Journal of Autonomous Intelligence (2019) Volume 2 Issue 1.
doi:10.32629/jai.v2i1.36

Article

Learning Hand Latent Features for Unsupervised 3D Hand Pose Estimation

Jamal Banzi[1*2], Isack Bulugu[3], Zhongfu Ye[1]

[1]*School of Information Science and Technology, University of Science and Technology of China, 230026, China*

[2]*Sokoine University of Agriculture, Morogoro, 3167, Tanzania*

[3]*College of information and communication Technology, University of Dare-es-salaam, Dar-es-Salaam, 33335, Tanzania*

ABSTRACT

Recent hand pose estimation methods require large numbers of annotated training data to extract the dynamic information from a hand representation. Nevertheless, precise and dense annotation on the real data is difficult to come by and the amount of information passed to the training algorithm is significantly higher. This paper presents an approach to developing a hand pose estimation system which can accurately regress a 3D pose in an unsupervised manner. The whole process is performed in three stages. Firstly, the hand is modelled by a novel latent tree dependency model (LTDM) which transforms internal joints location to an explicit representation. Secondly, we perform predictive coding of image sequences of hand poses in order to capture latent features underlying a given image without supervision. A mapping is then performed between an image depth and a generated representation. Thirdly, the hand joints are regressed using convolutional neural networks to finally estimate the latent pose given some depth map. Finally, an unsupervised error term which is a part of the recurrent architecture ensures smooth estimation of the final pose. To demonstrate the performance of the proposed system, a complete experiment was conducted on three challenging public datasets, ICVL, MSRA, and NYU. The empirical results show the significant performance of our method which is comparable or better than the state-of-the-art approaches.

Keywords: *Hand Pose Estimation; Convolutional Neural Networks; Recurrent Neural Networks; Human-machine Interaction; Predictive Coding; Unsupervised Learning.*

ARTICLE INFO
Received: Apr 7, 2019
Accepted: Apr 28, 2019
Available online: May 6, 2019

***CORRESPONDING AUTHOR**
Jamal Firmat Banzi, School of Information Science and Technology, University of Science and Technology of China, 230026, China; Sokoine University of Agriculture, Morogoro, 3167, Tanzania;
jbanzi@mail.ustc.edu.cn;

CITATION
Jamal Banzi, Isack Bulugu, Zhongfu Ye. Learning Hand Latent Features for Unsupervised 3D Hand Pose Estimation *Journal of Autonomous Intelligence* 2019; 2(1): 1-10. doi: 10.32629/jai.v2i1.36

1. Introduction

Hand pose estimation from depth is the first step for several human-computer interaction applications. It has been widely applied to human-machine interaction (HMI) since it provides the possibility for future multi-touchless interfaces. An accurate hand pose estimation provides a natural way of interaction between human and virtual space that achieves greater user experience. Different from the conventional human-machine interactions which are limited to 2D plane display, and which are only suited where users sit behind the computing devices, hand pose estimation offers 3D user interaction without direct contact with the computing device. This provides a possibility for the new interface leading towards seamless human-computer interaction **Figure 1.**

However, hand pose estimation is still a difficult task owing to some challenges that a human hand possesses[1-3]. The hand is very dextrous, has many degrees of freedom. Similarly, fingers have high self-similarities and severe self-occlusion[4,5]. The input depth image is accompanied by the large amount of noise which will probably mislead the pose estimator and distort the output results[6].

Indeed, there are significant progress in developing fast and accurate hand pose estimation systems thanks to the advent of low-cost depth sensors[7-12]. State-of-the-art methods for 3D hand pose estimation from depth rely heavily on large numbers of depth images annotated with

However precise annotation of 3D hand joints on real data is difficult to come by and time-consuming. Additionally, the computation complexity of the annotation process increases the chance of generating multiple residual errors[19,25]. This reduces the utility of deep neural networks on hand pose estimation domain.

Inspired by the discussed challenges; this paper presents a deep neural network algorithm with a predictive coding model (deep PCM) to contends the overhead of annotating large numbers of training examples required for initial training of the supervised networks. The proposed model predicts hand joint positions recursively using deep recurrent convolutional networks with bottom up and top down connections in an unsupervised fashion. The LTDM hand topology improves hand detection with much better accuracy, discussed in detail in section 2. The generated hand topology is fed into deep predictive coding (Deep PCM), get encoded to generate a hand representation which will be mapped with the decoded depth maps. Finally, we regressively train end to end, a stable and accurate pose estimator based on depth images. The included long-short-term memory (LSTM) layer as a part of the recurrent architecture, provides an error correction population used to improve the estimation accuracy of the final hand pose.

Our contributions can be summarized as follows:

- We propose a new way of modeling a hand topology using LTDM. The LTDM transforms internal joint locations to an explicit representation, which is compact and invariant in scale and view angles.
- We innovatively integrate an LTDM with the deep PCM to learn the internal representation of the hand geometry that is well suited to subsequent recognition and decoding of the latent hand parameters which are used for hand pose estimation.
- The error regression scheme provides error back propagation that allows our network to learn from its own mistakes and automatically correct them to improve the accuracy of the estimation.

hand joints[10-15]. These methods have demonstrated promising results

using deep learning approaches[16-18] which are all fully supervised.

