3D Human Pose Estimation on a Configurable Bed from a Pressure Image

Henry M. Clever*, Ariel Kapusta, Daehyung Park, Zackory Erickson, Yash Chitalia, Charles C. Kemp

***Abstract*— Robots have the potential to assist people in bed, such as in healthcare settings, yet bedding materials like sheets and blankets can make observation of the human body difficult for robots. A pressure-sensing mat on a bed can provide pressure images that are relatively insensitive to bedding materials. However, prior work on estimating human pose from pressure images has been restricted to 2D pose estimates and flat beds. In this work, we present two convolutional neural networks to estimate the 3D joint positions of a person in a configurable bed from a single pressure image. The first network directly outputs 3D joint positions, while the second outputs a kinematic model that includes estimated joint angles and limb lengths. We evaluated our networks on data from 17 human participants with two bed configurations: *supine* and *seated*. Our networks achieved a mean joint position error of 77 mm when tested with data from people outside the training set, outperforming several baselines. We also present a simple mechanical model that provides insight into ambiguity associated with limbs raised off of the pressure mat, and demonstrate that Monte Carlo dropout can be used to estimate pose confidence in these situations. Finally, we provide a demonstration in which a mobile manipulator uses our network's estimated kinematic model to reach a location on a person's body in spite of the person being seated in a bed and covered by a blanket.**

I. Introduction

Various circumstances, such as illness, injury, or longterm disabilities can result in people receiving assistance in bed. Previous work has shown how robots can provide assistance with ADLs [1]–[3], but providing assistance to a person in bed can be challenging. Estimating the pose of a person's body could enable robots to provide better assistance. Typical methods of body pose estimation use line-of-sight sensors, such as RGB cameras, which can have difficulties when the body is occluded by blankets, loose clothing, medical equipment, over-bed trays and other common items in healthcare settings, such as hospitals. A pressure-sensing mat on the bed can allow for estimation of the body's pose in a manner that is less sensitive to bedding materials and surrounding objects [2], [4]–[6]. However, prior work with pressure images has not addressed a number of concerns key to the success of robot assistance in bed, namely (1) pose estimation in 3D for either flat or non-flat beds and (2) appropriately dealing with uncertainty when the pose estimate may be inaccurate.

In this work, we present a method for estimating the 3D joint positions in real time with a measure of confidence in each estimated position for a person in a configurable bed using a pressure-sensing mat. We provide evidence that our method works for some challenging scenarios, such as when the bed and human are configured in a seated posture, and when limbs are raised off of the pressure sensing mat. Further, we release a motion capture labeled dataset of over 28,000 pressure images across 17 human participants, in addition to our open-source code.

H. M. Clever, A. Kapusta, D. Park, Z. Erickson, Y. Chitalia and C. Kemp are with the Healthcare Robotics Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology,

*H. M. Clever is the corresponding author. henryclever@gatech.edu

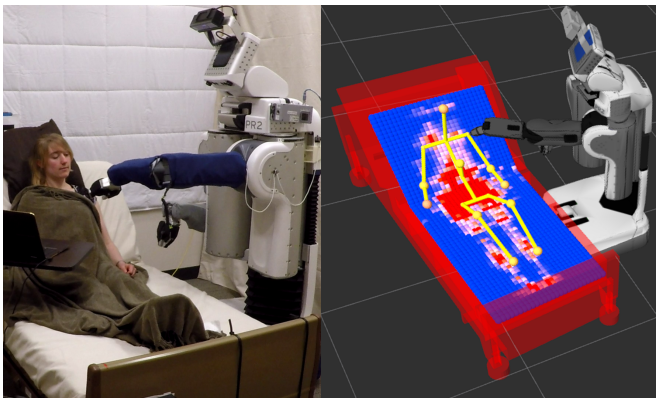

Fig. 1. We demonstrate how an assistive robot could use our 3D human pose estimation method. A PR2 robot uses our method's body pose estimation to reach to a person's shoulder.

In prior work, we used a pressure sensing mat on a configurable bed to estimate the location of a person's head for positioning a rigid geometric model of a human body [2]. In this paper, we propose and compare two convolutional neural network (ConvNet) architectures to learn a mapping from a pressure image and bed configuration to 3D human joint positions: the head, neck, shoulders, elbows, wrists, chest, hips, knees, and ankles. The first method directly regresses to the 3D ground truth labels (x, y, z). The second method embeds a 17-joint kinematics skeleton model of the human to the last layer of the ConvNet to enforce constraints on bone lengths and joint angles. This provides a more complete pose representation with additional unlabeled joints and latent space joint angle estimates. We introduce a new architecture that adjusts the skeleton model for differently sized people, while providing comparison to a constant link length skeleton model. We train the kinematics ConvNets end-to-end, backpropagating from 3D joint Euclidean error through the kinematics model. We compare our ConvNet methods to baseline data-driven algorithms, including ridge regression and k-nearest neighbors.

Our configurable bed, introduced previously as Autobed [2], [7], features adjustable height, leg rest angle, and head rest angle. Autobed can sense its own state, sense the pressure distribution of the person on the bed, and communicate with other devices. In this work, we estimate the human joint positions in two Autobed configurations by adjusting the bed's head rest: supine (0° flat) and seated (60° incline).

Lack of contact by limbs or other body parts presents a

challenging issue for the pressure image modality. In this work, we consider common poses where this issue arises, such as an arm raised in the air resembling a double inverted pendulum. Among other poses, this case demonstrates where the pressure image can be similar for different configurations of the arm. In such a case, the pressure data may be insufficient to confidently estimate the pose of the arm. An estimate of confidence or model uncertainty can be valuable, for example to allow an assistive robot to reject low-confidence estimates by removing them from a list of potential goals in task plan execution. To estimate model uncertainty, we use Monte Carlo dropout, a method proposed by Gal and Ghahmarani [8]. With this method, we perform a number of stochastic forward passes through the ConvNet during test time and compute the joint position and joint position confidence from the moments of the output distribution.

