Patient MoCap: Human Pose Estimation under Blanket Occlusion for Hospital Monitoring Applications

Felix Achilles[1,2], Alexandru-Eugen Ichim[3], Huseyin Coskun[1], Federico Tombari[1,4], Soheyl Noachtar[2], and Nassir Navab[1,5]

[1] Computer Aided Medical Procedures, Technische Universität München, Germany
[2] Department of Neurology, Ludwig-Maximilians-University of Munich, Germany
[3] Graphics and Geometry Laboratory, EPFL, Switzerland
[4] DISI, University of Bologna, Italy
[5] Computer Aided Medical Procedures, Johns Hopkins University, USA

Abstract. Motion analysis is typically used for a range of diagnostic procedures in the hospital. While automatic pose estimation from RGB-D input has entered the hospital in the domain of rehabilitation medicine and gait analysis, no such method is available for bed-ridden patients. However, patient pose estimation in the bed is required in several fields such as sleep laboratories, epilepsy monitoring and intensive care units. In this work, we propose a learning-based method that allows to automatically infer 3D patient pose from depth images. To this end we rely on a combination of convolutional neural network and recurrent neural network, which we train on a large database that covers a range of motions in the hospital bed. We compare to a state of the art pose estimation method which is trained on the same data and show the superior result of our method. Furthermore, we show that our method can estimate the joint positions under a simulated occluding blanket with an average joint error of 7.56 cm.

Keywords: pose estimation, motion capture, occlusion, CNN, RNN, random forest

1 Introduction

Human motion analysis in the hospital is required in a broad range of diagnostic procedures. While gait analysis and the evaluation of coordinated motor functions [1, 2] allow the patient to move around freely, the diagnosis of sleep-related motion disorders and movement during epileptic seizures [3] requires a hospitalization and long-term stay of the patient. In specialized monitoring units, the movements of hospitalized patients are visually evaluated in order to detect critical events and to analyse parameters such as lateralization, movement extent or the occurrence of pathological patterns. As the analysis of patient movements can be highly subjective [4], several groups have developed semi-automatic methods in order to provide quantified analysis. However, in none of the above works, a full body joint regression has been attempted, which would be necessary for

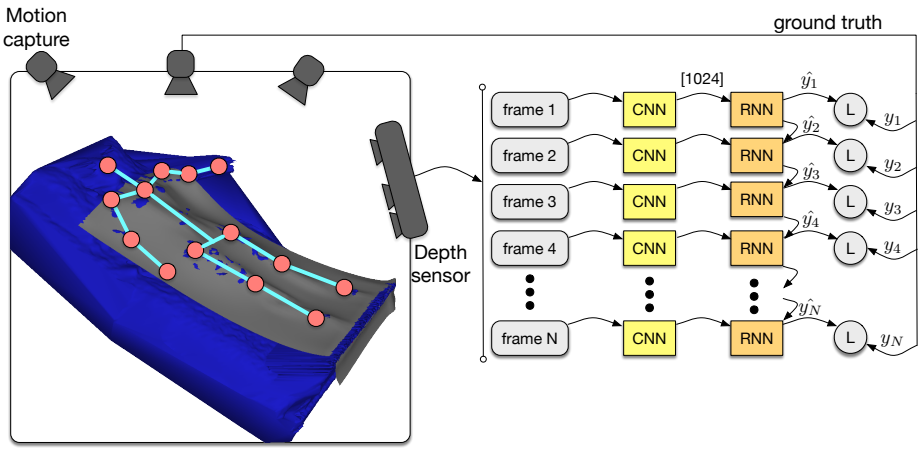

Fig. 1. Data generation and training pipeline. Motion capture (left) allows to retrieve ground truth joint positions y, which are used to train a CNN-RNN model on depth video. A simulation tool was used to occlude the input (blue) with a blanket (grey), such that the system can learn to infer joint locations $\hat{y}$ even under blanket occlusion.

automatic and objective quantification of patient movement. In this work, we propose a new system for fully automatic continuous pose estimation of hospitalized patients, purely based on visual data. In order to capture the constrained body movements in the hospital bed, we built up a large motion database that is comprised of synchronized data from a motion capture system and a depth sensor. We use a novel combination of a deep convolutional neural network and a recurrent network in order to discriminatively predict the patient body pose in a temporally smooth fashion. Furthermore, we augment our dataset with blanket occlusion sequences, and show that our approach can learn to infer body pose even under an occluding blanket. Our contributions can be summarized as follows: 1.) proposing a novel framework based on deep learning for real time regression of 3D human pose from depth video, 2.) collecting a large dataset of movement sequences in a hospital bed, consisting of synchronized depth video and motion capture data, 3.) developing a method for synthetic occlusion of the hospital bed frames with a simulated blanket model, 4.) evaluating our new approach against a state-of-the-art pose estimation method based on Random Forests.

