Vision-Based Observation Models for Lower Limb 3D Tracking with a Moving Platform

by

Richard Zhi-Ling Hu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Tracking and understanding human gait is an important step towards improving elderly mobility and safety. This thesis presents a vision-based tracking system that estimates the 3D pose of a wheeled walker user's lower limbs with cameras mounted on the moving walker. The tracker estimates 3D poses from images of the lower limbs in the coronal plane in a dynamic, uncontrolled environment. It employs a probabilistic approach based on particle filtering with three different camera setups: a monocular RGB camera, binocular RGB cameras, and a depth camera. For the RGB cameras, observation likelihoods are designed to compare the colors and gradients of each frame with initial templates that are manually extracted. Two strategies are also investigated for handling appearance change of tracking target: increasing number of templates and using different representations of colors. For the depth camera, two observation likelihoods are developed: the first one works directly in the 3D space, while the second one works in the projected image space. Experiments are conducted to evaluate the performance of the tracking system with different users for all three camera setups. It is demonstrated that the trackers with the RGB cameras produce results with higher error as compared to the depth camera, and the strategies for handling appearance change improve tracking accuracy in general. On the other hand, the tracker with the depth sensor successfully tracks the 3D poses of users over the entire video sequence and is robust against unfavorable conditions such as partial occlusion, missing observations, and deformable tracking target.

Acknowledgements

I am heartily thankful for my supervisor, Pascal Poupart, whose guidance, support, and encouragement enabled me to develop an understanding of the subject. I am deeply grateful for his effort to explain things simply and clearly, his great devotion to my thesis and the research work, and his important advice; all of which gave me confidence to explore the research subject without getting lost in the exploration. The thesis would not have been possible without my supervisor.

I would like to express my gratitude to my colleagues. I want to thank Adel Fakih for his guidance and advice in various aspects of the research. His knowledge of the project and thoughtful opinions have provided me with important insights and directions for the project. The work of Adel and his colleague Samantha Ng provides an important foundation for the thesis. I would like to acknowledge Adam Hartfiel, who offered significant contributions, ideas, and opinions to the project and made the publication of the work possible. I would also like to thank James Tung, who helped set up experiments for the project and offered important opinions in the perspective of engineering and kinesiology. It's been a pleasure to work with all of you.

I wish to express my appreciation to the readers of the thesis, Jesse Hoey and John Zelek, who provided important feedbacks on the thesis. Special thanks to Jesse for continually providing important ideas and guidance in many aspects of the thesis and steering my research in the right direction.

Outside the lab, plenty of important people offered informal and/or emotional support during my study. Among many others, I wish to thank Sishi Huang, Eric Fan, Siu Pak Mok, Alex Chan, Wilson Leung, and Marco Xie for keeping me happy and offering support outside of my study.

Finally, despite geographical distance, my parents were always nearby. I would like to show my deepest gratitude to my parents, and to thank them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

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 cameras mounted on the walker, as shown in Figure 1.1.

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 in various situations, such as recovery following orthopedic surgery (e.g. joint replacement). Zhou and Hu [73] 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 behavior

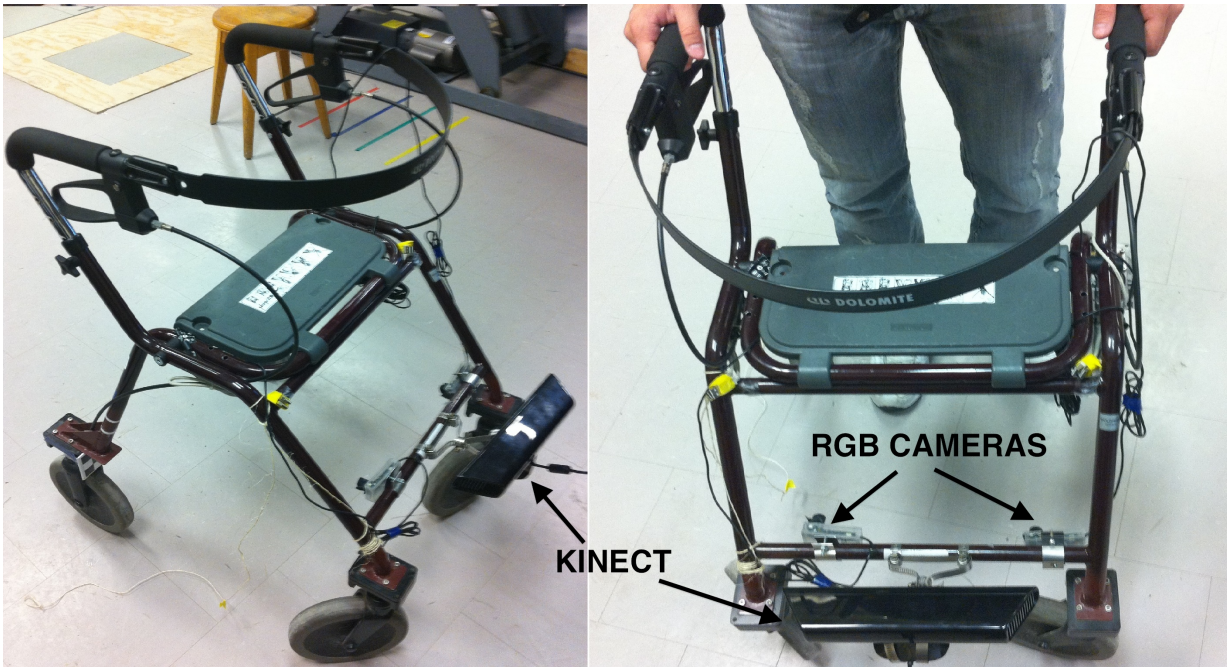

Figure 1.1: Setup of cameras on the walker.

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 and Optotrak 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.

If the system is able to run in real-time, the pose information can be incorporated into the walker systems (e.g. prompting system for fall prevention) for online assistance. First, the orientation and position of the legs during walking is a key determinant of the user's balance, and must be considered in the control of high level assistance. For example, navigational assistance to avoid obstacles may de-stabilize the user if the steering direction is not in line with the user's intent [5]. Second, the pose information can be used for online

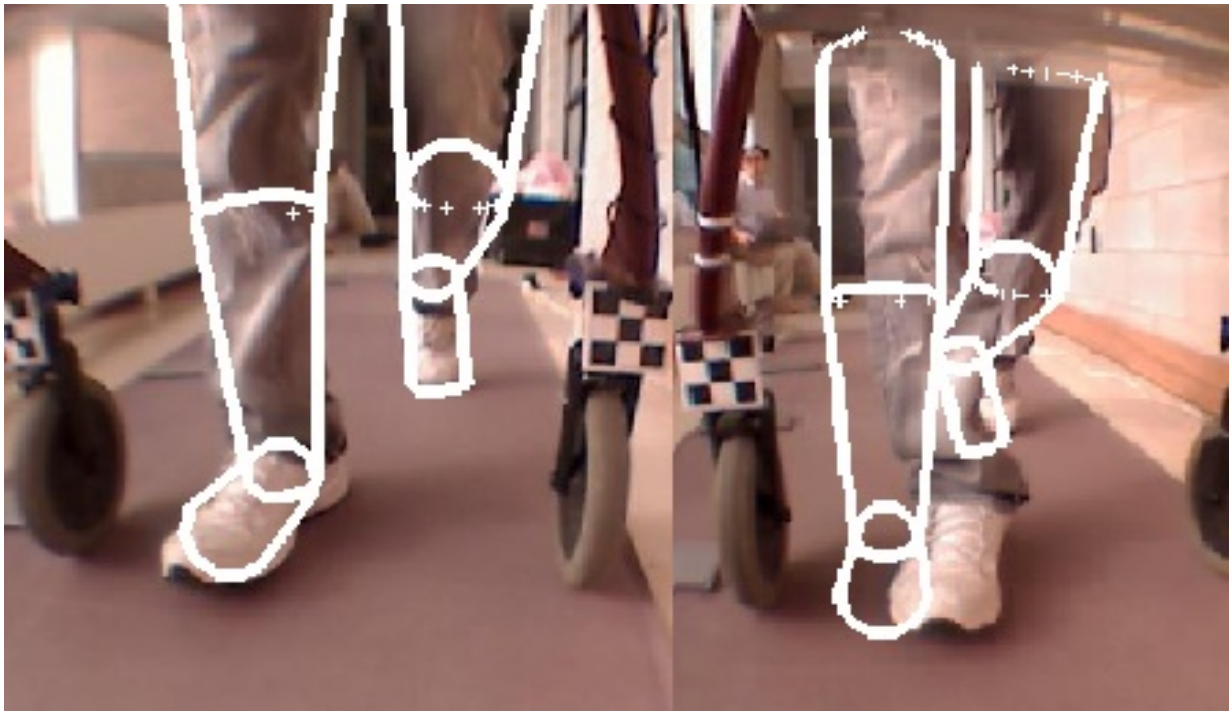

Figure 1.2: Cameras are installed to capture coronal views of the legs

fall prevention. There are often behavioral indicators which are precursors to falls, such as step width with high variability, that can be discovered through accurate pose tracking. Upon detecting these indicators, a prompting system can be used to warn the user of possible falls.

