3D Pose Tracking of Walker Users' Lower Limb with a Structured-Light Camera on a Moving Platform

Richard Zhi-Ling Hu[1], Adam Hartfiel[1], James Tung[1,2], Adel Fakih[3], Jesse Hoey[1] and Pascal Poupart[1]
[1]David R. Cheriton School of Computer Science, University of Waterloo
[2]Department of Systems Design Engineering
[3]Department of Kinesiology
{rzhu,ajhartfi,james.tung,afakih,jhoey,ppoupart}@uwaterloo.ca

Abstract

Tracking and understanding human gait is an important step towards improving elderly mobility and safety. Our research team is developing a vision-based tracking system that estimates the 3D pose of a wheeled walker user's lower limbs with a depth sensor, Kinect, mounted on the moving walker. Our tracker estimates 3D poses from depth images of the lower limbs in the coronal plane in a dynamic, uncontrolled environment. We employ a probabilistic approach based on particle filtering, with a measurement model that works directly in the 3D space and another measurement model that works in the projected image space. Empirical results show that combining both measurements, assuming independence between them, yields tracking results that are better than with either one alone. Experiments are conducted to evaluate the performance of the tracking system with different users. We demonstrate that the tracker is robust against unfavorable conditions such as partial occlusion, missing observations, and deformable tracking target. Also, our tracker does not require user intervention or manual initialization commonly required in most trackers.

1. Introduction

Falls and fall related injuries are the leading cause of injury-related hospitalization among seniors. In addition to physical consequences (e.g. hip fracture, loss of mobility), falls can cause a loss in confidence and activities, which may lead to further decline in health and more serious falls in the future. To improve mobility and safety of seniors, our research team is developing a smart walker that aims to provide navigation and stabilizing assistance to users. An important goal of the project is to track and understand the walker user's leg pose, based on analyzing image sequences extracted from a depth sensor mounted on the walker, as shown in Figure 1.

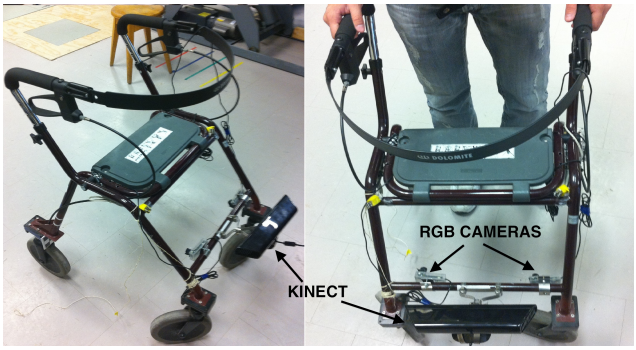

Figure 1. Setup of cameras on the walker. The Kinect is placed at the center capturing the legs in the coronal plane. Two RGB camera are placed at the side, and are used only as point of comparison with the Kinect in the experiment section.

Tracking leg pose in an uncontrolled environment has important applications in biomedical settings. Specifically for the walker, the tracking system allows assessment and monitoring of gait, such as recovery following orthopedic surgery (e.g., joint replacement). Zhou and Hu [20] presented a list of human motion tracking systems used for biomedical purposes. The best measurements currently available for gait parameters in uncontrolled environment are accelerometer-based temporal measures (e.g. step-time mean and variability), which lack reliable spatial estimates (e.g. step length and width). Temporal and spatial measures of gait are complementary indicators of balancing behaviour during walking, reflecting different strategies of maintaining stability. However, reliable spatial measures are only available with non-visual tracking systems such as inertial and magnetic sensors at the cost of restricting users to walk in a limited area and are thus not suitable in real, uncontrolled settings. Similarly, marker-based visual tracking systems such as VICON involve fixed sensors and require special markers on the user, which is unnatural to the user.

As a result, markerless visual based tracking is an important area of research for biomedical gait analysis.

The ability to work with 3D data, as opposed to 2D intensity/RGB information, is important in our application. First, reliance on color and gradient information is not robust in our setting, which involves a dynamic subject walking in a dynamic scene with varying lighting conditions. The dynamic background (due to moving cameras) offers significant distraction. Common techniques to eliminate distractions such as background subtraction are not applicable or reliable with dynamic scenes in RGB images. In addition, due to physical constraints of the walker, the camera can only be mounted to capture the frontal view of the legs. In this view, the greatest motion during walking is perpendicular to the image plane, so it is difficult to observe movement with regular RGB cameras, and depth measurement becomes crucial.