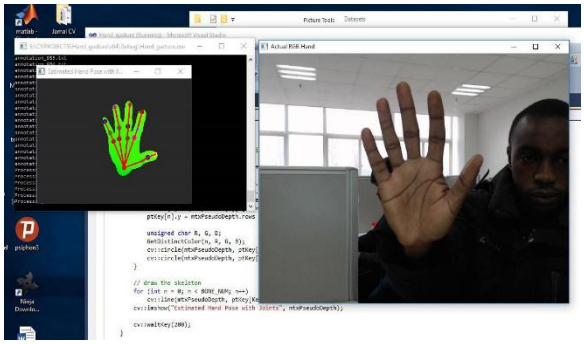

Figure 1. Demonstration of our hand pose estimation system

2. The Proposed Approach

This section explains in details about the whole process of hand joints position estimation in the given hand depth. Ideally, we formulate the hand topology of a human hand using a novel LTDM. Then, the data-dependent method[29] is used to jointly learn LTDM which generates latent variables of the posterior pose appearance, and the deep PCM which generates pose configuration. The mapping is performed between the two individual domains.

Eventually, the complete network is then trained end to end for the pose estimation process.

A. Representation of a hand topology

Using LTDM to represent hand structure is a viable choice. More precisely, unlike previous works[30-33] which utilized the traditional latent tree model (LTM) to represent a hand topology, the proposed LTDM model is devoted to resolving ambiguity, avoid redundant hidden nodes and more certainly reduces complexity. Furthermore, LTDM models dependencies between random variables with a structure that can change dynamically based on the variable values. It is therefore capable of modeling context-specific independencies.

Early works, e.g. Wang *et al*[31] employed LTM to represent the articulation of body parts[31]. They applied recursive grouping and Chow-Liu Grouping (CL grouping)[32] to learn structures directly from observation. However, their method had the following drawbacks:

- The learned LTM contain no latent node, all node represents observable variables i.e. skeletal parts.

- They defined 14 single parts and 10 combined parts to mimic the latent node, resulting in an LTM which diverges from true representative topology for the human body. To eliminate these limitations, Tan *et al*[32] applied LTM to represent human hand topology with a coarse-to-fine search paradigm to reduce the training samples and testing complexity. The method learns an LTM automatically to capture hand topology in a coarse-to-fine manner. Their proposed LTM requires no prior knowledge of physical joint connections nor predefined combined parts and can be applied to many articulated objects. However, Tan's LTM algorithm is still complex and requires many computations, there is a possibility for redundancy for hidden joints, and also the dependencies between variables are fixed. Inspired by the work of Tan *et al*[32], we propose the latent dependencies tree model LTDM as a new type of modeling hand topology. LDTMs encode the relations between variables with a dependency tree. A distribution over all possible dependency trees given the current assignment of variables is specified using the first-order non-projective Dependency Grammars (DG) presented in the literature of[32]. The probability of a complete assignment can then be computed by adding up weights of all the dependency trees. We provide an example of using LTDM to compute joint probability of the assignment of a given variables in **Figure 2**.

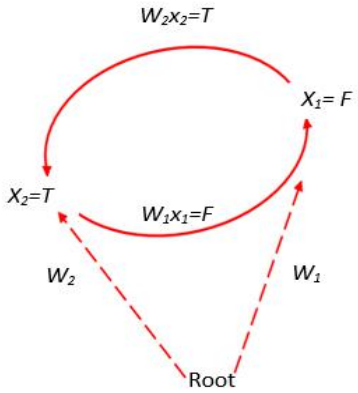

Figure 2. An example of using LTDM to compute joint probability from three joint representations (Thumb, Little, and Middle)

These three joints are more reliable and are chosen as the foundation to construct new coordinates(branches). X_1 and X_2 are variables which form a pairwise dependency WnX_n. Each dependency has a weight which is used to grow a tree and calculate the probability of joint distribution.

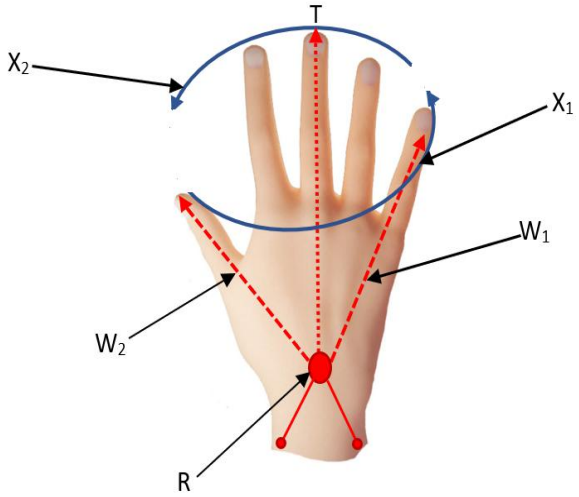

Figure 3. LDTM representing a hand model geometry

Compared with the existing probabilistic models, the LTDM has the following unique features;

