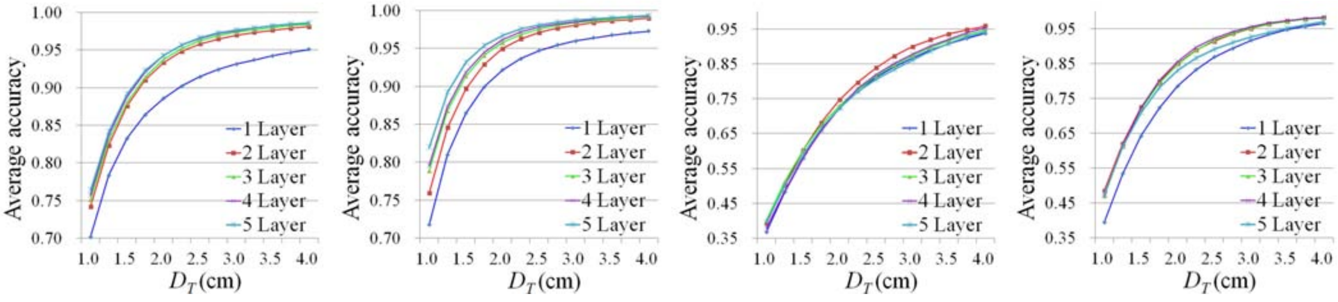
We test the performance of the Mean-Shift algorithm and the MPF algorithm with the per-pixel votes from the $N_L$-layered forest for different $N_L$. Fig. 13 illustrates the results on the synthesized and real datasets for $N_L = 1, ..., 5$. Overall, $N_L = 2$ improves the performance by the largest margin on all the tests. On the synthesized dataset we observe that the prediction accuracy keeps improving when $N_L$ increases, and the hand part classification accuracy shows the same trend, as in Table IV. However, on the real dataset the joint prediction accuracy increases very little for $N_L > 2$ or even begins to drop for large $N_L$. This may indicate that using too many layers of the forest tends to overfit the synthesized training data, and does not necessarily improve the performance for real inputs.

TABLE IV
THE AVERAGE HAND PARSING ACCURACIES ON THE SYNTHESIZED DATASET FOR DIFFERENT $N_L$.

| Layer Num | 1 Layer | 2 Layer | 3 Layer | 4 Layer | 5 Layer |
|---|---|---|---|---|---|
| Accuracy | 81.96% | 86.18% | 88.31% | 89.42% | 90.14% |

### F. Evaluations on the Parameters

In this part we investigate the impact of the following parameters: the number of pixels used for voting $N$, the maximum number of per-pixel votes $J$ and the maximum number of EM iterations. Besides, we also investigate the impact of the amount of the synthesized training data on the performance



(a) Syn. dataset, MS (b) Syn. dataset, MPF (c) Real dataset, MS (d) Real dataset, MPF

Fig. 12. Comparison of the average prediction accuracies on the synthesized and real datasets for different $N_L$.

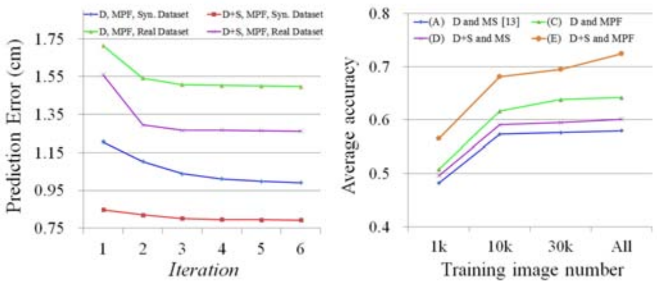
on the real dataset. For each test the uninvestigated parameters take the same values as in Section VI.B.

**Number of voting pixels** $N$**.** Fig. 13 (a-b) illustrates the prediction accuracy changes with respect to different $N$ on the synthesized and real datasets, where we use $D_T = 1.5cm$. All the methods show the same trend on both datasets, *i.e.*, the more pixels used for voting, the better the prediction accuracy. While the computational cost for per-pixel regression is approximately linear with $N$, it should be noted that all the pixels need to be classified in advance if the semantic context is to be used, which leads to a relatively constant time cost regardless of $N$ for hand parsing. This makes the methods “$\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MS”and “$\boldsymbol{\mathcal{D}}+\boldsymbol{\mathcal{S}}$ and MPF”inefficient especially when only a small number of pixels are used for voting.