In this paper, we present a vision-based, markerless, tracking system that works with 3D points obtained from a single structured-light camera, Kinect. We adopt a top-down probabilistic approach based on particle filtering that generates particles (i.e., possible poses) according to a simple constant velocity model and then weights the particles based on two distance metrics that measure the distance between the generated leg model and the observed 3D points. A series of experiments are performed to evaluate the robustness of the system.

2. Related Works

In general, the problem of pose tracking is approached in one of two ways: top-down or bottom-up. Tracking with 3D points can be formulated as a bottom-up, model-fitting problem by working directly with the sensor data points. Knoop et al. [13] use the Iterative Closest Point (ICP) algorithm to find the optimal translation and rotation matrix that minimizes the sum of squared distances between data points obtained from a time-of-flight camera and a stereo camera, and model points from the degenerate cylinder model. Muhlbauer et al. [15] exploit the structure of the human body to search for the ideal pose by fitting a body pose to the 3D data points obtained from a stereo camera. They use a hierarchical scheme by looking for the head first, then the torso, and limbs are then fitted iteratively starting with the joint closest to the torso and search outwards. Fua et al. [9] formulate a least-square optimization problem to adjust the model's joint angles by minimizing the distance of their model points to the 3D points obtained by a stereo camera. In addition, they use a skeleton model combined with soft, deformable surface to simulate the behavior of bone and muscle. Kehl et al. [12] obtain 3D-data from a volumetric reconstruction based on multiple camera views. They formulate tracking as a minimization of a scalar objective function, using gradient descent with adaptive step sizes. They also use a hierarchical tracking approach by first fixing the torso and then tracking the limbs. Cheung et al. [6] present a real time system that fits a 3D ellipsoid model to 3D voxel data obtained by 5 cameras. Instead of comparing camera projections of the model with the silhouettes, fitting is done directly in 3D space. A real-time full body tracking software has been developed by PrimeSense [2] specifically designed for its depth sensing devices including the Kinect. However, the software is not suitable for our application because it requires the torso to be visible and the feet are not tracked at all.

Pose tracking can also be formulated as a top-down approach with a hidden Markov model that generates a distribution of predictions according to some motion model, and then evaluates the probability of each prediction according to the likelihood of the sensor data given the prediction. Kalman filtering for Gaussian distributions and particle filtering [10] for general distributions are two predominant top-down approaches in the tracking literature. Multiple variants of these two approaches have been published for pose tracking [19, 18, 7, 8]. While the literature on top-down tracking approaches is quite vast for regular RGB cameras, it is less common for 3D sensors. Ziegler et al. [21] formulate the tracking problem as the registration of two point sets in an Unscented Kalman filtering approach. ICP is used to obtain a refined set of model points, and the measurement model is based on the distance between this refined set of model points and measured 3D points obtained by a stereo camera. Mikic et al. [14] present a full-body tracking algorithm with 3D voxel data obtained from multiple cameras. Their approach involves an automatic initial fitting of the cylindrical model to the data points in the first frame, subsequent frames are then tracked using the extended Kalman filter (EKF). Jojic et al. [11] also use EKF for tracking the upper body with 3D data obtained from stereo cameras. A statistical image formation model that accounts for occlusion plays a central role in their tracking system.

The trade-off between top-down and bottom-up approaches concerns speed and accuracy. While bottom-up approaches are fast and many of them are implemented in near real-time, top-down approaches are known to produce better, more stable results, due to the ability to incorporate temporal smoothness and maintain a distribution of predictions instead of just one prediction. For our application, we choose accuracy over speed as tracking is usually done offline for assessment purposes. Moreover, the unique setup of our camera makes bottom-up approaches difficult. First, only the lower limbs are visible in our problem, which prevents us from using many bottom-up techniques that first locate salient parts such as the torso or head. Also, we observe that legs frequently occlude each other in the camera view during walking, and that there is significant missing

data in the Kinect image due to close-object sensing and dress folding. In particular, dress folding of the pants during walking leads to complex surfaces that makes it difficult for the Kinect to retrieve depth information in every pixel. The missing data problem leads to isolated regions of data points which belong to the legs in the image, and this is problematic for approaches based on hierarchical search. Since bottom-up approaches in general are sensitive to noise and missing data, we opt for a top-down approach. We use a particle filter due to its ability to handle multi-modal distributions. Such distributions can occur when the observation is inherently ambiguous where multiple states can fit a single observation.

