# Hand Pose Estimation and Hand Shape Classification using Multi–layered Randomized Decision Forests

Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun

Boğaziçi University, Computer Engineering Department, Istanbul, Turkey
keskinc@cmpe.boun.edu.tr,{kiracmus,yunus.kara,akarun}@boun.edu.tr

**Abstract.** Vision based articulated hand pose estimation and hand shape classification are challenging problems. This paper proposes novel algorithms to perform these tasks using depth sensors. In particular, we introduce a novel randomized decision forest (RDF) based hand shape classifier, and use it in a novel multi–layered RDF framework for articulated hand pose estimation. This classifier assigns the input depth pixels to hand shape classes, and directs them to the corresponding hand pose estimators trained specifically for that hand shape. We introduce two novel types of multi–layered RDFs: Global Expert Network (GEN) and Local Expert Network (LEN), which achieve significantly better hand pose estimates than a single–layered skeleton estimator and generalize better to previously unseen hand poses. The novel hand shape classifier is also shown to be accurate and fast. The methods run in real–time on the CPU, and can be ported to the GPU for further increase in speed.

## 1 Introduction

Hand gestures are a natural part of human interaction. In addition to their complementary roles in speech based interaction, they play a primary role when speech is absent, as in sign language based interaction. Attempts to use the hand gesture modality in human computer interaction (HCI) has intensified research efforts for articulated hand pose tracking and hand shape recognition in the last decade.

Two developments have recently accelerated implementations of HCI using human body and hand gestures: The first is the release and widespread acceptance of the Kinect depth sensor. With its ability to generate depth images in very low illumination conditions, this sensor makes the human body and hand detection and segmentation a simple task. The second development is the emergence of fast discriminative approaches using simple depth features coupled with GPU implementation; enabling real time human body pose extraction [1, 2].

The approaches for human body pose detection using the Kinect camera use a variety of techniques: Shotton *et al.* [1] use a large amount of labeled synthetic images to train a randomized decision forest(RDF) [3] for the task or body part recognition. In a later study, Girschick *et al.* [2] use the same methodology, but

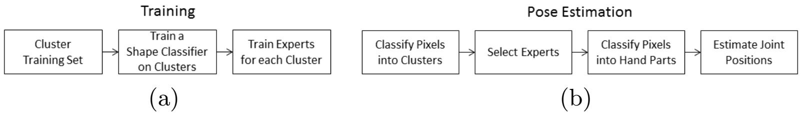

Training
Cluster Training Set
Train a Shape Classifier on Clusters
Train Experts for each Cluster
(a)
Pose Estimation
Classify Pixels into Clusters
Select Experts
Classify Pixels into Hand Parts
Estimate Joint Positions
(b)

**Fig. 1.** The flowchart of the training and pose estimation processes.

let each pixel vote for joint coordinates; and learn the voting weights from data. [4] relies on pre–captured motion exemplars to estimate the body configuration as well as the semantic labels of the point cloud. [5] uses an upper body model, and tracks it using a hierarchical particle filter. Although these ideas may be extended to extracting the 3D pose of the hand, the problem is made more difficult by the increased pose variability and self-occlusion.

Hand pose estimation studies have initially relied on 2D models [6]. Although pose variability and occlusion limit the success of 2D approaches, successful models relying on partial models have been defined [7]. Approaches using articulated 3D models have relied on color images [8–12], as well as the use of multiple cameras or time-of-flight sensors [13, 14]. These approaches have achieved good performances even in the presence of occlusions and pose changes, though their time performances have limited their application in real time HCI applications. In a recent study Oikonomidis et al. [15] present a solution that makes use of both depth and color images. They propose a generative single hypothesis model-based pose estimation method. They use particle swarm optimization for solving the 3D hand pose recovery problem, and report accurate and robust tracking in near real–time (15 fps), with a GPU based implementation.