Tracking with a monocular RGB camera for our problem is inherently difficult for two main reasons:

- Since the camera is rigidly attached on the walker, the camera moves over the course of tracking. As a result, both the lower limbs and the background move with respect to the camera's reference frame, making it difficult to separate the background from the limbs. Moreover, since the tracking system is designed for uncontrolled environments, a dynamic background often offers unpredictable distractions that can confuse the tracker. The appearance of the tracking targets (i.e. lower limbs) may

also undergo significant appearance change over the course of tracking. This is due to both change in lighting conditions and gait motions.

- Images of the coronal plane of the body are acquired (Fig. 1.2) since it is not possible to install a camera on the side due to physical constraints of the walker. 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.

The goal of this thesis is to build and evaluate a system for tracking human lower limbs with cameras attached to a moving walker operating in an uncontrolled environment. To achieve this goal, this thesis presents a vision-based probabilistic tracking system with particle filtering. With the above challenges, it is clear that better camera systems than monocular vision are needed. This thesis evaluates three different camera setups:

- Monocular RGB Camera: obtains a 640 by 480 image of RGB values in each pixel. This is the baseline setup which suffers from significant depth ambiguity. Dynamic background and changing appearance of the foreground can negatively impact tracking accuracy.

- Binocular RGB Cameras: includes two separate RGB cameras that are jointly calibrated. Although depth information is not explicitly computed and the system relies only on 2D image appearance (i.e. color and gradient) of the tracking targets, combining the likelihood functions from both cameras implicitly handles depth information in a probabilistic fashion. Nevertheless, this setup still suffers from the problem caused by a dynamic scene.

- Monocular Depth Camera: obtains a 640 by 480 image of depth values for each pixel. Depth information is explicitly stored, but there is no RGB information.

1.2 Contribution

The contributions of the thesis are the following:

- Novelty of the problem: the setup of the cameras, dynamic scenes, and requirement of depth information present significant challenges which, taken together, are not common to most tracking applications in the literature. The problem of tracking 3D pose in a dynamic environment has applications in vision systems for robotics and moving vehicles. Also, the design of an effective approach for tracking the lower limbs based on each camera setup is novel in body modeling and system engineering.

- Evaluation and clinical implication: a thorough evaluation of different camera systems for a novel application of tracking human lower limb pose with a moving walker is presented. More specifically, the difficulty of tracking with an RGB camera in a dynamic environment and possible strategies to mitigate the problem are investigated. Also, tracking with a single depth camera, Kinect [3], is demonstrated to produce high accuracy even under unfavorable conditions. This tracking system 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 tests and avenues for improvement remain highly promising.

Chapter 2

Background

2.1 Tracking with a single RGB camera

The literature of tracking human pose from a single RGB image is vast. In this thesis, the literature will be presented to focus on two specific aspects relevant to our problem: dynamic scene and depth ambiguity.

2.1.1 Dynamic Scene

Dynamic Background

While there exists a large body of literature on human tracking, the majority focuses on a fixed camera setup in which the background remains stable over the course of tracking. The unchanged background can be exploited by background subtraction [54] and optical flow [13] algorithms to significantly narrow the search for the ideal pose and eliminate distractions offered by the background. For moving cameras, this problem becomes more difficult.

Jung [35] and Cohen [20] attempt to remove the camera motion from the images so that moving objects can be detected using optical flow. Camera motion parameters are

estimated by computing the transformation of the corresponding background image feature sets detected in two consecutive frames. In order to separate the features between the background and the object of interest, application-specific heuristics/assumptions are needed. Jung and Sukhatme [35] use an outlier detection approach to detect foreground regions, assuming that the portion of moving objects in the image is relatively small compared to the background.

Sheikh et al. [60] build a model of background by estimating from trajectories of salient features across frames. The authors use background trajectories and foreground trajectories to create background and foreground appearance models.

Hayman et al. [24] present a statistical background subtraction method for mobile observers. It involves a 2-part algorithm that accounts for camera pan and tilt motions and a Bayesian approach about whether the background has been uncovered by a moving foreground object. However, this algorithm is applicable only when the motion of camera involves pan and tilt only (i.e. no translation).

Although we did not incorporate background subtraction in this thesis, it remains an important task for future work, since the dynamic background provides significant distractions for tracking by presenting objects or image patches similar to the tracking target. Having a background/foreground segmentation may eliminate most false positives induced by the background.

Dynamic Foreground

A dynamic scene with a moving subject affects the appearance of the tracking target due to lighting changes and geometric changes. Such appearance change of target can confuse the tracker.

One technique to handle the problems of lighting and geometric changes is to dynamically update the appearance model based on the mean prediction of the tracker in a particle filter framework [51]. A drawback with this approach is that it suffers from a chicken-and-egg problem that may lead to divergence, so careful preventive techniques need to be incorporated to prevent updating the appearance model when tracking is unstable.

Another method is to accumulate an appearance model over time. Jepson et al. [34] formulates measures of stableness for image features used in tracking to adaptively adjust the image appearance model. To achieve the goal, an appearance model is constructed as a mixture of three components: an image structure that is learned over a long period of time, a two-frame motion, and an outlier process. The model is built upon wavelet appearance which provides a significant degree of amplitude and illumination independence. The parameters of the appearance model are estimated online using Expectation Maximization (EM).

Lepetit et al. [41] introduce a framework that treats the 3D pose estimation problem as a key-point classification problem, with the randomized forest as the classification tool. For each randomized tree, each non-terminal node contains a simple test that splits the image space, and the test compares the brightness of three pixels in small neighborhoods around the keypoints. The leaf node contains a distribution of the classes. Since building the optimal tree is hard, multiple randomized trees are used to approximate the optimal tree. Due to the requirement of a large number of training images for robustness with respect to pose and illumination changes, the authors synthesize new views of objects using rendering techniques to generate different viewing conditions for each keypoint. The authors show that their approach is robust to deformation, lighting changes, motion blur, and occlusion.

Color space selection is an important strategy for tracking under dynamic scene with varying lighting. Moreno-Noguer et al. [45] present a scene-dependent color space that is computed as a calibration procedure before tracking. The criterion of a good color space is one that maximizes the distance of the color points between the target and the background. This color space computes a hyperplane that best separates the two classes through linear discriminant analysis. The authors demonstrate that the resulting color space is quite robust to lighting changes, and performs better than RGB and HSV space. Under the assumption that the color distribution changes continuously in time in small amounts, Stern et al. [66] present an algorithm that adaptively switches color space for face tracking.

More complex handling of color changes and shadowing uses detailed 3D body models and exploits shadowing effects for robust pose and light source recovery, but these methods require intensive optimization and are not suitable for online use at this point [9].

In this thesis, different techniques are employed to minimize the effect of changing appearance. First, we use a combination of gradient and color information to track. Second, to handle changes in lighting, different color spaces are explored. Third, multiple target templates are used to correct for appearance changes due to walking motion.

2.1.2 Resolving Depth Ambiguity

Inferring 3D pose from a single RGB image is an underdetermined problem due to the lack of depth information with a single 2D image. As a result, there is significant ambiguity in the projection of human poses onto the image plane, and additional information besides image observation is needed to disambiguate between the possible poses. Viewing under a Bayesian perspective, such information often takes the role of prior knowledge that encodes likely human motions, capitalizing on the fact that the set of typical human poses is far smaller than the set of possible ones. This section explores major approaches in monocular human pose tracking, with an emphasis on how the approach resolves depth ambiguity.

For example, regression based and exemplar based methods predict poses given the existing database of motion capture data or ground truth data, ensuring that the prediction is probable given the training data [4, 59]. Alternatively, one can generate possible poses first, then prune or eliminate unlikely poses based on kinematic constraints or probabilistic models. Hierarchical search of pose space and variants of this approach limit the search space to satisfy certain kinematic or application-specific constraints [47, 64]. Such approaches are especially attractive to estimate high dimensional poses. In a top-down approach such as particle filtering or Kalman filtering, the prior knowledge takes in the form of a dynamic model, which generates better possible poses for evaluation in the next time step.