II. Related work

Markerless human pose estimation is a challenging problem complicated by environment factors, the human pose configurations of interest, and data type. Relatively few researchers have used pressure images for human pose estimation in bed [4]–[6], while many used cameras in myriad environments and poses [9]–[21]. Researchers have increasingly explored data-driven methods such as ConvNets, from model-free networks to inclusion of models in various architectures. Here we discuss research with pressure images, data-driven methods, and measuring network uncertainty.

A. Pressure-Image-based Work

Prior pressure-image-based pose estimation work has fit 2D kinematic models to pressure image features. In a series of papers including [4], Harada et al. used a kinematic model to create a synthetic database for comparison with ground truth pressure images. Grimm et al. [5] identified human orientation and pose using a prior skeleton model. Similar to our motivation and findings, they used a pressure mat to compensate for bedding occlusion and observed higher error for lighter joints (e.g. the elbow), which had a relatively low pressure. Liu et al. [6] generated a pictorial structures model to localize body parts on a flat bed.

A few researchers have also looked at human posture classification from pressure images [5], [22], [23]. Posture classification is a different problem from 3D body pose estimation, but it can be used in a complementary way, e.g. by providing a prior on the model used for pose estimation.

B. Data-driven Human Pose Estimation

Like the pressure-image-based research, we use a human body model, but take a data-driven approach more common in vision-based work. While infrared and depth images have seen recent attention in human pose estimation [9], a large body of vision-based work uses monocular RGB cameras [10]. Our method builds upon research with monocular RGB image input; we note the following similarities and differences:

- *Single image.* Both monocular RGB-image-based work and our pressure-image-based work has a single input array.
- *Under-constrained.* For 3D human pose, both monocular RGB images and single pressure images are under-constrained.
- *Data content.* The data encoding is fundamentally different. For example, in the context of pose estimation, light intensity in an RGB image is highly disconnected from pressure intensity.
- *Dimensionality.* RGB images typically have more features and a higher resolution than pressure images.
- *Warped spatial representation.* Calibrating RGB cameras to alleviate distortion is straightforward. In contrast, it is challenging to determine the configuration of a cloth pressure sensing mat from its pressure image in the case of folds or bends.

While the differences may limit the transferability of methods across data types, we find that some data-driven methods previously used for vision modalities are applicable to pressure images. Researchers have performed 3D human pose estimation with monocular RGB images using standard machine learning algorithms such as ridge regressors [11]–[13]. In particular, Okada and Soatto [11] use kernel ridge regression (KRR) as well as linear ridge regression (LRR). Further, Ionescu et al. [12] compare K-Nearest Neighbors (KNN) and ridge regression on the Human3.6M dataset. We use these classical approaches to provide a baseline comparison for our proposed method.

Recently, with the advent of high quality, labeled synchronous datasets such as Human3.6M [12], many researchers have explored deep learning methods such as end-to-end training of ConvNets [14]–[17]. Two common ConvNet approaches include direct regression to joint labels [14], [21] and regression to discretized confidence maps [15], [16], [18]. Within 3D human pose estimation research, confidence map approaches include Pavlakos et al. [16], who train a ConvNet end-to-end on a 3D confidence voxel space, and Zhou, Zhu et al. [19] and Bogo et al. [20] who fit a 3D model to 2D confidence maps. However, the high dimensional output space of confidence maps can make real-time pose estimation difficult. Li and Chan [21] used rapid direct regression to 3D Cartesian joint positions; we implement a similar architecture because real-time estimation is important to our planned application. Zhou, Sun et al. [17] take a hybrid approach, by training a ConvNet end-to-end and enforcing anthropomorphic constraints with an embedded human skeleton kinematics model with constant link lengths. We implement a method of this form for comparison and introduce a new architecture with variable skeleton link lengths to allow the model to adapt to differently sized people.

C. Monte Carlo Dropout in ConvNets

We use Monte Carlo dropout to measure network uncertainty, a method introduced by Gal and Ghahramani [8]. Monte Carlo dropout has been applied to measure uncertainty for camera relocalization [24] and semantic segmentation [25]. We use this method to estimate pose and provide a measure of confidence.

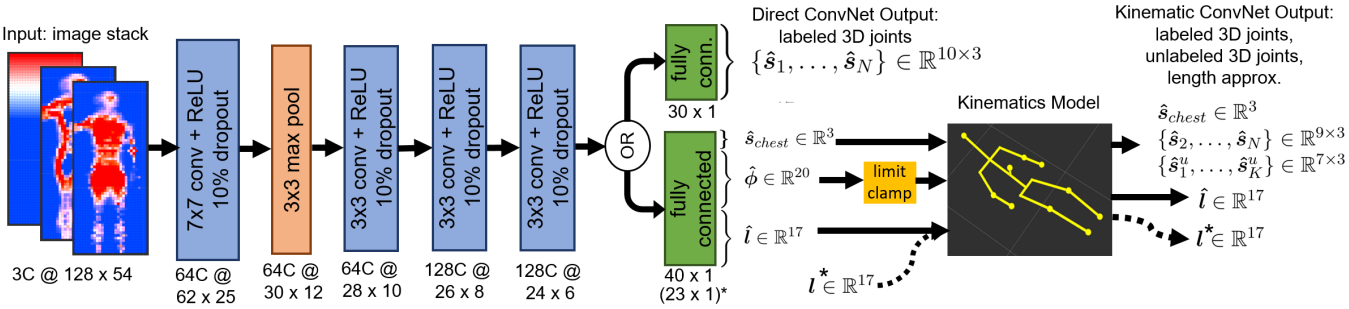

Fig. 2. Two ConvNet architectures. The *Direct ConvNet* directly regresses to ten motion capture labeled global joint positions. The *Kinematic ConvNet* embeds a kinematics skeleton model into the last fully connected network layer, parameterized in the latent space by a root joint global position, joint angles, and skeleton link lengths. The *Kinematic ConvNet* also outputs unlabeled joint position estimates. We explore architectures with both variable link length (shown) and constant link length (dashed arrows, defined by $*$.)

