Pose Estimation on Depth Images with Convolutional Neural Network

Jingwei Huang
Stanford University
jingweih@stanford.edu

David Altamar
Stanford University
daltamar@stanford.edu

Abstract

In this paper we propose a method of human pose estimation with single depth image based on convolutional neural network. We build a dataset that contains various human poses, captured in different camera view with different human body shapes, considering no such depth images dataset is large enough for the deep learning task. With the large scale of the dataset and power of convnet, our model can do holistic reasoning for occluded human joints, and handles full range of human body shapes in different camera view. We introduce an original loss function that fits our 3D joints estimation task. Our loss function is simple, but is demonstrated to be powerful to describe the hierarchical information of the human pose. We benefit from this strategy to learn accurate pose estimation model.

1. Introduction

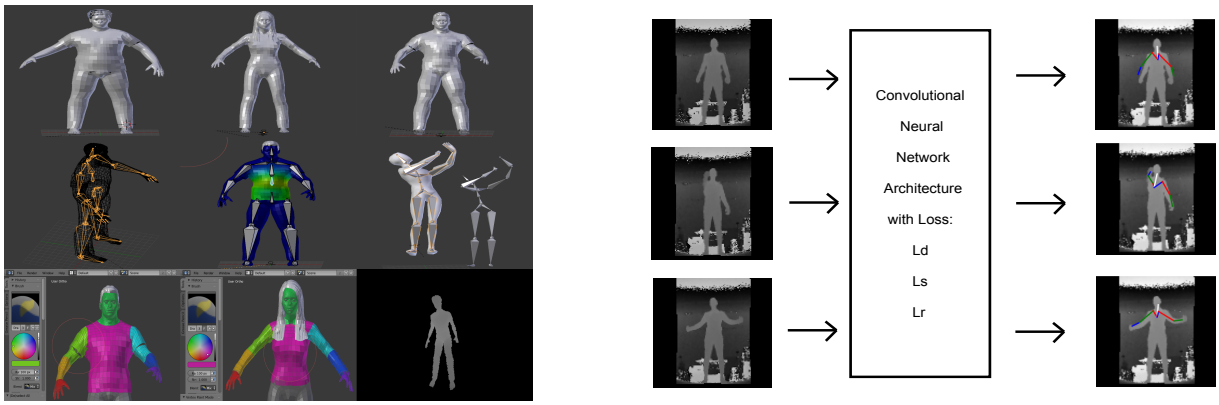
Pose estimation is a well-known research topic in computer vision, with many applications in the real world like human activity analysis, human-computer interaction, gaming and security. Precise estimation of human pose remains challenging in noisy environments with human joints occluded. Previous works provide various approaches for human pose estimation in both color and depth images.

For color images, local features are used to model articulation to match human joints [5][15]. Holistic reasoning techniques [13][20] are proposed to analyze occluded human joints. With the recent technique of convolutional neural networks, deep neural network architectures improve task dramatically with high-level feature extraction [25]. The challenge of pose estimation on 2D images is obvious. Skeleton information in 3D space is destroyed in 2D images: Length of limbs are invariant with different poses in 3D space, but can change a lot in image coordinates. Although some of skeleton information like parent-child joint dependencies are carefully analyzed with Markov Field [24], these dependencies are still hard to use considering poses captured in different angles.

The availability of depth camera make it possible for 3D human skeleton analysis on depth images [7][10]. Microsoft released a Kinect System that predict 3D human joints in real-time [21] with random forest. Instead of predicting human joints, Jiu *et al.* [9] used a convolutional neural network to do human body segmentation. However, the lack of dataset limit its ability to do segmentation with high accuracy. Depth images have advantage over color images that it segmented objects well by depth and are not influenced by different light conditions. Depth images allow 3D reconstruction directly from image coordinates and depth values. Therefore, human hierarchy information can be utilized in 3D space.

However, no previous work tries to apply deep architectures on depth images to predict 3D human joints, probably due to the unavailability of 3D joints and depth images dataset. To handle this problem, we carefully build a synthetic depth images dataset. The dataset contains depth images of 12 different human models making general poses in captured with different camera view. Fig. 1 (a) shows our 3D human models with annotated and rendered depth images. We further adjust the convnet [11] used for pose estimation in color images to our task, by modifying the architecture and proposing an original loss function. Fig. 1 (b) shows depth images sent to convnet and predicted results.

