A Deep Learning Approach to 3D Human Walking Pose Estimation From SensFloor Capacitive Signals

Anonymous Author(s)

Abstract

While camera-based systems are the dominant approach for human pose estimation, they face challenges in terms of privacy concerns and occlusion problems. These issues are of particular relevance in domains such as elderly care, where pose estimates can be used to monitor residents health or analyze incidents retrospectively. To assess an alternative to camera-based pose estimation, this paper aims to predict 3D walking poses using SensFloor: a capacitance-based floor which registers movement activity. We analyze the potential to utilize the floor's low-resolution signals to estimate poses and to what extent certain joint positions can be predicted accurately. For this purpose, we collected synchronized SensFloor signals and video data, from which we extracted 3D human poses using MediaPipe to serve as ground-truth targets for training. These signals and their corresponding targets were then used for supervised training of an LSTM neural network. To estimate the person's position on the floor, a Kalman filter was applied to smooth the noisy SensFloor measurements. Our results demonstrate that it is possible to predict human walking poses using the proposed methods, establishing a proof-of-concept for an alternative way of activity monitoring.

Keywords

SensFloor, Human pose estimation, Deep Learning, LSTM, Kalman Filter, Capacitive Floor, MediaPipe

Code: https://github.com/sensfloor

Create README for public organization view

1 Introduction

With the ageing of the population elderly care is becoming more important than ever. However, the amount of missing workforces is increasing and is expected to be between 280 000 and 690 000 in 2049 in Germany [1]. Therefore, the demand for AI-based technologies that support caregivers by automating monitoring tasks is growing, as they help improve patient safety even with limited personnel.

As monitoring patients for 24 hours continuously is particularly difficult with a limited workforce, research on estimating human movement and posture has been actively pursued.

In recent years, camera-based approaches for human pose estimation have been widely studied. However, such methods raise concerns regarding privacy and are sensitive to environmental factors such as occlusions and camera placement. Therefore, careful consideration is required before introducing these technologies in care facilities. To enable the practical implementation of automation in real care environments, it is necessary to explore alternative methods that can monitor patients while preserving privacy and mitigating occlusion issues.

compress

Considering these concerns, one promising alternative for automating tasks in the real care-giving is floor-based sensing technology. In this study, we focus on SensFloor as an example of a floor-based sensing system. SensFloor is primarily used in elderly care, as it provides a less intrusive form of monitoring compared to camera-based sensors. It is a capacitive sensing floor composed of rectangular patches, each containing eight triangular sensor fields . Each field registers changes in capacitance and outputs a signal between 127 and 255 sampled at 10 Hz, depending on the magnitude of the detected change. Since the human body contains a large amount of water, walking over changes the capacitance, which the SensFloor uses it to detect human movement.

Recently, various floor-based sensing approaches have been explored for human movement analysis. Several studies have estimated human posture or movement patterns using foot pressure-based sensors with neural network models, including intelligent carpets [1] and pressure-sensing insoles [2]. However, the pressure-based systems used in these studies capture richer information about foot–floor interactions than SensFloor . Moreover, existing studies using SensFloor [3] focus on recognizing gait patterns or walking conditions or estimating the age, rather than estimating full-body 3D human poses. Therefore, while prior work demonstrates the potential of floor-based sensing for human movement analysis, the feasibility of estimating full-body 3D walking poses from low-resolution capacitive floor sensor signals remains largely unexplored.

This study aims to answer the research questions of whether it is possible to use the SensFloor signals to estimate 3D human walking poses and how accurate they are. For this purpose we created a labeled dataset, trained a neural network that estimates poses, tuned a Kalman filter to smooth the extracted positions of the estimated poses on the floor and implemented a frontend that visualizes them.

2 Methods

To achieve this goal of estimating poses using a SensFloor, we split up the problem into two tasks, that are illustrated in Figure 1: pose estimation and localization. Pose estimation estimates the person's joint coordinates in a local coordinate system. For the pose estimation component, we implemented a supervised fine-tuning pipeline. For that we collected reference pose estimates from a MediaPipe to serve as targets. Using these targets, we trained a Long Short-Term Memory (LSTM) to predict human poses based on SensFloor activations within an extracted Region of Interest (ROI).