III. Method

Our ConvNets learn a function $f(\mathcal{P}, \theta_B)$ that estimates pose parameters of a person lying in a robotic bed, given a specified bed configuration θ_B and a 2D pressure image $\mathcal{P}$ from a pressure mat.

A. ConvNet Architecture

We explore two ConvNet architectures shown in Fig. 2. Our network includes four 2D convolutional layers with 64 output channels for the first two layers and 128 channels for the last two layers. The layers mostly have 3×3 filters, with a ReLU activation and a dropout of 10% applied after each layer. We apply max pooling, and the network ends with a linear fully connected layer.

To estimate a person's joint pose, we construct an input tensor for the ConvNet comprised of three channels, i.e. $\{\mathcal{P}, E, B\} \in \mathbb{R}^{128\times 54\times 3}$. Raw data from the pressure sensing mat is recorded as a (64×27)-dimensional image, which we upsample by a factor of two, i.e. $\mathcal{P} \in \mathbb{R}^{128\times 54}$. We use first order interpolation for upsampling. In addition to a pressure image, we also provide the ConvNet with an edge detection channel, $E \in \mathbb{R}^{128\times 54}$, which is computed as a Sobel filter over both the horizontal and vertical directions of the upsampled image. Empirically, we found this edge detection input channel improved pose estimation performance. In order to estimate human pose at different configurations of the bed (e.g. sitting versus lying down), we compute a third input channel, $B \in \mathbb{R}^{128\times 54}$, which depicts the bed configuration. Specifically, each element in the matrix B depicts the vertical height of the corresponding taxel on the pressure mat. When the bed frame is flat, i.e. $\theta_B = 0$, then B is simply the zero matrix.

B. Direct Joint Regression

The first proposed ConvNet architecture outputs an estimate of the motion capture labeled global 3D joint positions $\{\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_N\}$, where each $\hat{\boldsymbol{s}}_j \in \mathbb{R}^3$ represents a 3D position estimate for joint j. This direct ConvNet regresses directly to 3D ground truth label positions in the last fully connected layer of the network. We compute the loss from the absolute value of Euclidean error on each joint:

$$\text{Loss}_{\text{direct}} = \sum_{j=1}^{N} ||\boldsymbol{s}_j - \hat{\boldsymbol{s}}_j|| \tag{1}$$

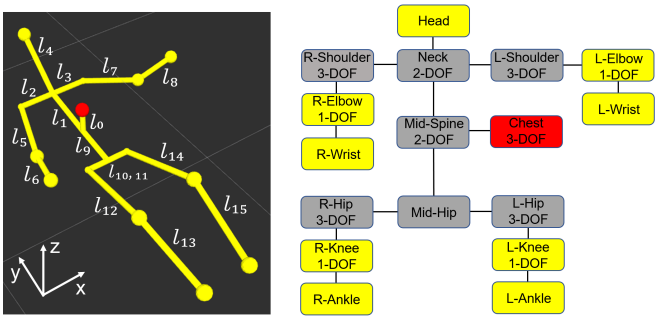

Fig. 3. Kinematics Model Parameters. The root joint $\boldsymbol{s}_{chest}$ is specified in red. Yellow boxes represent the remaining motion capture labeled joints $\{\hat{\boldsymbol{s}}_2, \ldots, \hat{\boldsymbol{s}}_N\}$. Grey boxes represent unlabeled joints $\{\hat{\boldsymbol{s}}^u_1, \ldots, \hat{\boldsymbol{s}}^u_K\}$, where the kinematic model adjusts to an approximate fit.

C. Deep Kinematic Embedding

We embed a human skeleton kinematics model into the last fully connected network layer to enforce geometric and anthropomorphic constraints. This creates an extra network layer where labeled joint estimates $\{\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_N\}$ and unlabeled joint estimates $\{\hat{\boldsymbol{s}}^u_1, \ldots, \hat{\boldsymbol{s}}^u_K\}$ are solved through forward kinematics equations depending on root joint position $\hat{\boldsymbol{s}}_1$, latent space joint angles $\hat{\phi}$ and an estimate of skeleton link length approximations $\hat{\boldsymbol{l}}$. We use $\hat{\boldsymbol{s}}_1 = \hat{\boldsymbol{s}}_{chest}$ as the root joint. We incorporate skeleton link lengths $\boldsymbol{l}$ into the loss function by pre-computing an approximation to the ground truth. While the kinematics functions are relative to a root joint, we learn root joint global position $\boldsymbol{s}_{chest}$ to put our output in global space. We compute a weighted loss from the absolute value of Euclidean error on each joint and error on each link length:

$$\text{Loss}_{\text{kin.}} = ||\boldsymbol{s}_1 - \hat{\boldsymbol{s}}_1|| + \alpha \sum_{j=2}^{N} ||\boldsymbol{s}_j - \hat{\boldsymbol{s}}_j|| + \beta |\boldsymbol{l} - \hat{\boldsymbol{l}}| \tag{2}$$

where α and β are weighting factors. We compare two variants of this loss function: The *variable link length* ConvNet is as