2 Related Work

Human pose estimation in the hospital bed has only been approached as a classification task, which allows to estimate a rough pose or the patient status [5, 6]. Li et al. [5] use the Kinect sensor SDK in order to retrieve the patient pose and estimate the corresponding status. However, they are required to leave the test subjects uncovered by a blanket, which reduces the practical value for real hospital scenarios. Yu et al. [6] develop a method to extract torso and head locations and

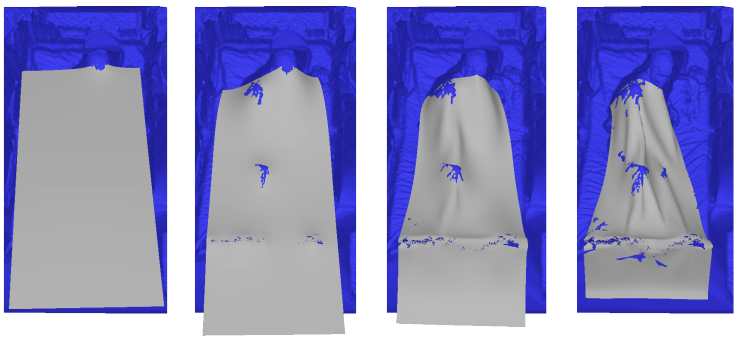

Fig. 2. Snapshots of iterations of the physics simulation that was used to generate depth maps occluded by a virtual blanket.

use it to measure breathing motion and to differentiate sleeping positions. No attempt was made to infer precise body joint locations and blanket occlusion was reported to decrease the accuracy of the torso detection. While the number of previous works that aim at human pose estimation for bed-ridden subjects is limited, the popularity of depth sensors has pushed research on background-free 3D human pose estimation. Shotton et al. [7] and Girshick et al. [8] train Random Forests on a large non-public synthetic dataset of depth frames in order to capture a diverse range of human shapes and poses. In contrast to their method, we rely on a realistic dataset that was specifically created to evaluate methods for human pose estimation in bed. Furthermore, we augment the dataset with blanket occlusions and aim at making it publicly available. More recently, deep learning has entered the domain of human pose estimation. Belagiannis et al. [9] use a convolutional neural network (CNN) and devise a robust loss function to regress 2D joint positions in RGB images. Such one-shot estimations however do not leverage temporal consistency. In the work of Fragkiadaki et al. [10], the authors rely on a recurrent neural network (RNN) to improve pose prediction on RGB video. However in their setting, the task is formulated as a classification problem for each joint, which results in a coarse detection on a 12×12 grid. Our method in contrast produces accurate 3D joint predictions in the continuous domain, and is able to handle blanket occlusions that occur in hospital monitoring settings.

3 Methods

3.1 Convolutional Neural Network

A convolutional neural network is trained for the objective of one-shot pose estimation in 3D. The network directly predicts all 14 joint locations y which are provided by the motion capture system. We use an *L2* objective during stochastic gradient descent training. Incorrect joint predictions $\hat{y}$ result in a gradient $g = 2 \cdot (\hat{y} - y)$, which is used to optimize the network weights via backpropagation. An architecture of three convolutional layers followed by two

fully connected layers proved successful for this task. The layers are configured as [9-9-64]/[3-3-128]/[3-3-128]/[13-5-1024]/[1024-42] in terms of [height-width-channels]. A [2x2] max pooling is applied after each convolution. In order to achieve better generalization of our network, we use a dropout function before the second and before the fifth layer during training, which randomly switches off features with a probability of 50 %. Rectified linear units are used after every learned layer in order to allow for non-linear mappings of input and output. In total, the CNN has 8.8 million trainable weights. After convergence, we use the 1024-element feature of the 4th layer and pass it to a recurrent neural network in order to improve the temporal consistence of our joint estimations. An overview of the full pipeline of motion capture and depth video acquisition as well as the combination of convolutional and recurrent neural network is shown in Figure 1.