**Maximum number of per-pixel votes** $J$**.** As defined in Section VI.B, at most three votes are stored for each joint in the leaf nodes by mode seeking in the samples. In the experiment we actually find most leaf nodes contain only one vote for each joint and the average number is about 1.2. Since the forest consists of four trees, the average number of $J$ should be between 4 and 5. Therefore, we investigate the performance of the MPF algorithm for $J = 1, ..., 5$, where $J = 1$ means there is no mode selection procedure as in Formula (11). Fig. 13(c-d) presents the prediction accuracies for $D_T = 1.5cm$ on the synthesized and real datasets, and ”All” means all the retrieved votes are used for the pixels. While $J = 1$ has the poorest performance, the accuracy does not improve much for $J \geq 3$ and $J = 3$ produces an average of $2.3\%$ improvement over $J = 1$.

**Maximum number of EM iterations.** The results in Section VI.D demonstrate the good performance of the MPF algorithm, and we thus investigate the drop of the average joint prediction error with respect to the number of EM iterations in MPF. Fig. 14 (a) illustrates the results on both datasets, which indicates the prediction error decreases quite fast within the first several iterations, especially on the real dataset. Since the prediction error changes very little after about three iterations, we choose the maximum number of EM iterations as three in all the other experiments.

**Synthesized training data size.** In Fig. 14 (b) we compare the prediction accuracy of the four methods on the real-world dataset when the regression forests are trained with different sizes of synthesized data, where $D_T = 1.5cm$. ”All” means the $91.4k$ training images are used, as in the other experiments. The results are consistent with that in Section VI.D, and our methods still outperform [13].



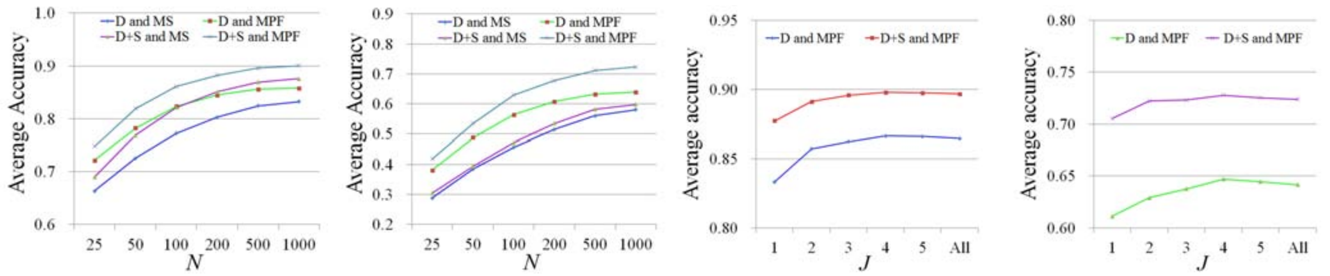
(a) (b)

Fig. 14. (a) Drop of the prediction error with more EM iterations on both the synthesized and real datasets. (b) Prediction accuracy on the real dataset with different numbers of synthesized training data.

## VII. Conclusion

In this paper we present a novel hand pose estimation scheme which utilizes the hand part correlations to improve the joint prediction accuracy from two different aspects. First, we use the parsed hand parts to extract the semantic context to construct more discriminative regression forest, which produces more compact per-pixel votes compared to using the depth image alone. Second, we propose a MPF algorithm to fuse the multimodal per-pixel pose votes subject to the learned hand motion constraints. The MPF algorithm is especially effective in handling the discrepancies between the synthesized training data and real-world inputs, and can be efficiently solved via Expectation-Maximization. Our methods have shown superior performances on both the synthesized dataset and the real-world dataset compared to the baselines [13] and [33]. Besides, we have extended our two-layered forest to even more layers and tested it on both datasets, while the results indicate it tend to overfit the synthesized training data and does not improve the performance for real inputs.

As our current method mainly works on single depth images and the hand motion dynamics are not exploited, we plan to analyze the dynamic constraints of the joint positions and track the hand joint positions in the successive input images to reduce the jitter of the predictions. In addition, we will further enlarge the synthesized training dataset to allow more



(a) $N$ Test, Syn. Dataset (b) $N$ Test, Real Dataset (c) $J$ Test, Syn. Dataset (d) $J$ Test, Real Dataset

Fig. 13. System parameter tests for voting pixel number $N$ and per-pixel vote number $J$. (a-b) Accuracy change of [13] and our three methods for different $N$ on both datasets. (c-d) Accuracy changes of the MPF algorithm for different $J$ on both datasets.

