DeepHand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features

Ayan Sinha* Chiho Choi* Karthik Ramani
Purdue University
West Lafayette, IN 47907, USA
{sinha12, chihochoi, ramani}@purdue.edu

Abstract

We propose DeepHand to estimate the 3D pose of a hand using depth data from commercial 3D sensors. We discriminatively train convolutional neural networks to output a low dimensional activation feature given a depth map. This activation feature vector is representative of the global or local joint angle parameters of a hand pose. We efficiently identify 'spatial' nearest neighbors to the activation feature, from a database of features corresponding to synthetic depth maps, and store some 'temporal' neighbors from previous frames. Our matrix completion algorithm uses these 'spatio-temporal' activation features and the corresponding known pose parameter values to estimate the unknown pose parameters of the input feature vector. Our database of activation features supplements large viewpoint coverage and our hierarchical estimation of pose parameters is robust to occlusions. We show that our approach compares favorably to state-of-the-art methods while achieving real time performance ($\approx$ 32 FPS) on a standard computer.

1. Introduction

Robust hand tracking is central to human-computer interaction interfaces and augmented reality applications. Although, there exists robust and accurate methods for full body tracking, hand tracking is far more challenging [10, 11, 26, 23, 29, 16, 14, 17, 18]. This is due to several reasons: (i) the hand pose exists in a high dimensional space because each finger and the palm is associated with several degrees of freedom, (ii) the fingers exhibit self similarity, are flexible and often occlude each other, (iii) noise in acquired data coupled with fast finger articulations confounds continuous hand tracking. Multi camera setups or GPU acceleration eases some of these challenges, but limits deployment to the general public.

We present a robust method for hand tracking with a single depth camera which achieves real time performance without a GPU. Specifically, we propose a novel matrix completion method to estimate the joint angle parameters on a per frame basis. Our method is flexible to operate with or without temporal information. This alleviates the need for explicit pose initialization if the method loses track or the hand disappears from the camera's view frustum. Furthermore, our pre-compiled database supports large viewpoint coverage and our hierarchical pose estimation from global to local parameters is robust to severe finger occlusions.

At the core of our approach lies a convolutional neural net (ConvNet) architecture to discriminatively reduce the dimensionality of the depth map. ConvNets have achieved ground-breaking performance in image classification [2, 24] and video recognition [8, 9]. A naive strategy to replace the classification layer in a deep neural net with a regression layer leads to errors, as the objective function often gets stuck in a local minima. Previous approaches have shown that this error decreases by incorporating a prior [15] or a intermediate heat map features [29] into the ConvNet architecture. Different from these approaches, we train several ConvNets to output a discriminative low dimensional activation feature in the penultimate fully connected layer. This activation vector represents either the global hand orientation or the local articulations of the five fingers, given a depth map. Our main insight is that a pool of (spatially or temporally) nearby activation features to an activation feature can better represent the hand pose. For generating a population of activation features from which such a pool is extracted, we render realistic depth maps covering a large range of hand articulations and feed them into a deep ConvNet. The ConvNets automatically learn the scope of training (local or global), the finger type (thumb, ring, index, middle or little), and prevalent occlusions by simply inputting the discretized class of the pose parameter values, and do not require any additional information. We then store the activation features from the ConvNets for each depth map in the training data to create a population

*These authors made an equal contribution.

database of activation features. We demonstrate these activation features can be re-purposed on generic databases in our experiments. Additionally, the low dimensionality of the activation feature, coupled with product quantization enables efficient retrieval of approximate nearest neighbors from the population at runtime.

A pose estimation matrix is imputed with the *deep* activation vectors of the nearest neighbor, their corresponding joint angles and the activation vector of the input depth map. This is similar in spirit of the collaborative filtering approach proposed in [1]. However, neither do we use low fidelity BRIEF descriptors for nearest neighbor retrieval, nor do we use inefficient iterations to factorize and complete the matrix. Instead, we estimate the unknown values in the incomplete matrix (*i.e*. pose parameters of input depth map) by assuming a low-rank matrix structure with missing entries. We also add some temporal neighbors from previous frames in the pose estimation matrix which act as a regularizer and reduce jitter of the estimated pose.

Following the success of cascaded approaches to hand pose estimation [23, 18], we hierarchically regress the hand pose from global to local joint angle parameters. The articulation complexity of the palm is lower than of the fingers, and hence, robust estimation of the global orientation is an easier task relative to that of the fingers. The ConvNet finetuned to the conditioned search space outputs more discriminative activation features for finger articulations. This in turn leads to better accuracy for finger parameter estimation. We demonstrate that the ConvNet architecture significantly outperforms PCA [23] and random forests (RF) [18] for global pose initialization. Our overall pipeline runs as 32 frame per second (FPS) on a standard computer. Our main contributions are summarized as follows:

1. Initialization of the pose matrix using a low dimensional and discriminative representation of the global orientation or finger articulations as an activation feature using deep ConvNets, which aids efficient retrieval of nearest neighbors from a large population of pre-computed activation features using product quantization.