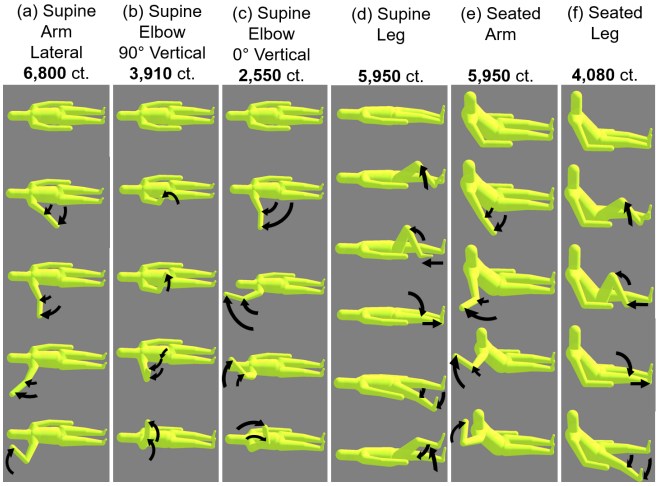

Fig. 4. Range of paths traversed by participants during training. Equivalent paths were traversed by the left limbs. Count represents both left and right data across 17 participants.

described, while for the *constant link length* ConvNet we set $\beta = 0$ and use a constant $\boldsymbol{l}^*$ input to the kinematics model. We compute $\boldsymbol{l}^*$ as the average of the approximations $\boldsymbol{l}$.

1) Human Kinematics Model: We represent the human body with a model similar to that used in other work [12], [17], [21], [26], with 17 joints to cover major links down to the wrists and ankles. We ignore minor links and joints. To train the networks that have link lengths as an output, we require ground truths for comparison. Some ground truth link lengths may be calculated directly from the dataset's labels, for example when motion capture gives the location of both ends of the link. We approximate the link lengths for links that are under-constrained in the dataset for the skeleton model. The link lengths are an output of our network as a $\boldsymbol{l} \in \mathbb{R}^{17}$ vector. We ignore unlabeled joints in the loss function.

The mid-spine is found by a vertical offset from the chest marker to compensate for the distance between marker placement atop the chest and the modeled bending point of the spine. We do not make offset corrections with other joints; these are more challenging than the chest and the effects are less noticeable.

2) Angular Latent Space: We define 20 angular degrees of freedom consisting of 3-DOF shoulders and hips, 1-DOF elbow and knee joints, a 2-DOF spine joint, and a 2-DOF neck joint. Fig 3 (b) shows this parameterization corresponding to labeled and unlabeled joints. For the spine of the model to better match the spine of a person seated in bed, we used two revolute joints about the x-axis. To account for head movement, we placed a neck joint at the midpoint of the shoulders with pitch and yaw rotation. We use PyTorch [27], a deep learning library with tensor algebra and automatic differentiation. We manually encoded the forward kinematics for the skeleton kinematic model. The network uses stochastic gradient descent during backpropagation to find inverse kinematics (IK) solutions.

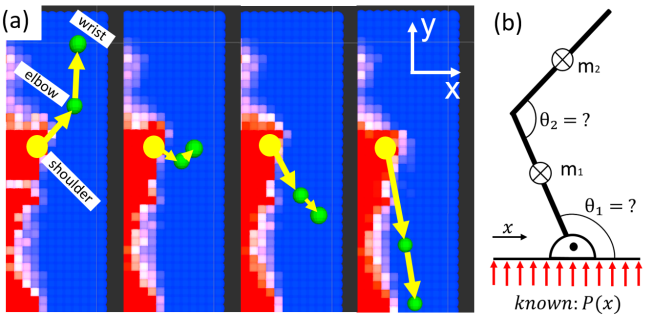

Fig. 5. (a) Crossover of the shoulder from the *supine elbow 0 vertical* traversal, both joints envisioned as 2-DOF inverted pendulum. Green dots represent left elbow and wrist ground truth markers projected in 2D, yellow dot and arrows indicate approximate shoulder and limb positions. (b) Model of 2-DOF inverted pendulum showing static indeterminacy. Known pressure distribution $P(x)$ is insufficient to solve θ_1, θ_2.

D. Pressure Image Ambiguity

Raising a limb off of a bed with pressure sensors can lead to a loss of information as the sensors can only sense pressure during contact. A similar loss of information can be seen when the limbs extend off the edge of the bed. Consider the movement shown in Fig. 4 (c) and an example of the pressure images associated with such a movement in Fig. 5 (a). Here, the pressure images appear nearly identical while the elbow and wrist positions change substantially. To better understand what is physically causing this phenomena, we can model the arm as a double inverted pendulum, shown in Fig. 5 (b). Here, the pendulum angles θ_1 and θ_2 are statically indeterminate given an underlying pressure distribution, until part of the arm touches the sensors.

Another challenge is pressure sensor resolution. Depending on the type of sensor, saturation can occur. A pressure image with higher spatial resolution, accuracy, and pressure range may result in less ambiguity. We note that a model of body shape (e.g. 3D limb capsules) might provide information that helps to resolve the ambiguity.

E. Uncertainty: Monte Carlo Dropout

To estimate both joint position and model uncertainty simultaneously, we apply Monte Carlo dropout from Gal and Ghahramani [8]. Monte Carlo dropout is the process of performing V forward passes through the network with dropout enabled. This results in V output vectors which may differ slightly due to the stochastic dropout of data during each forward pass. We can compute an estimated output as the average of all V outputs, corresponding to the first moment of the predictive distribution within the network. Similarly, the model's uncertainty corresponds to the second moment of the distribution, which we can compute as the variance of all V forward passes.