3. Camera Setup

The Kinect provides an inexpensive and fairly accurate 3D camera solution based on structured light. The Kinect uses a weak infrared laser to project a predefined pattern of dots of varying intensity [1]. This pattern provides a rich source of easily extracted features. The variation of these features compared against the known pattern for a fixed distance provides a method for depth reconstruction. The accuracy of the Kinect's depth output is exceptional at relatively close range, with increasing error as distance increases beyond a few meters.

The use of infrared light presents some weaknesses. For instance, bright sunlight may wash out the structured light pattern, effectively blinding the Kinect. Also, the camera is not always able to produce a depth value at every pixel and frequently leaves blank patches in the image due to occlusion and bumpy surfaces. Objects located too close to the sensor may not be detected since the Kinect has a minimum working distance of about 30 cm. In the experiment section, we show that the tracker works comfortably even with substantial missing data.

To calibrate the camera, we use the calibration software from [3]. The checkerboard method is suitable for the Kinect because the black and white pattern is clearly visible in the infrared stream. The intrinsic parameters obtained from the calibration procedure are used subsequently to convert the depth value of each pixel into 3D points, and to project the 3D cylinders of our model onto the image plane.

4. Tracking Framework: Hidden Markov Model

The pose tracking problem is formulated as a belief monitoring or filtering problem with a Hidden Markov Model (HMM). We use the particle filtering approach, using samples to represent the underlying distribution of target states. The HMM is specified by four elements:

Hidden State the leg pose of the walker user. The pose is represented as a state vector $\overrightarrow{X}$ defined in Section 4.1

Observation the data returned by the Kinect. The raw data is processed by first converting the raw value of each pixel into 3D coordinates and then classifying each 3D point as either foreground or background. Details are described in Section 4.2

Transition Function governs how the hidden states evolve over time. We use a simple constant velocity model as described in Section 4.3.

Likelihood Function the likelihood of an observation given a hidden state. In this work, we use two separate distance measures to compute how close the prediction is to an observation. These two functions are then combined assuming independence. Details are described in Section 4.4.

4.1. Hidden State: Physical Model

We adopt a model composed of tapered cylinders for the thigh, calf, and foot of each leg, as shown in Figure 2. To better model the feet, we use half-cylinders with a flat base. We define the state vector $\overrightarrow{X}$, from which the position and orientation of each cylinder in the model can be determined. There are 23 elements in the state vector $\overrightarrow{X}$: the spherical coordinates of the left hip relative to the right hip, the position of the right hip, the lengths and widths of the cylinders (assuming symmetry between left and right legs), 3 DoF joint angles for the hips, 1 DoF joint angles for the knees, and 2 DoF joint angles for the ankles. Various constraints are placed on the legs: maximum and minimum allowable values are enforced on the lengths, widths, and joint angles of each leg segment; there must always be at least one foot on the ground; and cylinders cannot intersect each other in 3D space.

4.2. Observation: Depth Image Processing

Each image capture by the Kinect camera corresponds to a 640x480-pixel frame in which each pixel (i, j) corresponds to an integer value d from 0 to 2047 representing the depth of the pixel relative to the camera center. Each raw depth value $d_{i,j}$ is first converted into millimeters according to the following equation:

$$z_{i,j} = \frac{1000}{-0.00307 d_{i,j} + 3.33} \tag{1}$$

where the constants of the equation come from [1] and are manually verified.

Given the depth value in metric space and the intrinsic parameters obtained from the calibration procedure, a 3D

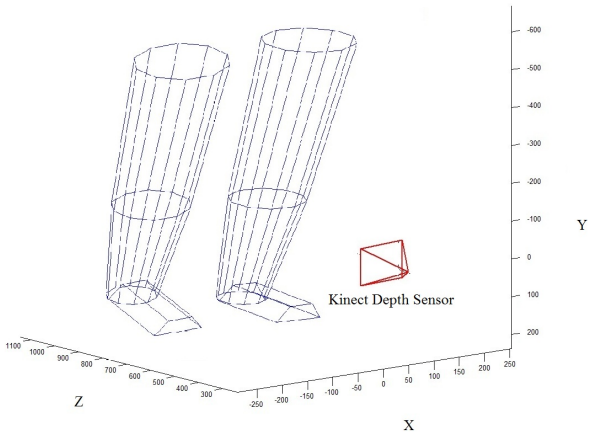

Figure 2. Graphical representation of the cylindrical model and the location of the sensor

point $(x_{i,j}, y_{i,j}, z_{i,j})$ in mm with respect to the camera center can be associated with each pixel according to the following equation:

$$(x_{i,j}, y_{i,j}, z_{i,j}) = (\frac{z_{i,j}(i - x_0)}{f_x}, \frac{z_{i,j}(j - y_0)}{f_y}, z_{i,j}) \quad (2)$$

where f_x and f_y are the focal lengths, and x_0 and y_0 are the camera centers in the x and y axis respectively.