2. An efficient matrix completion method for estimating joint angle parameters using the initialized pose matrix.

3. A hierarchical pipeline for hand pose estimation that combines the global pose orientation and finger articulations in a principled way while maintaining real-time frame rates on a standard computer.

The rest of this paper is organized as follows. In section 2, we review relevant literature on 3D hand pose estimation from depth sensors. Section 3 briefly describes our synthetic 3D hand model. The activation feature extraction using ConvNets is discussed in section 4. Section 5 introduces matrix completion for pose parameter estimation. Experimental results and evaluations are described in section 6. Finally, conclusions are presented in section 7.

2. Related Work

Approaches for hand-pose estimation can be broadly classified as either generative (model-based) or discriminative (appearance based) methods. We briefly discuss the generative and discriminative methods relevant to our work. We refer the readers to [5] for a comprehensive review on wearable, marker based and RGB input based techniques from single or multiple cameras and [31] for review on depth-based body pose estimation.

Generative methods An explicit hand model guides the optimization of an objective function in model-based methods to recover the hand pose. [16] use particle swarm optimization (PSO) and [14] use a Gauss-Seidel solver to recover the hand configuration. The objective function is based on the similarity of the depth map and an approximate depth map corresponding to the hand model. The accuracy of the these methods are highly reliant on the hand crafted similarity function. Moreover, these methods are susceptible to error accumulation when the previous estimates are inaccurate. To alleviate model drift prevalent in generative methods, recent approaches adopt the paradigm of *optimization + reinitialization*. These methods first create a population of hand poses and then select the hand pose that *best* fits the observed depth data. The heavy computational burden of this optimization means that the system either achieves low frame rates (12 FPS in [30]) or needs to be accelerated using a GPU (as in [18]).

Discriminative approaches Appearance based methods are proposed for hand pose estimation in [10, 11, 27] similar in spirit to human pose estimation in [19]. The low resolution of hand depth map, self-occlusion and rapid movements lead to large errors in these methods. Subsequently, local regression [3] based approaches were presented to improve the robustness to occlusions, but these methods [23, 26, 28] may suffer from jittering between frames. In [29], convolutional neural networks are used to infer 2D heat-maps corresponding to joint positions. However, their inverse kinematic approach for 3D pose recovery from a 2D image is inefficient in the presence of occlusion. Although our method is similar in spirit to regression, our deep activation features together with enforced temporal consistency in the matrix completion method suppress jitter. Also, the low rank assumption used for matrix completion implicitly allays outliers and aggravates inliers. Our method also shares relationship with the collaborative filtering model proposed in [1]. However, the small size of their database makes the method prone to errors when introduced to unknown poses.

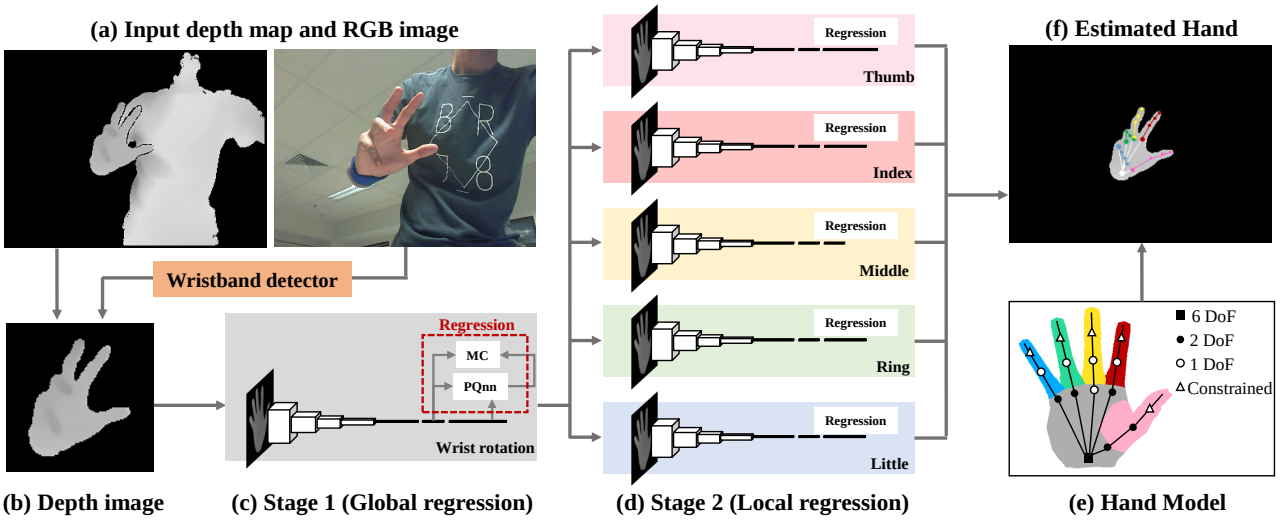

Figure 1: An overview of the proposed approach. In a real-setting, we extract region of interest using depth map and RGB-based wrist band detector (a)-(b). The obtained depth image is fed into a ConvNet which outputs an activation feature. This activation feature synchronizes with other features in a population database using our matrix completion method and the global pose parameters are estimated(c). Based on this global pose initialization, we estimate the rest of the local joint parameters in the same recursive manner (d). The final hand pose is displayed on a multimedia screen (f).