Our previous 3D hand pose estimation attempt in [16] adopted the methodology of body pose estimation used in [1]. In this work, large synthetic datasets were generated using a realistic hand model and an RDF was trained to assign each pixel a hand part label. Then, we applied the mean shift algorithm to estimate the centers of hand parts to form a hand skeleton. This method was shown to be robust to noise and worked in real time.

In this work, we first present a novel hand shape classifier, which is an adaptation of the RDF based hand pose estimation method of [16] to hand shapes. We call this new type of RDF a shape classification forest (SCF). Then, we use SCF in designing a multi–layered RDF network to tackle the articulated hand pose estimation problem. The idea is to divide the problem of modelling a large dataset into simpler sub–problems by clustering the dataset first. Then, each such cluster corresponds to a hand shape that can be recognized with an SCF, and a separate hand pose estimator is trained on each cluster, forming skeleton experts. A similar approach is used in [17], which detects the hands in the first layer and then classifies hand shapes in the second layer. Our method classifies hand shapes in the first layer and then estimates the hand pose in the second layer.

A flowchart is given in Figure 1. Training consists of three phases: First, the training set is clustered according to hand skeleton similarity. Then, an SCF is trained that can assign cluster labels to input images. Finally, RDFs are trained on each cluster, forming the experts. In the hand pose estimation process, there are four main steps: First, the SCF assigns a cluster label to each pixel. Then, either experts corresponding to the majority of the pixels are selected, or each pixel is assigned to its respective expert. The selected experts form a forest and infer hand part labels. Finally, the part labels are used to estimate the joint positions, forming the hand skeleton.

The performance of the novel shape classification and pose estimation methods are evaluated on real and synthetic images, respectively. In particular, SCF is tested on the publicly available ASL dataset of [18] and is shown to achieve a success rate of 97.8%. Multi-user ASL letter recognition is a difficult task, and comparable good results to ours have been reported in the literature on other datasets. [19] provides a good review of ASL letter recognition on depth data. We compare the success of the multi–layered RDF with the results in [16]. Whereas the method of [16] achieves a per pixel classification rate of 68% on a large synthetic dataset, the multi–layered method achieves a classification rate of 91.2%.

The paper is organized as follows: In Section 2 we describe the novel hand shape classification method. Then, we show how this model can be used to design a multi–layered expert network in Section 3. The multi–layered networks GEN and LEN and their detailed training procedures are also explained in this Section. Section 4 discusses the experiments: Parameter selection, the datasets used in evaluation, shape classification and hand pose estimation results. Finally, we conclude the paper and discuss future work in Section 5.

## 2 Hand Shape Classification

Hand shape classification is the act of assigning a class label $c$ to an input image $I$ representing a certain configuration of the hand. We propose an RDF model that uses scale invariant features extracted from depth images to infer the hand shape class. Inspired by the part classification approach of [1] and [16], we formulate an RDF for hand shape classification, in which every pixel votes for a hand shape label instead of a hand part label. The final class label is determined by majority vote.

### 2.1 Decision Trees

Decision trees consist of split nodes, which are the internal nodes used to analyze the data, and leaf nodes, which are the terminal nodes used to infer the posterior probability of the class label, based on statistics collected from past data. Each split node sends the incoming input to one of its children according to the test result. The test associated with a split node is usually of the form:

$$f_n(F_n) < T_n \tag{1}$$

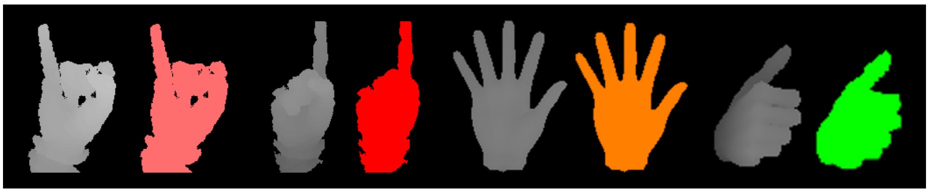


**Fig. 2.** SCT training images: The first four images are real depth images and their labels, and the rest of the images are synthetic depth images and their labels.

where $f_n(F_n)$ is a function on features and $T_n$ is a threshold, for the split node $n$. $f_n(F_n) = T_n$ defines a possibly complicated hyper–surface in the feature space, and the test determines, on which side of the hyper–surface the input is. The input is injected at the root node, which is forwarded by the split nodes according to the test results, and the posterior probabilities associated with the leaf node that is reached are used to infer the class label. Hence, the training of a decision tree involves determining the tests and collecting statistics from a training set in a supervised manner.