- LTDM models the latent dependencies between random variables i.e. dependencies are dynamically determined based on the assignment of random variables.
- LTDM considers all possible tree structure at the same time resulting in easier learning.
- LTDM removes all possible latent node dependencies, and hence improves hand joint detection as shown in **Figure 3.**

B. Model definition

An LTDM is a tree-structured graphical model where leaf nodes are observed, and internal nodes can either be observed or latent as for the conventional latent tree model, but further LTDM encodes relations between variables with a dependency tree. We denote the tree model as,

$$T = (X_1 \bigcup X_2 W) \quad (1)$$

where the vertices are composed of observable vertices X_1, and latent vertices $X_2 = \{x_2\}$, where $x_2 \subseteq X_2$; and W denotes pairwise dependencies among variables. The strength of each pairwise dependency is independent. The dependency strength from node x_i and node x_i is denoted by an edge weight w_{ii}.

C. Edge weight function

The edge weight function F_x is the sum over the weights of all possible dependency branches for a given

assignment x , which represents the weight of the assignment. It is given as;

$$F_x = \sum_{S \in S(N_x)} w(T) = \sum_{S \in S(N_x)(x_i, x_2) \in E_T} \prod w_{ij} \qquad (2)$$

where S is a spinning tree, and $S(N_x)$ is a set of all possible dependency nodes.

The LTDM model requires that the weight of each dependency (x_i, x_2) is the conditional probability. Therefore, the vertices X_1 and X_2 are given assignment $X_1 = x_i$ or the root node and its probability is given as w_{x_i/x_i} such that $0 \le w_{x_i/x_i} \le 1$. For all nodes, there are given assignment, $X = (x_1, x_2, ..., x_n)$ which are generated recursively in a top-down manner.

We grow a tree with $n+1$ nodes uniformly at random. The root node is given as x_0 . Then starting from this root node, we recursively traverse the tree in pre-order such that at each non-root node, a variable to value pair is generated conditioned on the variable to value pair of its parent node. The probability of generating an assignment x is given as:

$$p(\mathrm{x}) = C!\mathrm{F}_x \alpha \mathrm{F}_x \qquad (3)$$

Where C is constant, representing the uniform probability of the tree structure. Note that some variables may be assigned to multiple nodes and therefore there might be missed variables. However, since we are only interested in the node space of the valid assignments i.e. no redundancy nor missing variable assignment, we define the joint probability of a valid assignment x as;

$$\phi(\mathrm{x}) = \frac{p(\mathrm{x})}{\sum \mathrm{x} \in A^{p(\mathrm{x})}} = \frac{\mathrm{F}_x}{\gamma} \qquad (4)$$

Where A is the set of valid assignments and γ is the normalization factor.

D. Unsupervised learning

We describe an algorithm to learn LTDM from the depth where the dependency structure of each training instance is unknown. For each internal node as in the Chow-Liu tree[32], a recursive joining method scheme is applied by identifying its neighborhood. This method can produce consistent LTDM s without redundant latent nodes. Applying log-likelihood to the function given in equation (3), we obtain;

$$\sum_{n=1}^{|D|} \log p(x_n) = \sum_{n=1}^{|D|} \log \mathrm{F}_{x_n} + \mathrm{C} \qquad (5)$$

Where $D = \{x_n\}_{n=1}^{|D|}$ is the training sample. It is noticeable that the log-likelihood is computed on $p(\mathbf{x})$, the probability of generating an assignment, and not of $\varphi(\mathbf{x})$, the probability of a valid assignment. This makes our learning algorithm tractable, and also encourages the learned model to be more likely to produce valid assignments.

3. Deep PCM Network

We present the deep learning architecture with the predictive coding model for prediction and subsequent regression of hand joint positions, **Figure 4**.

A predictive coding[36,39] is an RNN with the following features:

- The start time, holds $t_0 = 0$, while $\tau = 1$, and τ is constant.
- The initial input state S_0 of a time series constitutes the initial components of the start vector x_0 .
- It has linear activation applied to all neurons
- The initial weight W^{in} and relative weights W^{re} are random, independent, and identically distributed from the standard normal distributions, whereas the output weight W^{out} are learned.
- The input and output are arbitrarily connected and there is no clear distinction between them.

A. Operation of Predictive coding Network

On the first time of operation, the input, prediction layer and the error representation layer are equivalent to a deep convolutional network[26]. On the other side, the recurrent representation layers are equivalent to a generative deconvolutional network with local recurrence at each stage. This architecture is general and can be adapted to model different kinds of data. The architecture was originally proposed by[38] and then modified to meet hand pose estimation demand, trained end to end using gradient descent, with a loss function implicitly embedded in the network as firing rates of the error neurons.

Basically, each module of our network consists of four basic deep layers:

- An input convolutional layer (P_i),
- An estimation (prediction) layer (P_l) ,
- An error regression layer (E_l),
- A recurrent representation layer (R_l)

For representation neurons, we explicitly use convolutional LSTM units[37]. This is a recurrent convolutional layer responsible for the generation of estimations from the input layer.