3. Preliminaries

In this section, we briefly describe our 3D hand model and discuss our method to extract the region of interest corresponding to the hand which serves as input to our hand pose estimation method.

Hand model We use a kinematic hand model with 21 degrees of freedom (DOF), represented as $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi})$, as standard in hand pose estimation literature (see Figure 1e). $\boldsymbol{\theta}$ denotes the set of 18 joint angle parameters and $\boldsymbol{\phi}$ is the set of 3 global translation parameters (x, y and z) of the hand.

Region of interest extraction Unlike the body, the hand occupies a relatively small region in the overall depth image obtained from the 3D depth camera. Hence, we preprocess the depth image to only include values that lie in the range of [50, 500] mm under the premise that the hand lies within this range. We then do a largest blob detection as an indicator of the hand segment, followed by median filtering for noise removal, depth normalization so that values lie in the range [0, 255], and finally resize the image while maintaining the aspect ratio to obtain a 64×64 depth image.The centroid of the blob in the original image marks the global position, $\boldsymbol{\phi}$. In more extreme settings (for ranges upto 2000 mm), we use a colored wristband as a simple indicator of the hand region as done in [17, 25]. Even in a close range scenario, the wristband helps removing extraneous pixels like those below the wrist, leading to better performance.

4. Dimensionality Reduction using Deep Learning

It is well known that the activation features from the intermediate hidden layers of a ConvNet can be re-purposed across domains [4, 6]. This suggests that the activation feature of a depth image itself contains discriminative cues about its overall shape and form of the hand, in the context of hand pose estimation. The thrust of our approach relies on the contention that a pool of nearby activation features is better able to reach consensus about the hand's orientation and shape. This introduces two challenges (1) The activation features in the population should conform to the activation features obtained from different individuals in diverse real settings. Additionally, they should be accurately annotated with their ground truth labels (joint angles or positions) (2) The population of activation features must be large enough to provide robust nearest neighbors to any input activation feature, however should be efficiently retrievable and consume limited memory. A straightforward approach is to directly use the depth data gathered from 3D sensors to train a ConvNet and store the corresponding activation features. However, creating a such database of hand poses to cover full range of hand articulations with accurate ground truth labels is a tedious task. In this section, we describe how we generate such a population of activation features from synthetic dataset, reflective of real data.

Gaussian noise	Classification accuracy
Yes	77.00%
No	44.88%

Table 1: The classification accuracy for the global rotation.

4.1. Synthetic population of realistic hand poses

We generate synthetic depth maps by first imposing static (*e.g.*, range of motion, joint length, location) and dynamic (*e.g.*, among joints and fingers) constraints listed in [13][1]. We then uniformly sample each of the 18 joint parameters in this restricted configuration space. This ensures that the depth maps are reflective of real poses covering a wide range of hand articulations. However, data from 3D sensors are prone to noise, distortion and additional artifacts. Hence, we add gaussian noise $N(0, \sigma^2)$ to the synthetic depth maps wherein the standard deviation σ is chosen from a range of [0, 2] by uniform sampling. We empirically validated the inclusion of Gaussian noise by testing the classification accuracy of the global rotation angles in the correct bin (total 144) for a real hand depth sequence captured using SoftKinect DS325 (2500 frames). The drastic improvement of classification accuracy in Table 1 highlights that our noise model if fairly reflective of real sensor noise. Our training dataset covers an entire camera viewpoint (coverage due to the 3 wrist rotation angles $\boldsymbol{\theta^W} = \{\theta_r^W, \theta_p^W, \theta_y^W\}$, where $\theta_r^W \in [-45, 135], \theta_p^W \in [-45, 180], \theta_y^W \in [-45, 180]$). Our large coverage ensures the robustness our method to camera viewpoint changes and not restricted to near frontal poses. We discuss the size of the synthetic population in context to ConvNets in the next subsection.

4.2. Activation features using ConvNet

ConvNet and its variants are the current state of the art architecture for numerous classification tasks such as object detection, scene recognition, texture recognition and fine grained classification. However, hand tracking is effectively a regression task. Our preliminary experiments with deep learning indicated that ConvNets do not adapt to regression as well as they do for classification as shown in Figure 2d. Consequently, our activation features are computed using ConvNet for classification instead of regression. These activation features feed into our matrix completion method which implicitly regresses and outputs the estimated joint angle parameters. The classification of joint angles into quantized bins, and hence, calculation of the activation feature in the penultimate layer, is performed by the ConvNet architecture displayed in Table 2. Observe that the penultimate layer corresponding to the activation feature is

[1]The availability of rigourous constraints in terms of joint angles is the main reason we choose angles over joint position in our hand pose method.