In the case of a randomized decision tree, $f_n$ operates on a subset of the features selected during training. This is done by randomly selecting multiple function candidates and choosing the one that best splits the data. This approach is particularly useful, when the feature space is large.

### 2.2 Shape Classification Forest

An SCF consists of trees, which we call the shape classification trees (SCT). The input to an SCT is a depth image $I$, and a pixel location $\boldsymbol{x}$. The output is a set of posterior probabilities for each shape class label $C_k$. The model is trained on a dataset consisting of depth image–class label pairs. Unlike the RDFs in [1] and [16], an image is given a single hand shape label. Exemplary input images are given in Figure 2. The first four images are real depth images retrieved from Kinect, and the rest of the hand images are synthetic. Each color corresponds to a different hand shape class.

SCT uses the same features as in [1] and [16]. Given a depth image $I(\boldsymbol{x})$, where $\boldsymbol{x}$ denotes location, we define a feature $F_{\boldsymbol{u},\boldsymbol{v}}(I,\boldsymbol{x})$ as follows:

$$F_{\boldsymbol{u},\boldsymbol{v}}(I,\boldsymbol{x}) = I(\boldsymbol{x} + \frac{\boldsymbol{u}}{I(\boldsymbol{x})}) - I(\boldsymbol{x} + \frac{\boldsymbol{v}}{I(\boldsymbol{x})}) \qquad (2)$$

The offsets $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors relative to the pixel in question, and normalized according to the depth at $\boldsymbol{x}$. This ensures that the features are 3D translation invariant. Note that, they are neither rotation nor scale invariant, and the synthetic training images should be generated accordingly. The depth of background pixels and the exterior of the image are taken to be a large constant.

Each split node is associated with a pair of offsets $\boldsymbol{u}$ and $\boldsymbol{v}$ and a depth threshold $\tau$. The data is split into two sets as follows:

$$C_L(\boldsymbol{u},\boldsymbol{v},\tau) = \{(I,\boldsymbol{x})|F_{\boldsymbol{u},\boldsymbol{v}}(I,\boldsymbol{x}) < \tau\} \tag{3}$$

$$C_R(\boldsymbol{u},\boldsymbol{v},\tau) = \{(I,\boldsymbol{x})|F_{\boldsymbol{u},\boldsymbol{v}}(I,\boldsymbol{x}) >= \tau\} \tag{4}$$

Here, $C_L$ and $C_R$ are the mutually exclusive sets of pixels assigned to the left and right children of the split node, respectively.

In the training phase, each split node randomly selects a set of features, partitions the data accordingly and chooses the feature that splits the data best. Each split is scored by the total decrease in the entropy of the label distribution of the data:

$$S(\boldsymbol{u},\boldsymbol{v},\tau) = H(C) - \sum_{s\in\{L,R\}} \frac{|C_s(\boldsymbol{u},\boldsymbol{v},\tau)|}{|C|} H(C_s(\boldsymbol{u},\boldsymbol{v},\tau)) \tag{5}$$

where $H(K)$ is the Shannon entropy estimated using the normalized histogram of the labels in the sample set $K$. The process ends when the leaf nodes are reached. Each leaf node is then associated with the normalized histogram of the labels estimated from the pixels reaching it.