B. Long short-term memory (LSTM)

LSTM model correlation between observed and hidden state with a memory unit. This provides significant improvement over RNNs. We apply LSTM to extend our predictive coding with two advantages. Firstly, induce the memory information. Secondly, handles sequential data better and avoids the gradient vanishing problem. We show how LSTM handles the vanishing problem and perform backpropagation error as in[37]. The core idea of LSTM is to encode the information of the inputs (x_t and y_{t-1}) that has been thrown away from the cell state C_t and output of y_t . Further LSTM has gates which control states with a sigmoid function. These are forget gate f_t , input gate i_t , output gate o_t , and modulation gate g_t . Therefore, for a given image sequence $\{x_1,.....x_T\}$, we have the following gate definitions;

$$f_t = \sigma(W_f \cdot [y_{t-1}, x_t] + b_f \quad (6)$$

$$i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i \quad (7)$$

$$g_t = \tanh(W_c x_t + W_y y_{t-1} + b_c) \quad (8)$$

$$o_t = \sigma(W_o[y_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t \text{gtanh}(C_t) \quad (10)$$

As for RNN, to make a prediction, we add a linear model over the hidden state h_t , and output the likelihood with softmax function.

$$z_t = soft\max(\mathrm{W}_h h_t + b_z) \quad (11)$$

Given ground truth at time t as y_t we can minimize the least square $\frac{1}{2}(y_t - z_t)^2$ to estimate hand parameters.

Hence for the top layer classification, with weight W_z , we now take derivates w.r.t z_t , and W_z respectively

$$dz_t = y_t - z_t \quad (12)$$

$$dW_z = \sum_t h_t dz_t \quad (13)$$

$$dh_T = W_z dz_T \quad (14)$$

Where the gradient is considered only w.r.t h_T . However, for any time step *t* its gradient will differ a little, see equation (15) below.

$$d\mathrm{h}_{t-1} = d\mathrm{h}_{t-1} + W_z dz_{t-1} \quad (15)$$

This will then be back propagating it with every time step *t*.

C. Error regression scheme

Initially, the input image sequence enters the model and the local estimation of this input is made. This estimated input is subtracted from the actual input and passed along to the next layer. The network takes the difference from the input P_i and the estimated hand $\hat{P}_l$, and output an error representation (E_l) which splits into separate rectified positive and negative error populations. The error E_l , is then passed forward through the convolutional layer to become the input of the next layer P_{i+1} .

The recurrent representation layer (R_l) receives a copy of the error signal E_l , along with the top-down input from the representation layer of the next level the network layer R_{i+1} as shown in **Figure 5.** To improve the accuracy of the location estimates of hand joints, this step is iterated several times while forwarding an error to an input to allow the network learn from its own previous mistake.

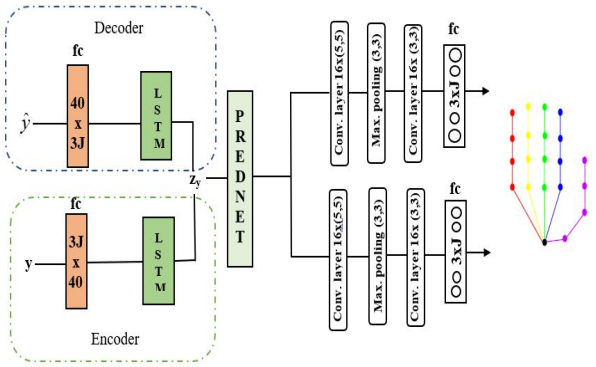

Figure 4. Overview of the proposed system showing the network architecture. Fc stands for fully connected layers and Conv. stands for convolutional layers

