Real-time Articulated Hand Pose Estimation using Semi-supervised Transductive Regression Forests

Danhang Tang
Imperial College London
London, UK
d.tang11@imperial.ac.uk

Tsz-Ho Yu
University of Cambridge
Cambridge, UK
thy23@cam.ac.uk

Tae-Kyun Kim
Imperial College London
London, UK
tk.kim@imperial.ac.uk

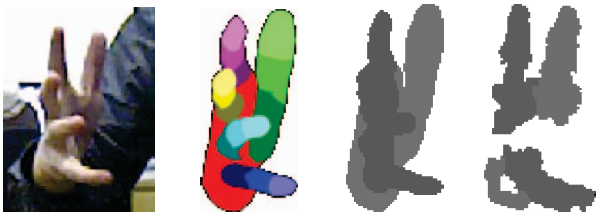
Abstract

This paper presents the first semi-supervised transductive algorithm for real-time articulated hand pose estimation. Noisy data and occlusions are the major challenges of articulated hand pose estimation. In addition, the discrepancies among realistic and synthetic pose data undermine the performances of existing approaches that use synthetic data extensively in training. We therefore propose the Semi-supervised Transductive Regression *(STR) forest which learns the relationship between a small, sparsely labelled realistic dataset and a large synthetic dataset. We also design a novel data-driven, pseudo-kinematic technique to refine noisy or occluded joints. Our contributions include: (i) capturing the benefits of both realistic and synthetic data via transductive learning; (ii) showing accuracies can be improved by considering unlabelled data; and (iii) introducing a pseudo-kinematic technique to refine articulations efficiently. Experimental results show not only the promising performance of our method with respect to noise and occlusions, but also its superiority over state-of-the-arts in accuracy, robustness and speed.*

1. Introduction

Articulated hand pose estimation shares a lot of similarities with the popular 3-D body pose estimation. Both tasks aim to recognise the configuration of an articulated subject with a high degree of freedom. While latest depth sensor technology has enabled body pose estimation in real-time [2, 24, 12, 26], hand pose estimation still requires improvement. Despite their similarities, proven approaches in body pose estimation cannot be repurposed directly to hand articulations, due to the unique challenges of the task:

(1) Occlusions and viewpoint changes. Self occlusions are prevalent in hand articulations. Compared with limbs in body pose, fingers perform more sophisticated articulations. Different from body poses which are usually upright and frontal [9], different viewpoints can render different depth images despite the same hand articulation.



(a) RGB (b) Labels (c) Synthetic (d) Realistic

Figure 1: The ring finger is missing due to occlusions in (d), and the little finger is wider than the synthetic image in (c).

(2) Noisy hand pose data. Body poses usually occupy larger and relatively static regions in the depth images. Hands, however, are often captured in a lower resolution. As shown in Fig. 1, missing parts and quantisation error is common in hand pose data, especially at small, partially occluded parts such as finger tips. Unlike sensor noise and depth errors in [12] and [2], these artefacts cannot be repaired or smoothed easily. Consequently, a large discrepancy is observed between synthetic and realistic data.

Moreover, manually labelled realistic data are extremely costly to obtain. Existing state-of-the-arts resort to synthetic data [16], or model-based optimisation [8, 15]. Nonetheless, such solutions do not consider the realistic-synthetic discrepancies, their performances are hence affected. Besides, the noisy realistic data make joint detection difficult, whereas in synthetic data joint boundaries are always clean and accurate.

Addressing the above challenges, we present a novel *Semi-supervised Transductive Regression* (STR) forest. This process is known as *transductive transfer learning* [21]: A transductive model learns from a *source domain*, *e.g.* synthetic data; on the other hand, it applies *knowledge transform* to a different but related *target domain*, *e.g.* realistic data, in the testing stage. As a result, it benefits from

the characteristics of both domain: The STR forest not only captures a wide range of poses from synthetic data, it also achieves promising accuracy in challenging environments by learning from realistic data. In addition, we design an efficient pseudo-kinematic joint refinement algorithm to handle occluded and noisy articulations. The STR forest is also semi-supervised, learning the noisy appearances of realistic data from both labelled and unlabelled datapoints. Moreover, generic pose estimation is facilitated by a wide range of poses from synthetic data, using a data-driven pose refinement scheme.

As far as we are aware, the proposed method is the first semi-supervised and transductive articulated hand pose estimation framework. The main contributions of our work are threefold:

(1) Realistic-Synthetic fusion: Considering the issue of noisy inputs, we propose the first transductive learning algorithm for 3-D hand pose estimation that captures the characteristics of both realistic and synthetic data.

(2) Semi-supervised learning: The proposed learning algorithm utilises both labelled and unlabelled data, improving estimation accuracy while keeping a low labelling cost.

(3) Data-driven pseudo-kinematics: The limitations of traditional Hough forest [11] against occlusions is alleviated by learning a novel data-driven pseudo-kinematic algorithm.

2. Related Work

Hand pose estimation Earlier approaches for articulated hand pose estimation are diversified, such as coloured markers [6], probabilistic line matching [1], multi-camera network [13] and Bayesian filter with Chamfer matching [25]. We refer the reader to [10] for a detailed survey of earlier hand pose estimation algorithms.