The originality of our loss function is to use simple metric to incorporate human hierarchy information in 3D space. We notice that lengths of limbs are invariant to poses for each human. Therefore, our network should be able to learn the knowledge of limb length of a human. We also notice that rotations of articulations are independent to each other in 3D space. There are two potential ways to utilize these facts. We may directly predict the limb lengths and directions of joints relative to its parent which is denoted by angles. However, this task is much harder than directly predicting 3D joint positions. Intuitively, there is a linear function to map joint positions on to the image space, while mapping of limbs lengths and angles can be nonlinear and much more complex. Instead, we minimize the errors of limbs lengths, joints locations and rotation of each articulation relative to its parent with linear combination of



(a) Dataset Examples (b) Pose Estimation Pipeline

Figure 1. Framework Overview: (a) we build a depth images dataset which contains poses from 12 different humans captured in different views. (b) We use a deep convnet architecture to predict 3D joint positions from depth images. We minimize the losses of limbs lengths, joints angles and joints distances.

three L2-loss functions. We carefully compare different prediction methods and loss functions to demonstrate that our choices can maximize the prediction accuracy.

2. Related Work

Human pose estimation can be solved in several ways. The idea of Pictorial Structures is introduced by Fishler *et al.* [5]. They view human pose as a tree consisting of different human parts. Researchers then focused on finding features to represent this tree-based model on images [19][4][1]. Yang *et al.* [26] design a general model to capture contextual co-occurrence relations between parts, which is better to represent spatial relations and local rigidity.

Toshev *et al.* [25] first introduce ConvNet into the task of human pose estimation. They use multiple ConvNet regressors to refine the positions of human joints. Carreira *et al.* [2] combine joint heat map with color images and introduce a top-down feedback strategy into the ConvNet training process to refine the joint positions. Tompson *et al.* [24] integrate joint dependencies into ConvNet by introducing Markov Random Field. Temporal information can also be used to help the estimation. Pfister *et al.* [17] apply ConvNet in the task of human pose estimation in video by taking multiple neighbor frames into the input layer. A per-video mean and training augmentation strategy is used to alleviate the inference of the background. Pfister *et al.* [16] uses temporal information in video by analyzing the optical flow. It is used to reinforce the heat map of human joints and reduce the prediction of kinematically impossible poses. Pose estimations can also be analyzed in a higher level. Li *et al.* [11] propose a multi-task CNN framework that can train 3D joint location and body part segmentation at the same time. The parameters of the framework is acquired from pre-trained CNN in the object segmentation task. Lillo *et al.* [12] learn human activities by analyzing poses and actions in different semantic levels. They estimate the poses and classify them to the actions. By analyzing actions distributed in temporal and spacial dimensions, they acquire complex human activities. Because of the limitation of color images, none of these architectures try to analyze human hierarchy information in 3D space. Our method shows that hierarchy information can be analyzed easily in 3D space.

Other works focus on pose estimation on depth images [7][10][6][18][27][22]. Shotton *et al.* [21] recognize human pose from single depth image by solving a per-pixel body par classification problem using random forest classifier. Their big dataset allows their predictions to be invariant to body shape, pose and cloths. Jiu *et al.* [9] used a convolutional neural network to do human body segmentation on depth images. However, they train their results using the CDC4CV dataset [8], which is too small to train a deep Convnet. We build a large synthetic dataset in order to train a Convnet that is robust for human joints estimation on depth images.

3. Approach

In this section, we introduce our approach to learn the human pose and to build a synthetic depth images dataset.

3.1. Architecture

Motivated by the Li *et al.* [11], we propose our Convnet architecture as shown in Fig. 2. We use a similar architecture with Li *et al.* [11]. Differently, we take depth images with a single channel instead of three. We omit the normalization strategy proposed by them after conv2. While they use this strategy to make the Convnet robust to various intensity of the image, we want to keep the real depth values in the depth image. Because of the availability of our large synthetic dataset, we increase the parameter space by re-

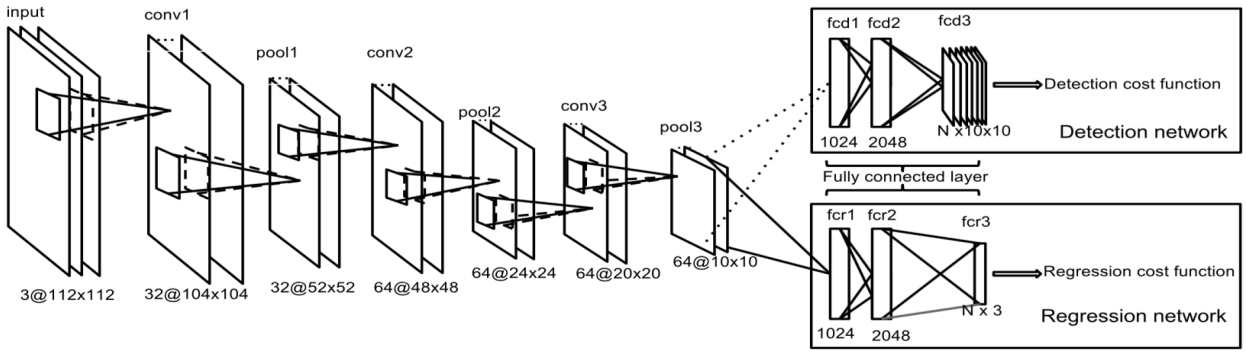


Figure 2. Convnet Architecture: we use a similar architecture with Li *et al.* [11]. Differently, we take depth images with a single channel instead of three. We omit the normalization strategy proposed by them after conv2. We replace depth of conv3 and pool3 with 128, in order to increase our ability to handle various poses estimation.

placing depth of conv3 and pool3 (64 in their network) with 128, in order to increase our power to handle various poses.