IV. Evaluation

We recorded a motion capture labeled dataset with over 28,000 pressure images from 17 different human participants.[1] We conducted this study with approval from the Georgia

[1]Dataset: ftp://ftp-hrl.bme.gatech.edu/pressure_mat_pose_data

Institute of Technology Institutional Review Board (IRB), and obtained informed consent from all participants. We recruited 11 male and 6 female participants aged 19-32, who ranged 1.57-1.83 m in height and 45-94 kg in weight. We fitted participants with motion capture markers at the wrists, elbows, knees, ankles, head, and chest. We used a commercially available 64×27 pressure mat from Boditrak sampled at 7 Hz. We asked each participant to move their limbs in 6 patterns, 4 while supine and 2 while seated, to represent some common poses in a configurable bed. The movement paths are shown in Fig. 4. We instructed participants to keep their torso static during limb movements.

We trained six data-driven models: three baseline supervised learning algorithms and the three proposed ConvNet architectures.[2] We designed the network using 7 participants (5M, 2F); we performed leave-one-participant-out cross validation using the remaining 10 participants (6M, 4F).

A. Data Augmentation

At each training epoch for the ConvNets, we selected images such that each participant would be equally represented in both the training and test sets. We augmented the original dataset in the following ways to increase training data diversity:

- *Flipping.* Flipped across the longitudinal axis with probability $P = 0.5$.
- *Shifting.* Shifted by an additive factor $sh \sim \mathcal{N}(\mu = 0cm, \sigma = 2.86cm)$.
- *Scaling.* Scaled by a multiplicative factor $sc \sim \mathcal{N}(\mu = 1, \sigma = 0.06)$.
- *Noise.* Added taxel-by-taxel noise to images by an additive factor $\mathcal{N}(\mu = 0, \sigma = 1)$. Clipped the noise at min pressure (0) and at the saturated pressure value (100).

We chose these to improve the network's ability to generalize to new people and positions of the person in bed. We did not shift the seated data longitudinally or scale it because of the warped spatial representation.

B. Baseline Comparisons

We implemented three baseline methods to compare our ConvNets against: K-nearest neighbors (KNN), Linear Ridge Regression (LRR), and Kernel Ridge Regression (KRR). For all baseline methods, we used histogram of oriented gradients (HOG) features [28] on $2\times$ upsampled pressure images. We applied flipping, shifting, and noise augmentation methods. We did not use scaling, as it worsened performance.

1) K-Nearest Neighbors: We implemented a K-nearest neighbors (KNN) regression baseline using Euclidean distance on the HOG features to select neighbors, as [12] did. We selected $k = 10$ for improved performance.

2) Ridge Regression: We implemented two ridge-regression-based baselines. Related work using these methods are described in Section II. We trained linear ridge regression (LRR) models with a regularization factor of $\alpha = 0.7$. We also train Kernel Ridge Regression (KRR) models with a radial basis function (RBF) kernel and $\alpha = 0.4$. We manually selected these values of α for both LRR and KRR. We also tried linear and polynomial kernels for KRR, but found the RBF kernel produced better results in our dataset.

[2]Code release: https://github.com/gt-ros-pkg/hrl-assistive/tree/indigo-devel/hrl_pose_estimation

C. Implementation Details of Proposed ConvNets

During testing, we estimated the joint positions with $V = 25$ forward passes on the trained network with Monte Carlo dropout for each test image. For each joint, we report the mean of the forward passes as the estimated joint position. We use PyTorch and ADAM from [29] for gradient descent.

1) Pre-trained ConvNet: We created a pre-trained ConvNet that we use to initialize both Kinematic ConvNets, with regressed and constant link length. The pre-trained network used the kinematically embedded ConvNet and the loss function in Eq. 2, with $\alpha = 0.5$ and $\beta = 0.5$. This network was trained for 10 epochs on the 7 network-design participants with a learning rate of 0.00002 and weight decay of 0.0005.

2) Direct ConvNet: We trained the network for 300 epochs directly on motion capture ground truth, using the sum of Euclidean error as the loss function. We used a learning rate of 0.00002 and a weight decay of 0.0005.

3) Kinematic ConvNet, Constant Link Length: We trained the network through the kinematically embedded ConvNet, used the loss function in Eq. 2, with $\alpha = 0.5$ and $\beta = 0$. This value for β means the network would not regress to link length, leaving it constant. We initialized the network with the pre-trained ConvNet, but we separately initialized each link length as the average across all images in the fold's training set for each fold of cross validation.

4) Kinematic ConvNet, Regress Link Length: We trained the network through the kinematically embedded ConvNet, used the loss function in Eq. 2, with $\alpha = 0.5$ and $\beta = 0.5$. Joint Cartesian positions and link lengths in the ground truth are represented on the same scale. We initialized with the pre-trained ConvNet.

D. Measure of Uncertainty

Here we show an example where ambiguous pressure mat data has a high model uncertainty. We compare two leg abduction movements from the *supine leg* motion, shown in the bottom two columns of Fig. 4 (d). We sample 100 images per participant: half feature leg abduction contacting the pressure mat, and half with elevated leg abduction. For each pose, we use $V = 25$ stochastic forward passes and compute the standard deviation of the Euclidean distance from the mean for abducting joints, including knees and feet. We compare this metric between elevated and in-contact motions.