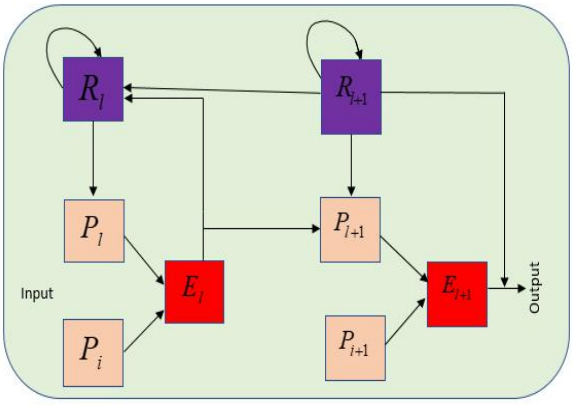

Figure 5. Illustration of the data flow within the proposed deep PCM

ALGORITHM 1: Updating States of the PCM

Input: A sequence of image x_t

Output: An estimated hand pose P_l^t

Procedure update R_l^t states

Let $i = 0,$ and $t = 0$ *Initialize i and t*

$P_0^t \leftarrow x_t$

$R_l^0, E_l^0 \leftarrow 0$ *Initial Estimates*

for $t = 1$ to T **do**

for $l = L$ to 0 **do**

if $l = L$ **then**

$R_L^t = \mathrm{LSTM}(E_t^{t-1}, R_L^{t-l})$

First iteration

else

$R_l^t = \mathrm{LSTM}(E_t^{t-1}, R_l^{t-l}, UPSAMPLE(R_{l+1}^t))$

for $l = 0$ to L do Second iteration

If $l = 0$ to L **then**

$P_0^t = RELU(CONV(R_0^t))$

else

$P_l^t = RELU(CONV(R_l^t))$

$E_l^t = [(E_l) - (E_{t+l})]$

if $l < L$ **then**

$P_{l+1}^t = MAXPOOL(CONV(E_t^l)$

$P_l^{t+1} = E_t^l + R_l$ *Total Estimation + error*

Send R_l *to the next iteration*

Return

end

4. Experiment

We perform a complete experiment to unveil the performance of the proposed approach.

A. Experimental settings

The Model was implemented with the python library, using Theano[35] and Keras[21]. The model parameters are optimized using gradient descent Adam algorithm[34] with all parameters set to default values.

System requirement	Specifications
Central processing unit (CPU)	Intel(R)Core i7-4790@ 3.6GHz
Operating system (OS)	Microsoft window
Random access memory (RAM)	16GB
System architecture	64bit

Table 1. Implementation specification of our system.

B. Data pre-processing

Before the training stage, we first need to pre-process the raw data from the dataset. The input of the pose estimator is the cropped image, but the original ground truth of the image is the used. absolute position in the entire raw image Therefore, there is a need to first transform the ground truth into a relative position with respect to the center of the hand. Finally, the cropped images are re-sized to 128X128 as the input of a deep PCM.

C. Regressive training

We train our deep PCM regressively with the learning time gradually decreased.

The model was based on the pre-trained model[38] and was trained to predict hand joints position. The loss was taken as the sum of the firing rates of the error neurons in the zeroth pixel layer. A random hyperparameter search was performed over fourth- and fifth-layer models of the posterior position. The deep PCM model consists of 5 layers with 3 by 3 filter sizes of all convolutions and stack size per layer of (1,32,64,64,128,256). The initial training rate is set to 0.001 dropped by learning ratio of 10 after every 60 epochs.

5. Evaluation with the state-of-the-art

This section discusses the comparison of our approach with the existing state-of-the-art approaches. We evaluate the performance of our approach on three publicly available datasets for hand pose estimation: The NYU datasets[22], ICVL dataset[20], and MSRA dataset[15]. Table 2 below presents the details of the datasets used.

Dataset	No. Subjects	No. joints	No. Frame	Depth resolution
ICVL	10	16	17K	320x320
NYU	2	36	81K	640x480
MSRA1	9	21	76K	320x240
MSRA	6	21	2K	320x240

Table 2. Datasets used for validation of the experiment

A. Evaluation metrics

Two different commonly used criteria to evaluate our method, namely:

- The fraction of sample error distance within a threshold. Here we measure the fraction of success frames whose error distance of each joint is less than a certain threshold. This is the most challenging evaluation criterion since the single mistaken joint may decline the judgment of the entire hand pose.

- Mean error distance of different joints and their average. This is recognized as the most commonly used criteria in the literature of hand pose estimation and allows comparison with many contending baselines, because of the simplicity of the evaluation.

B. Self-Comparisons

We first evaluate the impact of modeling hand topology using LTDM. We utilize NYU dataset to depict the number of success frame over a certain threshold. As illustrated in **Figure 6**, the impact of LTDM based on hand topology on the estimated joints is presented, it shows that LTDM performed better than the traditional LTM. Also, the proportion of success frame is higher with the LTDM than LTM and assumption-based method. This indicates the effectiveness of LTDM in modeling the hand topology.

*C. Comparison with the state-of-t*he-art

The proposed deep PCM is compared with 12 state-of-the-art methods: Feedback loop[24], 3D CNN[27], Bighands[45], DeepPrior++[28], Regional Ensemble[23], Tang *et al*[20], Tompson *et al*[22], PointNet[46], Zhou *et al*[40], Hand3D[43], CrossingNets[46] and Madadi *et al*[41].

The proportions of good frames over a certain error threshold are presented in **Figure 7.** Generally, the empirical results show an outstanding performance of the proposed method over many contending approaches while it works comparably with few methods that also have attained the state-of-the-art performance.