	Layers	# Kernels	Filter size	Stride	Pad
1	Conv	16	5×5×1	1	2
2	Pmax			2	0
3	ReLU				
4	Conv	32	5×5×16	1	2
5	ReLU				
6	Pmax			2	0
7	Conv	32	5×5×32	1	2
8	ReLU				
9	Pmax			2	0
10	Conv	64	5×5×32	1	2
11	ReLU				
12	Pmax			2	0
13	Conv	128	4×4×64	1	0
14	ReLU				
15	Conv	32	1×1×128	1	0
16	ReLU				
17	Conv	144	1×1×32	1	0
18	Smax				

Table 2: Overall architecture of our convolutional networks. (Conv: convolutional layer, Pmax: max pooling layer, ReLU: rectified linear units layer, Smax: softmax layer)

a 32 dimensional vector of the sixth convolutional layer so as to reduce memory usage in storing the population of activation features. We use these activation features in a collaborative spatio-temporal fashion to estimate pose parameters using efficient nearest neighbor search and out novel matrix completion model.

There are two extremal strategies for quantization. The first strategy is to quantize each joint angle separately for a total of 21 ConvNets. However, this is inefficient both in terms of speed and memory. The second is to use an all-in-one strategy to train all joint angle parameters simultaneously. However, it would be impossible to learn an accurate classifier in such a high dimensional space even with a nominal number of bins. Hence, we use a 2-stage hierarchical strategy which satisfactorily balances computational time, memory requirement and classification accuracy.

In *Stage 1* the activation feature associated with the 3 global rotation angles, $\boldsymbol{\theta^W} = \{\theta_r^W, \theta_p^W, \theta_y^W\}$ is calculated and input into the matrix completion method along with a pool of nearest neighbors. The output of the matrix completion method is used to infer the correct rotation bin. For each rotation bin, five ConvNets are trained to output the activation feature associated with each of the five fingers. The ConvNets in *Stage 2* are trained on images within the bin to simplify learning and also on images in adjacent bins to prevent boundary errors. We used 200K images for *Stage 1* global regression (see Figure 1c) wherein the roll, pitch, yaw angles were quantized into 144 bins. Subsequently, 5 Convnets for each of the 144 bins were trained on 10K im-

Model	Accuracy	Memory	Settings
RF	57.45 %	1.30 GB	22 Depth, 70 Trees
	59.04 %	1.87 GB	22 Depth, 100 Trees
ConvNet	71.01 %	2.12 MB	20 Epochs
	72.30 %	2.12 MB	25 Epochs
PCA	5.72 %	None	

Table 3: Accuracy and memory comparison of global pose initialization.

ages within the bin and 10K randomly chosen images in adjacent bins. Training converged after 20 Epochs for the global bin and approximately 10 Epochs for the local rotation bins. The discrete quantization over the joint angle values for each finger is as follows: thumb (144), index (144), middle (36), ring (144), and little (144).

The activation feature associated with the global rotation is critical to the overall accuracy of our approach because this step influences all subsequent ones. To demonstrate the efficacy of ConvNet relative to other approaches, we detail the classification accuracy of ConvNet for global rotation relative to PCA [23] and random forest (RF) [18]. We used 100K depth images because of RF's memory constraints. Table 3 shows that ConvNet achieves a very high accuracy with minimal memory requirement.

5. Matrix Completion for Regression

The matrix completion algorithm runs 6 times: once for the 3 global rotation angles and 5 times for estimating the 15 joint angle parameters associated with the fingers. An iterative approach as the one in [1] is inefficient. Instead we evaluate the unknown parameters in a single shot by assuming a low rank matrix. We discuss the details of our nearest neighbor retrieval to create a pool of activation features followed by the matrix completion method below.

5.1. Extracting pool of activation features

Our matrix completion method takes spatio-temporal nearest neighbors as input. Acquiring temporal nearest neighbors are trivial as they are simply the activation features from the previous frames. However, brute force nearest neighbor evaluation from say the 200K global activation vectors introduces a computational bottleneck unsuitable for realtime application. Our solution to alleviate this problem is to use the top classes predicted by the softmax function in ConvNet to first reduce the search space. We then use highly efficient product quantization based nearest neighbor approximation [7] with 8 subquantizers to retrieve the desired number of nearest neighbors. Details of product quantization are skipped for brevity. In practice, we found retrieving a higher fraction of approximate nearest neighbors by product quantization and then selecting the desired number of nearest neighbors using brute force search from this reduced subset to be more robust than direct retrieval.