To position the local pose estimate in the global coordinate system of the floor we extract positions using a clustering approach and smooth the resulting positions using a Kalman filter. Finally, the combined data of pose and position is streamed to a frontend application for real-time visualization.

By separating these tasks, the system is able to use a small localized active area of the floor for pose estimation. Afterward this pose is mapped across the global floor coordinates. This enables the system to be ported to various floors of different dimensions.

2.1 Data collection

Make definition of terms "patch" and "field" more obvious

Before was too general

Human tracking paper, shortly mention it exists

Mention assumption of only one person walking on the floor

Rephrase

find better transition

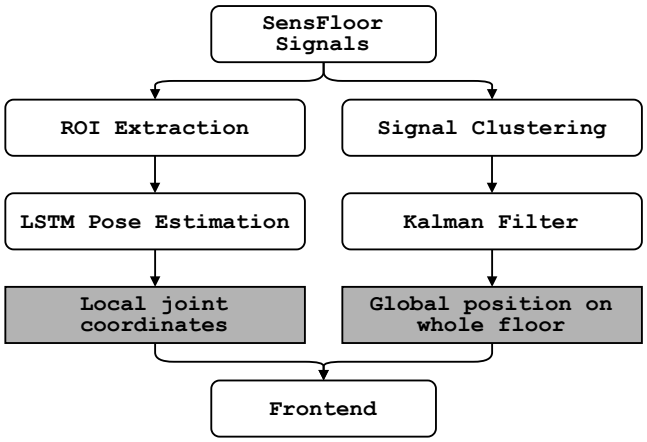

Figure 1: Overview of the SensFloor-based pose estimation pipeline — The left branch estimates the pose in a local, position-independent coordinate system, while the right branch localizes the person in global floor coordinates.

The model developed in this paper takes SensFloor signals as input to predict the 3D joint coordinates of a person walking on the floor. To train and evaluate such a model, we recorded in two sessions a total of eight hours of synchronized SensFloor signals and video at 15 FPS from three participants. In addition to this, there are further short recordings of three other people that we used for testing. To synchronize the data, we mapped the SensFloor signals to the corresponding video frame number at their time of occurrence. In post-processing, we used MediaPipe to extract 3D pose estimates from the video to serve as training targets. It is important to mention here that the origin of MediaPipe's pose estimates coordinate system is located at the center of the hips. This allows us to decompose the paper's problem into the previously mentioned two components of pose estimation relative to the hip and localization on the whole floor.

Look up actual duration

cite

2.2 Training Data preprocessing

Before giving the data to our model we preprocess it in several ways. First we filter signals that are beneath a threshold and likely correspond to noise from the floor. We defined the threshold to be $\tau = 140$, but it could vary depending on the floor setup and its surface material. Then we select the frames, at which a significant SensFloor signal was measured and assign them the pose of that frame, ending up with a dataset containing both significant signals as input and a pose as the target. Next, we split the dataset sequence-wise with 80% for training, 10% for validating and 10% for testing. When we load one item of this set we create a ROI history, which is a temporal sequence of areas of highest activation. We defined the length of the history as 50 and the ROI is a 4×4 grid of the floor with the signal values for this frame. The history is created by going back 50 frames and for each frame taking the signals it received or no signals if no signals were in this frame. All signals are normalized to the range $[0, 1]$, where values of 127 will be 0 and values of 255 will be 1. Because the ROI is independent in the rest of the floor, we randomly rotate full ROI histories by $\{0, 90, 180, 270\}$ degrees for a more robust training. Finally, we use 13 of the 33 joints given by MediaPipe, excluding eyes, ears, mouth and fingers because they are irrelevant for our goal. Additionally, we exclude the foot heel and foot index, because when analyzing the data we found that MediaPipe predicted these bones unreliably to lengths of 3 to 18 cm.

