ISMSIT2017

1st International Symposium on Multidisciplinary Studies and Innovative Technologies Proceedings Book

(November 2-4, 2017, Tokat, Turkey)

Symposium homepage: www.ismsitconf.org

Structured Deep Learning Supported with Point Cloud for 3D Human Pose Estimation

Erdal Özbay[1], Ahmet Çınar[2] and Zafer Güler[3*]

[1,2] Computer Engineering Department/Fırat University, Turkiye
[3] Software Engineering Department/Fırat University, Turkiye
**(zaferguler@firat.edu.tr)*

Abstract – In this paper, a structural-output is obtained to estimate 3D human pose using 3D human point cloud and monocular images. The Neural Network takes a human image and 3D pose as inputs and gives outputs a score value. Conditional Random Field (CRF) approach is using to semantically classify human limbs in its point cloud for 3D human pose production. The voxel cloud connectivity segmentation (VCCS) is used as the segmentation method that voxelisation of the 3D point cloud. The network structure consists of a convolutional neural network for image feature extraction and pose into a joint embedding. The score function is calculation from the dot-product between the images and pose embeddings which is high when the image-pose pair matches and low otherwise. Image-pose embedding and score function are jointly trained using the max-margin cost function. Finally we present visualizations of the image-position placement field, showing that the network has learned a high level embedding of body orientation and pose configuration.

Keywords – Pose estimation, Point Cloud, Deep learning, Structured learning, 3D

I. Introduction

The human pose estimation, which has been in the works for decades, it has been the scene of different work in the past few years. These studies are influenced by industrial camera capabilities. In particular, having knowledge of human skeletal joints facilitates pose estimation in many studies. Due to the dependencies among the joint points, it can be considered as a structured-output task. In general, two different approaches can be taken to estimate human pose:

The first of these prediction-based methods; the second is optimization-based methods. The approach of the first method is to address the problem of regression or detection in human pose estimation [1]–[4]. The goal of the method is to learn the matching 2D or 3D joint points on the target with the image features from the input space or classifiers to identify certain body parts in the image. Such methods are usually simple and work fast during the evaluation phase. Toshev et al. have set up a cascaded network to refine 2D joints locations at the image level [2]. However, this approach does not explicitly consider the structured constraints of human pose. Inference studies on 2D joint positions included joint predictions of the relationship between them [4, 5]. Constraints of prediction-based methods include: the manually designed constraints may not exactly match the dependencies between the body joints points; poor scalability to 3D joint estimation when the search space needs to be discretized; only single pose can be predicted in situations where multiple poses may be valid due to partial self-occlusion.

The second approach is learning a score function instead of directly estimating the target, which takes both an image and a pose as input, and produces a high score for the matched pair of correct images and poses and low scores for unmatched image-pose pairs. Given an input image x, the estimated pose y^* is the pose that maximizes the score function.

$$y^* = \underset{y \in \mathcal{Y}}{\arg\max}\, f(x, y), \tag{1}$$

where $\mathcal{Y}$ is the pose space, if the score function can be normalized properly, then it can be interpreted as a probability distribution, either a conditional distribution of poses given the image, or a joint distribution over both images and joints.

One of the different popular approaches use pictorial constructions [6]. Accordingly, dependencies between joints are represented by edges in a probabilistic graphical model [7]. The structured output SVM is an alternative to the generative models [8], which is a distinctive way to learn the score function. This method ensures a large margin between the score values for correct input pairs and for incorrect input pairs [9, 10].

As a score function is treated both image and pose as input there are various methods for presenting the information together with the source of the image and pose, e.g., the image features extracted around the input joint positions could be viewed as the joint feature representation of image and pose [11, 12]. Alternatively, the features in the image and the features in the pose can be extracted separately and then combined. The score function can be trained to combine those together [13]. However, in studies that follow all these methods, features are handcrafted and system performance is largely depends on the quality of the features. From the other side, it is shown to be more efficient in extracting informative high-level features of deep neural networks [14].