3.2. 3D Human Pose Estimation Prediction

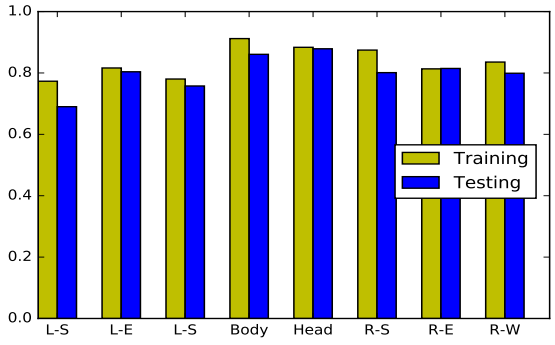
Given a depth image I, our task is to estimate the human pose, which is described by joint positions in the camera coordinate system $J_i^w = (J_{i,x}^w, J_{i,y}^w, J_{i,z}^w)$. The joint position relative to its parent can be denoted as $J_i^p = J_i^w - J_{p(i)}^w$, where $p(i)$ means the parent of joint i. We define the ground truth of J as $\hat{J}$.

Li *et al.* directly learn the relative 3D positions J_i^p, given the reason that length of J_i^p usually remain constant, which narrows the range of prediction. However, our experiment suggests prediction of 3D joints directly (J_i^w) performs better results than J_i^p. One possible explanation is that our Convnet learns high level features of joints in the depth images and it is more straightforward for the neural network to locate absolute locations from these features rather than taking into account hierarchy information and learn relative locations.

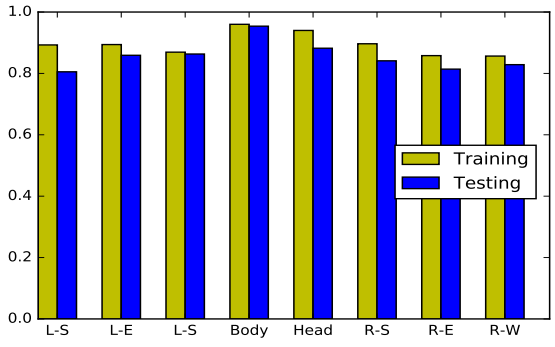
Therefore, we directly predict 3D joint positions in camera coordinate system J_i^w, which is similar to previous Convnet works [25][2] on color images where outputs are joint locations in the image coordinate system.

Prediction Analysis: In order to decide the best way to predict joints locations, we do two different experiments. Both the experiments are trained on our 550k-images training set and tested on our testing set. In the first experiment, we use our architecture but replace the loss function to L_p. Second, we change the output of our Convnet to be relative joint positions J_i^p, and do the training and testing again with loss function as L_p. By fixing the loss function, we can easily compare these two ways of joint predictions. We show the results of training and testing accuracy (defined in Sec. 5) with every epoch in Fig. 3.

We can conclude from Fig. 3 that with the same architecture and loss function, higher training and testing accuracy can be achieved by directly predicting absolute joint locations instead of relative joint locations. Thus we believe it is better to predict J_i^w than J_i^p.



(a) Training and Testing Accuracy with J_i^w



(b) Training and Testing Accuracy with J_i^p

Figure 3. Prediction Analysis: We show training and testing accuracy with predictions of J_i^w and J_i^p on synthetic data. By predicting absolute joint locations, we can get higher training and testing accuracy.

Train/Val Accuracy		α				
		10	20	30	40	50
β	1.0	0.831 / 0.793	0.828 / 0.804	0.843 / 0.816	0.848 / 0.823	0.837 / 0.815
	3.0	0.843 / 0.812	0.839 / 0.808	0.849 / 0.827	0.857 / 0.833	0.852 / 0.831
	5.0	0.858 / 0.830	0.866 / 0.824	**0.883 / 0.846**	0.876 / 0.835	0.867 / 0.829
	7.0	0.853 / 0.826	0.859 / 0.831	0.871 / 0.841	0.862 / 0.830	0.854 / 0.834
	9.0	0.823 / 0.805	0.819 / 0.793	0.802 / 0.784	0.809 / 0.796	0.794/ 0.778