explain roi gird not triangular and use image and

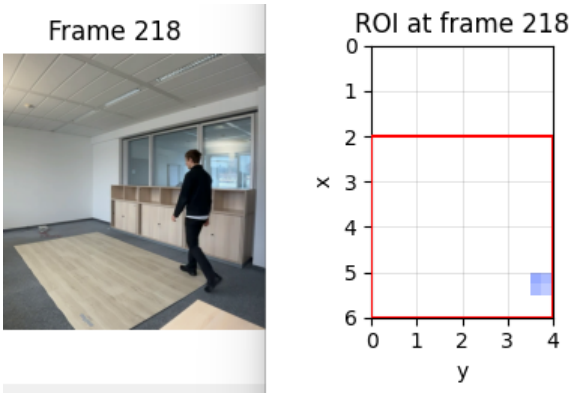

Figure 2: Example of one ROI in the sequence given to the model. The highlighted rectangle is the area of the ROI and the signals of the floor are in blue, showing the step of a person

2.3 Pose estimation model

As our model we chose an LSTM architecture. It is based on the model from [1]. We adapted in a way that we added a CNN layer before the LSTM and increased the model size. The model's full architecture and parameters are illustrated in figure 3.

Mention regression model

Mention Epoch and such?

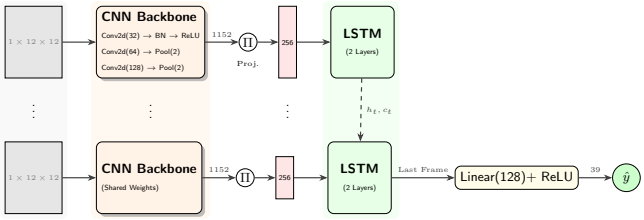

Figure 3: Overview of the model. We used a CNN-LSTM architecture with a self attention layer

The CNN-LSTM model receives the signals in the ROI frame by frame and produces a vector of continuous values, representing the x, y and z coordinate predictions for each joint. These coordinates are given into the loss function. The total loss is defined as

$$L_{total} = L_{\text{WMSE}} + \lambda \cdot L_{\text{link}}, \tag{1}$$

where λ is a fixed scalar for weighting L_{link}. For our training we set $\lambda = 0.1$. The L_{WMSE} (Weighted Mean Squared Error) calculates the squared difference between predictions and the targets, the joint coordinates, weighted by specific landmark importance and is defined as

$$L_{\text{WMSE}} = \frac{1}{B \cdot J \cdot 3} \sum_{i=1}^{B} \sum_{j=1}^{J} \sum_{k=1}^{3} w_j (\hat{y}_{i,j,k} - y_{i,j,k})^2, \tag{2}$$

where B is the batch size, J the number of joints, k indexing the x, y and z coordinates of each joint, $\hat{y}_{i,j,k}$ the model's predictions and $y_{i,j,k}$ the targets. We increased the weight for the knees and feet to 5 compared to 1 for all other joints, because they are the most active components in a walking scenario. For each estimated joint-pair we furthermore calculate the Euclidean distance d (e.g. the lower leg length). We use it in the link loss function which we adopted from [2] defined as

$$L_{\text{link}}(d) = \begin{cases} k_{\min} - d, & \text{if } d < k_{\min} \\ d - k_{\max}, & \text{if } d > k_{\max} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $k_{\min}$ and $k_{\max}$ are the 3^{th} and 97^{th} percentiles of the link lengths in the training set. During the training we calculate the following metrics in each epoch: Mean joint position error (MJPE) and percentage correct keypoits (PCK) with a 5 and 10 threshold

2.4 Localization

To integrate the local pose estimate from the model into the global coordinate system of the floor, we developed a pipeline that tracks the subject's hip position relative to the SensFloor. We do this by extracting a raw pose estimate through clustering the measured sensor activations and applying a Kalman filter to smoothen the trajectory.