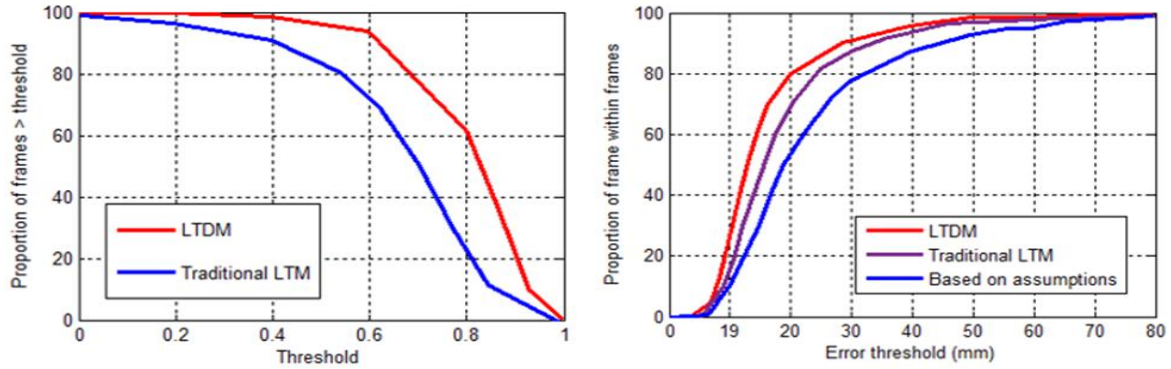

Figure 6. Left: The success rate of the hand detection for the two methods. Right: Error rate of the two methods of modeling a hand

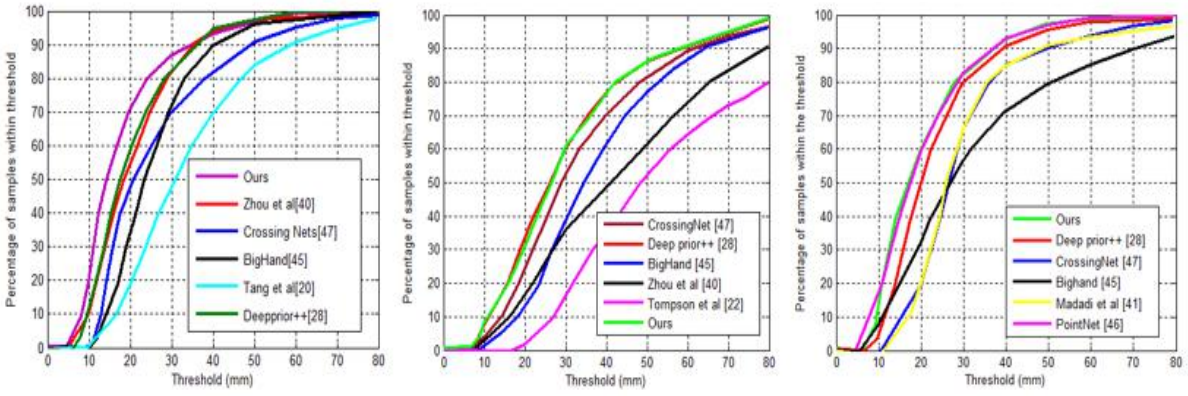

Figure 7. Comparison with the state-of-art on ICVL [20] (left), NYU [22] (middle), MSRA [15] right. It shows the fraction of samples whose distance between all estimated joints and ground truth is less than a certain threshold

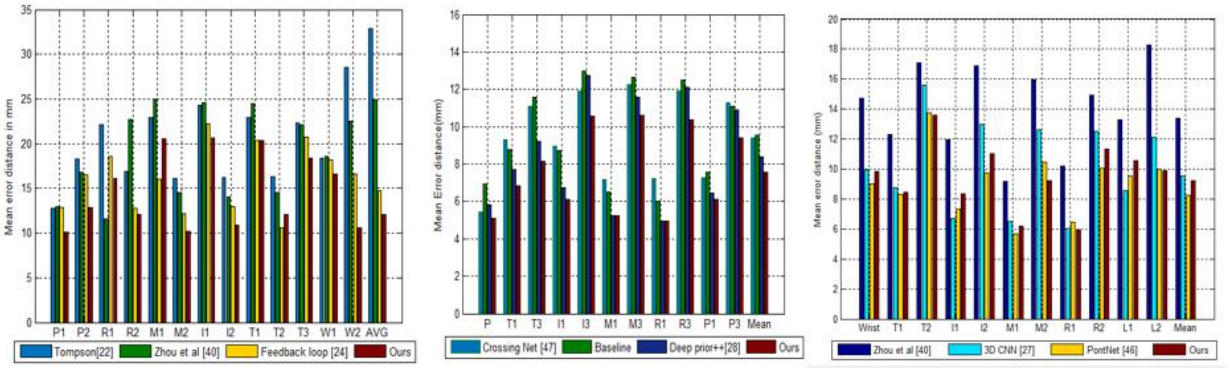

Figure 8. Comparison with the state of the arts on NYU [22] (left), ICVL [20] (middle), MSRA [15] (right). It measures the error distance of different joints and their average. (P: palm, T: Thumb, I: Index, M: Middle, L: Little)

The reason for this superiority is attributed by the following. Firstly, the robust hand topology using a novel LTDM capable of modeling context-specific independencies, such that the estimation of the final pose of the hand is based on the prior knowledge of the detected hand representation. This is different from many previous works which are based on a very weak detector or based on the strong assumption that the nearest object behind the camera is the hand. Secondly, the proposed approach has an intrinsic error regression paradigm which smooths the estimated values allowing the network to learn its own mistakes and rectifies to finally increase the accuracy of estimation. For example, on ICVL dataset[20], we compare our work with the five state-of-the-arts works. We use the three joints to define the pose space based on deep PCM. **Figure 7** (left) illustrates the result that show the achievement of our method using first evaluation metric. For example, when the error threshold is 20mm, the proportions of good frames of our approach achieve 10% and 15% better than DeepPrior++[28] and Zhou *et al*[40] respectively, whereas others more than 20%. This shows that the proposed deep PCM works very well.