5.2. Matrix Completion

Let n be number of spatial nearest neighbors, $\mathbf{D_1} \in \mathbb{R}^{n\times r}$ be the r dimensional activation vectors and $\mathbf{P_1} \in \mathbb{R}^{n\times m}$ be the m desired joint angle parameters being estimated of the n neighbors. In addition, let vector $\mathbf{d_2} \in \mathbb{R}^{1\times r}$ be the r dimensional activation feature output from ConvNet. Let vector $\mathbf{p_2} \in \mathbb{R}^{1\times m}$ be the unknown parameters.

$$\mathbf{M} = \begin{bmatrix} \mathbf{D_1} & \mathbf{P_1} \\ \mathbf{d_2} & \mathbf{p_2} \end{bmatrix} \tag{1}$$

Our task is to estimate $\mathbf{p_2}$ given the other 3 block matrices. Assuming a low rank structure of matrix M this reduces ro solving:

$$\mathbf{p_2} = \mathbf{d_2}(\mathbf{D_1})^{-1}\mathbf{P_1}, \tag{2}$$

The proof of the above result is detailed in the supplementary material.

In practice, we observed that kernelizing the feature matrix and regularizing it by adding a small constant, c to the diagonal, in the spirit of ridge regression makes the output more robust. This parameter c is set to 0.001 in all our experiments. We use the RBF kernel with sigma equal to the variance of the dataset ($\sigma = 200$).

A straightforward extension beyond including just the spatial neighbors is to also include t temporal neighbors from previous frames. This reduces jitter and improves the final quality of our solution. We use 60 nearest neighbors and 16 temporal neighbors for the global parameter estimation. For the 15 local angles, we use 24 nearest neighbors and 4 temporal neighbors. The choice of these parameters is empirically validated in the supplementary material.

6. Experiments

We conduct a comprehensive evaluation with state-of-the-art approaches as well as self-generated baselines on the synthetic and real datasets to demonstrate the efficacy of our solution. We first describe the datasets and baselines.

6.1. Datasets

We split our evaluation into two stages. First, we use synthetic data to compare our method to baselines. This comparison validates the rationale of our specific approach against other choices. This data is generated using the same approach as described in Section 3 to generate our database, albeit continuity constraints are enforced. Two synthetic sequences are generated which are 2.5K frames long at standard rates (approximately 80 seconds each). The advantage

of these synthetic sequences are that they are already labeled, avoiding tedious ground-truth assignment.

Next, for fair comparison to other methods, we evaluate the performance of our method on two publicly available datasets: Dexter1 [21] and NYU [29]. The Dexter1 dataset consists of seven gestures (*i.e.*, adbadd, flexex1, pinch, fingercount, tigergrasp, fingerwave, and random) with high inter-gesture verifiability, however, mostly from frontal viewpoints. Hence we use the NYU dataset for a more thorough evaluation of the method. As we shall shortly show, our method remarkably achieves state-of-art performance without fine-tuning on their training dataset.

Although the authors are aware of other datasets like ICVL [26], MSRA14 [17], or MSRA15 [23] in the literature, we do not use them for one or more of the following reasons: (1) the depth pixels of the body are included with the hand depth map. Recall we use a heuristic method for segregating the hand from the rest of the body and a wrist band under more extreme conditions. We did not find a straightforward way to segregate the data without incurring loss. (2) The hand poses are enforced using muscular labor, i.e., hand configurations wherein one or more finger applies pressure on another. These configurations are not accounted for in our joint angle modeling framework to render synthetic depth maps, however, modeling additional constraints to account for such hand poses is plan of future work. Also note that we use the SoftKinetic's DethSense DS325 for all our real demonstrations.

6.2. Baselines for method validation

There are three salient features of our approach which we rigorously validate. First, a hierarchical approach is justified in spite of the computational overload it introduces. Second, a pool of activation features is better at estimating the hand pose than a single activation feature or a direct regression based approach using ConvNets. Third, our choice of imputing the matrix with spatio-temporal neighbors and kernelizing the features provides superior performance. We naturally perform this validation by comparing to the following three baselines: (a) *Holistic* which evaluates all parameters in an all-in-one approach using a single activation feature. We also compare it to *JMFC* which also performs a matrix update using a single feature vector, although using computationally expensive iterations in [1] (b) *Conv-PQ* which directly estimates the pose parameters to be the nearest neighbor and *Regression* which directly regresses pose parameters using ConvNets with L2 loss are used to validate our choice of pool of activation feature, and finally (c) *No-temporal* which contains only spatial neighbors for matrix completion, *Non-kernel* which uses feature matrix without kernelization, and *Weighted* which finds pose parameters using Gaussian similarity between activation features as weights are used to validate our matrix completion approach. The validation is done in terms of one or more of the following standard error metrics popular for pose estimation problems: (a) the average joint angle error in degrees, (b) the average joint distance error in millimeters, (c) the maximum allowed joint angle error in terms of a threshold ε_A, and (d) the maximum allowed joint distance error in terms of a threshold ε_D. Broadly speaking, the first two metrics evaluate performance at a local joint level whereas the the other measure global robustness of an approach. We employ the appropriate metric based on the context of the evaluation. Although our angle based method is particularly effective in minimizing joint angle errors, yet we choose joint distances as our error metric on public datasets to demonstrate the overall robustness of our approach.