In this paper, we propose a framework for structured learning with deep neural networks for 3D human pose estimation. Our presented framework jointly learns the image and pose feature representations and produce a score function. In particular, our network first extracts separate feature embeddings from the image input and from the 3D pose input. Point cloud data is used for 3D human pose input. Generation of human 3D pose estimation will be performed with 3D point cloud segmentation Conditional Random Fields (CRF). Human3.6m and i3Dpost datasets are used for matching of monocular images and 3D human pose as well as real human 3D point cloud to be produced with the Kinect sensor camera. The score function is then the dot-product between the image and pose embeddings, which is efficient to compute. The score function and feature embeddings are trained using a maximum-margin criteria, resulting in a discriminative joint-embedding of image and 3D pose.

II. Materials and Method

Human body part segmentation is one important focus in pose estimation. Segmentation is the process of splitting the image into multiple regions by extracting the object from a background. In this section segmentation is being performed over the point cloud for 3D pose estimation. A segmentation method is presented for semantic human point cloud classification to the body parts as seen in Fig. 1. The first output of this phase is a 3D point cloud that will be semantically categorized into body parts. Our aim is to classify the 3D human point cloud and assign one label for each piece of the joint part. The normal and curvature features of the point cloud points are used for this purpose.

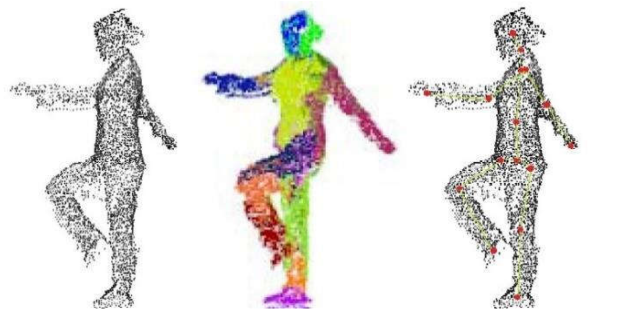

Fig. 1 Point cloud, segmented point cloud, and skeleton extraction of points

A. 3D Pose Extraction

CRF is a graphical-based method that is often used in partitioning studies. In this study, a simplified CRF extraction is used to best divide the 3D point cloud data. The inference is the average field approach in the CRF graph. Graph nodes are supervoxel from the past step. Computation of the output with a fully connected CRF model has a large computational complexity with many nodes in the original point cloud. For this reason, the CRF graph is defined on overly divided regions originating from the past step. As such, by replacing the supervoxel with singular point cloud as graph nodes, the number of graph nodes and the computational complexity of inference reduce.

In the proposed method by constructing the graph with every supervoxel in its nodes, pair wise potentials are some of Gaussian kernels as shape, appearance, smoothness and also the geodesic distance between each node [15]. The mesh between the nodes is calculated for the calculation of the geodetic distance. Poisson surface reconstruction is used to calculate the mesh between each supervoxel with the PCL library. Dijkstra's shortest path algorithm [16] is used to calculate the geodetic distance between nodes. The open source implementation of the Dijkstra algorithm is used as part of the C library [17].

The CRF model has both single and pairwise potentials defined for the 3D human point cloud. The single potentials should best be calculated using point features, and a probability must already be defined on the label assignments for each point.

In general for a fully dependent CRF model;

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2}$$

The pairwise edge potential $\psi_p(x_i, x_j)$ is defined as a linear combination of Gaussian kernels $k^m(f_i, f_j)$ [18].