Finally, we classify each pixel/3D point as foreground or background by applying two filters. First, a background frame is generated before tracking to capture the floor (i.e., no objects or people in the field of view near the camera). For subsequent frames during tracking, we subtract the raw depth value of each pixel from the raw depth value of the same pixel in background frame. If the absolute difference is below a certain threshold, the pixels are classified as belonging to the floor and thus background. Assuming the ground plane remains flat, the first filter aims to remove pixels that correspond to the ground only. Note that this background frame needs to be generated only once and is used throughout all walks to remove floor pixels. For the second filter, remaining points are classified as background if they are outside the region-of-interest defined by a 3D bounding box in front of the camera. This bounding box extends 1500mm to the front of the camera (Z-axis), 245mm to the left and 175mm to the right of the camera (X-axis), and no limit on the Y-axis (height). The limit on the X-axis (width) aims to ignore points that are outside the walker frame, since most gait motions are performed between the legs of the walker (in the X axis). Also, the limit on the Z-axis (depth) is sufficient for most users since it is difficult for most users to hold the walker and yet be more than 1500 mm away from the camera mounted on the walker. After the two filters, all remaining points are classified as foreground, which should correspond to the user's legs only. Pixels with missing data have a raw depth value of 2047 and are automatically ignored to avoid further processing.

4.3. Transition Function

We adopt a constant velocity model on the joint angle parameters:

$$\overrightarrow{X}_{t+1} = 2\overrightarrow{X}_t - \overrightarrow{X}_{t-1} + \epsilon, \quad (3)$$

where ϵ is a zero-mean Gaussian noise with manually adjusted variance. All other non-angle parameters follow a Gaussian noise model with manually adjusted variance. The constant velocity model is appropriate in our application because the motion of walker users is typically slow and it ensures smoothness in leg motions. However, this model is a rough estimate of the actual gait motion, which follows a cyclic pattern and involves sudden changes in velocity at certain points in the gait cycle (e.g. ground contact of the foot; foot lifting off from the ground). Since one leg may occlude the other leg due to the coronal field of view, it is desirable to use a motion model with a prior over likely poses to continue tracking during complete occlusion. Nevertheless, we show in the experiments that the simple constant velocity model is sufficient to enable tracking when one of the legs is partially occluded.

4.4. Likelihood Function

We define two different likelihood functions, which are combined to produce a single final weight for the particle, assuming independence between them. We will also give a brief description of the likelihood functions for the binocular RGB cameras installed on the walker. The results are included as a point of comparison with the Kinect in the experiment section.

4.4.1 Average distance in the 3D space

The first likelihood function is based on the 3D distance between the predicted leg model and the foreground 3D points. We adopt a skeleton representation of the model by selecting the centroid of the top and bottom surfaces of the tapered cylinder as end points. Afterwards, n points are generated uniformly on the line segment defined by the two end points, and together the $n + 2$ points represent the skeleton in the center of the cylinder. To incorporate the width of a tapered cylinder, note that the width changes in a linear fashion along the skeletal points, as shown in Figure 3. Therefore, we can associate each skeleton point m with a distance w_m to ensure that each skeletal point is at a w_m distance away from the closest observed foreground points. To ensure that observed foreground points (F) and model points (M) are close to each other, we need a two-

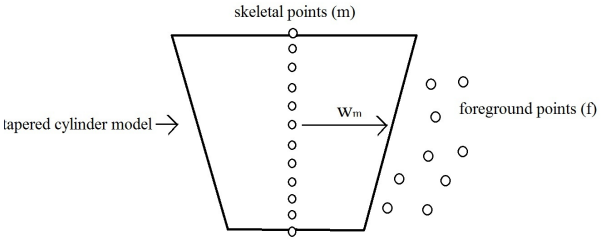

Figure 3. The skeleton representation of the cylinders. The distance functions aim to favor state hypothesis with foreground points close to the front cylinder surface of each leg segment.