V. Results

In Table I , we present the mean per joint position error (MPJPE), a metric from literature to represent overall accuracy [12], [21]. Fig. 6 shows the per-joint position error across all trained models, separated into supine and seated postures. The error for the direct ConvNet and the kinematic ConvNet with length regression is significantly lower than the other methods. The results of knees and legs show that more distal limbs on the kinematic chain do not necessarily result

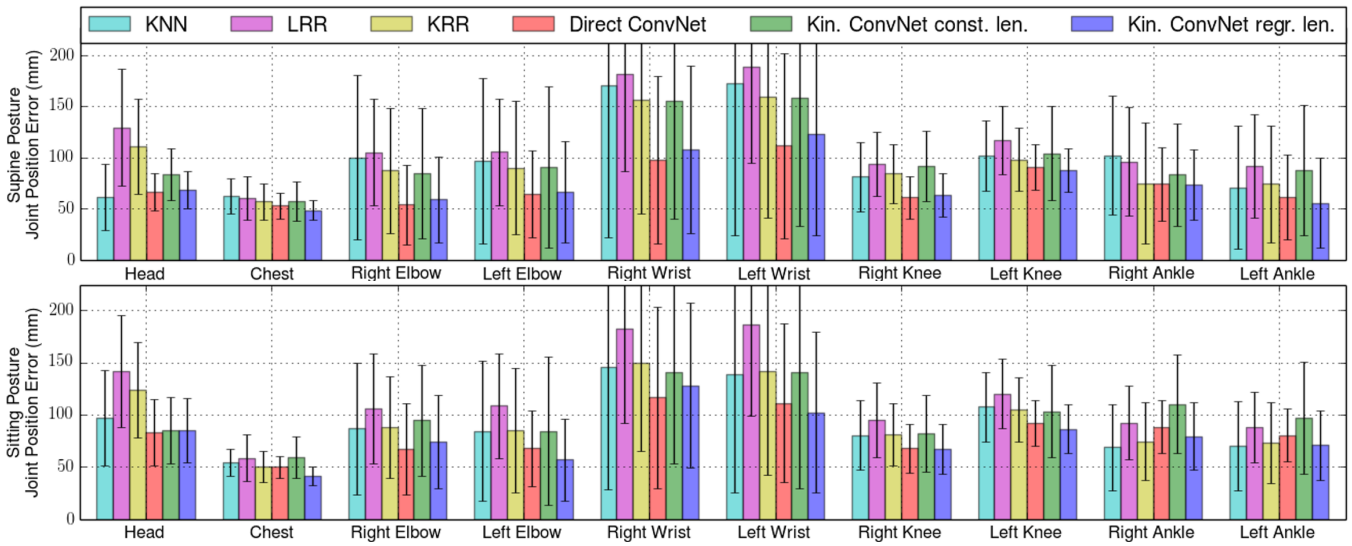

Fig. 6. Per-joint error mean and standard deviation for leave-one-participant-out cross validation over 10 participants for a sitting and a supine posture in bed. Lower is better. Our methods outperformed baseline methods.

TABLE I: Mean Per Joint Position Error.

Method	MPJPE Supine (mm)	MPJPE Seated (mm)	MPJPE Overall (mm)
K-Nearest Neighbors	102.01	93.42	99.06
Linear Ridge Regression	117.01	114.76	116.24
Kernel Ridge Regression	99.25	97.10	98.51
Direct ConvNet	73.49	82.44	**76.56**
Kinematic ConvNet, avg. $\boldsymbol{l}$	99.74	99.70	99.72
Kinematic ConvNet, regr. $\boldsymbol{l}$	75.43	79.19	**76.72**

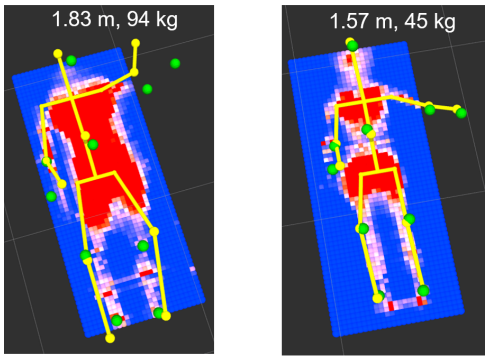

Fig. 7. Comparison of the heaviest and tallest participant with the lightest and shortest participant. Our kinematics ConvNet with link length regression appears to adjust for both sizes.

in higher error. The wrists are both distal and light, and have higher error than the other joints. Fig. 7 shows the kinematics ConvNet with length regression adjusting for humans of different sizes and in different poses. Furthermore, we can perform a pose estimate with uncertainty using $V = 25$ stochastic forward passes in less than a half second.

A. Measure of Uncertainty

We performed a t-test to compare uncertainty in elevated leg abduction and in-contact leg abduction. We compared each knee and ankle separately. We found that the standard deviation of the Euclidean distance from the mean of $V = 25$ forward passes with Monte Carlo dropout is significantly higher for joints in the elevated position. Further, we note that variance in the latent angle parameters $\boldsymbol{\theta}$ compounds through the kinematic model, causing more distal joints in the kinematic chain to have higher uncertainty. This phenomena is further described in Fig. 8, which shows limbs removed from the mat that have a high variance.

VI. Discussion and Limitations

A. Network Architecture Considerations

While the results for the direct ConvNet architecture were marginally better than the kinematic ConvNet with variable lengths, the latter has other advantages. First, there can be value in getting a skeletal model from the network, providing a more complete set of parameters including 20 angular DOFs and a total of 17 joint positions. Second, unsurprisingly, requiring that outputs from the ConvNet satisfy kinematic constraints means that outputs will be constrained to plausible looking body poses. Without those constraints the ConvNet could produce unrealistic outputs.

While limiting our skeleton model to 20 angular DOFs promotes simplicity, it has some hindrance to generalizability. For example, the mid-spine joint lacks rotational DOFs about the y- and z-axes. Adding these DOFs would allow the model to account for rolling to a different posture and laying sideways in bed.