Learning Based

Under the bottom-up approach of creating an image-to-pose mapping, one way to resolve depth ambiguity is to learn a general model that maps 2D image observations to likely 3D poses from data that contains the likely motion one wants to capture.

Agarwal et al. [4] use a learning based method to track 3D human body poses from a histogram of edges of silhouettes extracted from images. A database of motion capture poses together with the histogram of edges for each corresponding silhouette is stored before tracking. The histogram extracted during tracking is then fed into the Relevance Vector Machine to regress a pose.

Howe et al. [32] present a system for single-camera tracking using a learning-based approach, relying on prior information learned from a labeled training set. The system tracks joints and body parts in the image plane as they move in the 2D video. Since 3D reconstruction from 2D data is under-determined, ambiguity is resolved by a training set of 3D human motions to probabilistically eliminate implausible poses. This is done by modeling the training data as a mixture of Gaussians, and Expectation-Maximization (EM) is used to find the most probable 3D pose based on the 2D information.

Elgammal et al. [25] infer 3D body poses from silhouettes using activity manifold learning. The authors learn view-based representations of the activity manifold and the mapping function between the manifold and the visual input as well as the 3D body pose.

Rosales et al. [57, 56] present a non-linear supervised learning architecture, called Specialized Mappings Architecture (SMA), that recovers body poses from monocular images. In general this is a divide-and-conquer algorithm that splits the input space (e.g. Hu moments of the silhouette) into several regions and approximates simpler functions to fit the input-output relationship specific to a region. This architecture consists of forward (input-to-output) mapping functions and a feedback matching function that are estimated from data. A probabilistic model for the architecture along with the EM algorithm for estimating the model's parameters are presented.

Brand [11] uses an HMM model to represent the mapping from 2D silhouettes to 3D poses. To resolve the ambiguity in projection, information over the entire sequence is integrated. The function between inputs and outputs is a dynamic manifold modeled from data with an HMM model obtained by entropy minimization.

Since bottom-up image-to-pose approaches are different from the top-down approach presented in this thesis, it would be interesting to combine them with our top-down approach [62]; however, we leave this as future work. For instance, the top-down filtering

approach can fail when subject motion does not follow the dynamic model (i.e. particles are not generated in the right location). In such case, a bottom-up image-to-pose approach may be able to compensate.

Exemplar Based

Exemplar based methods are similar to the learning based methods in that the pose is estimated based on a large set of examples given the input image observation. In contrast to learning methods that construct an explicit description of the target function, exemplar-based methods simply store the training examples. Generalizing beyond these examples is delayed until a new instance needs to be classified. The difference with learning based methods is that the 3D pose of each instance is estimated locally and differently instead of estimating the target function once from the training data. This is advantageous when the target function is complex, but 3D poses can be estimated by a set of less complex local approximations. On the other hand, there are a few disadvantages. First, the cost of estimating a new instance can be high since all computation takes place during tracking instead of learning. Second, due to the ambiguity of 3D projection onto a single image, common exemplar-based techniques like nearest-neighbor may find close neighbors in the input (image) space that do not necessarily result in a desirable output (3D pose). In other words, there is a many-to-one relationship between the 2D observations and the 3D poses.

Shakhnarovich et al. [59] address both efficiency and depth ambiguity in their paper. The authors learn a set of hashing functions that efficiently index examples and find approximate neighbors in time sub-linear in the number of examples. Moreover, the hash functions are sensitive to the similarity in the output space. This is accomplished by learning a new feature space from examples that links inputs to outputs and reflect the proximity in output space. Finally, the pose estimate is produced by locally weighted regression that uses the approximate neighbors retrieved from the hash function to dynamically build a model of the neighbors of the input.

Mori et al. [46] store a number of 2D exemplar views along with manually labeled 2D body joints. During tracking, test images are matched to 2D exemplars using shape context matching based on sample points from the contour of an object. The labeled 2D

body joints from the exemplar are transfered to the test image. Finally, based on the 2D body joints, a 3D configuration is inferred using Taylor's method [68]. The 2D-to-3D process is inherently ambiguous with a single view, so the authors make some additional assumptions such as an orthographic projection model of the camera, and are able to infer depth up to some ambiguity in scale.

Athitsos et al. [7] present an exemplar based algorithm for simultaneous 3D hand pose and camera viewpoint estimation. Based on image similarity, through edge location/orientation, finger location, and geometric moments, the most similar matches from the synthesized database are retrieved, and their labels are returned as estimates of hand pose. The ill-posed problem of recovering depth information directly from input image is avoided with the camera information provided.

Temporal information can be used to resolve some depth ambiguity from a single image. Howe et al. [32] use a direct silhouette lookup table with Chamfer distance to select a candidate pose. To resolve the many-to-one relationship between silhouettes and 3D poses, the algorithm is integrated with a Markov chain for temporal propagation of 3D poses. Additional smoothing is employed to ensure plausible human motions. Ong et al. [52] investigate viewpoint invariant representations of the mapping from images to poses. To ensure viewpoint invariance of the 3D poses, each 3D pose exemplar is associated with contour information from multiple views. The exemplars are coupled into a particle filter, with the dynamic model providing additional constraints. Dimitrijevic et al. [23] incorporate dynamic information directly into the exemplars, by incorporating spatial-temporal templates. This is done by including visual information from three consecutive frames into a single template. Temporal information can be incorporated in a probabilistic fashion as suggested by Sminchisescu et al. [63]. A mixture of experts is used to model the probability that a particular exemplar generates some visual input. Based on 2D inputs of motion sequence, the best exemplar is retrieved.

For our problem, tracking with an RGB camera is done by comparing the color and gradient information in the projected image region against appearance templates specified in the first frame. This framework is extended to include 30 templates in 30 consecutive frames that correspond to a gait cycle. The templates are stored along with the associated state vector. During tracking, the color and gradient information of a projected hypothesis

will be compared against the template with the closest state vector in terms of Euclidean distance.

Exploiting Kinematic Model

Another popular approach to resolve depth ambiguity is to limit or to better explore the search space of likely human poses by exploiting a Kinematic model, since most candidate poses are unnatural or anatomically impossible and can be eliminated from the search. This includes techniques such as eliminating anatomically impossible/unlikely poses, imposing limits on pose parameters, building a hierarchical structure for body parts. Depth ambiguity is a significant problem for human poses due to the high degree of freedom that can generate similar observations in an image. Hierarchical structures and partitioning of the search space exploits properties of the kinematic model to speed up and better guide the search for a good local minimum in the underlying state distribution.

Mori et al. [47] present a human body configuration recovery algorithm that localizes joints and limbs. First, a segmentation method based on Canny edge and Normalized cut [42] is applied to obtain a segmentation of the background/foreground and parts of the body. Then, a set of low level cues is computed to classify these segments. The cues are based on contour, shape, shading, and focus, and the cues are combined into a probability-like quantity. Given the possible body part candidates, a set of body configurations is generated, pruned, and scored to give the best configuration. Pruning is done based on constraints on body configurations and part-based score. In addition, hierarchical search of body parts during the generation of body configurations reduces the search space considerably.

Lee et al. [40] employ a hierarchical approach for tracking human pose using edge-based features during the coarse stage and later finer features for global optimization. First, foreground blobs are extracted by background subtraction. Then, the blobs are tracked by fitting an ellipse in the image, which provides a coarse estimate of the body pose. Afterwards, components of the body are searched in a hierarchical fashion, starting from the head, to shoulder, and finally to the limbs. This is done by detecting features (e.g. skin color for the head, contour for the shoulder) for each component, generating multiple

candidates for each component of the body, and an optimization process is used to align the candidates to the image features. The set of candidates is evaluated based on edges and foreground matching, and the optimal 2D pose is extracted using dynamic programming. 3D pose inference is done by adding physical constraints of human kinematics, and using the detected 2D pose to bootstrap the search in a MCMC method. The authors noted that the prior 2D inference accuracy needs to be high in order to overcome the depth ambiguity.

Sminchisescu et al. [64] investigate a novel method to resolve the forward/backward (motions in depth) ambiguity of limbs resulting from ambiguous 3D projection. This ambiguity results in 2 configurations of 3D pose having the same projection on the image. The authors present a tree-based search algorithm to produce an efficient "kinematic jump" sampler for the different configurations that speeds up the search for local minima. In each branch of the tree, the algorithm generates a hypothesis that has an approximately-correct image projection. This is achieved by constructing a 3D sphere centered on the currently hypothesized position, and inferring the 2 possible 3D positions of the endpoint in the radius that are consistent with the image observation. The hypotheses are thus of high quality since they match the image observation. The hypotheses are then evaluated using a likelihood function, and further pruned according to kinematic constraints and density propagation to look for the correct minima.