way distance metric. The two directed average distances are computed as follows:

$$d_{FM} = \frac{\sum_{f \in F} |w_m - \min_{m \in M}(||f - m||)|}{|F|} \quad (4)$$

$$d_{MF} = \frac{\sum_{m \in M} |w_m - \min_{f \in F}(||f - m||)|}{|M|} \quad (5)$$

We combine the two directed distances into a single distance metric as follows:

$$d_1 = \frac{d_{FM} + d_{MF}}{2} \quad (6)$$

The likelihood given the distance is

$$P(I_1|\overrightarrow{X}) = exp^{(-\lambda_1 d_1)} \quad (7)$$

where λ_1 is manually set to $1/5$.

To improve performance, we use relatively few points from the model (n=10) and the foreground points (randomly choosing 10% of all the foreground points). While including more points improves tracking results, we observed that the loss of accuracy is negligible with the chosen parameters.

The parameters λ_1 and λ_2 of the exponential functions in Equations 7 and 9 are manually adjusted. At one extreme, if the parameter is too small, the values returned by the exponential function will be closer to 1 and very similar, making it difficult to distinguish between good and bad particles. At the other extreme, if the parameter is too big, then it is possible to run into numerical problems where the numbers returned for most particles will be very close to 0, or exactly 0 due to the representation accuracy of computers. The parameters are basically manually adjusted to balance between the two extremes.

4.4.2 Pixel-wise distance in the 2D image plane

The second likelihood function is based on the pixel-wise depth distance between the model-projection image and the Kinect image. The projection of 3D cylinders follows the standard pipeline in 3D graphics. First, cylinders are represented using planar rectangular polygons that circumscribe the surface in the 3D model. Polygons that are not visible to the camera are removed using the backface culling algorithm. Afterwards, the 3D polygons are projected onto the image plane space. They are then clipped at the boundary of the image plane, triangulated, and rasterized using the Z-buffer algorithm, which computes the depth value along with each rasterized pixel. We use a simple distance metric that sums the metric depth distance between the projected model image P and the Kinect image K at each pixel (i,j), with resolution of 640 by 480:

$$d_2 = \frac{\sum_{(i,j)} |z^P_{i,j} - z^K_{i,j}|}{(640 \times 480)} \quad (8)$$

Pixels that do not belong to any rasterized polygons (i.e. background pixels) have a depth value of 0 in image P. Likewise, pixels that correspond tothe background in image K have a depth value of 0. As a result, mismatched foreground/background pixels in the two images will correspond to a high distance. This scheme effectively favors particles with projections overlapping the foreground pixels from the Kinect image.

The likelihood given the distance is

$$P(I_2|\overrightarrow{X}) = exp^{(-\lambda_2 d_2)} \quad (9)$$

where λ_2 is manually set to $1/3$.

4.4.3 Combining the likelihood

Assuming conditional independence, we combine the likelihood probabilities based on the two distance functions as follows:

$$P(I|\overrightarrow{X}) \propto P(I_1|\overrightarrow{X})P(I_2|\overrightarrow{X}) \quad (10)$$

While the independence assumption allows easy addition of image cues and camera observations, this assumption is not true in general since the likelihood functions are calculated from the same underlying observation. When the independence assumption does not hold, a state hypothesis that results in low distance for one image cue is likely to result in a low distance for a second one. Simply multiplying the likelihood functions may lead to sharp peaks in the likelihood distribution. For our tracker, the projection cue focuses more on the X-axis by giving much higher preference for the particles that have perfectly aligned projections with the foreground even if the depths of those particles are not close to the Kinect values. On the other hand,

the 3D cue does not focus on the lateral error as much as the projection cue does, so the lateral error can be compensated by comparatively smaller errors in the Z-axis. Even though both likelihoods are based on the 3D distance of model points/pixels to Kinect points/pixels and are thus not independent, they focus on different dimensions of the same error and as a result the accuracy improves when they are combined. The results on step width (X-axis) and length (Z-axis) in section 5 support this reasoning.

4.4.4 Likelihood for the RGB cameras

In addition to the Kinect camera, we also installed 2 RGB cameras to the sides of the Kinect, approximately 20 cm apart. We perform tracking with these 2 cameras, separately from Kinect, as a point of comparison in the experiment section.