Starting at the root node of each SCT, each pixel $(I,\boldsymbol{x})$ is assigned either to the left or the right child until a leaf node is reached. There, each pixel is assigned a set of posterior probabilities $P(c_i|I,\boldsymbol{x})$ for each hand shape class $c_i$. For the final decision, the posterior probabilities estimated by all the trees in the ensemble are averaged:

$$P(c_i|I,\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N} P_n(c_i|I,\boldsymbol{x}) \tag{6}$$

where $N$ is the number of trees in the ensemble, and $P_n(c_i|I,\boldsymbol{x})$ is the posterior probability of the pixel estimated by the tree with index $n$. We call this ensemble a Shape Classification Forest (SCF). To determine a final hand shape label, the posterior probabilities of every pixel in the input image are averaged, and the label that maximizes this term is selected:

$$c^* = \arg\max_{c_i} \frac{1}{M}\sum_{m=1}^{M} P(c_i|I,\boldsymbol{x_m}) \tag{7}$$

where $M$ is the number of foreground pixels in the input image, and $c^*$ is the determined hand shape class label.

## 3 Hand Pose Estimation

The RDF proposed in [16] used for hand pose estimation is structurally similar to the shape classifier introduced in Section 2.2. While the SCF classifies hand shapes, the pose estimating RDFs classify each pixel into hand parts. Hence, the ground truth labels should indicate hand parts instead of poses. We call this model a Part Classification Forest (PCF) to make the distinction explicit.

### 3.1 Pose Estimation using PCF

As in the case of SCFs, classification of a pixel $(I, \boldsymbol{x})$ is performed by starting at the root node and assigning the pixel either to the left or the right child until a leaf node is reached. Each leaf node is associated with a set of posterior probabilities $P(c_i|I, \boldsymbol{x})$ for each hand part label $c_i$, which are estimated from the normalized histograms during training. Some examples of ground truth labels are given in Figure 3. The final decision for the hand part label is made by averaging the posterior probabilities estimated by all the trees in the ensemble:

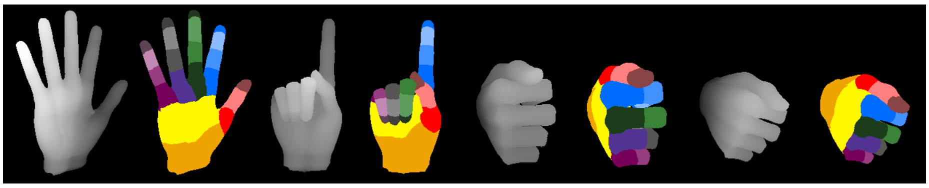
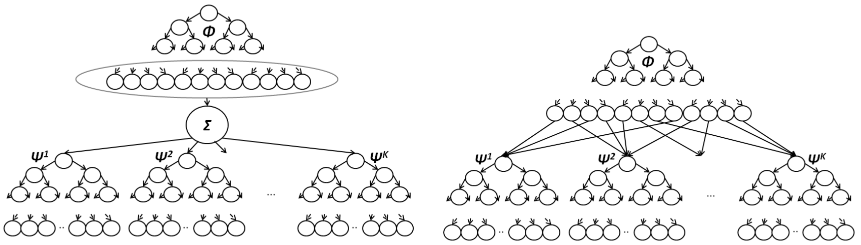


**Fig. 3.** Four examples of synthetic depth images and the corresponding ground truth labels that are used to train the pose estimation RDFs.

$$P(c_i|I, \boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} P_n(c_i|I, \boldsymbol{x}) \tag{8}$$

where $N$ is the number of trees in the ensemble. In PCFs, the final phase of the SCF classification, i.e. averaging over all the pixels, is replaced with a joint position estimation step. The mean shift local mode finding algorithm [20] is used to estimate the mode of the probability density of each class label, formed by placing weighted Gaussian kernels on each pixel. The bandwidth of each hand part is manually selected based on the size of each hand part. The weight of the kernel is set to be the pixel's posterior probability $P(c_i|I, \boldsymbol{x})$ corresponding to the class label $c_i$, times the square of the depth of the pixel, which is an estimate of the area the pixel covers, indicating its importance. Starting from a point estimate, the mean shift algorithm uses a gradient ascent approach to locate the nearest maximum point. As the maxima are local, several different starting points are used and the one converging to the highest maximum is selected. Finally, a decision regarding the visibility of the joint is made by thresholding the highest score reached during the mean shift phase. The joint positions estimated in this manner are then connected according to their configuration in the hand skeleton, forming the final pose estimate.