The algorithm used to estimate the inference function is the backward message passing method. The algorithm is given as follow;

Algorithm 1. Mean field in fully connected CRFs [19].

```
Initialize $Q$
$Q_i(x_i) = \frac{1}{z_i}\exp\{-\phi_u(x_i)\}$
While not converged do;
Message passing from all $x_j$ to all $x_i$.
$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j\neq i} k^{(m)}(f_i, f_j) Q_j(l)$ for all $m$
Compatibility conversion
$\hat{Q}_i(x_i) \leftarrow \sum_{l\in L} \mu^m(x_i, l) \sum_m \omega^m \tilde{Q}_i^{(m)}(l)$
Local update;
$Q_i(x_i) \leftarrow \exp\{-\psi_u(x_i) - \hat{Q}_i(x_i)\}$
Normalize $Q_i(x_i)$
End while
```

According to algorithm, compatibility conversion and local update work in a fairly efficient manner within a linear time. But, the message passing step for each variable, requires a more computational complexity. Since evaluating a sum over all other variables has a quadratic complexity in the number of variables *N*. High-dimensional filtering can be used to reduce the computational cost of message passing from quadratic to linear [20].

B. Image Feature Extraction

The aim of the image extraction subnetwork is to transform the raw input image into a more compact representation with the pose information preserved. In proposed method we use a deep CNN which is consisting of 3 sets of convolution and max-pooling layers, to extract image features from the image. The rectified linear units (ReLU) [21] are using as the activation function in the first 2 layers, and the linear activation function in the 3rd layer.

$conv^j\ (x)$ is defined as a feature maps. The output of the pooling layers is a set of these feature maps, where j is the layer number. Fig. 2 shows detailed information about feature map and convolutional filter sizes. Each feature in the map has a receptive field in the input image, with higher layer features having larger receptive fields. Intuitively, higher-layer features contain general information about the pose. This will be useful for distinguishing between different poses. Also, lower-layer properties, contain more detailed information about the pose that will help distinguish similar poses.

C. Image-Pose Embedding

Since the image and pose are in different spaces, a common area is needed. The image-pose embedding sub-network is to reflect the intended image features and the 3D pose where they can be compared effectively. Fig. 2 shows the basic architecture of the image and pose embedding network. We inspired from Sun, and Li et al., both middle and upper layers of convolution were used [22, 3].

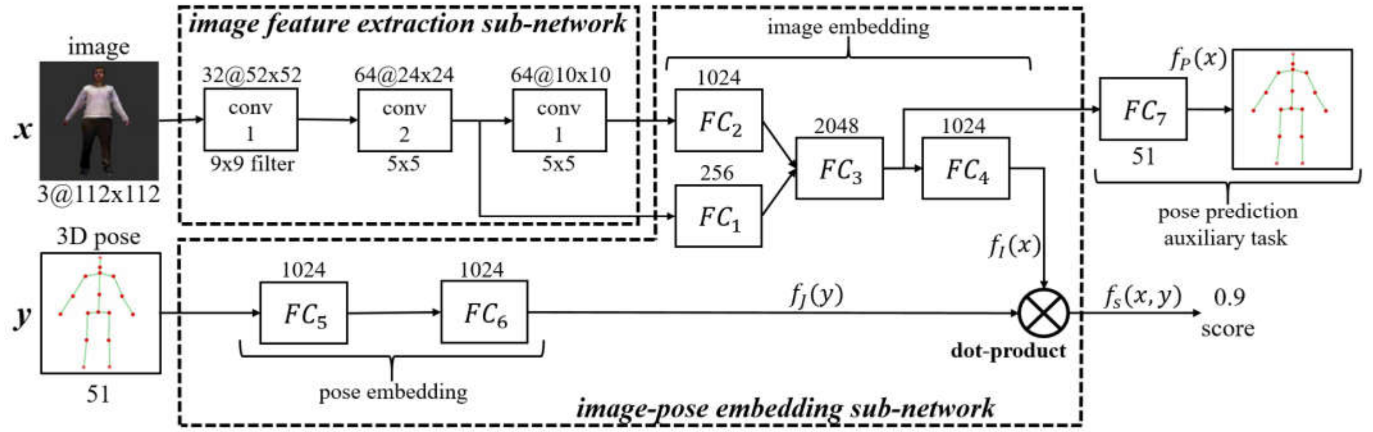

Fig. 2 Flow diagram of deep-network score function. Image input is fed over a set of convolution layers to extract image features. Two separate sub-networks are used to embed the image and pose in a common area, and the score function is the dot product between the two embeddings. An auxiliary 3D body-joint prediction process is performed which will help to network to find a good image features. Unscratched max-pooling layer follows the each convolution layer to reduce clutter