3.2 Recurrent Neural Network

While convolutional neural networks have capability of learning and exploiting local spatial correlations of data, their design does not allow them to learn temporal dependencies. Recurrent neural networks on the other hand are specifically modeled to process timeseries data and can hence complement convolutional networks. Their cyclic connections allow them to capture long-range dependencies by propagating a state vector. Our RNN is built in a Long Short Term Memory (LSTM) way and its implementation closely follows the one described in Graves et al. [11]. We use the 1024-element input vector of the CNN and train 128 hidden LSTM units to predict the 42-element output consisting of x-, y- and z-coordinate of each of the 14 joints. The number of trainable weights of our RNN is around 596,000. During training, backpropagation through time is limited to 20 frames.

3.3 Patient MoCap Dataset

Our dataset consists of a balanced set of easier sequences (no occlusion, little movement) and more difficult sequences (high occlusion, extreme movement) with ground truth pose information. Ground truth is provided through five calibrated motion capture cameras which track 14 rigid targets attached to each subject. The system allows to infer the location of 14 body joints (*head, neck, shoulders, elbows, wrists, hips, knees and ankles*). All test subjects (5 female, 5 male) performed 10 sequences, with a duration of one minute per sequence. Activities include *getting out/in the bed, sleeping on a horizontal/elevated bed, eating with/without clutter, using objects, reading, clonic movement* and a *calibration* sequence. During the *clonic movement* sequence, the subjects were asked to perform rapid twitching movements of arms and legs, such as to display motions that occur during the clonic phase of an epileptic seizure. A calibrated and synchronized Kinect sensor was used to capture depth video at 30 fps. In total, the dataset consists of $180,000$ video frames. For training, we select a bounding box that only contains the bed. To alleviate the adaption to different hospital environments, all frames are rendered from a consistent camera viewpoint, fixed at 2 meters distance from the center of the bed at a 70 degree inclination.

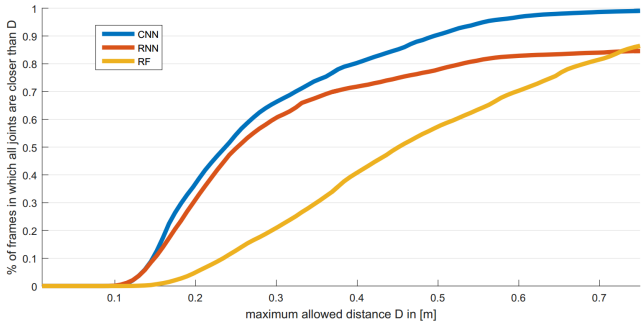

Fig. 3. Worst case accuracy computed on 36,000 test frames of the original dataset. On the y-axis we plot the ratio of frames in which all estimated joints are closer to the ground truth than a threshold D, which is plotted on the x-axis.

3.4 Blanket Simulation

Standard motion capture technologies make it impossible to track bodies under blankets due to the necessity of the physical markers to be visible to the tracking cameras. For this reason, we captured the ground truth data of each person lying on the bed without being covered. We turned to physics simulation in order to generate depth maps with the person under a virtual blanket. Each RGB-D frame is used as a collision body for a moving simulated blanket, represented as a regular triangle mesh. At the beginning of a sequence, the blanket is added to the scene at about 2 meters above the bed. For each frame of the sequence, gravity acts upon the blanket vertices. Collisions are handled by using a sparse signed distance function representation of the depth frame, implemented in OpenVDB [12]. See Figure 2 for an example rendering. In order to optimize for the physical energies, we employ a state-of-the-art projection-based dynamics solver [13]. The geometric energies used in the optimization are triangle area preservation, triangle strain and edge bending constraints for the blanket and closeness constraints for the collisions, which results in realistic bending and folding of the simulated blanket.

4 Experiments

As to validate our method, we compare to the regression forest (RF) method introduced by Girshick et al. [8]. The authors used an RF to estimate the body pose from depth data. At the training phase, random pixels in the depth image are taken as training samples. A set of relative offset vectors from each sample's 3D location to the joint positions is stored. At each branch node, a depth-difference feature is evaluated and compared to a threshold, which determines if the sample is passed to the left or the right branch. Threshold and the depth-difference feature parameters are jointly optimized to provide the maximum information gain at the branch node. The tree stops growing after a maximum depth has been reached or if the information gain is too low. At the leaves, the sets of offsets