B. Data Augmentation Challenges

Data augmentation cannot easily account for a person sliding up and down in a bed that is not flat. Vertical shifting augmentation for non-flat beds would not match the physical effects of shifting a person on the pressure mat. Augmentation by scaling also has problematic implications, because a much smaller or larger person may have a weight distribution that would not scale linearly at the bending point of the bed.

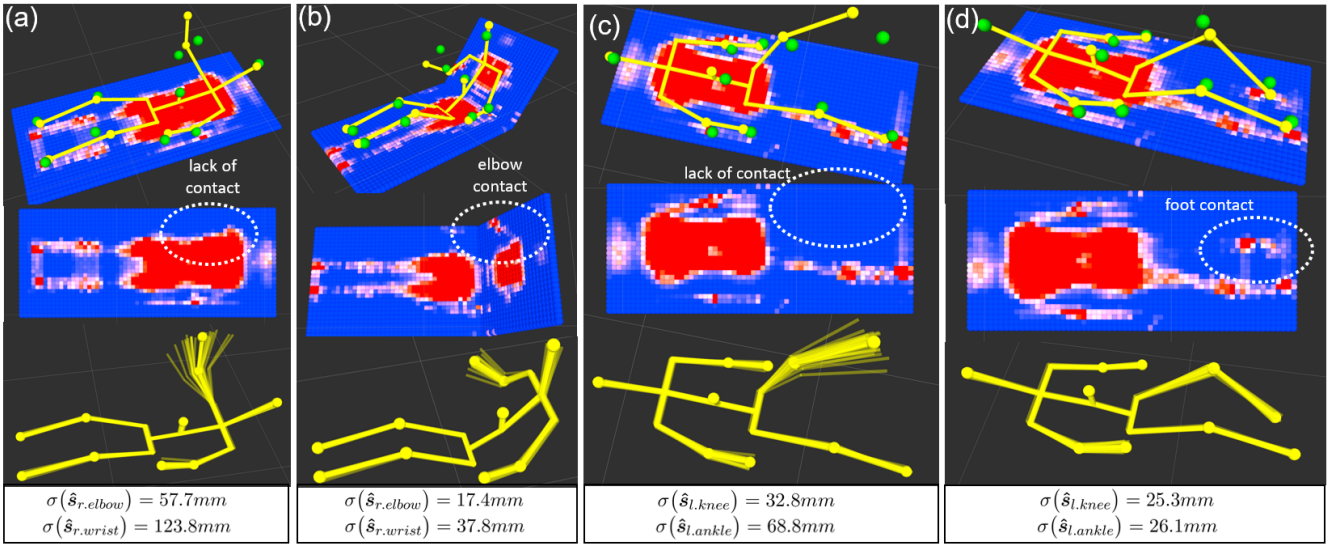

Fig. 8. Illustration of mean and standard deviation of kinematic ConvNet ($\boldsymbol{l}$ regression) output. $V = 25$ forward passes with Monte Carlo dropout shown with thin translucent skeleton links. Spheres indicate joint position estimates. (a) Right arm thrust into the air. (b) Right forearm extending off the side of the pressure mat. (c) Left leg extended in the air and abducted. (d) Left knee extended in the air, left foot contacting pressure mat. Note: (c) and (d) show the same participant, others are different.

Simulation might resolve these issues by simulating placing a weighted human model of variable shape and size placed anywhere on a simulated pressure mat, with many possibilities of bed configurations.

C. Dataset Considerations

The posture and range of paths traversed by participants may not be representative of other common poses, and we expect our method to have limited success in generalizing to body poses not seen or rarely seen in the dataset. While a participant is moving their arm across a specified path, other joints remain nearly static, which over-represents poses with the arms adjacent to the chest and legs straight. We found over-represented poses to generally have a lower uncertainty. Interestingly, with our current sampling and training strategy, over-representation and under-representation is based on the percentage of images in the dataset a joint or set of joints is in a particular configuration. Additional epochs of training or directly scaling the size of the dataset does not change these effects on uncertainty of pose representation. Our method could be improved by using weighting factors or sampling strategies to compensate for this effect.

In our evaluation, some limb poses occur in separate training images, but do not occur in the same training image. For example, we recorded one participant moving both arms and both legs simultaneously. Fig. 9 shows that our method has some ability to estimate these poses.

The skeleton model has offset error in addition to the ground truth error reported. While we attempted to compensate for the chest marker offset, the other markers were more challenging. This may have caused some inaccuracy in the link length approximations.

D. Removal of High Variance Joints

Fig. 10 shows that joints with high uncertainty have a higher average error. Discarding these joint estimates can decrease

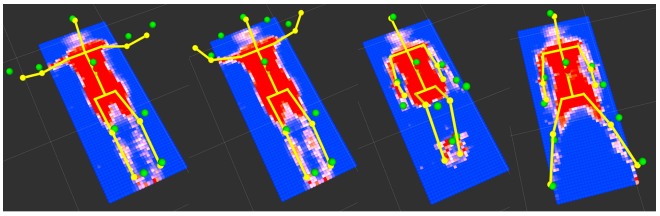

Fig. 9. Dual arm and dual leg traversals. While these are excluded from the training set, our methods can provide a reasonable pose estimate.

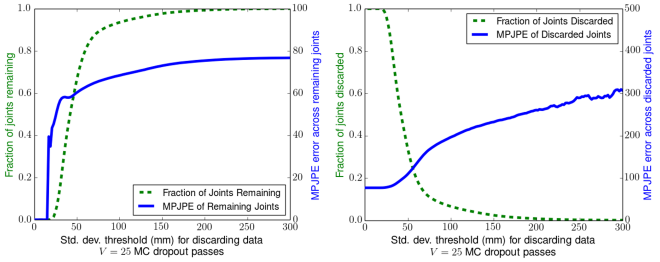

Fig. 10. Discarding joints with a higher uncertainty can decrease error, with a tradeoff to the number of joints remaining.

the average error of the model. As an example application, a robot using our method's estimated poses for task and motion planning might want to require low uncertainty before plan execution.