On NYU dataset[22], when the error threshold is taken between 20mm and 30mm the proportion of good frames of our method is comparable to DeepPrior ++[28], and 10% better than the CrossingNet[47]. Similarly, On MSRA dataset[15], when the error threshold is considered between 20mm and 30mm, the proportion of good frames is comparable to PointNet[46], 5% better than DeepPrior++[28] and 25% better than CrossingNet[47], BigHand[45], and Madadi *et al*[41]. These results indicate that our proposed method is robust and it can accurately estimate the location of the hand joint positions. On the other hand, using the second metric, the error distance of different joints and their average are presented, **Figure 8.** We only compare the mean error distance of 11 joints as most of works did[25,42]. The results show that our method outperformed state-of-the-art methods on showing the lowest error of 12.2mm on NYU dataset, 7.4mm on ICVL dataset. However, on MSRA dataset, PointNet[46] performed better than our method in overall mean error attained lowest error of about 8.4mm where as ours is 8.9mm. Nevertheless, if the error distance of different joints is considered, our method achieves better results than theirs for most of the significant joints.

The visualization of some of the samples drawn from the proposed system are presented in **Figure 9** showing accurate joint location of most of the poses.

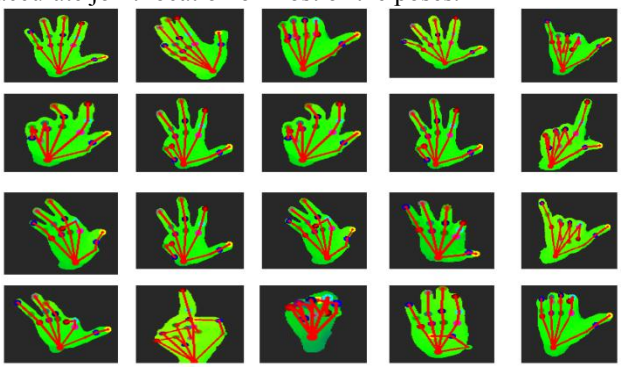

Figure 9. Visualization of the estimated results of our method

6. Conclusion

In this paper, we present a novel approach to model hand topology based on LDTM which captures hand latent features and observable features to construct hand joint representation. Then we integrate LDTM with the deep PCM using a data-independent method to encode the hand

representations and map with the decoded hand depth map. Finally, the multi-layered convolutional neural network based on deep PCM was utilized to regress a 3D pose space based on the joints location. As a result, our system can accurately estimate a hand pose based on the prior knowledge of the hand representation. This confers robust and reliable hand pose estimation system that can achieve greater user experience.

Acknowledgement

This research is supported by the Fundamental Research Funds for the Central Universities (Grant no. WK2350000002). The authors acknowledge a fellowship by the Chinese Academy of Science and The World Academy of Science (CAS-TWAS).

References

1. Barsoum E., 'Articulated Hand Pose Estimation Review'2016, arXiv: 1604.06195 (preprint) pp. 1–50.
2. Erol A., Boyle R. D. *et al*. Vision-based hand pose estimation: A review. Computer Vision and Image Understanding. 2007; 108(1-2): 52–73.
3. Sridhar. S, *et al*. "Fast and robust hand tracking using detection-guided optimization." In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, 07-12-June-2015, pp. 3213–3221.
4. Krejov. P, Andrew. G, and Richard. B, "Combining discriminative and model-based approaches for hand pose estimation." In 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG); IEEE, 2015; vol. 1, pp. 1-7
5. Tracewski, L., Bastin, L., & Fonte, C. C. Repurposing a deep learning network to filter and classify volunteered photographs for land cover and land use characterization. Geo-spatial information science, 2017; 20(3), 252-268.
6. Yu, H., *et al*. Analysis of large-scale UAV images using a multi-scale hierarchical representation. Geo-spatial Information Science, 2018., 21(1), 33-44.
7. Chen, *et al*. Learning a deep network with spherical part model for 3D hand pose estimation, Pattern Recognition, 2018, 80, 1-20.
8. Zimmermann .C, and Brox.T "Learning to Estimate 3D Hand Pose from Single RGB Images," in Proceedings of the IEEE International Conference on Computer Vision, 2017, vol. 2017–October, pp. 4913–4921.
9. Tagliasacchi. A, *et al*, Robust articulated-ICP for real-time hand tracking," Eurographics Symp. Geom. Process., 2015, 34(5) pp. 101–114.
10. Patel. H, *et al*, Neural network with deep learning architectures, Journal of Information and Optimization Sciences, 2018, 39 (1), pp 31-38.
11. Ye. Q, Yuan. V, and Kim. T, Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation Computer vision and pattern recognition, 2016.; arXiv:1604.03334,
12. Sun. X, *et al*, "Cascaded hand pose regression," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June-2015, pp. 824–832.
13. Sinha. A, Choi. C, and Ramani. K, "Deep Hand: Robust Hand Pose Estimation by Completing a Matrix Imputed with Deep Features," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4150–4158.
14. Choi. C, *et al*, "A collaborative filtering approach to real-time hand pose estimation," in Proceedings of the IEEE International Conference on Computer Vision, 2015, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 2336–2344.
15. Oikonomidis. M, Lourakis I., and AR gyros. A. , "Evolutionary quasi-random search for hand articulations tracking," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 3422–3429.
16. Krejov. P, Gilbert. A, and Bowden. R, Guided optimisation through classification and regression for hand pose estimation, Comput. Vis. Image Underst., 2017.; 155, pp. 124–138.
17. Qian *et al*, Realtime and robust hand tracking from depth, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 1106–1113.
18. Tang, T. H. Yu, and T. K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3224–3231.
19. Banzi. J, Zhongfu. Ye, and Bulugu. I, "A novel hand pose estimation using discriminative deep model and Transductive learning approach for occlusion handling and reduced discrepancy," in proceedings of IEEE International Conference on Computer and Communications, 2017, pp. 347–352.
20. Tang. D *et al*, "Latent regression forest: Structured estimation of 3D hand poses," IEEE Trans. Pattern Anal. Mach. Intell. 2017., 39(7), pp. 1374–1387.
21. F. Chollet. Keras, 2016.
22. Tompson. M, Stein. Y, and Perlin. K, "Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks," ACM Trans. Graph., 33(5), pp. 1–10, 2014.
23. Guo. H, *et al*. "Region ensemble network: Improving convolutional network for hand pose estimation." 2017 IEEE International Conference on Image Processing (ICIP), pp. 4512-4516.
24. Oberweger. M, Wohlhart. P, and Lepetit.V, "Training a Feedback Loop for Hand Pose Estimation," 2015