6.3. Comparison to Baselines

In this section, we quantitatively evaluate our method with respect to the baselines on the synthetic datasets. Figure 2 shows that our method significantly outperforms the proposed baselines both in terms of local as well global error metrics. The performance markup over the *Conv-PQ* approach as seen in Figure 2c indicates that a ConvNet by itself would do a poor job of inferring a complex articulated structure such as the hand. The performance improvement over Holistic in the zone of small angles is also intuitive. It indicates that the global activation feature contains some latent information about the local joint angles, but this information is better revealed by a hierarchical estimation procedure. This is also validated in Figure 2a and 2b where we see a significant performance improvement in terms of joint angles for finger portions that are frequently occluded such as the middle finger. It is also noteworthy to note that the similarity of these plots in terms of error ranges to plots on real hand sequences implicitly validate our data creation process. *Regression* [2] for joint angle prediction resulted in worse performance than even *Conv-PQ* baseline (nearest activation feature) as shown in Figure 2d. We adopted different approaches, *e.g.,* fine-tuning our ConvNets, L1 loss, etc.to ensure that direct regression is indeed suboptimal. We contend that as joint angles are a function of relative joint points,learning joint angles is harder compared to joint positions, and hence, resulted in inferior performance. Figure 2e shows the performance of matrix-completion baselines relative to our proposed approach. The figure validates that constructing a kernel, incorporating temporal information and using matrix completion instead of simple weighted regression are all critical to good performance.

6.4. Comparison with the state-of-the-arts

Having validated the rationale of our approach, we now compare our method to other state-of-the-art approaches

[2] the penultimate layer is of dimension 2048 as we do not need nearest neighbor retrieval