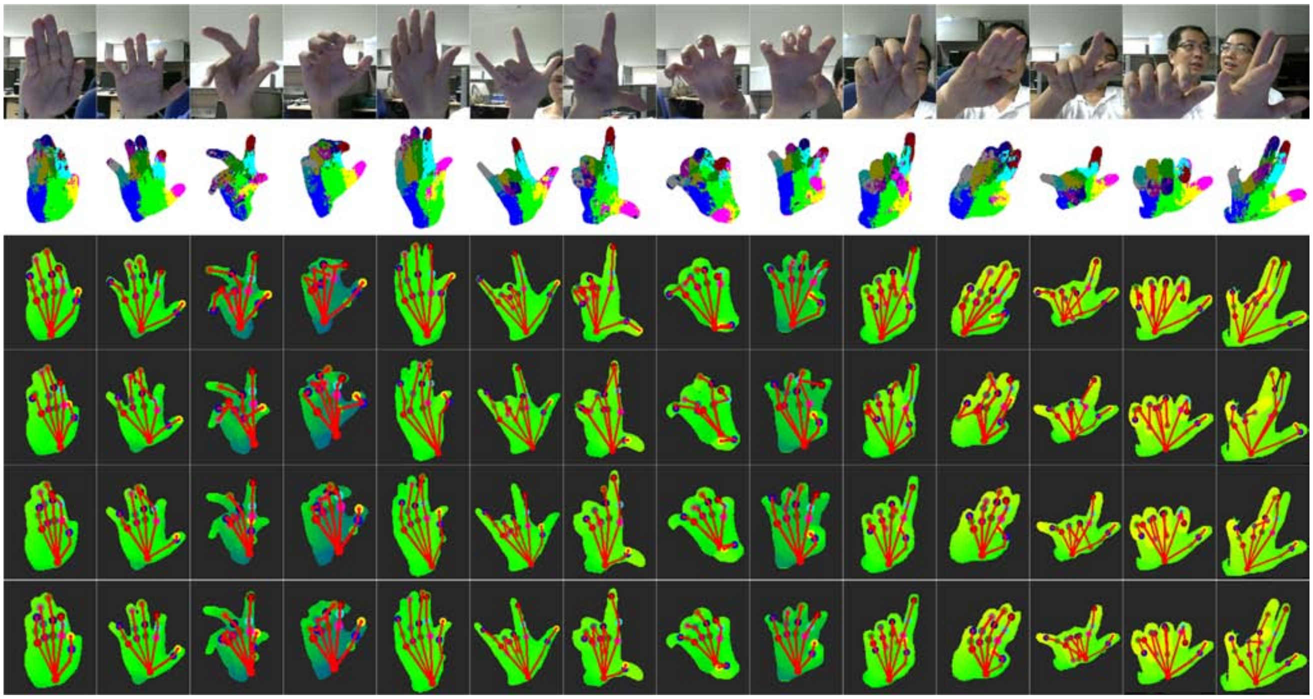


Fig. 15. More results on the real dataset. **Second row**: parsed hand parts with the RDF. **Third row**: ground truth joint annotations. **Fourth row**: predictions obtained with [13]. **Fifth row**: predictions obtained with (C). **Sixth row**: predictions obtained with (E).

complex hand configurations, and develop a practical hand-based human computer interaction system.

## Appendix

Note that Formula (12) is in the quadric form of the variable $\alpha$, we can thus find the optimal solution of $\alpha$ by setting its derivative with respect to $\alpha$ to zero. The derivative of the function to be minimized in Formula (12) with respect to the coefficient $\alpha_x$ takes the following form:

$$\begin{aligned}
\frac{\partial \log P}{\partial \alpha_x} &= 2\sum_k\sum_i w_{ik}\frac{\boldsymbol{e}_{x,k}^T\left(\sum_m \alpha_m \boldsymbol{e}_{m,k}+\mu_k-\boldsymbol{v}_{ijk}^*\right)}{\delta^2} \\
&\propto \sum_k\sum_i w_{ik}\frac{\sum_m \alpha_m \boldsymbol{e}_{x,k}^T \boldsymbol{e}_{m,k}}{\delta^2} - \sum_k\sum_i \frac{w_{ik}\boldsymbol{e}_{x,k}^T\left(\boldsymbol{v}_{ijk}^*-\mu_k\right)}{\delta^2} \\
&\propto \sum_k\sum_m \alpha_m \boldsymbol{e}_{x,k}^T \boldsymbol{e}_{m,k} - \sum_k\sum_i w_{ik}\boldsymbol{e}_{x,k}^T\left(\boldsymbol{v}_{ijk}^*-\mu_k\right) \\
&= \sum_m \alpha_m \sum_k \boldsymbol{e}_{x,k}^T \boldsymbol{e}_{m,k} - \sum_k\sum_i w_{ik}\boldsymbol{e}_{x,k}^T\left(\boldsymbol{v}_{ijk}^*-\mu_k\right) \\
&= \alpha_x - \sum_k\sum_i w_{ik}\boldsymbol{e}_{x,k}^T\left(\boldsymbol{v}_{ijk}^*-\mu_k\right)
\end{aligned}$$