Model-based tracking methods are popular among recent state-of-the-arts. Hypotheses are generated from a visual model, *e.g*. a 3-D hand mesh. Hand poses are tracked by fitting the hypotheses to the test data. For example, De La Gorce *et al*. [8] use a hand mesh with detailed simulated texture and lighting. Hamer *et al*. [14] address strong occlusions using local trackers at separate hand segments. Ballan *et al*. [3] infer finger articulations by detecting salient points. Oikonomidis *et al*. [20] estimate hand poses in real-time from RGB-D images using particle swarm optimisation. Model-based approaches inherently handle joint articulations and viewpoint changes. However, their performances depend on the previous pose estimations, output poses may drift away from groundtruth when error accumulates over time.

Discriminative approaches learn a mapping from visual features to the target parameter space, such as joint labels [24] or joint coordinates [12]. Instead of using a predefined visual model, discriminative methods learn a pose estimator from a labelled training dataset. Although discriminative methods have proved successful in real-time body pose estimation from depth sensors [24, 12, 2, 26], they are less common than model-based approaches with respect to hand pose estimation. Recent discriminative algorithms for hand pose estimation include approximate nearest neighbour search [23, 27] and hierarchical random forests [16].

Discriminative methods rely heavily on the quality of training data. A large labelled dataset is necessary to model a wide range of poses. It is also costly to label sufficient realistic data for training. As a result, existing approaches resort to synthetic data by means of computer graphics [23, 16], which suffers from the realistic-synthetic discrepancies. On the positive side, discriminative methods are frame-based such that there exists no track drifting issue.

Kinematics Inverse kinematics is a standard technique in model-based and tracking approaches for both body [28, 22] and hand poses estimation [8, 15, 25]. Lacking an articulated visual model, only a few discriminative methods consider the physical properties of hands. For instance, Girshick *et al*. [12] estimate body poses using a simple range heuristic, yet it is inapplicable to hand pose due to self-occlusions. Wang *et al*. [27] detect joint using a coloured-glove and match them from the groundtruth database.

Transfer Learning Transductive transfer learning is often employed when training data of the *target domain* are too costly to obtain. It has seen various successful applications [21], still it has not been applied in articulated pose estimation. In this work, realistic-synthetic fusion are realised by extending the idea of Bronstein *et al*. [5] to the proposed STR forest, where the training algorithm preserves the associations between cross-domain data pairs.

Semi-supervised and Regression Forest Various semi-supervised forest learning algorithms have been proposed. Navaratnam *et al*. [19] sample unlabelled datapoints to improve Gaussian processes for body pose estimation. Shotton *et al*. [7] measure data compactness to relate labelled and unlabelled datapoints. Leistner *et al*. [18, 17] design a margin metric to evaluate with unlabelled data. On the other side, regression forest is widely adopted in body pose estimation, *e.g*. [12, 26]. The STR forest adaptively combines the aforementioned semi-supervised and regression forest learning techniques in a single frame work.

3. Methodology

The concept of STR learning is illustrated in Fig. 2. For each viewpoint, training data are collected from a partially labelled target domain (realistic depth images) and a fully labelled source domain (synthetic depth images). These domains are explicitly related by establishing associations from the labelled target datapoints to their corresponding source datapoints, as shown in the figure.

The STR learning algorithm introduces several novel

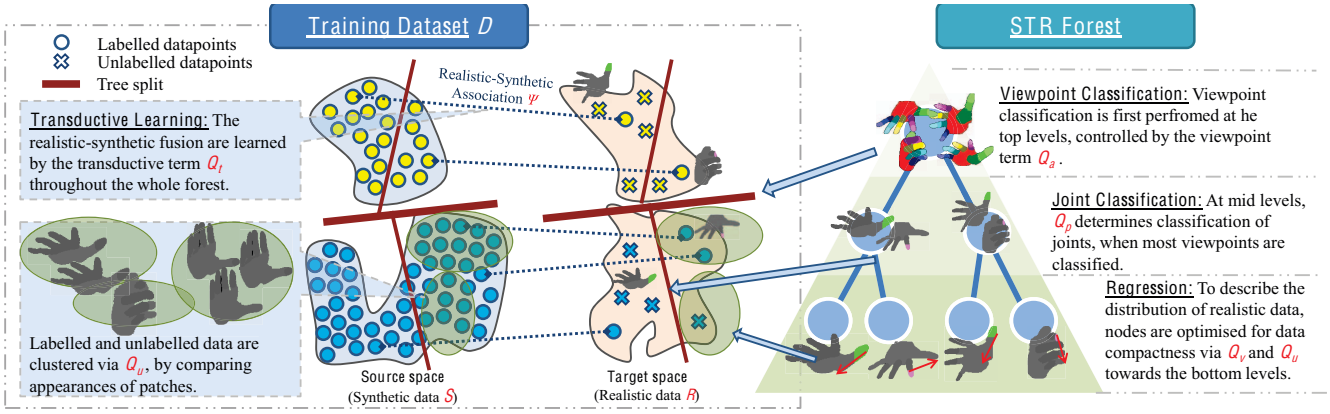


Figure 2: The proposed STR learning model.

techniques to the traditional Hough/regression forest [11]. Firstly, transductive realistic-synthetic associations are preserved, such that the matched data are passed down to the same node. Secondly, the distributions of labelled and unlabelled realistic data are modelled jointly in the proposed STR forest using unsupervised learning. Thirdly, viewpoint changes are handled alongside with hand poses using an adaptive hierarchical classification scheme. Finally, we also propose an *data-driven, kinematic-based* pose refinement scheme.