In the first frame before tracking, we manually label the two images by specifying the image regions that correspond to each leg segment. Based on the labelled region, we construct a histogram of colors (HoC) in the HSV space and histogram of oriented gradients (HoG) for each leg segment. For subsequent frames during tracking, HoC and HoG will be constructed from the pixels belonging to the predicted model projection. The newly constructed HoC and HoG will then be compared against the template HoC and HoG constructed in the first frame by taking a L_1 distance of each bin in the histograms. The likelihood given the distance follows an exponential distribution with manually adjusted parameters similar to the likelihood functions we use for the Kinect.

Since two cameras are installed on the walker, the number of observations and the corresponding likelihood is doubled. With this formulation, depth information is incorporated implicitly in a probabilistic fashion, since states with depth errors will generally not fit both image observations. The likelihood for each distance measure and camera observations are combined assuming independence, as described in the previous section.

5. Experiments

In the following experiments, we measure step width and step length errors of the mean prediction over 5000 particles against ground truth obtained with a GaitRITE mat (array of pressure sensors that measures the spatial location of the feet when they are on the mat). Although we are interested in validating the entire 3D model, in the experiments we only report step length and step width measures for two reasons. First, as important determinants of the stabilizing torques required to maintain whole-body balance, step length and width are important biomechanical measures of gait. From a clinical perspective, physical therapists routinely use step length and width to assess gait recovery.

Table 1. Number of frames and steps for each subject in the experiment.

Subject	# of Frames	# of Step Frames	# of Steps
1	309	59	12
2	500	39	10
3	437	89	19

Second, through visual inspection we see that most errors happen at the feet instead of other leg segments. Thus, the step measures can be seen as upper bounds on the error we expect across the whole 3D model.

We collected data with 3 subjects who walked forward and then backward on the GaitRITE mat in an indoor environment. In order to synchronize the tracking data with the GaitRITE data, we manually extract frames in which the user made a step on the mat (i.e. when both feet are on the mat) and compute step length/width measures only on those frames. The total number of frames, number of frames corresponding to steps on the mat, and the number of steps are shown in Table 1.

The mean and standard deviation of the errors in step length and step width are summarized in Tables 2 and 3 for different likelihood functions. The results suggest that the distance metric computed in 3D space (cue 1) has lower error in step length, while the one computed in 2D image space by projection (cue 2) has lower error in step width. More importantly, the result of the combined function is generally close to, and mostly better than, the best results of the two separate cues within each subject.

In order for the tracker to be used for clinical studies, the step length/width errors need to be smaller than the variability in step length/width in the tested population. Owings [16] reported step length variability of 1.4 cm to 1.6 cm and step width variability of 1.4 cm to 2.5 cm with subjects walking on a treadmill under different conditions. According to Table 2 and 3, the error of our tracker with the combined likelihood is slightly larger than the reported variability in their study.

Note that the RGB cues have significantly higher error than the Kinect cues for both measures. We observe that there are two reasons for this. First, due to the changing background, from time to time there are new regions in the image that have similar color as the leg segments. These regions pull the leg predictions away from the true location of the legs and create instability for the tracker. Second, changing lighting conditions and walking motions in uncontrolled environments change the gradient and color information of the tracking targets dramatically in comparison to the reference template. These two factors significantly contribute to the poor results for the RGB cameras.

Readers are invited to check the supplementary material for video results of the combined cue. The tracker successfully tracks the legs over the entire sequence for each of the

Table 2. Mean and standard deviation of step length error of each cue in cm

Subject	Measure	Cue 1	Cue 2	Combine	RGB
1	Mean	4.67	7.11	2.73	21.90
	Std	1.66	3.11	2.31	9.95
2	Mean	3.88	15.46	3.87	10.66
	Std	4.69	6.98	2.27	8.67
3	Mean	4.60	4.77	3.48	16.84
	Std	2.28	4.23	2.02	10.36

Table 3. Mean and standard deviation of step width error of each cue in cm

Subject	Measure	Cue 1	Cue 2	Combine	RGB
1	Mean	8.71	2.37	2.87	6.51
	Std	1.07	1.91	2.50	4.55
2	Mean	7.00	3.45	3.04	6.28
	Std	2.01	2.43	2.52	6.00
3	Mean	4.56	2.36	1.75	6.76
	Std	2.48	1.22	1.17	4.44

3 subjects. As demonstrated in Figure 4, tracking is successful even when there is significant missing data, shown as black patches in the color-coded depth image. During the few frames when more than half of the points are missing, the prediction of the foot goes off-track slightly. However, tracking recovers quickly when the points are observable again in the last 3 frames.