In [16], a single PCF is used to estimate the hand pose. This means that, each tree is trained on a huge dataset, and therefore, they need to be sufficiently complex, which translates into larger memory requirements. If we limit the depth of the trees, the accuracy is dropped.



(a) (b)

**Fig. 4.** Multi–layered RDF networks. $\Phi$ depicts the SCF, and $\Psi^i$ depicts the expert PCF corresponding to the cluster $C_i$. a) Global Expert Network: The experts are selected according to the pose label. b) Local Expert Network: Each pixel is sent to its own expert.

### 3.2 Pose Estimation using a Multi–layered RDF Network

Here, we propose a novel multi–layered approach to tackle the complexity problem. The idea is to reduce the complexity of the model by dividing the training set into smaller clusters, and to train PCFs on each of these compact sets. Thus, the PCFs need to model only a small amount of variation, requiring smaller memory. These *experts* accurately model a specific subset of the data, and infer significantly better pose estimates. The main challenge is to direct the input towards the correct experts, which can be done by training an SCF on the clusters.

The SCF assigns a cluster label to each pixel in an input image. This information can be used in two different ways: i) a pose label for the entire image can be estimated via voting; ii) individual pixels can be sent to the corresponding expert PCFs according to their labels. We call these the Global Expert Network (GEN) and Local Expert Network (LEN) respectively. These networks are illustrated in Figure 4.

The training of the multi–layered model requires three steps: i) clustering of the training data, ii) training an SCF with the clusters as shapes as in Section 2.2, iii) training separate PCFs on each cluster.

**Clustering training data** Spectral clustering is employed to form the pose clusters in the hand skeleton configuration space. Spectral clustering methods are based on the Min–Cut algorithm, which partitions graph nodes by minimizing a certain cost associated with each edge in the graph [21]. This is a binary clustering method, which can be used to hierarchically cluster data into multiple clusters. A related algorithm has been proposed by Meila and Shi [22], which can estimate multiple clusters. In this method, a similarity matrix is formed for the samples to be clustered, where each entry $S_{ij}$ in the matrix corresponds to the similarity of samples $i$ and $j$. As the similarity measure, the reciprocal of distance can be used.

The distance between two skeletal configurations is taken to be the weighted sum of the absolute differences of each angle pair. The clustering procedure is as follows:

$$D_{ij} = ||\mathbf{W}(\boldsymbol{v_i} - \boldsymbol{v_j})||_1 \tag{9}$$

$$\alpha = \max(\mathbf{D}) \tag{10}$$

$$S_{ij} = 1 - \frac{1}{\alpha} D_{ij} \tag{11}$$

$$R_{ii} = \sum_j S_{ij} \tag{12}$$

$$\mathbf{P} = \mathbf{S}\mathbf{R}^{-1} \tag{13}$$

Here, $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ are the vectors formed by all the angles of a skeleton. $\mathbf{W}$ is a diagonal matrix, such that $W_{ii}$ is the weight of the angle $i$. $\alpha$ is the maximum amount of distance recorded in $\mathbf{D}$. $\mathbf{S}$ is the similarity matrix formed by normalizing $\mathbf{D}$ by $\alpha$ and subtracting each element from 1. Then, each column $c_i$ of $\mathbf{S}$ is normalized using the sum of elements in row $r_i$ to form the matrix $\mathbf{P}$. The eigenvectors corresponding to the $m$ largest eigenvalues of this matrix are then found in the form of a $N \times m$ matrix. Each row of this matrix is an $m$ dimensional representative of one of the $N$ samples. To create the final clusters, the rows are clustered using the $k$–means method.