The middle and top layer features are passed through each independent fully connected layers and then they are joined together. It is passed over two more fully connected layers to obtain the embedding image $f_I(x)$.

$$f_I(x) = h_4\left(h_3\left(\begin{bmatrix} h_1(conv^2(x)) \\ h_2(conv^3(x)) \end{bmatrix}\right)\right), \qquad (3)$$

where the activation function $h_i(x) = ReLU(W_i^T x + b_i)$ is a rectified linear unit with weight matrix W_i and bias b_i.

The input pose y is represented by the 3D coordinates of body-joint locations. Dimensions are strongly related to dependencies between joints. The pose is matched as a non-linear embedding so that it can be more easily combined with the image embedding. 2 completely connected layers are used for this transformation.

$$f_J(y) = h_6(h_5(y)). \qquad (4)$$

D. Score Prediction

The score function $f_S(x, y)$ is represented between the image and the pose inputs as the inner product between the image embedding $f_I(x)$ and the pose embedding $f_J(y)$. i.e.;

$$f_S(x, y) = \langle f_I(x), f_J(y) \rangle. \qquad (5)$$

The advantage of inner-product is used to facilitate alignment between the two embeddings. This allows direct interaction of the corresponding dimensions of the image and the pose embedding vectors. Another advantage of the method is that the calculation yields very effective results. Pose calculation works independently of image features. This means that the candidate pose can be computed offline when the pose is stabilized.

In network training, the image and pose are matched to similar embedding spaces that their dot-product similarity serves as an appropriate score function. This situation can be loosely expressed as learning a multi-view "kernel" function, where the "high-dimensional" feature space is the learned joint embedding. At the same time the score function can be interpreted as the linear combination of the multiple pose-score function. Each dimension of the pose embedding corresponds to a pose-score function, which indicates how well the input pose belongs to a specific pose subspace (i.e.; poses facing the camera).

The calculation of linear combination weights is decided by image-feature embedding, which controls which pose-subspaces are to be well matched by the input pose.

E. Maximum Margin Cost

Structured SVM is used to calculate the maximum margin cost function to learn the score function [23]. The maximum margin cost calculation ensures that the difference between the scores of the two input pairs has at least a particular value (margin). Unlike known standard SVMs, the structured-SVM may have a margin that changes values based on the dissimilarity between the two input pairs. In addition, a margin that recalculates the rescaling role is used, similar to the structured-SVM,

$$\mathcal{L}_M(x, y, \hat{y}) = \max(0, f_S(x, \hat{y}) + \triangle\,(\hat{y}, y) - f_S(x, y)). \qquad (6)$$

where (x, y) is a training image-pose pair, $\triangle\,(\hat{y}, y)$ is a nonnegative margin function between two poses, and max (a, b) returns the maximum value of a and b. In other words, max $(0, x)$ is a rectified linear function. $\hat{y}$ is the pose that most violates the margin constraint. $\hat{y}$ depends on the input (x, y) and network parameters θ to reduce clutter, we write $\hat{y}$ instead of $\hat{y}\,(x, y, \theta)$ when no confusion arises.

$$\begin{aligned} \hat{y} &= \arg\max_{y \in \mathcal{Y}} f_S(x, y') + \triangle\,(y, y') - f_S(x, y) \\ &= \arg\max_{y \in \mathcal{Y}} f_S(x, y') + \triangle\,(y, y') \end{aligned} \qquad (7)$$