In the above derivation we use the property of the principal components, *i.e.* $\sum_k e_{x,k}^T e_{m,k}$ equals to 1 for $m = x$ and 0 otherwise. By setting $\partial f/\partial \alpha_x = 0$, we can get the solution for the optimization problem:

$$\begin{aligned}
\alpha_x^* &= \sum_k \boldsymbol{e}_{x,k}^T \sum_i w_{ik}\left(\boldsymbol{v}_{ijk}^*-\mu_k\right) \\
&= \sum_k \boldsymbol{e}_{x,k}^T \left[\sum_i w_{ik}\boldsymbol{v}_{ijk}^*-\mu_k\right]
\end{aligned}$$

## Acknowledgment

This research, which is carried out at BeingThere Centre, is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

## References

[1] A. Aristidou and J. Lasenby, Motion Capture with Constrained Inverse Kinematics for Real-Time Hand Tracking, in ISCCSP 2010.

[2] Cyberglove 2, http://www.cyberglovesystems.com/.

[3] R. Y. Wang and J. Popovic, Real-Time Hand-Tracking with a Color Glove, in in Siggraph, 2009.

[4] G. Shakhnarovich, Fast pose estimation with parameter sensitive hashing, in Proc. Int'l Conf. on Computer Vision, 2003.

[5] H. Guan, J. S. Chang, L. Chen, R. S. Feris and M. Turk, Multi-view Appearance-based 3D Hand Pose Estimation, in Proc. CVPR, 2006.

[6] Y. Wu and T. S. Huang, Capturing Articulated Human Hand Motion: A Divide-and-Conquer Approach, in Proc. Int'l Conf. on Computer Vision, 1999.

[7] I. Oikonomidis, N. Kyriazis and A. A. Argyros, Efficient model-based 3D tracking of hand articulations using Kinect, in BMVC, 2011.

[8] B. Stenger, A. Thayananthan, P. H. S. Torr and R. Cipolla, Model-Based Hand Tracking Using a Hierarchical Bayesian Filter, in IEEE Trans. Pattern Analysis and Machine Intelligence, 2006.

[9] 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.

[10] C. Keskin, F. Kirac, Y. E. Kara and L. Akarun, Real-time Hand Pose Estimation Using Depth Sensors, in Proc. Int'l Conf. on Computer Vision, 2011

[11] 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.

[12] J. Xu, Y. Wu and A. Katsaggelos, Part-based Initialization for Hand Tracking, in ICIP, 2010.

[13] R. Girshick, J. Shotton, P. Kohli, A. Criminisi and A. Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images, in Proc. Int'l Conf. on Computer Vision, 2011.

[14] R. Rosales and S. Sclaroff, Inferring body pose without tracking body parts, in Proc. of CVPR, 2000.

[15] J. Romero, H. Kjellstrom and D. Kragic, Monocular Real-Time 3D Articulated Hand Pose Estimation, in Proc. Int'l Conf. on Humanoid Robots, 2009.

[16] P. F. Felzenszwalb, Pictorial Structures for Object Recognition, in Int'l Journal on Computer Vision, 2005.

[17] C. S. Chua, H. Guan and Y. K. Ho, Model-based 3D hand posture estimation from a single 2D image, in Image and Vision Computing, 2002.

[18] B. Stenger, P. R. S. Mendonqa and R. Cipolla, Model-Based 3D Tracking of an Articulated Hand, in Proc. CVPR, 2001.

[19] J. Y. Lin, Y. Wu and T. S. Huang, 3D Model-Based Hand Tracking Using Stochastic Direct Search Method, in FG, 2004.

[20] A. Baak, M. Mller, G. Bharaj, H. P. Seidel and C. Theobal, A data-driven approach for real-time full body pose reconstruction from a depth camera, in Proc. Int'l Conf. on Computer Vision, 2011.

[21] S. Sridhar, A. Oulasvirta and C. Theobalt, Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data, in Proc. Int'l Conf. on Computer Vision, 2013.

[22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp and M. Finocchio, Real-time Human Pose Recognition in Parts from Single Depth Images, in Proc. CVPR, 2011.