We identify the subject's contact points with the floor by grouping adjacent active sensor fields into clusters. The hip position is then calculated the following way: during a step, when both feet touch the ground, two clusters can be measured. In this case, we define the hip position as the midpoint between the centroids of the two clusters. If only one cluster is detected, meaning one foot is off the ground or both feet are close together, we assume the hip is located directly above that single centroid. This logic is illustrated in Figure 4 and allows us to extract the hip position from the received signals.

Make horizontal

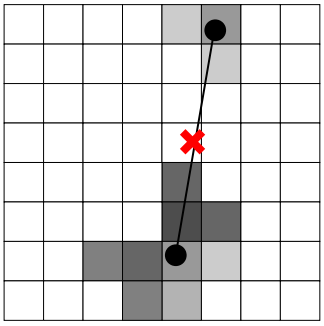

Figure 4: Calculation of the global position — This visualization shows two clusters of SensFloor activations. For both the centroid calculated. The final position is then determined by taking the mean position between both clusters.

Because the raw position estimates are noisy and lead to erratic, unnatural movement in the visualization, we apply a Kalman Filter to ensure a smooth trajectory. Similar to [3, 255 ff.], the filter's state vector is defined as $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$, where x and y are the position coordinates and $\dot{x}$ and $\dot{y}$ the corresponding velocities. Due to the absence of ground-truth tracking data for exact calibration, we tuned the filter's noise parameters empirically until the trajectory visually matched the subject's actual movement in the recorded videos. A comparison of the Kalman-filtered position estimates versus the raw measured positions is shown in Figure 5

move maybe to results

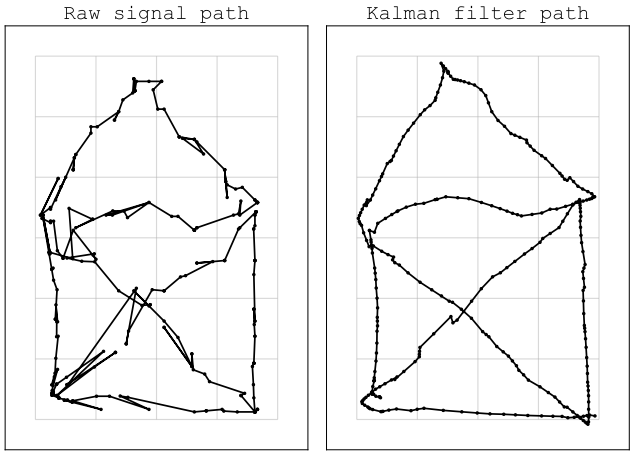

Figure 5: Effect of the Kalman filter — The left part of the image shows the raw signals that we obtain from the clustering of the activated fields from the walk of a person on the floor. On the right side are the signals processed by the Kalman filter which show a much smoother and realistic trajectory.

2.5 Visualization of predicted 3D Pose

For visual evaluating and demo-purposes of the pose estimation pipeline, we furthermore implemented a frontend. The estimated pose and its position are streamed to the frontend using a WebSocket. The frontend then visualizes a simulated environment of the floor and the person's pose moving on it.

3 Results

In this section, we will first visually compare the estimated

4 Discussion and Conclusion

References

[1] Raoul Hoffmann, Hanna Brodowski, Axel Steinhage, and Marcin Grzegorzek. 2021. Detecting Walking Challenges in Gait Patterns Using a Capacitive Sensor Floor and Recurrent Neural Networks. 21, 4 (2021), 1086. doi:10.3390/s21041086

[2] Yiyue Luo, Yunzhu Li, Michael Foshey, Wan Shou, Pratyusha Sharma, Tomas Palacios, Antonio Torralba, and Wojciech Matusik. 2021. Intelligent Carpet: Inferring 3D Human Pose from Tactile Signals. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA, 2021-06). IEEE, 11250–11260. doi:10.1109/CVPR46437.2021.01110

[3] Roger R. and Labbe. 2024. Kalman and Bayesian Filters in Python. (2024). https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python