Likewise, the back leg is periodically occluded by the front leg during walking. Such occlusion is most severe when the subject makes a step that has high step length and low step width. As shown in Figure 5, the tracker is able to infer the location of the partially occluded leg segments in the first 5 frames. In the next 4 frames, tracking temporarily fails for one of the legs, in which the foot is totally occluded and the calf is heavily occluded as well. The tracker mistakenly predicts the foot is in the air as opposed to on the ground. Nevertheless, tracking resumes successfully when the leg is visible again as shown in the last 3 frames of the figure.

Note that all the images in Figure 5 and 4 correspond to the second subject who wears baggy pants that violate our cylindrical model of the legs. Although this subject has the highest error in both step length and step width for the combined cue as shown in the tables, the difference is small, and the visual results suggest that the limbs are tracked successfully over the entire sequence with few off-tracked frames. In summary, this preliminary experiment shows that our tracker is robust against moderate missing data, partial occlusion, and deformable tracking targets.

The software is implemented in Matlab, where a portion of the code involving distance calculation and 3D projection is written in C++ that interface with Matlab through mex files. The current running time to process one frame with 5000 particles, including the computation of both distances and segmentation, ranges from 14 to 19 seconds on a modern 2.5 Ghz computer, with parallelized computation of particles over 2 cores using Matlab's parallelization facility. We believe that a significant speed-up is possible by: parallel computation of particles in C++ instead of Matlab, rasterization through the GPU OpenGL instead of CPU for the 2D cue, and a better data structure for finding the nearest neighbor for the 3D cue such as storing the points in a 3D-tree instead of a list. One advantage of our tracker is that it does not require any user intervention or manual initialization as commonly required in many trackers. As tracking is usually done offline by Kinesiologists for assessment purposes, real-time tracking is not necessary and speed is not the primary concern at this point.

6. Conclusion and Future Work

In this paper we designed and evaluated a tracker to estimate the 3D pose of the lower limbs of walker users. The tracker uses a real-time structured-light camera to capture the scene, which is segmented based on depth. We employ a particle filter that combines two likelihood functions designed to complement each other. Our experiment shows that the tracker successfully tracks the 3D poses of users over the entire video sequence. We also demonstrate that the tracker is robust against unfavorable conditions such as occlusion, missing observations, and deformable tracking targets. The system described and tested in the current paper represents a significant advance in ambulatory lower limb tracking. Not only does the system provide spatial measures that accelerometer-based systems do not provide, users are also free from donning sensors and/or markers on the body. While the errors of the system are large compared to clinically relevant values, the initial system tests and avenues for improvement remain highly promising. In future work, we plan to improve the motion model by using physics-based models that better respect the laws of physics and therefore produce gaits that better resemble human motion [5, 4]. Another direction is to learn the dynamics from data on a lower dimensional space even if the pose space is high dimensional [17]. Finally, we believe there is still room for improvement in the likelihood model. In this paper we use cylinders to represent the legs, which may not be suitable if the user wears loose-pants or skirts. We plan to use deformable or data-driven models in the future.

References

[1] Open kinect. http://openkinect.org/wiki.

[2] Prime sense technology. http://www.primesense.com.

[3] J. Bouguet. Camera calibration toolbox for matlab, 2001.

[4] M. Brubaker and D. Fleet. The kneed walker for human pose tracking. In *Computer Vision and Pattern Recognition (CVPR)*, Anchorage, 2008.

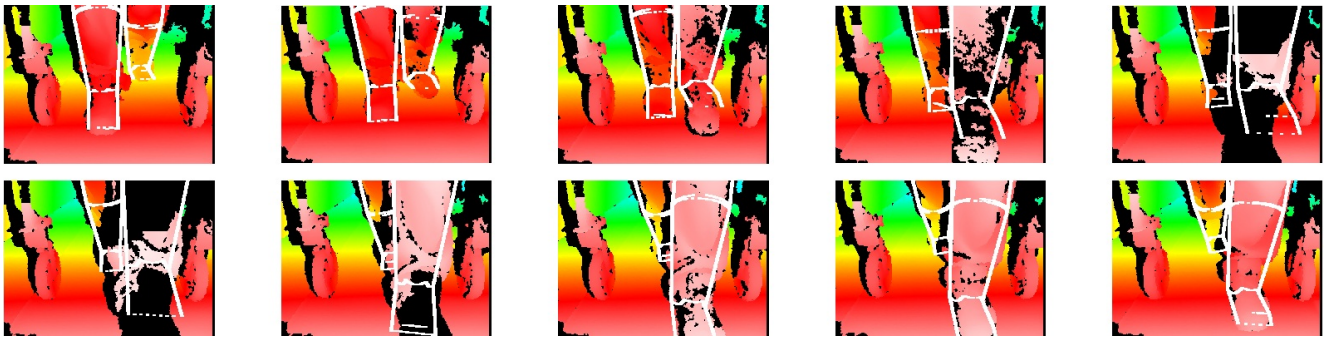

Figure 4. Images with significant missing data shown as black patches. Tracking is generally successful even under such unfavorable conditions.