3.1. Training datasets

The training dataset $\mathcal{D} = \{\mathcal{R}_l, \mathcal{R}_u, \mathcal{S}\}$ consists of both realistic data $\mathcal{R}$ and synthetic data $\mathcal{S}$. A small potion of $\mathcal{R}$ is labelled, where the labelled and the remaining unlabelled subsets are denoted by $\mathcal{R}_l$ and $\mathcal{R}_u$ respectively. All datapoints in $\mathcal{S}$ are labelled with groundtruths. The subset of labelled data in $\mathcal{D}$ is defined as $\mathcal{L} = \{\mathcal{R}_l, \mathcal{S}\}$.

Each datapoint in $\mathcal{D}$ is an image patch sampled randomly from foreground pixels in the training images. The size of a patch is 64×64 which is comparable to the patches in [24]. The number of datapoints roughly equals 5% of foreground pixels in the depth images.

Every datapoint in $\mathcal{R}_l$ or $\mathcal{S}$ is assigned to a tuple of labels $(\mathbf{a}, p, \mathbf{v})$. Viewpoint of a patch is represented by the roll, pitch and yaw angles, which are quantised into 3, 5 and 9 steps respectively. The view label $\mathbf{a} \in \mathcal{A} : \mathbb{N}^3$ indicates one of the 135 quantised viewpoints. A datapoint is also given the class label of its closest joint, $p \in \{1 \dots 16\}$, similar to [24]. Furthermore, every labelled datapoint contains 16 vote vectors $\mathbf{v} \in \mathbb{R}^{3\times 16}$ from the patch's centroid to the 3-D locations of all 16 joints as in [11].

Realistic-synthetic associations are established through matching datapoints in $\mathcal{R}_l$ and $\mathcal{S}$, according to their 3-D joint locations. The realistic-synthetic association $\Psi : \mathcal{R}_l, \mathcal{S} \rightarrow \{1, 0\}$ is defined as below:

$$\Psi(r \in \mathcal{R}_l, s \in \mathcal{S}) = \begin{cases} 1 \text{ when } r \text{ matches } s \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

3.2. STR Forest

Building upon the hybrid regression forest by Yu *et al.* [29], the STR forest performs classification, clustering and regression on both domains in one pose estimator, instead of performing each task in separate forests. We grow N_t decision trees by recursively splitting and passing the current training data to two child nodes. The split function of a node is represented by a simple two-pixel test as in Shotton *et al.* [24]. Instead of using a typical metric such as information gain or label variance [7], we propose two new quality functions. The quality function is selected at random between Q_{apv} and Q_{tss} for training in Equation 2.

$$\begin{cases} Q_{apv} = \alpha Q_a + (1-\alpha)\beta Q_p + (1-\alpha)(1-\beta) Q_v \\ Q_{tss} = Q_t^{\omega} Q_u \end{cases} \quad (2)$$

where Q_{apv} is a combined quality function for learning classification-regression decision trees, and Q_{tss} enables transductive and semi-supervised learning. Given the training data $\mathcal{D} = \{\mathcal{R}_l, \mathcal{R}_u, \mathcal{S}\}$, the quality functions are defined as below.

Viewpoint classification term Q_a: Traditional *information gain* is used to evaluate the classification performance of all the viewpoint labels $\mathbf{a}$ in dataset $\mathcal{L}$ [4]. Since this term is applied on the top of the hierarchy, a large amount of training samples needs to be evaluated. Inspired by [12], reservoir sampling is employed to avoid memory restriction and speed up training.

Patch classification term Q_p: Similar to Q_a, it is the information gain of the joint labels p in $\mathcal{L}$. It measures the performance of classifying individual patch in $\mathcal{L}$. Thus, Q_a and Q_p optimises the decision trees by *classifying* $\mathcal{L}$ their viewpoints and joint labels.

Regression term Q_v: This term learns the *regression* aspect of the decision trees by measuring the compactness of vote vectors. Given the set of vote vectors $\mathcal{J}(\mathcal{L})$ in $\mathcal{L}$, regression term Q_v is defined as:

$$Q_v = \left[1 + \frac{|\mathcal{L}_{lc}|}{|\mathcal{L}|}\Lambda(\mathcal{J}(\mathcal{L}_{lc})) + \frac{|\mathcal{L}_{rc}|}{|\mathcal{L}|}\Lambda(\mathcal{J}(\mathcal{L}_{rc}))\right]^{-1} \quad (3)$$

where $\mathcal{L}_{lc}$ and $\mathcal{L}_{rc}$ are the training data that pass down the left and right child nodes respectively, and $\Lambda(\cdot) = \mathrm{trace}(\mathrm{var}(\cdot))$ is the trace of variance operator in [11]. Q_v increases with compactness in vote space and converges to 1 when all votes in a node are identical.