Lee et al. [40] use a multi-level state representation that provides a hierarchical estimation of 3D pose. A graphical model is used, such that each node represents a joint, with a corresponding observation function. Humans are first tracked in blobs to get the coarse position, then parts such as face and limbs are inferred using specialized part detectors. These detections of body components are combined by grid-based belief propagation to infer 2D joint positions. Finally, the detected parts are used to bootstrap the data-driven MCMC to infer a 3D pose. Flip kinematic jumps from [64] are used to explore the depth space.

Stenger et al. [65] propose a tree-based representation of the state distribution. In their problem of tracking the hand pose in 3D, each node in the tree is a non-overlapping set in the state space, defining a partition of the state space based on rotation angle. The posterior for each node is evaluated, and nodes with low posterior are not processed further, so that regions with low probability can be quickly discarded in a hierarchical search. To

further resolve ambiguous situations, temporal information is incorporated in a Bayesian framework of filtering.

The kinematic model of lower limbs in our problem incorporates two constraints. First, we impose a range of values that each parameter of the state vector can take to avoid anatomically implausible poses. Second, we check for model limb segment intersection. Particles whose segments intersect will be assigned a probability of zero, so that the particle will not be re-sampled in the next time step. As future work, the hierarchical structure of human limbs can also be exploited to better place the particles in the state space.

Prior and dynamic Model

Top-down approaches for tracking start the process from the 3D model. They first generate hypotheses of pose estimates according to a prior or dynamic model, and then evaluate the hypotheses according to an observation likelihood. The dynamic model provides an important way to compensate for observation ambiguities due to missing depth information, since it provides a way to encode temporal information or knowledge about human motion to better explore the pose space. Simple zero-order (e.g. Gaussian noise) and first-order (e.g. constant velocity) dynamic models are easy to implement, but are too simple to capture human motions. More sophisticated forms of dynamic models are designed to better capture human motion and density propagation.

Fleet and colleagues [13, 12, 69, 72] develop a series of dynamic models for tracking articulated body pose within the particle filtering framework. Brubaker et al. [13] present a physics-based tracker of the lower body in a fixed environment. Their dynamic model follows Newton's laws of motion. Force during collision, momentum of swing, and the "toe-off" event are modeled by solving the equations of motion in a Lagrangian formulation. The results show that tracking is successful even under total occlusion. Nevertheless, errors in depth are still significant under monocular tracking. The work was further extended to include the knee in the motion equation to better model walking motion [12]. Scaled Gaussian Process Latent Variable Models (SGPLVM) are proposed by Urtasun et al. [69] to learn a prior model of 3D human pose. This model simultaneously optimizes a low-dimensional embedding of the high dimensional pose data and a function that provides

a mapping from the low-dimensional space to the high dimensional space. This model is further developed into Gaussian Process Dynamic models (GPDM) [72]. Like SGPLVM, GPDM provides a low dimensional representation of human motion data that gives higher probability to poses and motions that are closer to the training data. However, SGPLVM is a static model, which does not produce a smooth latent path in the low dimensional space from time-series data. GPDM gives such additional path to provide a dynamic model in the latent space, which results in a smoother trajectory. With Bayesian model averaging the GPDM can be learned with a relatively small amount of data and yet yields good generalization.

The particle filter described in this thesis uses a simple constant velocity dynamic model. This is a weak first-order model that encodes smoothness of human motion. Behaviors associated with typical walking motions (e.g. toe-off, heel strike) are not captured by this dynamic model. In the future, more complex forms of dynamic models as described above can be implemented to obtain a better distribution of the particles. This is especially important for a monocular RGB camera since the observations are inherently ambiguous. Hence a better dynamic model is needed to compensate for the weak likelihood model.

2.2 Tracking with Multiple RGB Cameras

Multiple cameras can help resolve the depth ambiguity problem because information from multiple views can be integrated. This section presents two popular approaches to tracking with multiple cameras: shape-from-silhouettes and stereo.

2.2.1 Shape-from-Silhouettes

Shape-from-silhouettes is a 3D reconstruction technique that attempts to rebuild a 3D model of objects of interest through multiple views. The silhouette image is a background/foreground image, and the 2D foreground mask is the silhouette. The silhouette together with the camera parameters define a backprojection cone, and the intersection of the cone from each camera define a visual hull, or the bounding geometry of the 3D object.

Cheung et al. [18] recover shape, motion, and joints of human body from multiple silhouette images. The algorithm segments points on the silhouettes corresponding to each part of the object and estimates the motion of each individual part using the segmented silhouette. Once the motion of each part is recovered, the joints are estimated by articulation constraints.

Cheung et al. [17] 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.

Miki et al. [44] present a tracking algorithm based on a 3D voxel reconstruction of the person's body from 2D silhouettes extracted from four cameras. In the first frame, template fitting is applied to first fit the head, torso, and then the limbs against the 3D voxels for initialization. The initial estimate of location, orientation, and size for each body part is then used as a measurement for the Extended Kalman Filter (EKF) to ensure a valid body model during tracking. To obtain a new measurement in the next frame, the voxels are assigned to body parts by minimizing the distance from the EKF prediction.

Kehl et al. [36] present a shape-from-silhouette tracking algorithm from 4 cameras using stochastic meta descent (SMD) optimization, a variant of gradient descent with local step-size adaptation. To make the SFS algorithm more accurate, the authors assign a representative color to each surface voxel during initialization and compare observed colors to the representative colors during tracking. The color model is represented by a Gaussian distribution in YUV space, and the mean is updated during tracking to account for occlusion and lighting changes, through a weighted combination of the old mean and the new value.

One problem of the Shape-from-Silhouettes technique is that it assumes the background/foreground image can be obtained. This is not true in our application since the dynamic scene makes background subtraction difficult.

2.2.2 Stereo Camera

The stereo system is an attractive alternative setup for the walker due to its low cost and low energy consumption. Based on the known parameters of the camera system, a dense depth map can be computed with a stereo matching algorithm. Beymer and Konolige [10] install a stereo camera on a mobile robot that tracks people while moving. A dense depth map is produced with the stereo camera to separate the background from the object of interest. Working with a stereo camera is similar to working with a depth camera as both outputs are dense depth map.

Unfortunately, stereo matching and epipolar geometry are not suitable for our application. One main reason is that most people tend to wear pants that are of uniform color and lack gradient information. Also, a significant portion of the image is occupied by the ground that is often featureless. The lack of features in a large region of the image poses a significant challenge for window-based stereo-matching algorithms [58]. The Markov Random Field based formulation may address this uniformity problem by adding a smoothness term to take neighboring gradient information into account [38], however it is very expensive computationally.

In our work, we include an additional camera in a probabilistic framework by combining the likelihood function from separate cameras assuming independence between them. Depth ambiguity is thus resolved in a probabilistic fashion. Details are described in Section 3.4.2.

2.3 Tracking with a Single Depth Sensor

Tracking with 3D images presents significant advantages over regular RGB images as depth information is made explicit.

2.3.1 Model Fitting

Tracking with 3D points can be formulated as a bottom-up, model-fitting problem through optimization or searching a 3D pose that minimizes the distance between the model and

the observed points.

Knoop et al. [37] 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 or a stereo camera, and points from a tapered cylinder model. Grest et al. [30] use ICP to find the correspondence between the observed 3D points and the model points. Afterwards, nonlinear optimization is used to find the optimal pose parameters that map the initially known model points to the observed points. Real-time performance is achieved by analytically deriving the Jacobian matrix and a highly optimized correspondence search. Fua et al. [27] 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. In addition, they use a skeleton model combined with a soft, deformable surface to simulate the behavior of bones and muscles.

The structure of the human body can be used to guide the search for an optimal pose by fitting the model against observed points in a hierarchical fashion. Muhlbauer et al. [48] exploit the structure of the human body to search for the ideal pose by fitting a body pose to the 3D data points obtained. They use a hierarchical scheme by looking for the head first, then the torso, and the limbs, which are then fitted iteratively starting with the joint closest to the torso and continuing outwards. Zhu et al. [74] present a hierarchical tracker for tracking 8 joints of the upper body by feature detection. First, the torso is initialized by finding the foreground mass center and fitting a rectangle on the observed 3D points. Then, a head-neck-trap template is fitted above the torso, and the likelihood of the template is evaluated based on how many 3D points the template overlaps. Finally, the limbs are located by first finding the end-points, which ideally correspond to the hands. The elbows and shoulder are located by tracing the 3D points from the hands back to the torso template.