Intuitively, a pose with a high predicted score, but that is far from the ground-truth pose, is more likely to be the most violated pose [24]. Here we use the mean per joint position error for the margin function, (MPJPE), i.e.;

$$\triangle\,(y, y') = \tfrac{1}{J}\textstyle\sum_{j=1}^{J} \left\| (y_j,\ y'_j) \right\| \qquad (8)$$

where y_j indicates the 3D coordinates of j'th joint in pose y, and J is the number of body-joints. When the loss function in (6) is zero, then the score of the ground-truth image-pose pair (x, y) is at least larger than the margin for all other image-pose pairs (x, y');

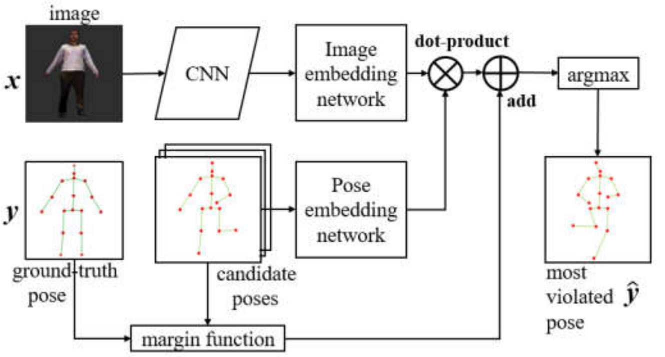

Fig. 3 Network structure for calculating the most violated pose. For a given image, the score values are predicted for a set of candidate poses. The re-scaling margin values are added, and the largest value is selected as the most-violated pose. Thick arrows represent an array of outputs, with each entry corresponding to one candidate pose

$$f_S(x,y) \geq f_S(x,y') + \Delta\,(y',y),\ \forall y' \in \mathcal{Y}. \tag{9}$$

On the other hand, if (6) is greater than 0, then there exists at least one pose y' whose score $f(x,y')$ violates the margin. Using maximum-margin training, our score function can be interpreted as a SSVM, where the joint features are the element-wise product between the learned image and pose embeddings,

$$f'_S(x,y) = \langle w, f_I(x) \text{ o } f_J(y) \rangle. \tag{10}$$

where o indicates element-wise multiplication, and *w* is the SSVM weight vector. The equivalence is seen by noting that during network training the weights w can be absorbed into the embedding functions $\{f_I,\ f_J\}$. In our framework, these embedding functions are discriminatively trained.

F. Multi-task Global Cost Function

A training task system has been added to help of predicting the 3D pose what the 3D pose is to encourage the embedding of the scene in support of more pose information.

$$f_P(x) = g_7(h_3), \tag{11}$$

where h_3 is the output of the penultimate layer of the image embedding, and $g_i(x) = \tanh(W_i^T x + b_i)$ is the tanh activation function. The cost function for the pose prediction task is the square difference between the ground-truth pose and predicted pose,

$$\mathcal{L}_P(x,y) = \|f_P(x) - y\|^2. \tag{12}$$

Finally, given a training set of image-pose pairs $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$, our global cost function consists of the structured maximum-margin cost, pose estimation cost, as well as a regularization term on the weight matrices.

$$cost(\theta) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}_M\left(x^{(i)}, y^{(i)}, \hat{y}^{(i)}\right) + \frac{1}{N}\lambda \sum_{i=1}^N \mathcal{L}_P\left(x^{(i)}, y^{(i)}\right) + \gamma \sum_{j=1}^7 \left\|W_j\right\|_F^2 \tag{13}$$

where *i* is the index for training samples, λ is the weighting for pose prediction error, γ is the regularization parameter, and $\theta = \{(W_i, b_i)\}_{i=1}^7$ are the network parameters. Note that gradients from $\mathcal{L}_P$ only affect the CNN and high-level image features (FC$_1$-FC$_3$), and have no direct effect on the pose embedding network or image embedding layer (FC$_4$). Therefore, we can view the pose prediction cost as a regularization term for the image features. Fig. 3 and Fig. 4 show the overall network structure for calculating the max-margin cost function, as well as finding the most violated pose.