Unsupervised term Q_u: The appearances the target domain, *i.e*. realistic data, are modelled in an *unsupervised* manner. Assuming appearances and poses are correlated under the same viewpoint, Q_u evaluates the appearance similarities of all realistic patches $\mathcal{R}$ within a node:

$$Q_u = \left[1 + \frac{|\mathcal{R}_{lc}|}{|\mathcal{R}|}\Lambda(\mathcal{R}_{lc}) + \frac{|\mathcal{R}_{rc}|}{|\mathcal{R}|}\Lambda(\mathcal{R}_{rc})\right]^{-1}. \quad (4)$$

Since the realistic dataset is sparsely labelled, *i.e*. $|\mathcal{R}_u| \gg |\mathcal{R}_l|$, $\mathcal{R}_u$ are essential for modelling the target distribution. In order to speed up the learning process, Q_u can be approximated by down-sampling the patches in $\mathcal{R}$.

Transductive term Q_t: Inspired from cross-modality boosting in [5], the *transductive* term Q_t preserves the cross-domain associations Ψ as the training data pass down the trees:

$$Q_t = \frac{|\{r,s\} \subset \mathcal{L}_{lc}| + |\{r,s\} \subset \mathcal{L}_{rc}|}{|\{r,s\} \subset \mathcal{L}|} \quad (5)$$
$$\forall\, \{r,s\} \subset \mathcal{L} \text{ where } \Psi(r,s) = 1$$

The transductive term Q_t is hence the ratio of preserved association after a split.

Adaptive switching$\{\alpha, \beta, \omega\}$ A decision tree mainly performs classifications at the top levels, its training objective is switched adaptively to regression at the bottom levels (Fig. 2). Let $\Delta(\cdot)$ be the difference between the highest posterior of a class and the second highest posterior in a node. $\Delta_{\mathbf{a}}(\mathcal{L})$ and $\Delta_p(\mathcal{L})$ denote the margin measures of viewpoint labels $\mathbf{a}$ and joint labels p in $\mathcal{L}$. They measure the purity of a node with respect to viewpoint and patch label.

$$\alpha = \begin{cases} 1 \text{ if } \Delta_{\mathbf{a}}(\mathcal{L}) < t_\alpha \\ 0 \text{ otherwise} \end{cases} \quad \beta = \begin{cases} 1 \text{ if } \Delta_p(\mathcal{L}) < t_\beta \\ 0 \text{ otherwise} \end{cases} \quad (6)$$

where t_α and t_β are tunable thresholds that determine the structure of the output decision trees; both thresholds are 0.9 in this work. The parameter ω controls the relative importance of Q_t to Q_u.

3.3. Data-driven Kinematic Joint Refinement

Since the proposed STR forest considers joint as independent detection targets, it lacks structural information to recover poorly detected joints when they are occluded or missing from the depth image. Without having an explicit hand model as in most model-based tracking methods, we designed a data-driven, kinematic-based method to refine joint locations from the STR forest. A large hand pose database $\mathcal{K}$ is generated, such that $|\mathcal{K}| \gg |\mathcal{S}|$, in order to obtain the maximum pose coverage. The pose database $\mathcal{K}$ is generated using the same hand model as in the synthetic dataset $\mathcal{S}$, but $\mathcal{K}$ contains only the joint coordinates.

The procedures for computing the data-driven kinematic model $\mathcal{G}$ is described in Algorithm 1. $\mathcal{G}$ contains viewpoint-specific distributions of joint locations represented as a N-part *Gaussian mixture models (GMM)*.

Algorithm 1: Data-driven Kinematic Models.

Data: A joint dataset $\mathcal{K} \subset \mathbb{R}^{3\times 16}$ that contains synthetic joint locations, where $|\mathcal{K}| \gg |\mathcal{S}|$.
Result: A set of viewpoint-dependent distributions $\mathcal{G} = \{\mathcal{G}_{\mathbf{i}} | \forall \mathbf{i} \in \mathcal{A}\}$ of global poses.

1 Split $\mathcal{K}$ with respect to viewpoint label $\mathcal{A}$, such that $\mathcal{K} = \{\mathcal{K}_1 \dots \mathcal{K}_{|\mathcal{A}|}\}$
2 **forall the** $\mathbf{i} \in \mathcal{A}$ **do**
3 Learn a N-part GMM $\mathcal{G}_i$ of the dataset $\mathcal{K}_i$: $\mathcal{G}_{\mathbf{i}} = \{\mu_{\mathbf{i}}^1 \dots \mu_{\mathbf{i}}^n \dots \mu_{\mathbf{i}}^N; \Sigma_{\mathbf{i}}^1 \cdots \Sigma_{\mathbf{i}}^n \cdots \Sigma_{\mathbf{i}}^N\}$, where $\mu_{\mathbf{i}}^n$ and $\Sigma_{\mathbf{i}}^n$ denote the mean and *diagonal* variance of the n-th Gaussian component in $\mathcal{G}_{\mathbf{i}}$ of view $\mathbf{i}$.