Table 1. Train/Validation Accuracy with different α and β as hyperparameters

3.3. 3D Loss Functions

The most commonly used loss function L_d is described as the sum L2-distance of joint position errors in Eq. 1. The motivation of L_d is that prediction of joint positions should be as closer as possible to the ground truth. However, it does not consider human hierarchy information and is not straightforward to guide the Convnet towards better parameters.

$$L_d = \sum_i ||J_i^w - \hat{J}_i^w||^2 \tag{1}$$

Li *et al.* propose a relative loss function L_r, the sum of L2-distance of relative joint position errors, as shown in Eq. 2. This loss is better at measuring the wellness of joint prediction relative to its parent. However, they have a strong assumption that the root of the joint is always at the origin, which limits their loss to handle absolute 3D joint locations. In addition, the localization of joints can still be bad even if $e(J_i^p)$ is zeros, mostly because the estimation of parent joints is not well enough. Thus, the relative loss itself may not be good enough to train the Convnet.

$$L_p = \sum_i ||J_i^p - \hat{J}_i^w||^2 \tag{2}$$

The hierarchy information of a human pose can be denoted as limb lengths and rotations of limb lengths relative to its parent. Intuitively, the wellness of prediction for relative joint conditions depends on the loss of the hierarchy information. We define loss of limb length as the sum of L2-distances of limb length prediction (Eq. 3). For rotation loss, we use a cosine distance metric, as shown in Eq. 4.

$$L_s = \sum_i (||J_i^p|| - ||\hat{J}_i^p||)^2 \tag{3}$$

$$L_r = \sum_i (1 - \frac{J_i^p \cdot \hat{J}_i^p}{||J_i^p|| \cdot ||\hat{J}_i^p||}) \tag{4}$$

We combine the loss of joint locations and relative joint conditions by taking the linear combination of L_d, L_s and L_r, as shown in Eq. 5.

$$L = L_d + \alpha L_s + \beta L_r \tag{5}$$

Because it is hard to analytically determine the relationship between the three terms, we take α and β as hyperparameters and using experiment to achieve best results.

It is important to get reasonable hyper-parameters in order to make our loss function work in the best way. A somewhat good model usually have relatively small loss of limbs length L_s compared to L_d and L_r. Therefore, α is much bigger than β. Table 1 shows the training the testing accuracy after 500 epochs with different α and β.

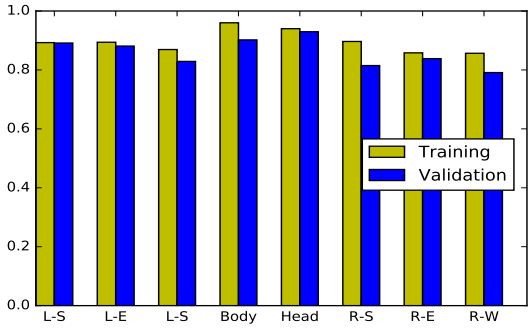
Based on the training/validation accuracy corresponding to different (α, β), we choose $\alpha = 30$ and $\beta = 5$ to achieve best prediction performance.

In order to see the contribution of L_d, L_s and L_r, we train four different models guided by four different loss functions: L, L_d, L_p, $\alpha \cdot L_s + \beta \cdot L_r$. Training and validation accuracies corresponding to these four loss functions at different epochs are shown in Fig. 4 (a), (b), (c) and (d) respectively.

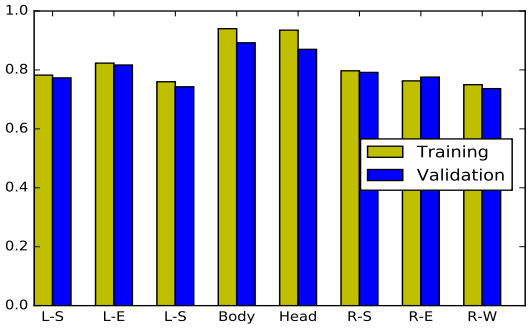
Comparing the Fig. 4 (c) and (d), we find that training and validation accuracy of our loss related to relative joints conditions is better than the loss proposed by Li *et al.* [11]. Also, the combination of absolute loss and that of relative joint conditions (Fig. 4 (a)) can achieve better results than separate losses (Fig. 4 (b) and (d)).

Fig. 5 gives an example where L has advantage over $\alpha \cdot L_s + \beta \cdot L_r$. This pose is also predicted by models trained by L_d, L and $\alpha \cdot L_s + \beta \cdot L_r$. We find that hands of prediction by Fig. 5 (c) is far from the ground truth. Fig. 5 (a) and (b) handles this situation better, because they take the absolute joint positions into consideration.