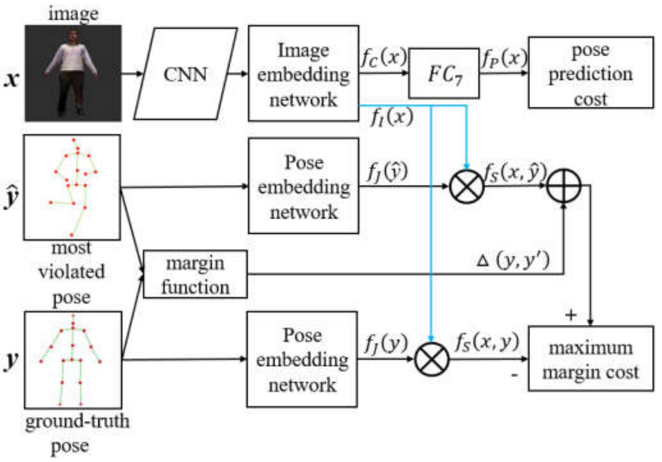

Fig. 4 Network structure for maximum-margin training. Given the most-violated pose, the margin cost and pose prediction cost are calculated, and the gradients are passed back through the network

G. Maximum-Margin Network Training

Stochastic gradient descent (SGD) is used to train the network. The system is similar to the SSVM procedure, as the most-violated pose finding (8), and the minimizing the cost function (13) is to update the network parameters that reduce the cost. The procedure is;

1- Find the most-violated pose $\hat{y}$ for each training pair (x, y) with the current network parameters using the pose selection network, Fig. 3.

2- Input into maximum margin into the training network $(x, y, \hat{y})$ and run the back rule to update the parameters Fig. 4.

The tuple $(x, y, \hat{y})$ values are introduced as extended training data. These training data is processed in mini-partitions. It has been discovered that using the weighted average of the current gradient and previous updates, the use of momentum among the mini-partitions that update parameters always hinders convergence. The reason for this is that the maximum margin cost is to choose the different poses most violated in each partitions. It also ensures that the direction of change is rapidly changing among the partitions.

Score function calculation is effective, but scanning of all poses increases the calculation time to find the most-violated pose. Instead, a set of candidates $\mathcal{Y}_B$ for each partitions are created and the most-violated poses in this set are searched. Candidates consist of *C* poses sampled from the pose space $\mathcal{Y}$. During the training, it was seen that the same poses were determined more than once as the most-violated poses. For this reason, a study set consisting of the most-violated poses was created and the *K* most frequently used most-violated pose were added to the candidate set.

The simplest solution to estimate pose is to train the pose with the maximum image of the test image *x*.

$$y' = \underset{y \in T}{\arg\max}\, f_S(x,y) \tag{14}$$

where *T* is the set of training poses. However, since the training and test sets contain different subjects, the poses in the training set will not perfectly match the subjects in the test set. Hence to allow for more pose variation, we compute the average of the A training poses with highest scores,

$$\bar{y} = \frac{1}{A}\sum_{i=1}^{A} argmax^{i}_{y\,\in\,T}\, f_S(x,y) \qquad (15)$$

where *argmaxⁱ* returns the pose with the i-th largest score. The motivation for averaging the top A poses is two-fold. First, averaging the poses with highest scores allows for interpolation of poses that are not in the training set are sshow in Fig. 5.