3.4. Testing

Joint Classification and Detection. Patches are extracted densely from the testing depth images. Similar to other decision forests, each patch passes down the STR forest to obtain the viewpoint $\hat{\mathbf{a}}$ and vote vectors $\hat{\mathbf{v}}$. The patch vote for all 16 joint locations according to $\hat{\mathbf{v}}$.

Kinematic Joint Refinement. The objective of kinematic joint refinement is to compute the final joint locations $\mathbf{Y} = \{\mathbf{y}_1 \dots \mathbf{y}_{\mathbf{j}} \dots \mathbf{y}_{16} | \forall \mathbf{y} \in \mathbb{R}^3\}$. Derived from the meanshift technique in [12], the distributions of votes vectors are evaluated as stated below: The set of votes received by the $\mathbf{j}$-th joint is fitted a 2-part GMM $\hat{\mathcal{G}}_{\mathbf{j}} = \{\hat{\mu}_{\mathbf{j}}^1, \hat{\Sigma}_{\mathbf{j}}^1, \hat{\rho}_{\mathbf{j}}^1, \hat{\mu}_{\mathbf{j}}^2, \hat{\Sigma}_{\mathbf{j}}^2, \hat{\rho}_{\mathbf{j}}^2\}$, where $\hat{\mu}$, $\hat{\Sigma}$, $\hat{\rho}$ denote the mean, variance and weight of the Gaussian components respectively. Fig. 3 visualises the two Gaussian components obtained from fitting the voting vectors of a joint.

A strong detection forms one compact cluster of votes, which leads to a high weighting and low variance in one of the Gaussians. On the contrary, a weak detection usually contains scattered votes, indicated by separated means with

similar weights. The **j**-th joint is of high-confidence when the Euclidean distance between $\hat{\mu}^1_{\mathbf{j}}$ and $\hat{\mu}^2_{\mathbf{j}}$ is smaller than a threshold t_q. For any high-confident j-th joint, the output location $\mathbf{y_j}$ is the mean of the dominating Gaussian in $\hat{\mathcal{G}}_{\mathbf{j}}$.

$$\mathbf{y_j} = \begin{cases} \hat{\mu}^1_{\mathbf{j}} & \text{if } ||\hat{\mu}^1_{\mathbf{j}} - \hat{\mu}^2_{\mathbf{j}}||^2_2 < t_q \text{ and } \hat{\rho}^1_{\mathbf{j}} \geq \hat{\rho}^2_{\mathbf{j}} \\ \hat{\mu}^2_{\mathbf{j}} & \text{if } ||\hat{\mu}^1_{\mathbf{j}} - \hat{\mu}^2_{\mathbf{j}}||^2_2 < t_q \text{ and } \hat{\rho}^1_{\mathbf{j}} < \hat{\rho}^2_{\mathbf{j}} \end{cases} \quad (7)$$

Subsequently, final locations of all high-confidence joints are determined. The joint refinement process is performed on the other low-confidence joints.

The nearest neighbour of the set of high-confidence joints are searched from its corresponding joint means $\{\mu^1_{\hat{\mathbf{a}}} \dots \mu^N_{\hat{\mathbf{a}}}\}$ in the kinematic model $\mathcal{G}_{\hat{\mathbf{a}}}$ using least squares with a direct similarity homography $\mathbf{H}$. Only the high-confident joint locations are used in the above nearest neighbour matching; the low-confident joint locations are masked out. Given the nearest Gaussian component $\{\mu^{nn}_{\hat{\mathbf{a}}}, \Sigma^{nn}_{\hat{\mathbf{a}}}\}$ of the high-confidence joints, each remaining low-confidence joint $\mathbf{y_j}$ are refined:

$$\{\tilde{\mu}, \tilde{\Sigma}\} = \underset{\{\mu,\Sigma\} \in \{\hat{\mu}^1_{\mathbf{j}}, \hat{\Sigma}^1_{\mathbf{j}}\}, \{\hat{\mu}_2\, \hat{\Sigma}_2\}}{\operatorname{argmin}} ||\mathbf{H}\mu - \mu^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}]||^2_2 \quad (8)$$

where $\{\tilde{\mu}, \tilde{\Sigma}\}$ is the Gaussian in $\hat{\mathcal{G}}_{\mathbf{j}}$ that is closer to the corresponding **j**-th joint location in $\{\mu^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}] : \mathbb{R}^3, \Sigma^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}] : \mathbb{R}^{3\times3}\}$. The final output of a low-confidence joint $\mathbf{y}_l$ is computed by merging the Gaussians in Equation 9.

$$\mathbf{y_j} = \left(\tilde{\Sigma}^{-1} + (\Sigma^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}])^{-1}\right)^{-1} \left(\tilde{\Sigma}\mu^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}] + \Sigma^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}]\tilde{\mu}\right) \quad (9)$$

Fig. 3 illustrates the process of refining a low-confidence joint. The index proximal joint is occluded by the middle finger as seen in the RGB image; the 2-part GMM $\hat{\mathcal{G}}_{\mathbf{j}}$ is represented by the red crosses (mean) and ellipses (variance). The final output is computed by merging the nearest neighbour obtained from $\mathcal{G}$, *i.e.* $\{\mu^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}], \Sigma^{nn}_{\hat{\mathbf{a}}}[\mathbf{j}]\}$ (the green Gaussian), and the closer Gaussian in $\hat{\mathcal{G}}_{\mathbf{j}}$ (the left red Gaussian). The procedures of refining output poses $\mathbf{Y}$ are stated in Algorithm 2.