**Training and Pose Estimation** We train an SCF ($\Phi$) on the clusters of the dataset $D^k$, $k = 1, \ldots, K$, using the method of Section 2.2. Next, $K$ PCFs are trained, depicted as $\Psi^k$, on the clusters $D^k$. In the case of GEN, $\Phi$ classifies the image into one of the $K$ clusters by assigning a label $C_i$ using the method in Section 2.2. Instead of estimating a single label $c^*$ to the image, the highest three average posterior probabilities are calculated:

$$\rho_j = \frac{1}{M} \sum_{m=1}^{M} P(c_j | I, \boldsymbol{x_m}) \tag{14}$$

Without loss of generality, we call the highest posterior probabilities $\rho_1$, $\rho_2$ and $\rho_3$, and the corresponding labels $C_1$, $C_2$ and $C_3$. The input image is sent to the experts $\Psi^1$, $\Psi^2$ and $\Psi^3$ to estimate the part labels for each pixel. The results of the experts are weighted with $\rho_1$, $\rho_2$ and $\rho_3$:

$$P(c_j | I, \boldsymbol{x}) = \sum_{i=1}^{3} \rho_i P(c_j | I, \boldsymbol{x}, \Psi^i) \tag{15}$$

Finally, the process continues with the joint estimation step explained in Section 3.1.

In the case of LEN, $\Phi$ follows the procedure in Section 2.2 until Equation 7, where the values are averaged over all the pixels. Then, each pixel is sent separately to an expert selected using to the posterior probability $P(c_i | I, \boldsymbol{x})$. Finally, the experts classify the pixels into hand parts as before.

The difference between GEN and LEN is that, GEN does not take local context into account when it directs the input to the experts, as it averages the posterior probabilities over all the pixels. LEN, on the other hand, makes use of local modes of the posterior probability distribution. The result is that GEN is more robust to noise, and LEN can generalize better to previously unseen data.

## 4 Experiments

In this work, three different models are introduced, namely SCF for shape classification, and the multi–layer RDF networks GEN and LEN for hand pose estimation. Several experiments have been conducted to verify the efficacy of these models.

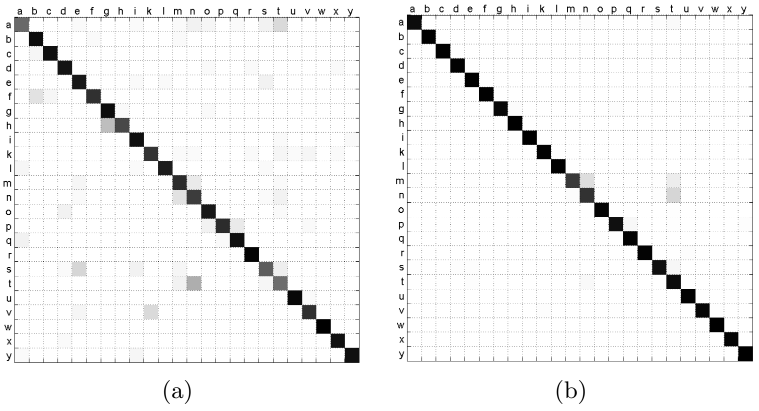
### 4.1 Shape Classification Performance

The accuracy of the SCF is tested on a dataset consisting of 65K depth images corresponding to 24 of the 26 ASL letters (omitting non–static letters $j$ and $z$) performed by five subjects [18]. Pugeault et al. reported their results on this dataset using both leave–one–subject–out cross–validation and by using half of the set for training and half for validation. For the former validation technique, we employed four trees of depth 20, and sampled 1000 features at each node. SCF achieved a recognition rate of 84.3%, while [18] report 47%. For the latter, an SCF consisting of a single tree reached 97.8%, compared to 69% using only depth features, and 75% using both depth and color features [18]. Even though SCF is a side product of the GEN and LEN models, it is accurate and fast. Moreover, SCF can be trained using real images, whereas synthetic images are needed to train GEN and LEN. We provide the confusion matrices in Figure 5.