However, the average pose $\bar{y}$ is not guaranteed to be a valid pose. To address this problem, we use the annealing particle filtering (APF).

$$y^{*} = \underset{y\,\in\,\mathcal{Y}}{\arg\min} \Delta\,(\bar{y}, y) \qquad (16)$$

Finally, we have obtained the relationship using the triangle inequality, $\|y_{gt} - y^{*}\| \leq \|y_{gt} - \bar{y}\| + \|\bar{y} - y^{*}\|$.Therefore, minimizing $\Delta\,(\bar{y}, y^{*})$ is equivalent to minimizing an upper bound of the MPJPE between the ground-truth pose y_{gt} and the valid pose prediction y^{*}.

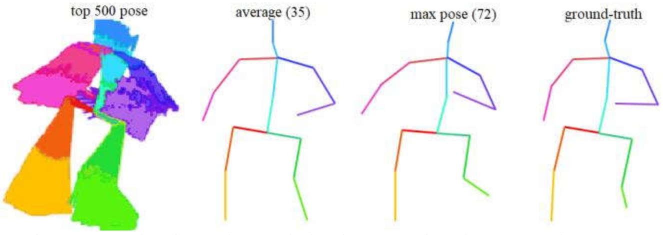

Fig. 5 Example of pose interpolation by averaging the top-scoring poses

III. Results

In this section, the maximum margin structured learning network is evaluated for human pose estimation. Human3.6M dataset was used for evaluation which contains around 3.6 million frames.

The input image is a cropped image around the human. The training images are re-sized by subtracting a square with the limiter provided by the Human3.6M data set. 112×112 sub-images are randomly selected for local translations from the training images. 3D pose entries are obtained with the body joints coordinate transformation of the human point cloud. The 3D pose input is a vector of the 3D coordinates of 17 joints.

The following inference methods are tested for pose estimation;

– **Max**: the training pose with maximum score in (14).

– **Avg**: the average of the top-500 training poses in (15), i.e., A = 500.

– **Avg-APF**: the valid pose after applying APFto the average pose in (16).

Table 1 shows the MPJPE results and the overall average in the test set for each action. Different estimation methods are compared for predicting pose. StructNet-Avg gives better results than StructNet-Max when all human action results are taken, and gives an overall reduction in error of about 10%. Furthermore, applying APF to the average pose from StructNet-Avg yields a valid pose with roughly the same MPJPE as StructNet-Avg.

Table 1. Human3.6m results: is calculated by MPJPE (mm) and standard deviation with parentheses.

Action	LinKDE (BS) [13]	Dconv [1] MPHML	StructNet-Max	StructNet-Avg	StructNet-Avg-APF
Walking	97 (37.1)	77.6 (23.5)	82.4 (27.4)	68.5 (21.4)	67.3 (22.2)
Discussion	183 (116.7)	148.7 (100.4)	147.9 (108.9)	133.1 (110.0)	132.9 (112.8)
Eating	132.5 (72.5)	104 (39.2)	108.7 (51.2)	97.9 (49.4)	96.1 (51.1)
Taking Photo	206.4 (112.6)	189 (93.9)	178.7 (93.5)	163.0 (90.6)	164.9 (92.9)
Walking Dog	177.8 (122.6)	146.6 (75.9)	146.0 (85.6)	131.3 (85.9)	131.3 (87.3)
Greeting	162.3 (88.4)	127.2 (51.1)	135.5 (64.7)	121.2 (61.8)	121.1 (64.5)
All	162.2 (104.4)	133.5 (81.3)	134.4 (86.6)	120.2 (85.6)	120.1 (87.9)

IV. Conclusion

In this paper, we propose a structured learning framework with deep neural network for human 3D pose estimation. The framework takes single human image and 3D pose (obtaining from classified 3D point cloud) as inputs and outputs a score value that represents a multiview similarity between the two inputs (whether they depict the same pose). The recurrent neural network takes both image-embedding and an initial pose as input and outputs a refined pose. The image and pose into a joint embedding, where the dot-product between the embeddings serves as the score function. The network using a max-margin cost function, which enforces a re-scaling margin between the score values of the ground-truth imagepose pair and other image-pose pairs. Finally, we demonstrade that the learned image-pose embedding encodes semantic attributes of the 3D pose, such as the orientation of the person and the position of the legs. Our proposed framework is general, and future work will consider applying it to other structured-output tasks. It is thought that the results obtained with the structural framework prepared by this study can be applied in future studies.