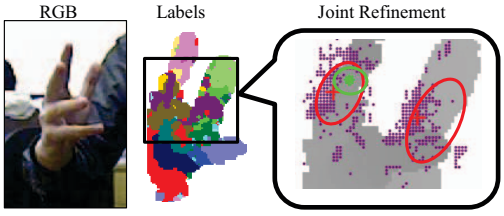


Figure 3: The proposed joint refinement algorithm.

Algorithm 2: Pose Refinement

Data: Vote vectors obtained from passing down the testing image to the STR forest.
Result: The output pose $\mathbf{Y} : \mathbb{R}^{3\times16}$.

foreach *Set of voting vectors for the* **j***-th joint* **do**
  Learn a 2-part GMM $\hat{\mathcal{G}}_{\mathbf{j}}$ of the voting vectors.
  if $||\hat{\mu}^1_{\mathbf{j}} - \hat{\mu}^2_{\mathbf{j}}||^2_2 < t_q$ **then**
    The **j**-th joint is a high-confidence joint.
    Compute the **j**-th joint location. (Equation 7)
  else
    The **j**-th joint is a low-confidence joint.
Find the Gaussian $\{\mu^{nn}_{\hat{\mathbf{a}}}, \Sigma^{nn}_{\hat{\mathbf{a}}}\}$ by finding the nearest neighbour of the high-confidence joints in $\mathcal{G}_{\hat{\mathbf{a}}}$.
Update the remaining low-confidence joint locations. (Equation 8 and 9)

4. Experiments

4.1. Evaluation dataset

Synthetic training data $\mathcal{S}$ were rendered using an articulated hand model(as shown in Figure 4). Each finger was controlled by a bending parameter, such that only the articulations that can be performed by real hands were considered. Different hand poses are generated by sampling the bending parameters randomly. Moreover, in order to capture hand shape variations, finger and palm shapes and sizes were randomised mildly in $\mathcal{S}$. As a result, the dataset $\mathcal{S}$ contains 2500 depth images per viewpoint, the size of $\mathcal{S}$ is $2500 \times 135 = 337.5K$.

Realistic data $\mathcal{R}$ were captured using a Asus Xtion depth sensor. This dataset contains 600 images per viewpoint, hence the size of $\mathcal{R}$ is $81K$. Not more than 20% of data in $\mathcal{R}$ were labelled. The number of labelled sample $|\mathcal{R}_l|$ is around $10K$. Since labels can be reused for the rotationally symmetric images (same yaw and pitch, different roll), only around $1.2K$ of data were hand-labelled.

For $\mathcal{R}_l$, visible joints were annotated manually using 3-D coordinates but occluded joints were annotated using the (x, y) coordinates only. Associations Ψ and the remaining z-coordinates in $\mathcal{R}_l$ were computed by matching visible joint locations with $\mathcal{S}$ using least squares with a direct similarity transform constraint. Consequently, each datapoint in $\mathcal{R}_l$ was paired with its closest match $\mathbf{x}_{syn} \in \mathcal{S}$, and its occluded z coordinates were approximated by the corresponding z coordinates of $\mathbf{x}_{syn}$. With joint locations as mean, each joint can be model as a 3D truncated Gaussian distribution, where variances can be defined according to hand anatomy. Foreground pixels are clustered into one of these distributions and therefore assigned with labels p.

For experiments, three different sequences (A,B and C) are captured and labelled with 450, 1000 and 240 frames respectively. Sequence A has only one viewpoint, B demonstrates viewpoint variation and C has more abrupt changes in both viewpoint and scale. In the experiments, 3 trees are trained with maximum depth varying from 16 to 24, as

in [24]. Since the training dataset contain a large amount of positive samples, a few trees are enough to average out noisy results. From the experimental results, adding extra trees did not improve the pose estimation accuracy.

4.2. Single View Experiment

The proposed approach was evaluated under the frontal view scenario, comparing with the traditional regression forest in [11] as a baseline. Since there was only one viewpoint in testing sequence A, Q_a in Equation 2 did not affect the experimental results. Performances of algorithms are measured by their pixel-wise classification accuracy per joint, similar to [24], hence only Q_p,Q_v, Q_t and Q_u were utilised in this experiment.