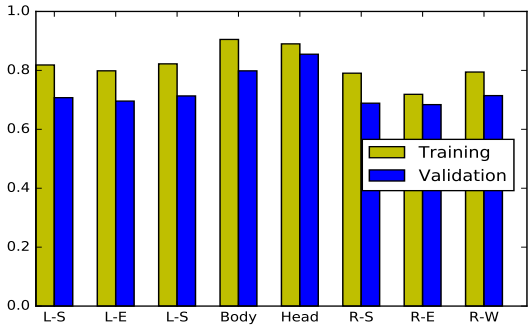
According from the discussions above, we conclude that our new loss function is better at representing the human hierarchy information in 3D space than Li *et al.*'s loss. Our combination of absolute loss and relative loss can further improve the results, possibly because it balances the information given by local area of the images to represent the joint positions, and higher level features in the image that implicitly denotes the structural information of human skeleton.



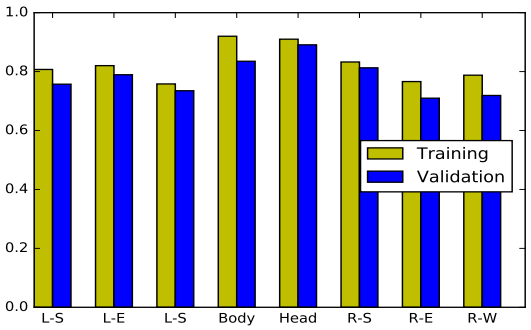
(a) Training/Validation Accuracy with L



(b) Training/Validation Accuracy with L_d

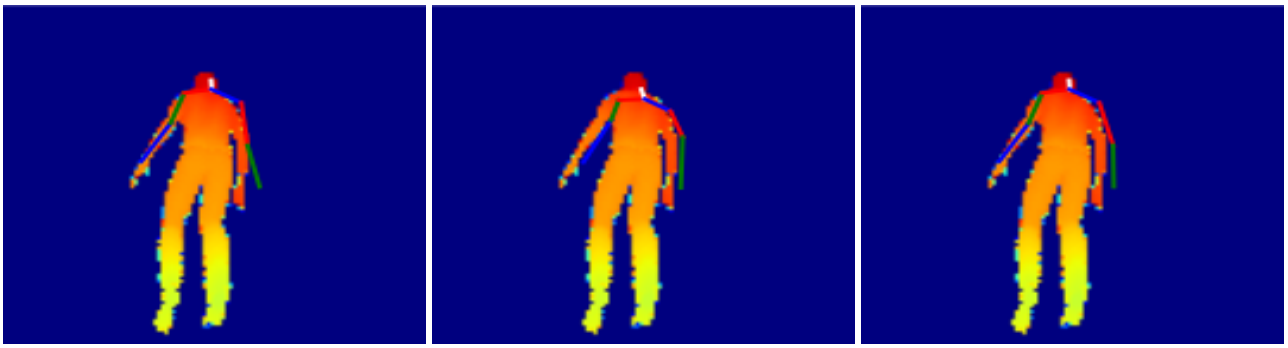


(c) Training/Validation Accuracy with L_p



(d) Training/Validation Accuracy with $\alpha \cdot L_s + \beta \cdot L_r$

Figure 4. Training/Validation Accuracy with different Loss Metrics. Accuracy in (d) shows better results than that in (c), suggesting that our relative joint condition loss can better represent the human hierarchy information. Loss L has the best performance.



(a) L_d prediction (b) L prediction (c) $\alpha \cdot L_s + \beta \cdot L_r$ prediction

Figure 5. Example pose predicted by models trained with different loss functions where L dominate $\alpha \cdot L_s + \beta \cdot L_r$.

