Structured Prediction of 3D Human Pose with Deep Neural Networks

Bugra Tekin*[1]
bugra.tekin@epfl.ch
Isinsu Katircioglu*[1]
isinsu.katircioglu@epfl.ch
Mathieu Salzmann[1]
mathieu.salzmann@epfl.ch
Vincent Lepetit[2]
lepetit@icg.tugraz.at
Pascal Fua[1]
pascal.fua@epfl.ch

[1] CVLab
EPFL,
Lausanne, Switzerland
[2] CVARLab
TU Graz,
Graz, Austria

Most recent approaches to monocular 3D pose estimation rely on Deep Learning. They either train a Convolutional Neural Network to directly regress from image to 3D pose [3], which ignores the dependencies between human joints, or model these dependencies via a max-margin structured learning framework [4], which involves a high computational cost at inference time. In this paper, we introduce a Deep Learning regression architecture for structured prediction of 3D pose from monocular images that relies on an overcomplete auto-encoder to learn a high-dimensional latent pose representation and account for joint dependencies.

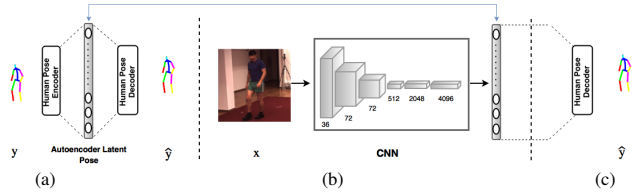

Figure 1: **(a)** An overcomplete denoising auto-encoder is trained. **(b)** A CNN is mapped into the latent representation learned by the auto-encoder. **(c)** The latent representation is mapped back to the original pose space using the decoder.

For this purpose, we first train an overcomplete auto-encoder that projects joint positions to a high dimensional space represented by its middle layer, as depicted by Fig. 1(a). We then learn a CNN-based mapping from the input image to this high-dimensional pose representation as shown in Fig. 1(b). This is inspired by Kernel Dependency Estimation (KDE) [2, 5], which maps both input and output to high-dimensional Hilbert spaces via kernel functions and learns a mapping between these spaces. In that, it can be understood as replacing kernels by the auto-encoder layers to predict the pose parameters in a high dimensional space that encodes complex dependencies between different body parts. As a result, it enforces implicit constraints on the human pose, preserves the body statistics, and improves prediction accuracy. Finally, as in Fig. 1(c), we connect the decoding layers of the auto-encoder to this network, and fine-tune the whole model for pose estimation. Our contribution is to show that combining traditional CNNs for supervised learning with auto-encoders for structured learning preserves the power of CNNs while also accounting for dependencies, resulting in increased performance.

Using Auto-Encoders to Learn Structured Latent Representations: We use a denoising auto-encoder that can have one or more hidden layers to model the dependencies between joints. We train our auto-encoder to take as input a noisy pose vector, $\tilde{y}$, and return a denoised y as output.

To learn the network parameters, θ_{ae}, we rely on minimizing the square loss between the reconstruction obtained by the auto-encoder mapping function, $f_{ae}(\tilde{y}, \theta_{ae})$, and the original input, y, over the N training examples. To increase robustness to small pose changes, we regularize the cost function by adding the squared Frobenius norm of the Jacobian of the hidden mapping $g(\cdot)$, that is, $J(\tilde{y}) = \frac{\partial g}{\partial \tilde{y}}(\tilde{y})$ where $g(\cdot)$ is the encoding function that maps the noisy input $\tilde{y}$ to the middle hidden layer, h_L. Training can thus be expressed as finding

$$\theta^*_{ae} = \underset{\theta_{ae}}{\operatorname{argmin}} \sum_i^N ||y_i - f_{ae}(\tilde{y}_i, \theta_{ae})||_2^2 + \lambda \|J(\tilde{y}_i)\|_F^2 \,, \qquad (1)$$

where λ is the regularization weight. Unlike when using KDE, we do not need to solve a complex pre-image problem to go from the latent pose representation to the pose itself. This mapping, which corresponds to the decoding part of our auto-encoder, is learned directly from data.

* indicates equal contribution

Regression in Latent Space: Once the auto-encoder is trained, we aim to learn a mapping between the image and the latent representation of the human pose. To this end, we make use of a CNN to regress the image, x, to the high-dimensional representation that was previously learned by the auto-encoder, h_L, with a mapping function $f_{cnn}(\cdot)$. Given N training examples, learning amounts to finding the model parameters, θ_{cnn}, by

$$\theta^*_{cnn} = \underset{\theta_{cnn}}{\operatorname{argmin}} \sum_i^N ||f_{cnn}(x_i, \theta_{cnn}) - h_{L,i}||_2^2 \,. \qquad (2)$$

Fine-Tuning the Whole Network: Finally, as shown in Fig. 1(c), we append the decoding layers of the auto-encoder to the CNN discussed above, which reprojects the latent pose estimates to the original pose space. We then fine-tune the resulting complete network for the task of human pose estimation. Denoting the complete set of model parameters by θ_{ft}, and the mapping function by $f_{ft}(\cdot)$, we minimize the squared difference between the predicted and ground-truth 3D poses.

$$\theta^*_{ft} = \underset{\theta_{ft}}{\operatorname{argmin}} \sum_i^N ||f_{ft}(x_i, \theta_{ft}) - y_i||_2^2 \,. \qquad (3)$$

Results: We evaluate our method on the Human3.6m dataset [2] and report our results along with three state-of-the-art approaches [2, 3, 4] in Table 1. Our method consistently outperforms all the baselines. Fig. 2 depicts example pose estimation results on Human3.6m.

Following [1], we show in Table 2 the differences between the ground-truth limb ratios and the limb ratios obtained from predictions based on KDE, CNN regression and our approach. These results evidence that our predictions better preserve these limb ratios, and thus better model the dependencies between joints.

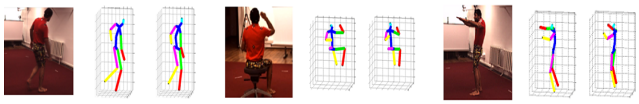

Figure 2: Example 3D pose estimation results of our approach. First skeleton depicts the ground-truth pose and the second one our prediction.

Model	Discussion	Eating	Greeting	Taking Photo	Walking	Walking Dog
LinKDE([2]	183.09	132.50	162.27	206.45	97.07	177.84
DconvMP-HML [3]	148.79	104.01	127.17	189.08	77.60	146.59
StructNet-Max [4]	149.09	109.93	136.90	179.92	83.64	147.24
StructNet-Avg [4]	134.13	97.37	122.33	166.15	68.51	132.51
OURS	**129.06**	**91.43**	**121.68**	**162.17**	**65.75**	**130.53**

Table 1: Average Euclidean distance in mm between the ground-truth 3D joint locations and those predicted by [2, 3, 4] and ours.

Model	Lower Body	Upper Body	Full Body
KDE [2]	1.02	7.18	16.43
CNN	0.57	6.86	14.97
OURS no FT	0.62	5.30	11.99
OURS with FT	0.77	5.43	11.90

Table 2: Sum of the log of limb length ratio errors for different parts of the human body.

[1] C. Ionescu, F. Li, and C. Sminchisescu. Latent Structured Models for Human Pose Estimation. In *ICCV*, 2011.

[2] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *PAMI*, 2014.

[3] S. Li and A.B. Chan. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In *ACCV*, 2014.

[4] S. Li, W. Zhang, and A. B. Chan. Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation. In *ICCV*, 2015.

[5] J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik. Kernel Dependency Estimation. In *NIPS*, 2002.