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 hierarchical 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 amount of 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 optimization or search-based approaches.

2.3.2 Point Classification

Tracking the body pose with a 3D image can be greatly simplified by first classifying each pixel according to the body part it belongs to. Once each pixel is classified, the joint location can be easily located.

Plagemann et al. [55] use graph algorithms to locate extrema 3D points by finding the longest connected path between two 3D points. The extrema points are candidates for hands, feet, or head, all of which are located at the ends of the human body. To classify the points to suitable classes, supervised learning is used to learn the features based on a set of labeled image patches around the designated 3D points. During testing, a square image patch is extracted centered on the extrema points and fed into the classifier. The orientation of the body parts is found by tracing back the path from the extrema points. The orientation is also used to normalize the shape descriptor for classification so that the classifier is robust against 3D rotations.

Anguelov et al. [6] present a data-point labeling algorithm that classifies each 3D point observed from the 3D sensor into a predefined, user-specified class. The algorithm is based on the Markov-Random Field (MRF) model with graph-cut, and the advantage of a MRF is that it ensures local consistency of labels. The authors define node features and edge features for the MRF tailored to the scene. For example, a simple edge-feature can be the Euclidean distance between the neighbor points, and the coordinates of the point can be a node feature. The weights associated with the node features and edge features are learnt through supervised learning with manual labels of each data point from a set of images. During the testing phase, features are computed and labels of each data point are computed using graph-cut with the weights provided by the learning phase.

Chen et al. [15] attempt to segment the 3D points of an object from time-of-flight cameras into meaningful parts according to the curves and shapes. They use a 3D mesh

watershed-based segmentation based on Gaussian curvature and concaveness estimation. The Gaussian curvature can identify elliptic and hyperbolic shapes of a 3D polygonal mesh. However, this method is only applicable to segment objects with salient geometric shapes.

Shotton et al. [61] present the tracking algorithm used by the XBox motion controller. In their work, no temporal information is encoded in that the pose is estimated independently in every frame. Tracking is formulated as a per-pixel classification problem using a decision forest, similar to [41], but the data for each pixel is 3D coordinates from the Kinect camera instead of RGB values. The feature used for each node in the decision forest is based on probing two other pixels that are at a certain offset from the current pixel and computing the depth difference. The feature is normalized by depth so the feature is translation invariant. During the training phase, 3 trees of depth 20 are trained from the CMU mocap database, and each tree randomly generates feature parameters (i.e. probing offset and thresholds) at the root node. Each internal node in the tree evaluates the feature with different offsets, and branches left or right according to the comparison to a threshold. The feature parameters in each internal node are learned by maximizing information gain. The distribution of feature values at each leaf node is learned by partitioning the dataset according to the evaluation of internal nodes, and the distribution is averaged over all trees. The authors avoid overfitting by simply training with a huge collection of images, close to 1 million training images. After each pixel is classified into a specific body part, the joint locations need to be inferred. The authors presented a weighted Gaussian Kernel of the image pixels to provide a density estimator of the body part that is robust against outlying pixels. Mean shift is then used to find modes in this density with a confidence estimate for each mode. The detected modes lie on the surface of the body, so the joint location is inferred by pushing the modes back with an optimized offset. The entire tracking process runs at 200 fps on the XBox GPU.

Although we do not use point classification in this thesis, it is a promising approach worth investigating in future work since it can simplify the tracking problem. The learning approach by Shotton can be directly applied to our problem once training data is available. However, under situations where there is significant missing data, the point probing strategy may be unstable. A better feature should be devised to suit our specific application.

2.3.3 Hidden Markov Model

While the literature on top-down tracking approaches is vast for regular RGB cameras, it is less common for 3D sensors.

Mikic et al. [43] present a full-body tracking algorithm with 3D data. 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. [50] also use EKF for tracking the upper body with 3D data. A statistical image formation model that accounts for occlusion plays a central role in their tracking system.

Ganapthi et al. [29] combine both a top-down with a bottom-up approach in their tracker with a 3D sensor. The top-down part involves a projection of model shapes onto the image. The best estimate of the joint location is found by hill-climbing on the likelihood distribution of the hypotheses. The bottom-up part involves detecting and classifying extremal parts (i.e. head, hands, feet) as in [55]. The detected joint locations are then compared against the projected model joint locations in the hill-climbing procedure so that only the one with the best likelihood is kept. The advantage of combining bottom-up joint detection into the tracker is that it allows recovery following tracking failure by the filtering method alone. The failure can be due to fast movements of the subject and occlusion.

Similarly, Siddiqui et al. [62] also combine a bottom-up with a top-down approach for tracking the upper body. The authors use a MCMC approach to find an optimal pose based on a likelihood derived from both 2D (silhouettes difference) and 3D (pixel-wise depth difference) information, that compares synthesized depth images to the observed depth image. Bottom-up detectors are used to locate candidate positions for the head by searching for the outline of a head shape in the Canny edges of the depth image, and for the forearms by locating endpoints of the foreground image skeleton. To speed up the convergence of the MCMC search, a Markov chain variant explores solutions about the detected parts, and thus combine bottom-up and top-down processing.

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. Future work needs to be done to investigate whether representing multi-modal distribution is necessary with the 3D image acquired. Otherwise, a more efficient

top-down approach, Kalman Filter, can be used instead.

2.4 Summary

The problem of 3D tracking with a monocular camera in a changing environment is still a challenging problem. In the literature, different approaches have been investigated to cope with the difficulty of dynamic background, changing appearance of the tracking target, and lack of depth information. Following the work of Ng et al. [49], this thesis continues exploring the feasibility of tracking under these challenges with a monocular camera setup. Moreover, the approach taken in the thesis for tracking with the binocular camera setup is to combine the observations from the two cameras together in a probabilistic fashion to resolve depth error. This is different than most of the multi-camera methods in the literature, which are based either on shape-from-silhouettes or stereo. Finally, tracking with a 3D sensor is gaining popularity due to the Kinect sensor becoming a household commodity. As discussed later in the experiment section, the images returned by the Kinect may have significant missing data and occlusion. As a result, while the bottom-up and point classification approaches are commonly used for tracking with a 3D sensor, a top-down approach (i.e. particle filter) is used in the thesis for robustness reasons.

Chapter 3

Framework - Hidden Markov Model

Pose tracking can be formulated as a belief monitoring problem with a Hidden Markov Model (HMM) 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. The HMM is specified by four elements:

Hidden State $\overrightarrow{X}$ the leg pose of the walker user. The pose is represented as a state vector $\overrightarrow{X}$ defined in Section 3.1

Dynamic Function $P(\overrightarrow{X}_t|\overrightarrow{X}_{t-1})$ governs how the hidden states evolve over time. We use a simple constant velocity model as described in Section 3.2.

Observation I_t the image data returned by the camera. The raw data is a 640 by 480 image that contains RGB values for an RGB camera, or depth values for the Kinect camera. The image is post-processed to extract meaningful data, such as histograms of color and histograms of oriented gradient for an RGB image, or foreground segmentation for a depth image. For the Kinect, the final observation corresponds to a scalar value that measures the distance between the 3D model and the foreground points. For the RGB camera, it corresponds to a scalar value that measures the distance between the projected 3D model appearance, based on color and gradient, and the template appearance. The camera parameters are needed to convert 3D points

into 2D pixel locations and vice versa, and the calibration procedure is described in Appendix A.1. Details are described in Section 3.3.

Likelihood Function $P(I_t|\overrightarrow{X}_t)$ computes the likelihood of an observation given a hidden state. In this thesis, all likelihoods follow an exponential function. As the observation corresponds to the distance between the 3D model and the expected location/appearance of the legs, an exponential function with manually adjusted parameters that assigns lower weight to higher distance is adequate. Details are described in Section 3.4

A tracking problem can be formulated as a belief monitoring or filtering problem in the HMM framework by evaluating the distribution $P(\overrightarrow{X}_t|I_1...I_t)$. This distribution can be computed according to Bayes rule as follows:

$$P(\overrightarrow{X}_t|I_{1:t}) = \sum_{x_{t-1}} P(\overrightarrow{X}_t|\overrightarrow{X}_{t-1})P(I_t|\overrightarrow{X}_t)P(\overrightarrow{X}_{t-1}|I_{1:t-1}) \tag{3.1}$$

Under this formulation, Kalman filtering for Gaussian distributions and particle filtering [33] are two predominant top-down approaches in the tracking literature. Multiple variants of these two approaches have been published for pose tracking [71, 70, 19, 22].