VII. Demonstration with PR2 Robot

We conducted a demonstration of how our method could inform an assistive robot trying to reach part a person's body. We conducted this study with approval from the Georgia Institute of Technology Institutional Review Board (IRB), and obtained informed consent from an participant. We recruited a single able-bodied participant who used a laptop computer running a web interface from [3] to command a PR2 robot

to move its end effector to their left knee and to their left shoulder. The robot's goal was based on the estimated pose of the person's body from our ConvNet with kinematic model regressing to link lengths. For the knee position, the participant raised her knee to the configuration shown in the 2nd image of Fig. 4 (f), and the participant was in the seated posture for both tasks. Using our 3D pose estimation method, the robot was able to autonomously reach near both locations. Fig 1 shows the robot reaching a shoulder goal while the participant is occluded by bedding and an over-bed table.

VIII. Conclusion

In this work, we have shown that a pressure sensing mat can be used to estimate the 3D pose of a human in different postures of a configurable bed. We explored two ConvNet architectures and found that both significantly outperformed data-driven baseline algorithms. Our kinematically embedded ConvNet with link length regression provided a more complete representation of a 17-joint skeleton model, adhered to anthropomorphic constraints, and was able to adjust to participants of varying anatomy. We provided an example where joints on limbs raised from the pressure mat had a higher uncertainty than those in contact. We demonstrated our work using a PR2 robot.

Acknowledgment

We thank Wenhao Yu for his suggestions to this work. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1148903, NSF award IIS-1514258, NSF award DGE-1545287, AWS Cloud Credits for Research, and the National Institute on Disability, Independent Living, and Rehabilitation Research (NIDILRR), grant 90RE5016-01-00 via RERC TechSAge. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Dr. Kemp is a cofounder, a board member, an equity holder, and the CTO of Hello Robot, Inc., which is developing products related to this research. This research could affect his personal financial status. The terms of this arrangement have been reviewed and approved by Georgia Tech in accordance with its conflict of interest policies.

References

[1] K. Hawkins, P. Grice, T. Chen, C.-H. King, and C. C. Kemp, "Assistive mobile manipulation for self-care tasks around the head," in *2014 IEEE Symposium on Computational Intelligence in Robotic Rehabilitation and Assistive Technologies*. IEEE, 2014.

[2] A. Kapusta, Y. Chitalia, D. Park, and C. C. Kemp, "Collaboration between a robotic bed and a mobile manipulator may improve physical assistance for people with disabilities," in *RO-MAN*. IEEE, 2016.

[3] P. M. Grice and C. C. Kemp, "Assistive mobile manipulation: Designing for operators with motor impairments," in *RSS 2016 Workshop on Socially and Physically Assistive Robotics for Humanity*, 2016.

[4] T. Harada, T. Sato, and T. Mori, "Pressure distribution image based human motion tracking system using skeleton and surface integration model," in *ICRA*, vol. 4. IEEE, 2001, pp. 3201–3207.

[5] R. Grimm, S. Bauer, J. Sukkau, J. Hornegger, and G. Greiner, "Markerless estimation of patient orientation, posture and pose using range and pressure imaging," *International journal of computer assisted radiology and surgery*, vol. 7, no. 6, pp. 921–929, 2012.

[6] J. J. Liu, M.-C. Huang, W. Xu, and M. Sarrafzadeh, "Bodypart localization for pressure ulcer prevention," in *EMBC*. IEEE, 2014, pp. 766–769.

[7] P. M. Grice, Y. Chitalia, M. Rich, H. M. Clever, and C. C. Kemp, "Autobed: Open hardware for accessible web-based control of an electric bed," in *RESNA*, 2016.

[8] Y. Gal and Z. Ghahramanim, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Conference on Machine Learning*, 2016, pp. 1050 – 1059.

[9] W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, and E.-h. Zahzah, "Human pose estimation from monocular images: A comprehensive survey," *Sensors*, vol. 12, no. 16, p. 1966, 2016.

[10] N. Sarafianos, B. Boteanu, C. Ionescu, and I. A. Kakadiaris, "3d human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, no. 152, pp. 1–20, 2016.

[11] R. Okada and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *ECCV*. Springer, 2008, pp. 434–445.

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[13] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 44–58, 2006.

[14] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*.

[15] 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.

[16] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *CVPR*. IEEE, 2017.

[17] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *ECCV 2016 Workshops*, 2016, pp. 186–201.

[18] R. V. Wei, Shih-En, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016, pp. 4724–4732.

[19] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video," in *CVPR*. IEEE, 2016, pp. 4966–4975.

[20] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *ECCV*. Springer, 2016, pp. 561–578.

[21] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in *Asian Conference on Computer Vision*. Springer, pp. 332–347.

[22] M. Farshbaf, R. Yousefi, M. B. Pouyan, S. Ostadabbas, M. Nourani, and M. Pompeo, "Detecting high-risk regions for pressure ulcer risk assessment," in *BIBM*. IEEE, 2013, pp. 255–260.

[23] S. Ostadabbas, M. B. Pouyan, M. Nourani, and N. Kehtarnavaz, "In-bed posture classification and limb identification," in *BioCAS*. IEEE, 2014, pp. 133–136.

[24] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *ICRA*. IEEE, 2016, pp. 4762–4769.

[25] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *CVPRW*. IEEE, 2016, pp. 680–688.

[26] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 44:1 – 44:14, 2017.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.