[23] X. Wei, P. Zhang and J. Chai, Accurate Real-time Full-body Motion Capture Using a Single Depth Camera, in Siggraph Asia, 2012.

[24] L. Breiman. Random forests. Mach. Learning, 45(1):5-32, 2001.

[25] D. Comaniciu and P. Meer, Mean shift: A Robust Approach toward Feature Space Analysis, in IEEE Trans. PAMI, vol. 24, no. 5, pp. 603-619, 2002.

[26] H. Liang, J. Yuan and D. Thalmann, Parsing the Hand in Depth Images, in IEEE Trans. Multimedia, 2014.

[27] C. Xu and L. Cheng, Efficient Hand Pose Estimation from a Single Depth Image, in ICCV, 2013.

[28] K. C. Sim, Discriminative Product-of-Expert Acoustic Mapping for Cross-lingual Phone Recognition, in IEEE Workshop on Automatic Speech Recognition & Understanding, 2009.

[29] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm, in Journal of the Royal Statistical Society, vol. 39, no. 1, pp. 1-38, 1977.

[30] Y. Yao and Y. Fu, Real-Time Hand Pose Estimation from RGB-D Sensor, in ICME 2012.

[31] J. Xu, J. Yuan and Y. Wu, Multimodal Partial Estimates Fusion, in ICCV, 2009.

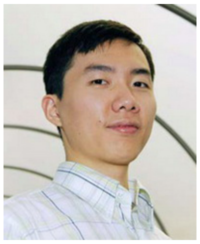
[32] M. Dantone, J. Gall, C. Leistner and L. V. Gool, Human Pose Estimation using Body Parts Dependent Joint Regressors, in Proc. of CVPR, 2013.

[33] F. Kirac, Y. E. Kara and L. Akarun, Hierarchically constrained 3D hand pose estimation using regression forests from single frame depth data, in Pattern Recognition Letters, Sep. 2013.

[34] http://www.softkinetic.com/products/iisumiddleware.aspx

[35] H. Liang, J. Yuan, D. Thalmann and Z. Zhang, Model-based Hand Pose Estimation via Spatial-temporal Hand Parsing and 3D Fingertip Localization, in the Visual Computer Journal, vol. 29, no. 6-8, pp. 837-848, June 2013.

**Hui Liang** received the B.S. and M.S. degrees in Electronics and Information Engineering from Huazhong University of Science & Technology (HUST), Wuhan, China, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree at Nanyang Technological University, Singapore. His research interests include computer vision and hand-based human computer interaction.



**Junsong Yuan** is a Nanyang Assistant Professor at Nanyang Technological University, leading the video analytics program at School of EEE. He obtained his PhD from Northwestern University, M.Eng. from National University of Singapore, and B.Eng. from special class for the gifted young at Huazhong University of Science and Technology, China. He has co-authors over 100 technical papers and filed 3 US patents and 2 provisional US patents. He received Outstanding EECS Ph.D. Thesis award from Northwestern University and Doctoral Spotlight Award from IEEE Conf. Computer Vision and Pattern Recognition Conference (CVPR'09). He is Organizing Chair of Asian Conf. on Computer Vision (ACCV'14), and co-chairs workshops at CVPR'12'13 and ICCV'13. He also serves as Area Chair for WACV'14, ACCV'14, ICME'14, and is an associate editor of Visual Computer Journal and Journal of Multimedia. He gives tutorials at IEEE ICIP'13, FG'13, ICME'12, SIGGRAPH VRCAI'12, and PCM'12.

**Prof. Daniel Thalmann** is with the Institute for Media Innovation at the Nanyang Technological University in Singapore. He is a pioneer in research on Virtual Humans. He has been the Founder of VRlab) at EPFL, Switzerland, Professor at The University of Montreal and Visiting Professor/ Researcher at CERN, University of Nebraska, University of Tokyo, and National University of Singapore. He is coeditor-in-chief of the Journal of Computer Animation and Virtual Worlds, and member of the editorial board of 6 other journals. Daniel Thalmann was Program Chair and CoChair of several conferences including IEEE VR, ACM VRST, ACM VRCAI, CGI, and CASA. Daniel Thalmann has published more than 500 papers in Graphics, Animation, and Virtual Reality. He is coeditor of 30 books, and coauthor of several books including 'Crowd Simulation' (second edition 2012). He received his PhD in Computer Science in 1977 from the University of Geneva and an Honorary Doctorate from University Paul- Sabatier in Toulouse, France, in 2003. He also received the Eurographics Distinguished Career Award in 2010 and the 2012 Canadian Human Computer Communications Society Achievement Award.