We employ the particle filtering approach, using samples to represent the underlying distribution of target states. At each discrete time step, each instance or particle is assigned a weight or probability based on the likelihood of observing the camera data given the state of the instance. This distribution, represented by weighted samples, will be re-sampled according to the weights, and the distribution will be propagated to the next time step according to the dynamic model which predicts the next location of each instance in the state space. The distribution will then be weighted by the observation likelihood again in the new time step and the cycle continues.

Because the distribution is approximated by samples, particle filtering can represent any type of distribution, and is especially suitable for tracking applications with multimodal distributions. Such distributions are common in pose tracking. For example, when the background provides significant distractions, the tracking target changes appearance,

or the pose is inherently ambiguous given the view of the camera. The trade-off is that the computational cost is often expensive. This is especially true for high-dimensional data, since the number of particles needed increases significantly with each added dimension, and the computational cost is proportional to the number of particles.

3.1 Hidden States

We adopt a model composed of tapered cylinders for the thigh, calf, and foot of each leg, as shown in Figure 3.1. 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. The algorithm for detecting the intersection of cylinders is described in Appendix A.3.

3.2 Dynamic Model

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

$$\overrightarrow{X}_{t+1} = 2\overrightarrow{X}_t - \overrightarrow{X}_{t-1} + \epsilon, \tag{3.2}$$

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

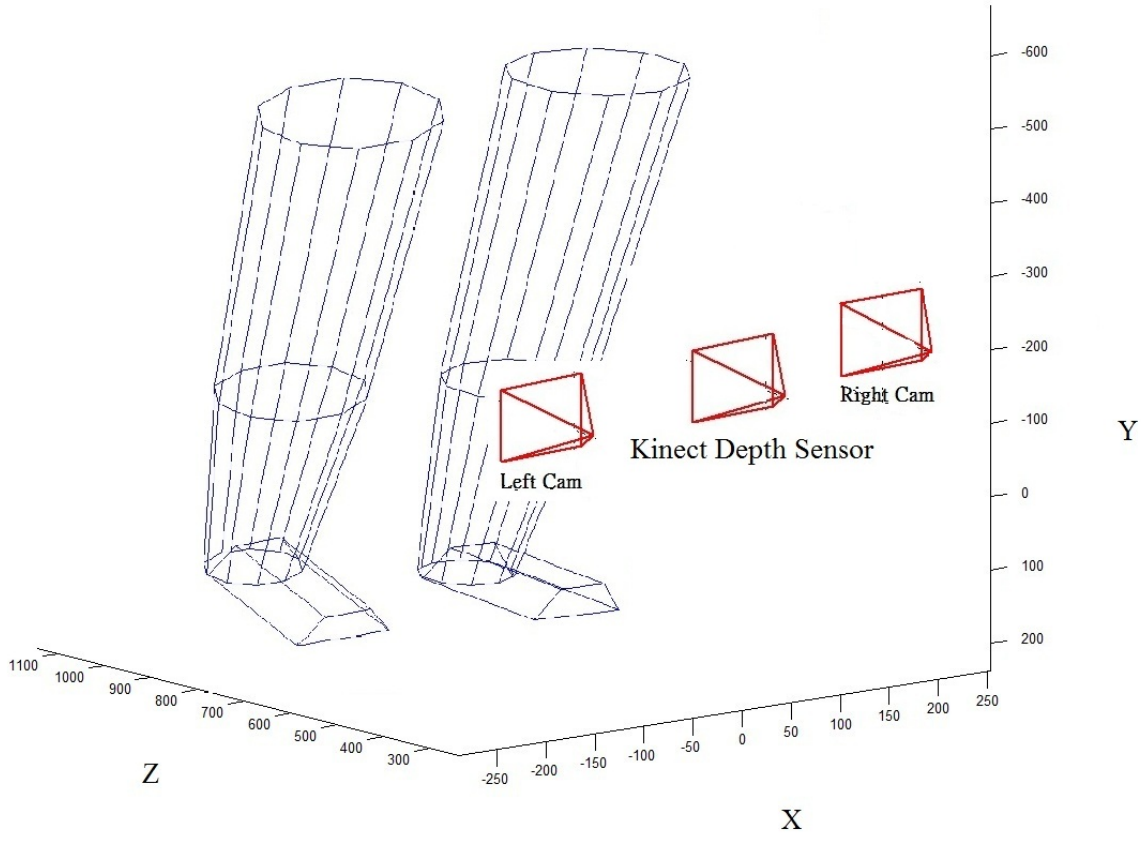

Figure 3.1: Graphical representation of the cylindrical model and the location of the sensors

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.

3.3 Observation

3.3.1 RGB Image

To represent the appearance of each limb segment in an RGB image, our program relies on both color and gradient information. Although not entirely independent, color and gradient information are complementary information that exhibit different behaviors. For example, the color information is robust to geometric changes of target due to motions and rotations. However, in a dynamic environment, lighting conditions can significantly change the color information of the tracking targets over time. This is demonstrated in Figure 3.2 for the histogram of colors in RGB space for the shoes. On the other hand, gradient orientation is known to be robust to lighting changes [14]. However, it is susceptible to geometric changes due to target movement as demonstrated in Figure 3.3. Thus, the two appearance models complement each other for our tracking problem, and it is best to use them together.

Strategies are employed to further correct for the weakness of color and gradient. For color, different representations of color information are investigated on 2 dimensions: distribution representation and color space. We also use a set of 30 templates of the legs in different poses as the appearance model to account for geometric changes.

For each pixel location i in the image plane of camera j, we define the function $s(i, j, \overrightarrow{X})$ to return 1 if the pixel lies within the perspective projection of $\overrightarrow{X}$ on the image plane of camera j, and 0 otherwise. Similarly, $s(i, j, \overrightarrow{C}_k)$ considers the projection of the k^{th} cylinder on the image plane.

Color

Distribution Representation: Histogram and Moments To construct a histogram of colors (HoC), the same procedure from [49] is followed. For the k^{th} leg segment captured by camera j in each color channel of the color space, each pixel i for which $s(i, j, \overrightarrow{C}_k) = 1$ is classified by color channel values into one of 32 bins. So the value of each bin corresponds to the number of pixels falling into the interval of the bin. Each HoC is then normalized

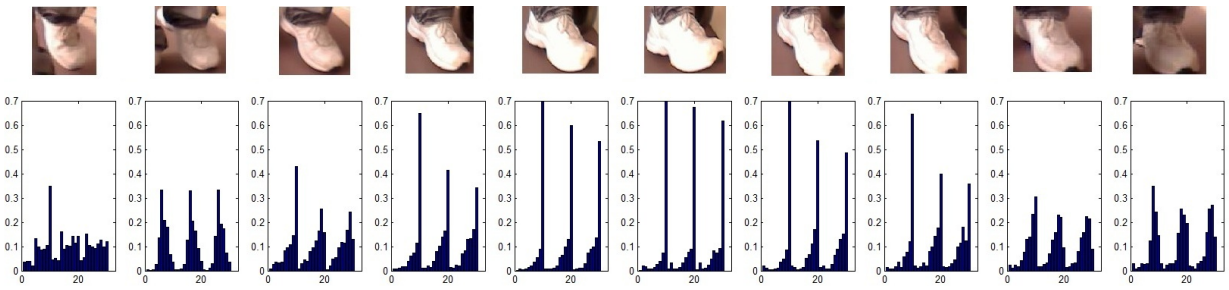

Figure 3.2: Appearance (HoC in RGB space, color channels concatenated) of shoes under different lighting conditions. The shape of the histograms is heavily influenced by the lighting conditions.

by the number of classified pixels. The distance d between the color histograms of a limb segment in two different frames can be measured by

$$d = \sum_{c \in \{R,G,B\}} \sum_{b=1}^{32} |HoC^1_{c,b} - HoC^2_{c,b}| \tag{3.3}$$

where c indicates a color channel and b indicates a bin number.

Histograms provide a descriptive representation of the color distribution and typical distance measurements such as L_1, which is used above, and Bhattacharyya distances involve simple bin-by-bin comparisons. However, such measurement does not take the color similarity between neighboring bins into account, so a slight shift of the histogram due to mild lighting changes can significantly increase the distance.