References

[1] S. Li, and A. B. Chan, *3d human pose estimation from monocular images with deep convolutional neural network*. In Asian Conference on Computer Vision, 2014, (pp. 332-347).

[2] A. Toshev, and C. Szegedy, *Deeppose: Human pose estimation via deep neural networks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, (pp. 1653-1660).

[3] S. Li, Z. Q. Liu, and A. B. Chan, *Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, (pp. 482-489).

[4] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, *Joint training of a convolutional network and a graphical model for human pose estimation.* In Advances in neural information processing systems, 2014, (pp. 1799-1807).

[5] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, *Learning human pose estimation features with convolutional networks*. 2013, arXiv preprint arXiv:1312.7302.

[6] P. F. Felzenszwalb, and D. P. Huttenlocher, *Pictorial structures for object recognition*. International journal of computer vision, 2005, 61(1), 55-79.

[7] D. Koller, and N. Friedman, *Probabilistic graphical models: Principles and techniques*. 2009, Cambridge: MIT Press.

[8] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, *Support vector machine learning for interdependent and structured output spaces*. 2004, In ICML

[9] J. A. Rodríguez, and F. Perronnin, *Label embedding for text recognition*. 2013, In BMVC

[10] C. Ionescu, L. Bo, and C. Sminchisescu, *Structural SVM for visual localization and continuous state estimation*. In ICCV 2009, (pp. 1157–1164).

[11] B. Sapp, and B. Taskar, *Modec: Multimodal decomposablemodels for human pose estimation*. In Proceedings of the IEEE conference on CVPR, 2013.

[12] Y. Yang, and D. Ramanan, *Articulated pose estimation with flexible mixtures-of-parts*. In CVPR, 2011, (pp. 1385 – 1392).

[13] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, *Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments*. IEEE TPAMI, 2014, 36(7), 1325–1339.

[14] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, *Better mixing via deep representations*. In ICML, 2013, (pp. 552–560).

[15] S. E. Nasab, S. Kasaei, E. Sanaei, A. Ossia, and M. Mobini, *Multiview 3D reconstruction and human point cloud classification*. 22nd Iranian Conference on Electrical Engineering (ICEE), 2014.

[16] "dijkstra algorithm." [Online]. Available: http://en.wikipedia.org/wiki/Dijkstra's_algorithm.

[17] "boost c++ library." [Online]. Available: http://www.boost.org/.

[18] J. Nation, *CRF Based Point Cloud Segmentation*. 2011.

[19] P. Krähenbühl and V. Koltun, *Efficient inference in fully connected crfs with gaussian edge potentials*, arXiv preprint arXiv:1210.5644, no. 4, pp. 1–4, 2012.

[20] A. Adams, J. Baek, and M. A. Davis, *Fast High-Dimensional Filtering Using the Permutohedral Lattice*. In Computer Graphics Forum 2010, (Vol. 29, No. 2, pp. 753-762).

[21] V. Nair, and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*. In Proceedings of the 27th international conference on machine learning 2010, (ICML-10) (pp. 807-814).

[22] Y. Sun, X. Wang, and X. Tang, *Deep learning face representation from predicting 10,000 classes*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, (pp. 1891-1898).

[23] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large margin methods for structured and interdependent output variables*. Journal of Machine Learning Research, 2005, 6, 1453–1484.

[24] S. Li, W. Zhang, and A. B. Chan, *Maximum-margin structured learning with deep networks for 3d human pose estimation*. In Proceedings of the IEEE International Conference on Computer Vision, 2015, (pp. 2848-2856). ISO 690