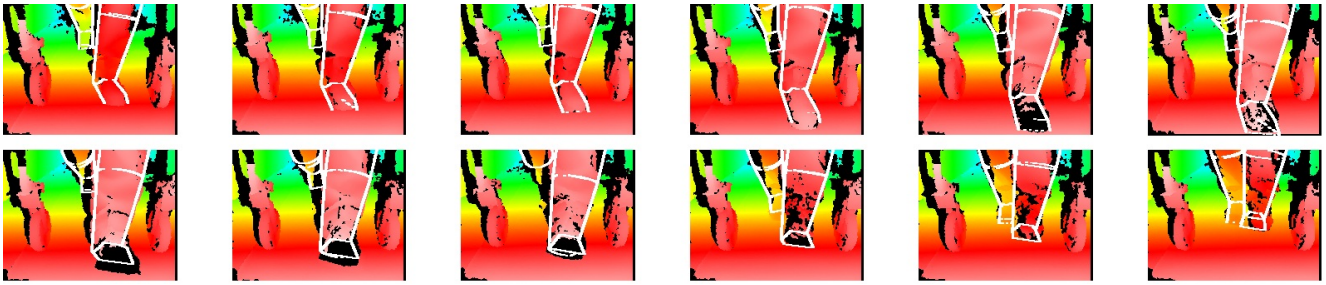

Figure 5. The left leg is significantly occluded by the right leg when the subject makes a backward step with small step width. Tracking is successful during partial occlusion (first 5 frames) of the foot, temporarily fails by hanging the foot in the air during total occlusion of the foot (next 4 frames), and successfully recovers when the leg is visible again (last 3 frames)

[5] M. Brubaker, D. Fleet, and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87:140–155, 2010.

[6] K. Cheung, T. Kanade, and J.-Y. B. M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Computer Vision and Pattern Recognition*, 2000.

[7] K. Choo and D. Fleet. People tracking using hybrid monte carlo filtering. In *International Conference on Computer Vision*, 2001.

[8] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition*, 2000.

[9] P. Fua, A. Gruen, N. DApuzzo, and R. Plankers. Markerless full body shape and motion capture from video sequences. In *Symposium on Close Range Imaging, International Society for Photogrammetry and Remote Sensing*, 2002.

[10] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. In *International Journal of Computer Vision*, 1998.

[11] N. Jojic, M. Turk, and T. S. Huang. Tracking articulated self-occluding objects in dense disparity maps. In *International Conference on Computer Vision*, 1999.

[12] R. Kehl, M. Bray, and L. VanGool. Full body tracking from multiple views using stochastic sampling. In *Computer Vision and Pattern Recognition*, 2005.

[13] S. Knoop, S. Vacek, and R. Dillmann. Modeling joint constraints for an articulated 3d human body model with artificial correspondences in icp. In *International Conference on Humanoid Robots(Humanoids)*, 2005.

[14] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. In *International Journal of Computer Vision*, 2003.

[15] Q. Muhlbauer, K. Kuhnlenz, and M. Buss. A model-based algorithm to estimate body poses using stereo vision. In *17th International Symposium on Robot and Human Interactive Communication*, 2008.

[16] T. Owings and M. Grabiner. Variability of step kinematics in young and older adults. *Gait Posture*, 9:20–26, 2004.

[17] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 238–245, 2006.

[18] R. van der Merwe, A. Doucet, J. F. G. de Freitas, and E. Wan. The unscented particle filter. In *Adv. Neural Inform. Process. Syst*, 2000.

[19] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. In *Computer Vision and Image Understanding*, 1999.

[20] H. Zhou and H. Hu. Human motion tracking for rehabilitation: A survey. In *Biomedical Signal Processing Control*, 2007.

[21] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. In *Computer Vision and Pattern Recognition*, 2006.