This thesis explores one possible alternative representation of distribution: the central moments. A probability distribution is uniquely characterized by its central moments. By interpreting the color distribution of an image as a probability distribution, we compute the first three moments (mean, variance, skewness) of each color channel in the color space. By working with only a subset of the moments, this representation does not have the power to model all different types of distribution. In other words, it is possible that two non-identical color distributions may produce the same first three moments. On the other hand, although one can use as many moments as desired, the discriminative power of the statistics

diminishes as more moments are included, and the computational requirement may increase significantly. The trade-off is that while the moment representation is less descriptive than the histogram representation, it is efficient to compute and is more robust to lighting changes [67]. Other alternative representations of color distribution may also be possible, such as a probabilistic formulation of the histograms, or a cumulative histogram [67]. These alternative representations are possible venues for future work.

To compute the color moments for the k^{th} leg segment on camera j, let p_{ci} denote the value of pixel i in color channel c when $s(i, j, \overrightarrow{C}_k) = 1$. Let N be the number of pixels in the leg segment, then the measurements corresponding to the first three moments can be defined as

$$\alpha_c = \frac{1}{N} \sum_{i=1}^{N} p_{ci}, \tag{3.4}$$

$$\beta_c = (\frac{1}{N} \sum_{i=1}^{N} (p_{ci} - \alpha_c^2))^{\frac{1}{2}}, \tag{3.5}$$

$$\gamma_c = (\frac{1}{N} \sum_{i=1}^{N} (p_{ci} - \alpha_c^3))^{\frac{1}{3}}. \tag{3.6}$$

The distance between two sets of moments is defined as

$$d = \sum_{c \in \{H,S,V\}} w_{c,1}|\alpha_k^1 - \alpha_c^2| + w_{c,2}|\beta_c^1 - \beta_c^2| + w_{c,3}|\gamma_c^1 - \gamma_c^2| \tag{3.7}$$

The weights w are currently manually specified.

Color Space: RGB and HSV A color space consists of a color model and a mapping of the model onto an absolute color space. There are 5 major models: CIE, RGB, YUV, HSV, and CMYK. In this thesis, the two most common color spaces are investigated: RGB and HSV. In the future, other color spaces should be investigated.

RGB is an additive color space based on the three primary colors: red, green, and blue. This space can produce any color that can be made from the three primary colors. HSV stands for hue, saturation, and value. It is a transformation of RGB space that

rearranges the geometry of the RGB to be more perceptually relevant in human perception. One advantage of the HSV space is that the impact of lighting changes differs on each dimensions. More specifically, the V channel is especially sensitive to change in lighting [67]. This knowledge can be exploited with the moment representation by setting lower weights for all moments in the V channel.

Histogram of Oriented Gradients

To represent texture in each leg segment, we construct a histogram of gradients (HoG), following the same procedure as in [49]. The image is first converted to gray scale and then two gradient filters are applied to obtain the gradient magnitude and orientation measurement for each pixel in the image. To compute HoG for the k^{th} leg segment on camera j, each pixel i for which $s(i, j, \overrightarrow{C}_k) = 1$ is classified into one of 10 bins according to the orientation value and its corresponding magnitude is added to the bin. The HoG is then normalized so that the sum of all bin values is equal to 1. The distance between two HoGs can be computed by

$$d = \sum_{b=1}^{10} |HoG_b^1 - HoG_b^2| \tag{3.8}$$

where b indicates the orientation bin number.

Under typical walking motions, the HoG of limbs changes according to a cyclic pattern corresponding to gait motion. This pattern is evident from the cyclic distance between the first and subsequent hand labeled limb parts as shown in Figure 3.3. In Figure 3.3, a video sequence is hand labeled by manually adjusting the state vector until its projection fits the observation from 2 cameras. The first N labeled states are selected as templates for which subsequent labeled states are compared with. Ideally, the distance should be close to 0 between any 2 labeled states. The left graph shows a cyclic trend in HoG distance that corresponds to a gait cycle with 1 template. With 30 templates this trend is partially corrected with an overall reduction in distance. The right graph shows that the overall HoG distance decreases as the number of templates increases.

To correct for this cyclic pattern, we include a total of 30 templates as the appearance model, where the 30 templates correspond to a full gait cycle of straight walking. We

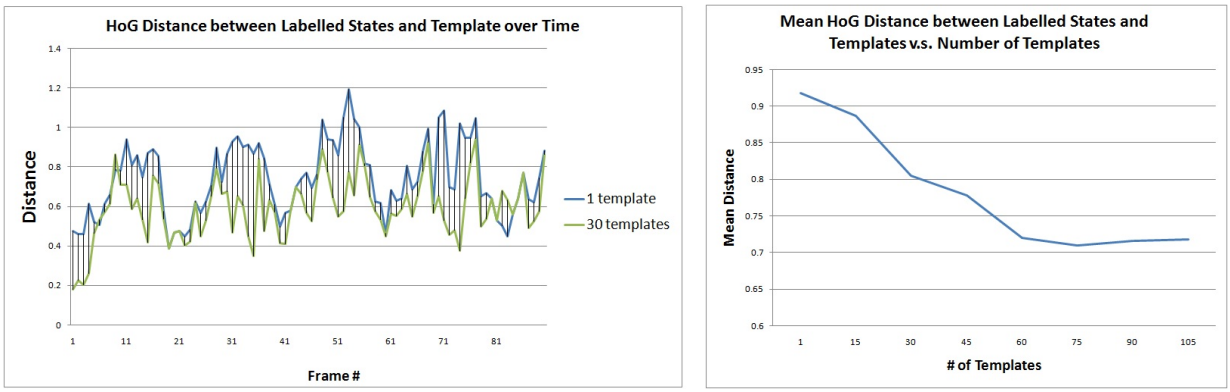

Figure 3.3: HoG distance of labeled ground truths against templates, as a function of number of templates. In the left graph, the cyclic pattern of the distance is partially corrected with 30 templates with lower distance in each frame. In the right graph, the mean distance between labelled ground truth and templates decreases gradually as the number of templates increases.

compute the Euclidean distance between the angles of a state and the angles of each of the 30 templates. The template with the smallest distance is retained to compute the HoG distance according to equation (3.8). One disadvantage of this strategy is that the system may need a lot of templates to cover different types of gait motions.

3.3.2 Depth Image

The depth image from the Kinect is processed as follows. First, a 3D point is associated with each pixel according to camera calibration parameters. Second, the 3D points are classified into foreground and background. Finally, two different distance measures are computed that estimate how close the observed points are with respect to the model cylinder surface.

3D Points Reconstruction

Each image captured 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:

$$z_{i,j} = \frac{1000}{-0.00307 d_{i,j} + 3.33} \tag{3.9}$$

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 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:

$$(x_{i,j}, y_{i,j}, z_{i,j}) = (\frac{z_{i,j}(i - x_0)}{f_x}, \frac{z_{i,j}(j - y_0)}{f_y}, z_{i,j}) \tag{3.10}$$

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.

Background subtraction

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 1500 mm to the front of the camera (Z-axis),

245 mm to the left and 175 mm 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.

Average distance in the 3D space

The first distance measure is based on the 3D distance between the predicted leg model and the foreground 3D points. The distance function aims to favor state hypothesis with foreground points close to the front cylinder surface of each leg segment. 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 3.4. 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-way distance metric. The two directed average distances are computed as follows:

$$d_{FM} = \frac{\sum_{f \in F} |w_{m^*} - ||f - m^*|||}{|F|} \tag{3.11}$$

where $m^* = arg\,min_{m \in M} ||f - m||$

$$d_{MF} = \frac{\sum_{m \in M} |w_m - \min_{f \in F}(||f - m||)|}{|M|} \tag{3.12}$$

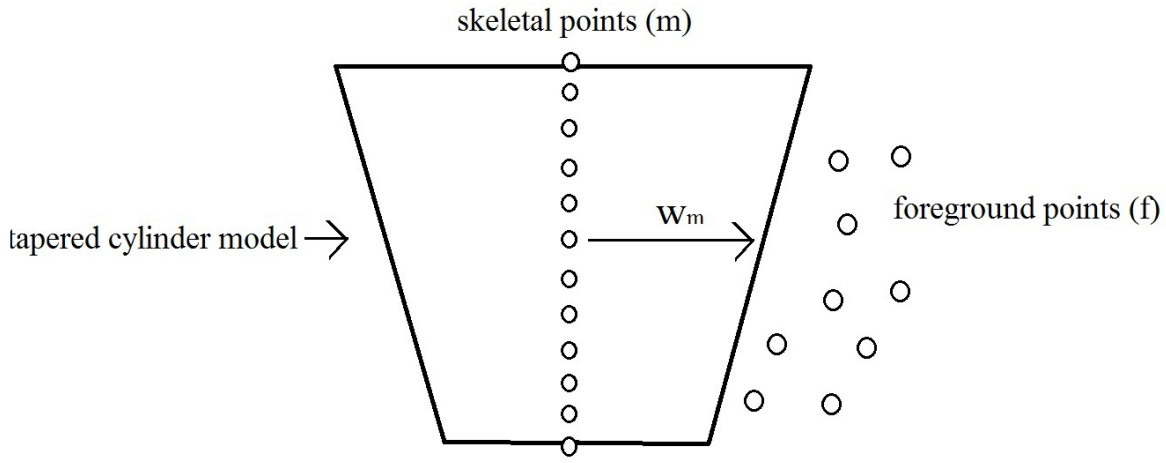

Figure 3.4: The skeleton representation of the cylinders. A set of model points are generated at the center of a particle's cylinder model, where each model point is associated with a different width w_m. The purpose of the observation is to keep the foreground points and the skeletal points at a width distance.