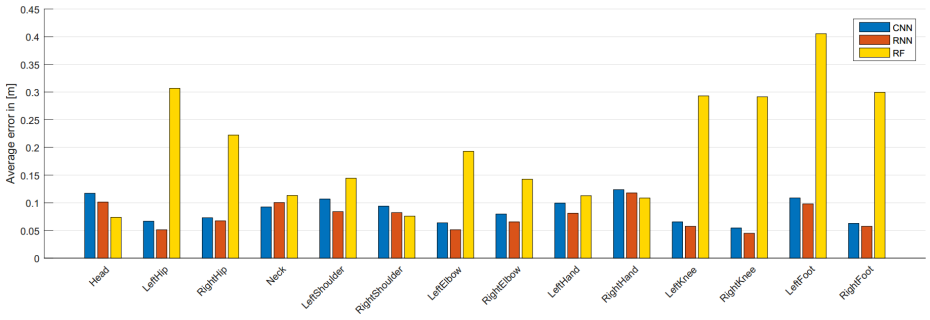

Fig. 4. Average per joint error on the blanket occluded sequence.

vectors are clustered and stored as vote vectors. During test time, body joint locations are inferred by combining the votes of all pixels via mean shift. The training time of an ensemble of trees on >100k images is prohibitively long, which is why the original authors use a 1000-core computational cluster to achieve state-of-the-art results [7]. To circumvent this requirement, we randomly sample 10k frames per tree. By evaluating the gain of using 20k and 50k frames for a single tree, we found that the accuracy saturates quickly (compare Figure 6 of [8]), such that using 10k samples retains sufficient performance while cutting down the training time from several days to hours.

4.1 Comparison on the Patient MoCap Dataset

We fix the training and test set by using all sequences of 4 female and 4 male subjects for training, and the remaining subjects (1 female, 1 male) for testing. A grid search over batch sizes B and learning rates η provided $B = 50$ and $\eta = 3 \cdot 10^{-2}$ as the best choice for the CNN and $\eta = 10^{-4}$ for the RNN. The regression forest was trained on the same distribution of training data, from which we randomly sampled 10,000 images per tree. We observed a saturation of the RF performance after training 5 trees with a maximum depth of 15. We compare the CNN, RNN and RF methods with regard to their average joint error (see Table 1) and with regard to their worst case accuracy, which is the percentage of frames for which all joint errors satisfy a maximum distance constraint D, see Figure 3. While the RNN reaches the lowest average error at 12.25 cm, the CNN appears to have less outlier estimations which result in the best worst case accuracy curve. At test-time, the combined CNN and RNN block takes 8.87 ms to infer the joint locations (CNN: 1.65 ms, RNN: 7.25 ms), while the RF algorithm takes 36.77 ms per frame.

4.2 Blanket Occlusion

A blanket was simulated on a subset of 10,000 frames of the dataset (as explained in Section 3.4). This set was picked from the *clonic movement* sequence, as it is

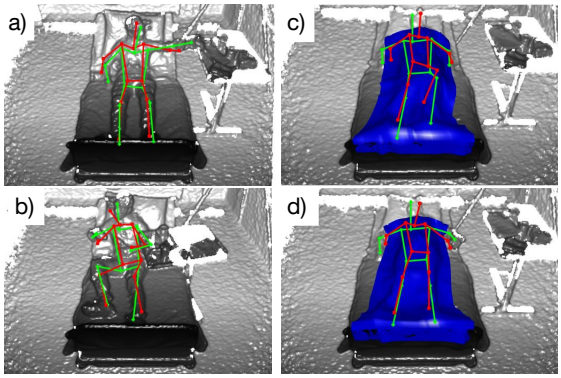

Fig. 5. Examples of estimated (red) and ground truth skeletons (green). Pose estimations work without (*a,b*) and underneath (*c,d*) the blanket (blue).

Table 1. Euclidean distance errors in [cm]. Error on the occluded test set decreases after retraining the models on blanket occluded sequences (+r).

sequence	CNN	RNN	RF
all	12.69	**12.25**	28.10
occluded	**9.05**	9.23	21.30
occluded+r	8.61	**7.56**	19.80

most relevant to clinical applications and allows to compare one-shot (CNN and RF) and time series methods (RNN) on repetitive movements under occlusion. The three methods were trained on the new mixed dataset consisting of all other sequences (not occluded by a blanket) and the new occluded sequence. For the RF, we added a 6th tree which was trained on the occluded sequence. Figure 4 shows a per joint comparison of the average error that was reached on the occluded test set. Especially for hips and legs, the RF approach at over 20 cm error performs worse than CNN and RNN, which achieve errors lower than 10 cm except for the left foot. However, the regression forest manages to identify the head and upper body joints very well and even beats the best method (RNN) for head, right shoulder and right hand. In Table 1 we compare the average error on the occluded sequence before and after retraining each method with blanket data. Without retraining on the mixed dataset, the CNN performs best at 9.05 cm error, while after retraining the RNN clearly learns to infer a better joint estimation for occluded joints, reaching the lowest error at 7.56 cm. Renderings of the RNN predictions on unoccluded and occluded test frames are shown in Figure 5.