Fig. 4 shows the classification accuracy of the experiment. It demonstrates the strengths of realistic-synthetic fusion and semi-supervised learning. Accuracy of baseline method was improved by simply including both domains in training without any algorithmic changes. Transductive learning (Q_t) substantially improved the accuracy, particularly for the finger joints which were less robust in the baseline algorithms. By coupling realistic data with synthetic data, the transductive term Q_t effectively learns the discrepancies between the domains, which is important in recognising noisy and strongly occluded fingers. Some joints are often mislabelled as other "stronger" joints after transductive learning, *e.g*. joints L3 and I1. Nevertheless, the data-driven joint refinement scheme significantly improved the performance of these joints.

4.3. Multi-view Experiment

In the multi-view experiment, the proposed approach was compared with the state-of-the-art by FORTH [20] under a challenging multi-view scenario. Quantitative and qualitative evaluations were performed to provide a comprehensive comparison of the methods.

Hand articulations are estimated from the multi-view testing sequences (sequence B and C) by both of the methods. Since FORTH require manual initialisation, the testing sequences used are designed such that they start with the required initialisation pose and position, making a fair comparison. Same as [20], performances of pose estimation were measured by *joint localisation error*.

Quantitative Results Fig. 5 shows the average localisation errors of the two testing sequences. It also demonstrates a representative of error graphs from a stable joint (palm, P) and a difficult joint (index finger tip, $I3$). The proposed STR forest, with the data-driven kinematic joint refinement, outperforms FORTH in all three statistics, especially for the finger tip joints that are noisy and frequently occluded. Even though a few large estimation errors are observed, our frame-based approach is able to recover from errors quickly.

Sequence C further confirms the major advantage of our approach over its tracking-based counterpart—In the first 200 frames, with kinematic joint refinement, STR forest approach performs just slightly better than FORTH. However, localisation errors in FORTH accumulate after an abrupt change and have not been recovered since then. As model-based tracking approaches rely on previous results to optimise the current hypothesis iteratively, estimation errors amass over time. On the other hand, frame-based discriminative approaches consider each frame as an independent input, enabling fast error recovery at the expense of a smooth and continuous output.

The proposed joint refinement scheme increases the joint estimation accuracy in general, as shown in Fig. 5. Some of the large classification errors, *e.g*. Fig. 5c, are fixed after applying joint refinement. It implies that the joint refinement process not only improves the accuracy of joint, but also avoids incorrect detections by validating the output of STR forest with kinematic constraints.

Qualitative Analysis The experimental results are also visualised in Fig. 6 for qualitative evaluation. Fig. 6a to e show the pose estimation results from different view points. Fig. 6f shows a frame at the beginning of test sequence B, both FORTH and our method obtains accurate hand articulations. Nonetheless, the performance of FORTH declines rapidly in the middle of the sequence when its tracking is lost and failed to recognise Fig. 6g, yet our approach still gives correct results. Conceptually, the proposed method is similar to Keskin *et al*. [16], where both approaches describe a coarse-to-fine hand pose estimation algorithm. However, our method is based on a unified, single-layered STR forest, which is trained on realistic and synthetic data, while Keskin *et al*. [16] is multi-layered, using only synthetic data in training. The STR forest achieves real-time performance, as it runs at about 25FPS on an Intel I7 PC without GPU acceleration, whilst the FORTH algorithm runs at 6FPS on the same hardware configuration plus NVidia GT 640.

5. Conclusions

This paper presents the first semi-supervised transductive approach for articulated hand pose estimation. Despite their similarities with body pose estimation, techniques for articulated hand pose is still far from mature, primarily due to the unique issues of occlusion and noise issues in hand pose data. On the other hand, the discrepancies between realistic and synthetic data also undermine the performances of state-of-the-arts.

Addressing the aforementioned issues, we propose a novel discriminative approach, STR forest, to estimate hand articulations using both realistic and synthetic data. With transductive learning, the STR forest recognises a wide range of poses from a small number of labelled realistic data. Semi-supervised learning is applied to fully utilise