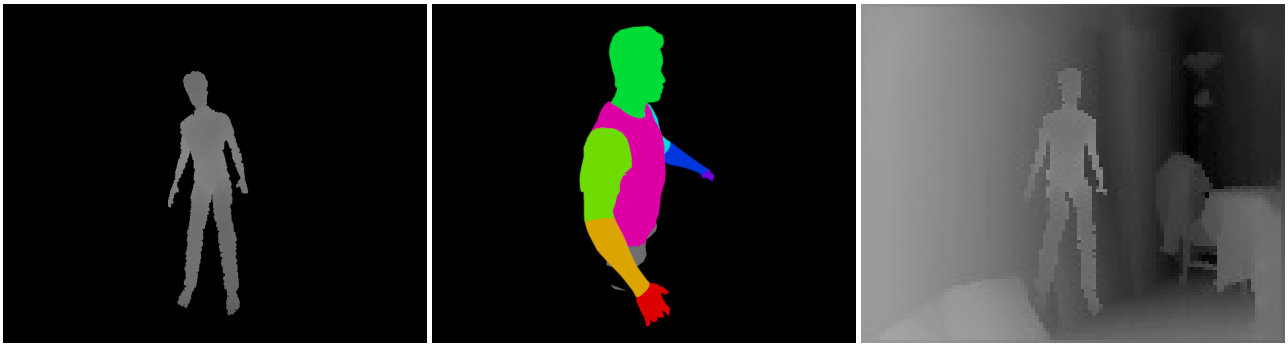
4. Dataset

Convnet training requires a large scale dataset. CDC4CV dataset [8] contains human poses and depth images, but the scale of the dataset is considered too small. Li *et al.* also provide a dataset with depth images and human joints, but it is also not large enough, and the joint positions are relative to the root rather than absolute. Shotton *et al.* generate a synthetic dataset to train the random forest. However, their dataset is not public available. Considering this, we generate our own dataset with twelve human models doing different poses.

We use MakeHuman software [23] to create humans with different body shapes, as shown in Fig. 8. In order to combine human models with various different poses, we

(a) Skeleton for Human Model (b) Retargeting to Mocap Data (c) Labeling Different Parts

Figure 6. Human Animation: (a) We create a skeleton for each human model. (b) We retarget the created human model with Motion Capture data. (c) We paint different body parts with different colors for segmentation task in the future.



(a) Depth Image (b) Per-Pixel Labeling (c) Background

Figure 7. Dataset Results: (a) We create kinect-simulated depth image. (b) We create per-pixel labeling of different body parts. (c) We blend depth images with background.

Figure 8. Twelve human models created by MakeHuman.

include the technique of skinning mesh animations. We create a skeleton for each human model (Fig. 6 (a)) and retarget it to Motion Capture data (Fig. 6 (b)) as provided by Carnegie-Mellon Graphics Lab Motion Capture Database [3]. In order to compare with Shotton et al. [21], we enable our dataset to train a related task of body segmentation required by their method. Fig. 6 (c) shows our labeling of the model.

When generating the dataset, we export eight joint positions in camera coordinate system, a depth image (Fig. 7 (a)) and a per-pixel labeled color image (Fig. 7 (b)) for each camera view of a pose. We capture each pose from four different camera views. After that, we blend every depth image with randomly picked background depth image from dataset provided by Silberman *et al.* [14], as shown in Fig. 7 (c).

As a result, our dataset contains 650k human poses, with 68 subjects from 13 to 80 from the mocap data [3]. The scale of the dataset is large enough for our Convnet to achieve good results.

4.1. Splitting the Synthetic Dataset

To demonstrate the generality of our model, we ensure that our training and testing dataset are well splited by motion categories and body shapes. Therefore, we select 550k images from 11 models in the first 62 mocap subjects (13-74) as training set. Validation and testing sets are both composed of 50k images containing the left model with mocap subjects (75-77) and (78-80) respectively. This spliting

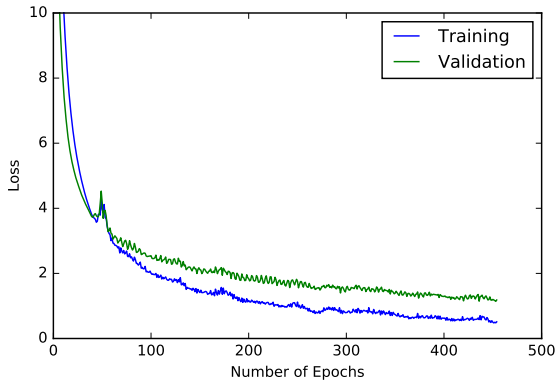


Figure 9. Training/Validation Loss Results on synthetic dataset

method ensures that the prediction is robust to general poses performed by new body shapes not included in the training set.

5. Results

In this section, we show several experiments to demonstrate the advantage of our approach and evaluate our results. Our experiments are performed on both our synthetic dataset and a real dataset provided by Ganapathi *et al.* [6]. We also train on our synthetic dataset, and then test the model with the testing set of our synthetic data and real dataset [6]. We evaluate the accuracy of joint positions by the proportion of predictions with error less than D=0.1m, as proposed by Shotton *et al.* [21]. We also use their mean average prediction metric to compare different methods.

5.1. Performance on Synthetic dataset

We train and test our model on the synthetic dataset. We show our loss on training, validation sets in Fig. 9. Our prediction results are shown in Fig. 10. As a result, our model can handle poses included in the synthetic dataset well. Even there are occlusions, our model may use holistic reasoning to predict a somewhat plausible result.

5.2. Comparison

In order to ensure that our synthetic dataset can be powerful enough to deal with real-world applications. We train and perform prediction results on a real depth images dataset provided by Ganapathi *et al.* [6]. Our loss and accuracy on training, validation and testing sets is shown in Fig. 13. We compare our performance with Shotton *et al.* [21] and Ganapathi *et al.* [6], and show the accuracies with different body parts in Fig. 11.