### 4.2 Hand Pose Estimation Performance

To test the accuracy of the GEN and LEN models, a synthetic dataset consisting of 60K images is generated. The network parameters are optimized through a grid search. The important parameters of SCFs and PCFs are: i) the tree height $h$; ii) limits of feature vectors $\boldsymbol{u}$ and $\boldsymbol{v}$; iii) limit of the depth threshold $\tau$; iv) the number of samples at each node $\epsilon$ v) and $K$, the number of clusters. We use $h = 20$, $\boldsymbol{u}, \boldsymbol{v} \in [-45, 45]$, $\tau \in [0, 60]$, $\epsilon = 1000$, and experiment with different values of $K$. Training takes around 4000 sec per tree on a quad core CPU.

Two new factors introduced by the multi–layered framework are the number of clusters $K$ and the diagonal matrix $\mathbf{W}$ used in estimating the pairwise distances of two skeletons. Increasing $K$ has a positive effect on the accuracy of both layers. By increasing $K$, we ensure that only similar configurations fall into the same clusters. This actually simplifies the clustering process, decreases the complexity of the SCF layer. Increasing $K$ also reduces the complexity of the PCFs in the second layer, thereby making the experts more accurate. For



(a)

(b)

**Fig. 5.** Confusion matrix for the ASL letter classification task using SCF on the Pugeault dataset with 24 letters and five subjects [18]. a) Leave–one–subject–out with a success rate of 84.3%. b) Half training–half validation, with a success rate of 97.8%. The main source of error is the similarity of the poses for the letters $M$, $N$ and $T$ in ASL.

instance, the classification accuracy of a five tree SCF of depth 20 in the first layer is 81.9% for $K = 5$, 96.2% for $K = 15$ and 98.0% for $K = 25$.

On the other hand, the individual elements of the weight matrix $\mathbf{W}$ determine the type of variation a cluster will contain: if we penalize the global rotation angles with large weights, pose clusters will contain variations in fingers mostly. Likewise, giving lower weights to the global rotation angles causes the clusters to contain more camera view point changes. By conducting several experiments, we determined that global rotation is the type of variation that is harder to capture by PCFs, mainly due to the rotation variant features used in the training phase. Therefore, we penalize the global angles with larger weights. We gradually decrease the weights from the palm to the fingers, allowing the finger tips to move rather freely.

Per pixel classification rates are given in Table 1. Our previous model from [16] achieves a success rate of 68% on this dataset, whereas GEN achieves 91.2% and LEN achieves 90.9%. As expected, the expert networks perform significantly better. On the other hand, the difference between GEN and LEN is negligible in this case. However, subjective real–time performance of LEN is better, since it can generalize better to previously unseen poses.

## 5 Discussions and Conclusion

In this work, novel models are introduced for hand shape classification and hand pose estimation problems that are accurate and efficient. For the hand shape recognition problem, SCF is proposed as an effective solution, which is demonstrated on a large ASL dataset. In contrast to the skeleton classification method

| Method | Single–layered RDF | GEN | LEN |
|---|---|---|---|
| Per Pixel | 68.0% | 91.2% | 90.9% |

**Table 1.** Per pixel classification rates of each hand pose estimation method. Single–layered RDF is the PCF as proposed in [16]. The accuracy of both GEN and LEN are substantially higher than a single PCF.

introduced in [16], SCF can be trained using real depth images and require a smaller training set.

For the hand pose estimation problem, we introduced two novel multi–layered RDF networks: Global Expert Network (GEN) and Local Expert Network (LEN). First, we clustered the large training sets using spectral clustering and trained expert PCFs on each cluster. Hence, we divided the complex problem into simpler subproblems and trained experts. Then, we trained an SCF that classifies the input images into clusters, which either determines a global cluster label for the image (GEN), or local clusters for individual pixels (LEN). We showed that this framework performs significantly better than the hand pose estimation method proposed in [16], in terms of accuracy, generalization power and memory requirements. In particular, GEN achieves 91.2% and LEN achieves 90.9% per pixel part classification rate, compared to the reported rate of 68% for the same dataset in [16].