5 Conclusions

In this work we presented a unique hospital-setting dataset of depth sequences with ground truth joint position data. Furthermore, we proposed a new scheme for 3D pose estimation of hospitalized patients. Training a recurrent neural network on CNN features reduced the average error both on the original dataset and on the augmented version with an occluding blanket. Interestingly, the RNN benefits a lot from seeing blanket occluded sequences during training, while the CNN can only improve very little. It appears that temporal information helps to determine the location of limbs which are not directly visible but do interact with the blanket. The regression forest performed well for arms and the head,

but was not able to deal with occluded legs and hip joints that are typically close to the bed surface, resulting in a low contrast. The end-to-end feature learning of our combined CNN-RNN model enables it to better adapt to the low contrast of occluded limbs, which makes it a valuable tool for pose estimation in realistic environments.

Acknowledgments. The authors would like to thank Leslie Casas and David Tan from TUM and Marc Lazarovici from the Human Simulation Center Munich for their support. This work has been funded by the German Research Foundation (DFG) through grants NA 620/23-1 and NO 419/2-1.

References

1. Stone, E.E., Skubic, M.: Unobtrusive, Continuous, In-Home Gait Measurement Using the Microsoft Kinect. Biomedical Engineering, IEEE Transactions on **60**(10) (2013)
2. Kontschieder, P., Dorn, J.F., Morrison, C., Corish, R., Zikic, D., Sellen, A., DSouza, M., Kamm, C.P., Burggraaff, J., Tewarie, P., Vogel, T., Azzarito, M., Glocker, B., Chen, P., Dahlke, F., Polman, C., Kappos, L., Uitdehaag, B., Criminisi, A.: Quantifying Progression of Multiple Sclerosis via Classification of Depth Videos. In: MICCAI 2014. Springer
3. Cunha, J., Choupina, H., Rocha, A., Fernandes, J., Achilles, F., Loesch, A., Vollmar, C., Hartl, E., Noachtar, S.: NeuroKinect: A Novel Low-Cost 3Dvideo-EEG System for Epileptic Seizure Motion Quantification. PLOS ONE **11**(1) (2015)
4. Benbadis, S.R., LaFrance, W., Papandonatos, G., Korabathina, K., Lin, K., Kraemer, H., et al.: Interrater reliability of eeg-video monitoring. Neurology **73**(11) (2009)
5. Li, Y., Berkowitz, L., Noskin, G., Mehrotra, S.: Detection of Patient's Bed Statuses in 3D Using a Microsoft Kinect. In: EMBC 2014, IEEE
6. Yu, M.C., Wu, H., Liou, J.L., Lee, M.S., Hung, Y.P.: Multiparameter Sleep Monitoring Using a Depth Camera. In: Biomedical Engineering Systems and Technologies. Springer (2012)
7. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-Time Human Pose Recognition in Parts from Single Depth Images. Communications of the ACM **56**(1) (2013)
8. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient Regression of General-Activity Human Poses from Depth Images. In: ICCV 2011, IEEE
9. Belagiannis, V., Rupprecht, C., Carneiro, G., Navab, N.: Robust Optimization for Deep Regression. In: ICCV 2015, IEEE
10. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent Network Models for Human Dynamics. In: ICCV 2015, IEEE
11. Graves, A.: Generating Sequences With Recurrent Neural Networks. arXiv preprint arXiv:1308.0850 (2013)
12. Museth, K., Lait, J., Johanson, J., Budsberg, J., Henderson, R., Alden, M., Cucka, P., Hill, D., Pearce, A.: OpenVDB: An Open-source Data Structure and Toolkit for High-resolution Volumes. In: ACM SIGGRAPH 2013 Courses, ACM (2013)
13. Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective Dynamics: Fusing Constraint Projections for Fast Simulation. ACM Transactions on Graphics (TOG) **33**(4) (2014)