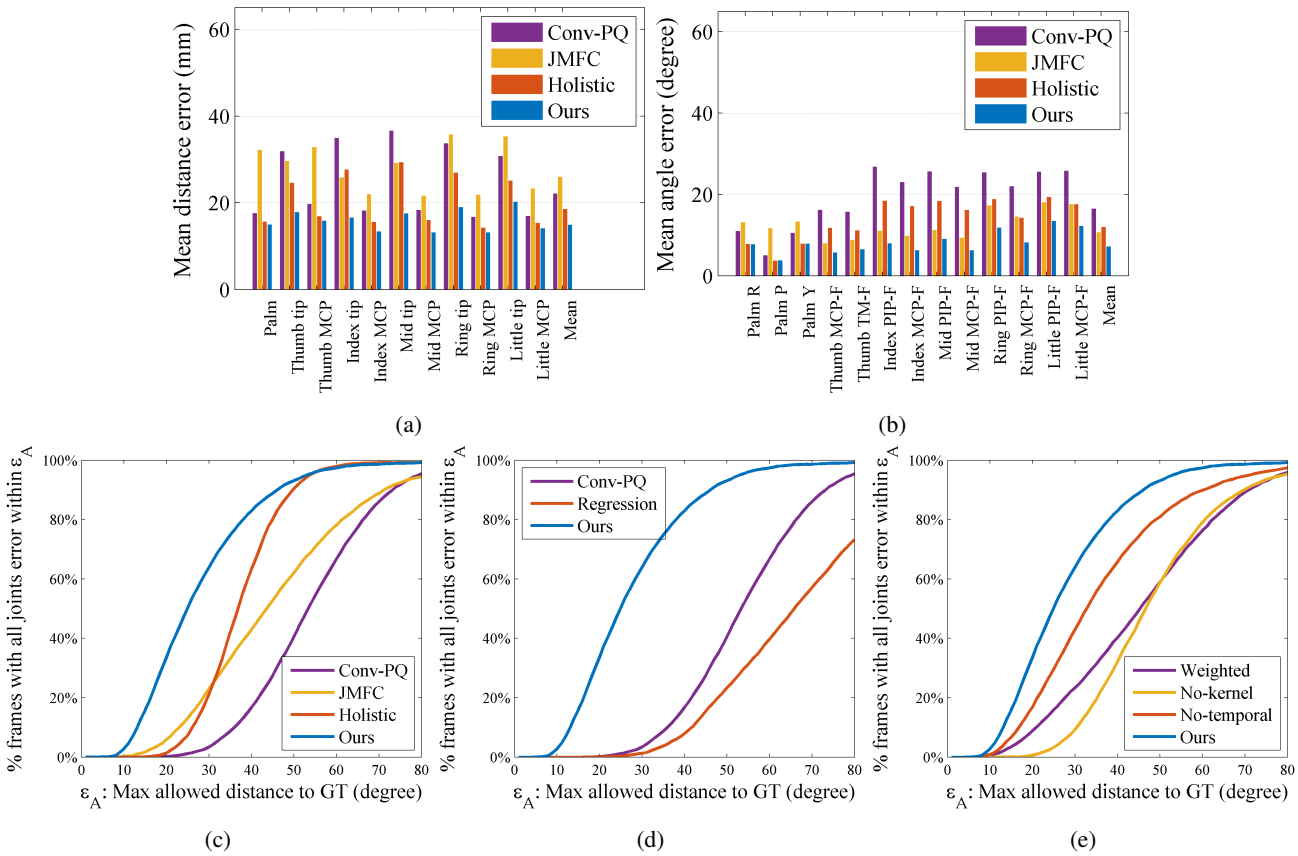

Figure 2: The results of quantitative evaluation on the synthetic dataset.

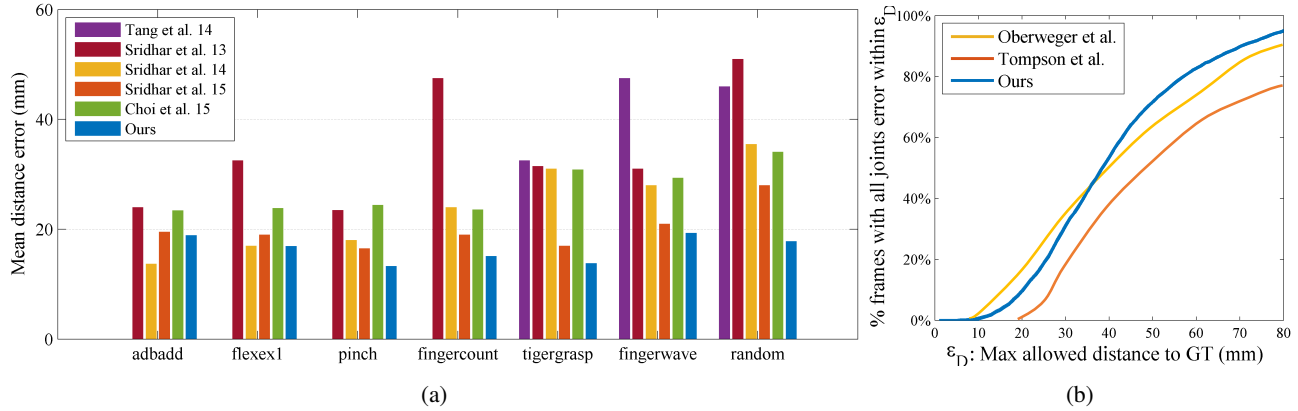

Figure 3: The results of quantitative evaluation on the public dataset. Note that the accuracies are directly estimated from corresponding figures (*i.e.*, figure 4 in [20] and figure 3a in [15]).

[29, 21, 26, 22, 20, 15, 1] on the Dexter1 and NYU datasets.

Quantitative Analysis We measured the average distance error of five fingertips (in mm) on the Dexter1 dataset to evaluate the overall robustness of our approach. Figure 3a shows the comparison of our approach to other methods which include both discriminative [26, 1] as well as generative [21, 22] methods. Not only does our method achieve the lowest overall error rate (see Table 4), we also achieve the lowest individual error rates for all but one gesture *i.e. adbadd*. This is because the particular gesture is especially hard to model in terms of joint angle constraints.

We evaluated our approach directly on the 8.2K of test depth maps from the NYU dataset. Figure 3b illustrates the maximum allowed error with respect to the distance

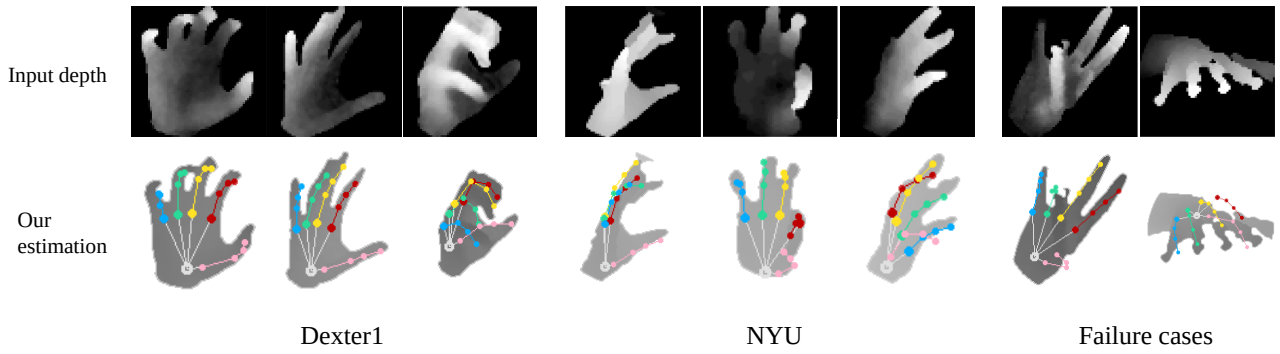

Figure 4: Qualitative evaluations are conducted on two public datasets, Dexter1 and NYU. The first row shows the input depth image, and corresponding estimation is presented in the second row.