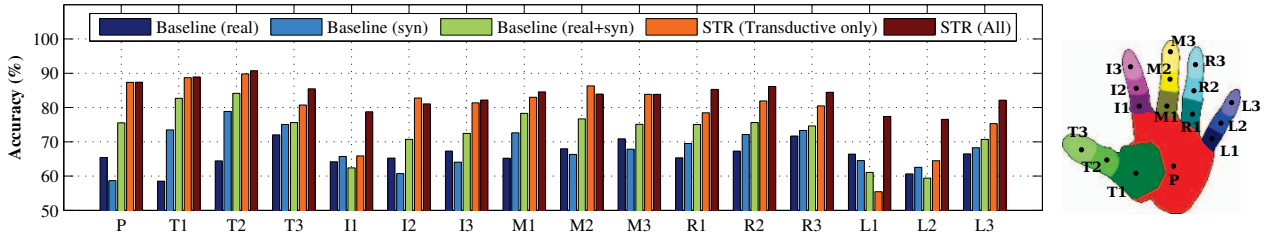
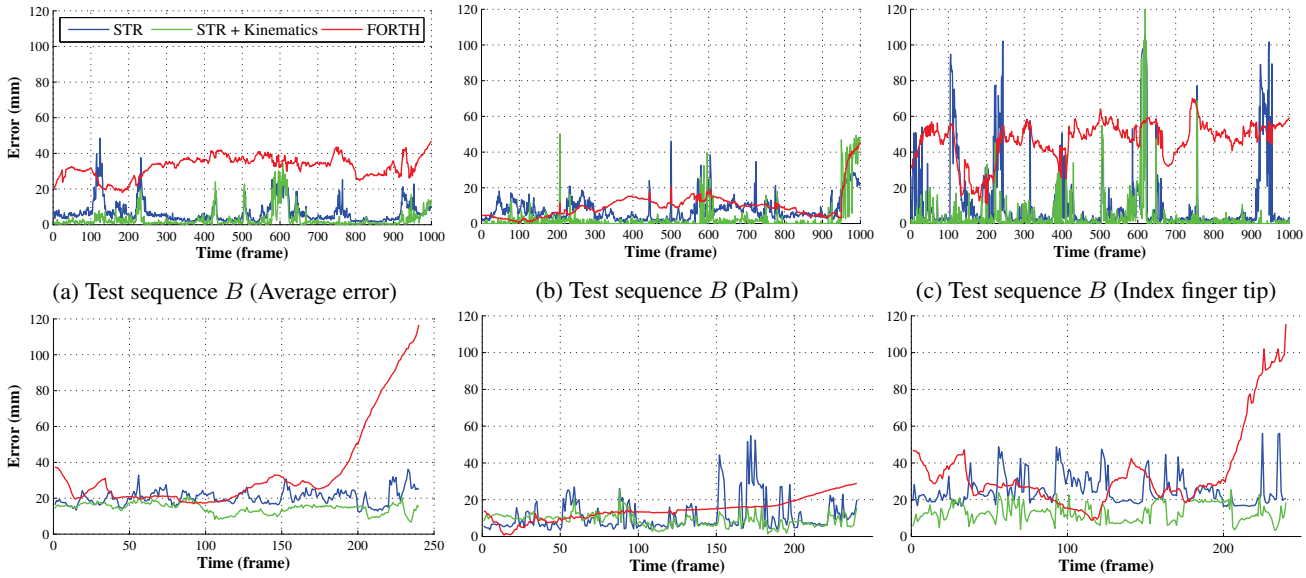


Figure 4: Joint classification accuracy of the single view sequence.



(a) Test sequence B (Average error)

(b) Test sequence B (Palm)

(c) Test sequence B (Index finger tip)

(d) Test sequence C (Average error)

(e) Test sequence C (Palm)

(f) Test sequence C (Index finger tip)

Figure 5: Quantitative results of the multi-view experiment.

the sparsely labelled realistic dataset. Besides, we also present a data-driven pseudo-kinematic technique, as means to improve the estimation accuracy of occluded and noisy hand poses. Quantitative and qualitative results demonstrate promising results in hand pose estimation from noisy and occluded data. It also attains superior performances and speed compared with state-of-the-art.

Acknowledgement

This work was supported by the Samsung Advanced Institute of Technology (SAIT).

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. *CVPR*, 2003.

[2] A. Baak, M. Müller, 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.

[3] 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.

[4] L. Breiman. Random forests. *Machine Learning*, 2001.

[5] M. M. Bronstein, E. M. Bronstein, F. Michel, and N. Paragios. Data fusion through crossmodality metric learning using similaritysensitive hashing. In *CVPR*, 2013.

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

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

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

[9] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *IJCV*, 2012.

[10] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 2007.

[11] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *PAMI*, 2011.

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

Figure 6: **Qualitative results of the multi-view experiment.** (a)-(e) are taken from sequence B and (f)-(g) are from sequence C. Hand regions are cropped from the originals for better visualisation (135×135 pixels for (a)-(e), 165×165 pixels for (f)-(g). The resolution of the original images are 640×480. Joint labels follow the color scheme in Figure 4.

[13] H. Guan, J. S. Chang, L. Chen, R. Feris, and M. Turk. Multi-view appearance-based 3d hand pose estimation. In *CVPR Workshops*, 2006.

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

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

[16] 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.

[17] C. Leistner, M. Godec, S. Schulter, A. Saffari, M. Werlberger, and H. Bischof. Improving classifiers with unlabeled weakly-related videos. In *CVPR*. IEEE, 2011.

[18] C. Leistner, A. Saffari, J. Santner, and H. Bischof. Semi-supervised random forests. In *ICCV*, 2009.

[19] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *ICCV*, pages 1–8, 2007.

[20] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, 2011.

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 2010.

[22] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixe, M. Muller, H.-P. Seidel, and B. Rosenhahn. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *ICCV*, 2011.

[23] J. Romero, H. Kjellström, and D. Kragic. Monocular real-time 3d articulated hand pose estimation. In *Humanoids*, 2009.

[24] 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.

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

[26] M. Sun and J. Shotton. Conditional regression forests for human pose estimation. *CVPR*, 2012.

[27] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 2009.

[28] A. Yao, J. Gall, and L. Gool. Coupled action recognition and pose estimation from multiple views. *IJCV*, 2012.

[29] T.-H. Yu, T.-K. Kim, and R. Cipolla. Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *CVPR*, 2013.