IEEE Int. Conf. Comput. Vis., pp. 3316–3324, 2015.
25. A. Dosovitskiy, *et al*, “Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks, 2014, ”38(9), pp. 1734–1747.
26. Chen. L, *et al* “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” IEEE Trans. Pattern Anal. Mach. Intell., 2018, 40(4), pp. 834–848.
27. Ge. L, *et al*. "3d convolutional neural networks for efficient and robust hand pose estimation from single depth images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1991-2000.
28. Oberweger. M, and Lepetit, V., 2017. DeepPrior++: Improving fast and accurate 3d hand pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, pp. 585-594.
29. Zhou. ZH, Feng J., Deep forest: Towards an alternative to deep neural networks, 2017, arXiv preprint arXiv:1702.08835.
30. Wang, F. and Li, Y., Beyond physical connections: Tree models in human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 596-603.
31. Tan, V.Y *et al*. Learning high-dimensional Markov forest distributions: Analysis of error rates. Journal of Machine Learning Research, 2011, 12(May), pp.1617-1653.
32. Chow C, Liu C. Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory. 1968 May;14(3):462-7.
33. Huang. Y and Rao R., “Predictive coding,” Wiley Interdiscip. Rev. Cogn. Sci., 2011., 2(5), pp. 580–593,
34. Kingma D and J. Ba. J, “Adam: a method for stochastic optimization,”, 2014, arXiv Prepr. arXiv1412.6980, pp. 1–13,
35. Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” 2016, arXiv e-prints.
36. Hill *et al*., “Deep Predictive Coding Network for Video Prediction and Unsupervised Learning, 2017, ICLR, pp. 1–9.
37. Srivastava, N., Mansimov, E, and Salakhutdinov R.“Unsupervised Learning of Video Representations using LSTMs,” 2015, BMVC2015, p. 2009,.
38. W. Lotter, G. Kreiman, D. Cox, ‘Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning’, pp. 1–12, 2016.
39. S. Frieder, O. Michael, and O. Obst. "Predictive Neural Networks." arXiv preprint arXiv:1802.03308(2018).
40. X. Zhou, Q. Wan, Z. Wei, X. Xue, and Y. Wei, “Model-based deep hand pose estimation,” in IJCAI International Joint Conference on Artificial Intelligence, 2016, vol. 2016–January, pp. 2421–2427.
41. Madadi, M., Escalera, S., Baró, X., & Gonzalez, J. (2017). End-to-end global to local CNN learning for hand pose recovery in-depth data. arXiv preprint arXiv:1705.09606.
42. Tzu-Yang Chen, Pai-Wen Ting, Mn-Yu Wu, Li-Chen Fu, Learning a deep network with spherical part model for 3D hand pose estimation,” pattern recognition. 2018, 33, pp.3203.
43. Deng, X., Yang, S., Zhang, Y., Tan, P., Chang, L. and Wang, H., 2017. Hand3d: Hand pose estimation using 3d neural network. arXiv preprint arXiv:1704.02224.
44. Yuan, S., *et al*. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4866-4874.
45. Ge, L., *et al*, Hand PointNet: 3d hand pose estimation using point sets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8417-8426.
46. Wan, C., *et al*,. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 680-689.