As a result, our approach is robust and able to predict real data with high accuracy. Some challenging examples can be well predicted using our algorithm (Fig. 12). The performance of our results is better than previous methods.

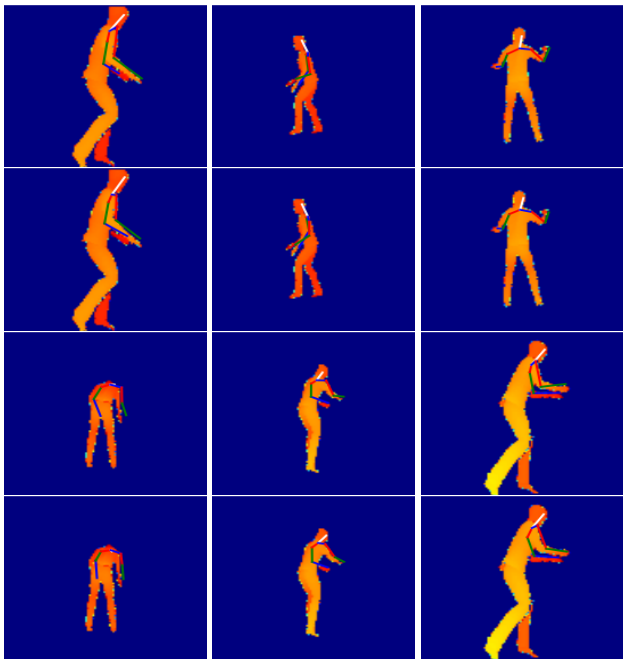


Figure 10. Examples that are well predicted by our approach in synthetic dataset. The first and third rows are predicted results. The second and fourth rows are from ground truth.

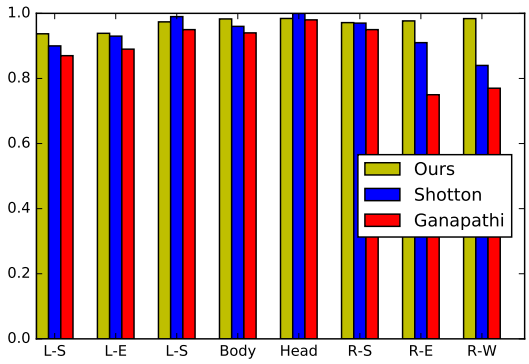


Figure 11. Comparison between our approaches with the state-of-art results on Ganapathi *et al.*'s dataset. Our mAP on this dataset is 95.4%. It shows that our approach can perform better prediction of different joint locations than previous methods.

We use the accuracy metric discussed in Sec. 5 to test our performance on our synthetic dataset and real dataset. As shown in Fig. 11, our mAP on the real dataset is 95.4%. We also show our train/validation loss and accuracy for our synthetic dataset in Fig. 13 (a) and (b).

However, we should admit that dealing with occlusion and prediction of hand is still a hard task. Our approach can fail in some cases, as shown in Fig. 14. To conclude, our model fails most often in two situations.

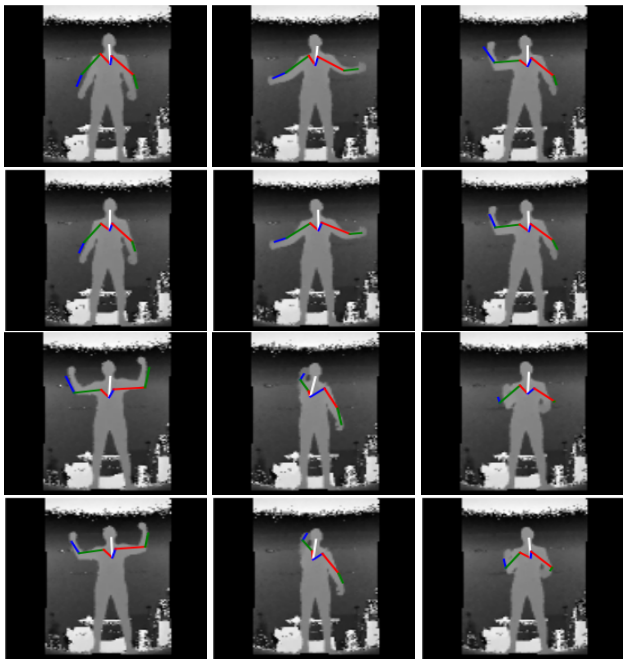


Figure 12. Examples that are well predicted by our approach in real dataset. The first and third rows are predicted results. The second and fourth rows are from ground truth.