## References

1. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR. (2011)
2. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient Regression of General-Activity Human Poses from Depth Images. In: in Proceedings Thirteenth IEEE International Conference on Computer Vision. ICCV 2011. Volume 2011., IEEE Comput. Soc (2011) 415–422
3. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
4. Ye, M., Wang, X., Yang, R., Ren, L., Pollefeys, M.: Accurate 3D pose estimation from a single depth image. In: in Proceedings Thirteenth IEEE International Conference on Computer Vision. ICCV 2011. Volume 2011., IEEE Comput. Soc (2011) 731–738
5. Lopez-Mendez, A., Alcoverro, M., Pardas, M., Casas, J.R.: Real-time upper body tracking with online initialization using a range sensor. In: in Proceedings Thirteenth IEEE International Conference on Computer Vision Workshops. ICCV 2011. Volume 2011., IEEE Comput. Soc (2011) 391–398
6. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108** (2007) 52–73
7. Singh, V.K., Nevatia, R.: Action recognition in cluttered dynamic scenes using pose-specific part models. In: in Proceedings Thirteenth IEEE International Conference on Computer Vision. ICCV 2011. Volume 2011., IEEE Comput. Soc (2011) 113–120

8. de Campos, T., Murray, D.: Regression-based Hand Pose Estimation from Multiple Cameras. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06) (2006) 782–789
9. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. (2003) II–432–9
10. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3D hand pose reconstruction using specialized mappings. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. Volume 2000., IEEE Comput. Soc (2001) 378–385
11. Stenger, B., Mendonça, P., Cipolla, R.: Model-based 3D tracking of an articulated hand. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, IEEE Comput. Soc (2001) II–310–II–315
12. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-Based 3D Hand Pose Estimation from Monocular Video. IEEE transactions on pattern analysis and machine intelligence (2011) 1–14
13. Mo, Z., Neumann, U.: Real-time Hand Pose Recognition Using Low-Resolution Depth Images. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06) (2006) 1499–1505
14. Malassiotis, S., Strintzis, M.: Real-time hand posture recognition using range data. Image and Vision Computing **26** (2008) 1027–1037
15. Oikonomidis, I., Kyriazis, N., Argyros, A.: Markerless and efficient 26-DOF hand pose recovery. In: Proceedings of the 10th Asian conference on Computer vision-Volume Part III, Springer (2011) 744–757
16. Keskin, C., Kirac, F., Kara, Y.E., Akarun, L.: Real-time hand pose estimation using depth sensors. In: Proceedings Thirteenth IEEE International Conference on Computer Vision Workshops. ICCV 2011, IEEE Comput. Soc (2011) 1228–1234
17. Ong, E.J., Bowden, R.: A boosted classifier tree for hand shape detection. In: Proceedings of the Sixth IEEE international conference on Automatic face and gesture recognition. FGR' 04, Washington, DC, USA, IEEE Computer Society (2004) 889–894
18. Pugeault, N., Bowden, R.: Spelling It Out: Real Time ASL Fingerspelling Recognition. In: in Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, in conjunction with ICCV 2011. Volume 2011., IEEE Comput. Soc (2011)
19. Uebersax, D., Gall, J., Van den Bergh, M., Van Gool, L.: Real-time sign language letter and word recognition from depth data. In: in Proceedings Thirteenth IEEE International Conference on Computer Vision. ICCV 2011. Volume 2011., IEEE Comput. Soc (2011) 383–390
20. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002) 603 –619
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97). CVPR '97, Washington, DC, USA, IEEE Computer Society (1997) 731–
22. Meila, M., Shi, J.: A random walks view of spectral segmentation. (2001)