Methods	[26]	[21]	[22]	[20]	[1]	Ours
Error	42.4	31.8	24.1	19.6	25.27	16.35

Table 4: The overall average error (mm) of the five fingertip positions on Dexter1. Ours shows the lowest error rate compared to the state-of-the-art methods.

threshold. The fact that our method performs better than [15] over a long range indicates the activation features we get from ConvNet can be used across domains and sensor types [3], and hence the activation features can potentially be made general purpose. This is encouraging in the context of progressively fine-tuning ConvNets with more information such as when new joint angle constraints or dynamic constraints become available. Furthermore, simulating principled noise models such as [12] corresponding to true sensor noise can further enhance the generality of these features in the context of hand pose estimation.

Qualitative Analysis We do a qualitative evaluation of our algorithm with the state-of-the-art methods on some public datasets. The top row of Figure 4 shows cropped 64x64 depth images which are used as input to our system, and the second row shows corresponding estimates with our matrix completion method (without temporal neighbors). All estimated poses are kinematically valid and follow a natural sequence. For the sake of completion, we also show some failure cases in the last two columns of Figure 4. In our system this happens when some unnatural pose (driven by muscular force) appears in front of the camera or when the image is severely affected by noise or has missing parts.

7. Conclusion

We present a novel framework for hand pose estimation using a deep convolutional neural network. Instead of using a single activation feature, we use a pool of activation features to synchronize and collectively estimate the hand configuration, all in real time. This pool is derived by training a deep ConvNet with a large database of synthetic hand poses and efficiently storing the activation feature corresponding to the penultimate fully connected layer. Careful thought was placed so that this database is reflective of real data. At runtime the pool of activation features in the spatial domain and temporal domain combine together in a hierarchical way to robustly estimate the hand pose. The derived activation features can be applied across domains and sensor types as demonstrated in our experiments. Furthermore, our method achieves state of the art performance. Although our approach is general, one limitation of our activation features is that the estimations are only valid in the joint angle domain. Future work will focus on ways such that people working in the joint angle or joint position domain can seamlessly fuse their models together to create even deeper and more robust models. Another line of future work is to investigate our matrix completion approach in a more general setting. The simplicity combined with its efficiency makes a promising alternative to standard regression techniques for a wide array of machine learning tasks.

Acknowledgements This work was partially supported by the NSF Award No.1235232 from CMMI and 1329979 from CPS, as well as the Donald W. Feddersen Chaired Professorship from Purdue School of Mechanical Engineering. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

[3] NYU dataset use PrimeSense to capture their data

References

[1] C. Choi, A. Sinha, J. H. Choi, S. Jang, and K. Ramani. A collaborative filtering approach to real-time hand pose estimation. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. 2, 5, 6, 7, 8

[2] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012. 1

[3] W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988. 2

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655, 2014. 3

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1):52–73, 2007. 2

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 3

[7] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, 2012. 5

[8] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013. 1

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[10] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*, pages 852–863. Springer, 2012. 1, 2

[11] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013. 1, 2

[12] M. J. Landau, B. Y. Choo, and P. A. Beling. Simulating kinect infrared and depth images. 2015. 8

[13] J. Lee and T. L. Kunii. Model-based analysis of hand posture. *Computer Graphics and Applications, IEEE*, 15(5):77–86, 1995. 4

[14] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013. 1, 2

[15] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015. 1, 7, 8

[16] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, volume 1, page 3, 2011. 1, 2

[17] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1106–1113. IEEE, 2014. 1, 3, 6

[18] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. K. C. R. I. Leichter, A. V. Y. Wei, D. F. P. K. E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. In *Proc. CHI*, volume 8, 2015. 1, 2, 5

[19] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. 2

[20] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015. 7, 8

[21] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013. 6, 7, 8

[22] S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *Proceedings of the International Conference on 3D Vision (3DV)*, Dec. 2014. 7, 8

[23] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015. 1, 2, 5, 6

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[25] A. Tagliasacchi, M. Schroeder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. Technical report, 2015. 3

[26] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3786–3793. IEEE, 2014. 1, 2, 6, 7, 8

[27] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3224–3231. IEEE, 2013. 2

[28] A. Thayananthan, R. Navaratnam, B. Stenger, P. H. Torr, and R. Cipolla. *Multivariate relevance vector machines for tracking*. Springer, 2006. 2

[29] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169, 2014. 1, 2, 6, 7

[30] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3456–3462. IEEE, 2013. 2

[31] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 149–187. Springer, 2013. 2