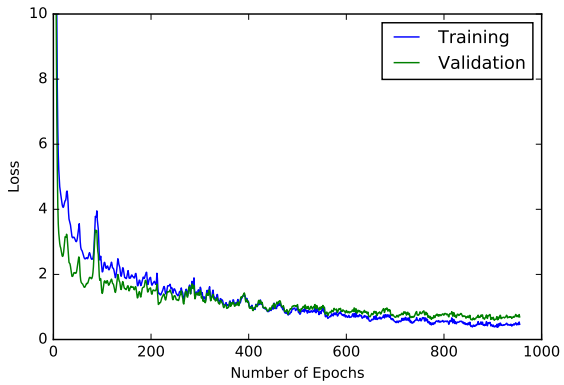


Figure 13. Training/Validation Loss on real dataset

- Elbows or hands are occluded by the body.
- Hands are too small for the model to analyze.

6. Conclusion and Future Work

In this paper, we introduce the convolutional neural network into the task of learning human pose from a single depth image. We carefully analyze and demonstrate the advantage of our prediction algorithm and choice of loss function. We build a large synthetic dataset which makes our algorithm robust to different human shapes, various backgrounds, different camera views and even occlusions.

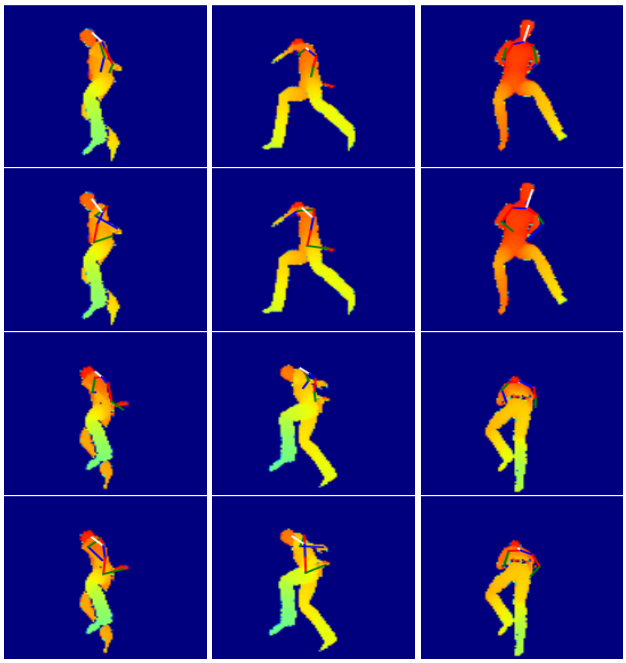


Figure 14. Cases where our approach fails in prediction. The first and third rows are predicted results. The second and fourth rows are from ground truth.

However, many approaches can be applied to depth image human estimation task. Multi-task learning of body segmentation and joint regression is worth trying. We also believe that incorporating the 3D human hierarchy information into the Convnet architecture may further improve the performance.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009.

[2] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015.

[3] CMU. Carnegie-mellon graphics lab motion capture database. `https://sites.google.com/a/cgspeed.com/cgspeed/motion-capture/daz-friendly-release`.

[4] M. Eichner, V. Ferrari, and S. Zurich. Better appearance models for pictorial structures. In *BMVC*, volume 2, page 5, 2009.

[5] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1):67–92, 1973.

[6] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 755–762. IEEE, 2010.

[7] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. In *Pattern recognition*, pages 285–292. Springer, 2005.

[8] B. Holt, E.-J. Ong, H. Cooper, and R. Bowden. Putting the pieces together: Connected poselets for human pose estimation. In *1st IEEE Workshop on Consumer Depth Cameras for Computer Vision (ICCV'11)*, Nov. 2011.

[9] M. Jiu, C. Wolf, G. Taylor, and A. Baskurt. Human body part estimation from depth images via spatially-constrained deep learning. *Pattern Recognition Letters*, 50:122–129, 2014.

[10] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1686–1691. IEEE, 2006.

[11] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Computer Vision–ACCV 2014*, pages 332–347. Springer, 2014.

[12] I. Lillo, A. Soto, and J. Niebles. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 812–819, 2014.

[13] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Computer VisionECCV 2002*, pages 666–680. Springer, 2002.

[14] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[15] R. Nevatia and T. O. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98, 1977.

[16] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[17] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Computer Vision–ACCV 2014*, pages 538–552. Springer, 2014.

[18] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113. IEEE, 2010.

[19] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2006.

[20] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 750–757. IEEE, 2003.

[21] 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.

[22] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2010.

[23] M. Team. Makehuman software. `http://www.makehuman.org`.

[24] 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*, pages 1799–1807, 2014.

[25] 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*, pages 1653–1660, 2014.

[26] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[27] Y. Zhu and K. Fujimura. Constrained optimization for human pose estimation from depth sequences. In *Computer Vision–ACCV 2007*, pages 408–418. Springer, 2007.