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

$$d_1 = \frac{d_{FM} + d_{MF}}{2} \tag{3.13}$$

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.

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 and Z-buffer rasterization of 3D

cylinders follow the standard pipeline in 3D graphics and is described in Appendix A.2. 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:

$$d_2 = \frac{\sum_{(i,j)} |z^P_{i,j} - z^K_{i,j}|}{(640 \times 480)} \tag{3.14}$$

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 to the 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.

3.4 Likelihood Function

3.4.1 Single RGB Camera

Let J be the set of pixels for which $s(i, j, \overrightarrow{X}_{true}) = 1$, where $\overrightarrow{X}_{true}$ is the manually adjusted state that corresponds to the lower limbs in the first image. For the first 30 images before tracking, the template appearance model is computed for each manually positioned leg segment defined by J. Let K be defined similarly for a predicted state representing $\overrightarrow{X}$ in a later frame. To evaluate the likelihood of a predicted state $\overrightarrow{X}$, the 3D model is projected onto the image plane and its appearance models are extracted. A distance measurement for the appearance model is computed between J and K for each leg segment according to Equations (3.3), (3.7), and (3.8). Given the distance measurements, the likelihood of a particle follows an exponential distribution which assigns weight inversely proportional to the distance.

We combine the likelihood from the appearance models based on color (color moments in the HSV space, HoC in the normalized log RGB space) and HoG together by assuming statistical independence between each image cue o:

$$P(I|\overrightarrow{X}) \propto \prod_{o=1}^{2} P(I_o|\overrightarrow{X}) \tag{3.15}$$

where

$$P(I_o|\overrightarrow{X}) = exp^{(-d_o)} \tag{3.16}$$

Here, d_o represents the distance measurement for image cue o ($o = 1$ for color, $o = 2$ for gradient). While the independence assumption allows easy addition of image cues and camera observations, this assumption is not true in general since the cues are somewhat dependent. In particular, a state hypothesis that results in low distance for one image cue is likely to result in a low distance for a second one, and the same is true for hypotheses that result in a high distance. As a result, when the two likelihood functions assign similar probabilities to the particles, simply multiplying the likelihoods may lead to sharp peaks in the likelihood distribution. Specifically, the different color representations are highly dependent on each other and should not be combined together. In the experiment, the different color representations will be evaluated individually and together with HoG, but at no time will two different likelihood functions based on color be combined together.

3.4.2 Multiple RGB Cameras

Including an additional camera in the particle filtering framework is straightforward assuming independence between them. Equation 3.15 is modified to reflect an additional camera:

$$P(I|\overrightarrow{X}) \propto \prod_{o=1}^{2}\prod_{j=1}^{2} P(I_{o,j}|\overrightarrow{X}) \tag{3.17}$$

where

$$P(I_{o,j}|\overrightarrow{X}) = exp^{(-d_{oj})} \tag{3.18}$$

Here, d_{oj} represents the distance measurement for image cue o with camera j. 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

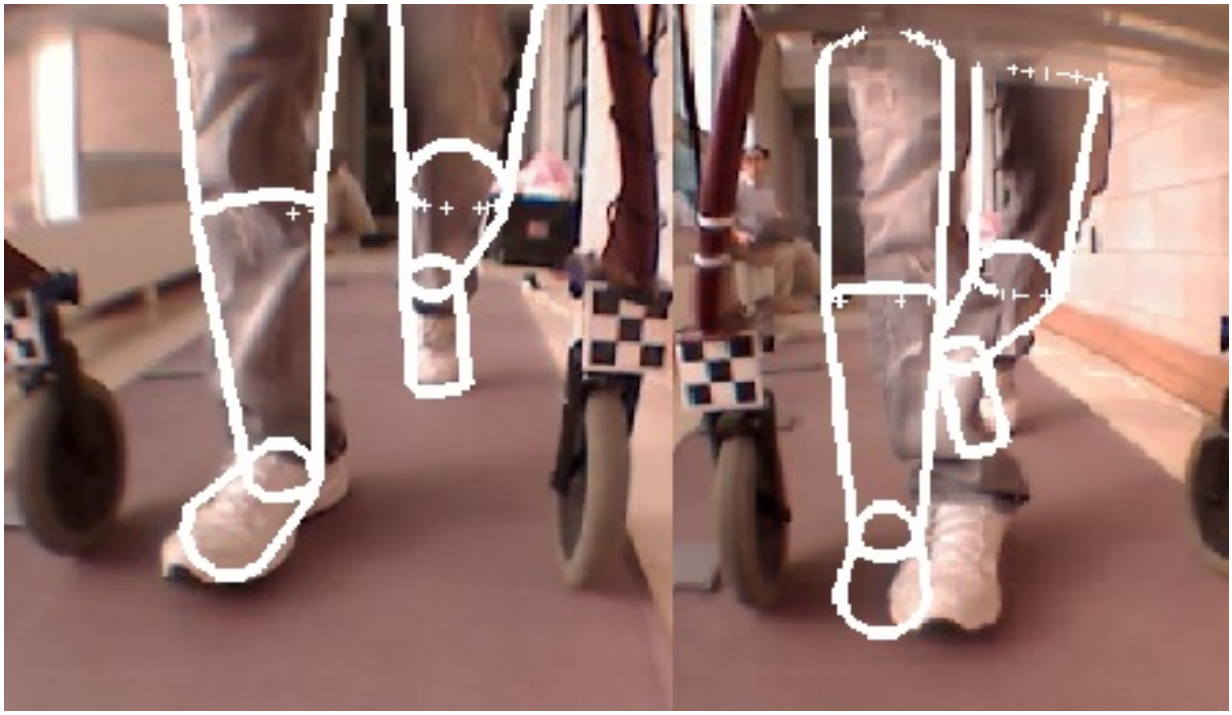

Figure 3.5: Depth error reflected in a binocular setup

in a probabilistic fashion, since states with depth errors will generally not fit both image observations, as shown in Figure 3.5, and therefore will be penalized. Obviously, it will be harder for bad states to falsely fit both image observations if the cameras capture very different or, to an extreme, perpendicular views of a scene. Hence, we maximized the angle between the two views given the physical constraints of the walker.

3.4.3 Single Kinect Camera

Given the two distance measures d_1 and d_2 in Equation 3.13 and 3.14. We define two different likelihood functions, which are combined to produce a single final weight for the particle, assuming independence between them.

$$P(I|\overrightarrow{X}) \propto P(I_1|\overrightarrow{X})P(I_2|\overrightarrow{X}) \tag{3.19}$$

where

$$P(I_1|\overrightarrow{X}) = exp^{(-d_1/5)} \tag{3.20}$$

and

$$P(I_2|\overrightarrow{X}) = exp^{(-d_2/3)} \tag{3.21}$$

The parameters assigned to the exponential functions (i.e. 5 and 3) in Equations 3.20 and 3.21 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.

Although the two likelihood functions are not independent, the formulation of the distance measures are designed to complement each other. For our tracker, the second distance measurement 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 for step length (Z-axis) and step width (X-axis) in Table 4.5 and 4.6 support this reasoning.

Chapter 4

Experiments

4.1 Evaluation of 2D Cues

In this section, we evaluate the effectiveness of each observation model for the RGB image. Five different distance measurements for the observation models are evaluated: histogram of colors in the HSV space (1), histogram of colors in the RGB space (2), moments in the HSV space (3), moments in the RGB space (4), and histogram of oriented gradients (5). The index corresponds to the distance measurements in Tables 4.1, 4.2, and 4.3. Also, the effect of increasing the number of templates is investigated by comparing the result of using one template versus thirty templates. For the evaluation, 100 consecutive frames are labeled by manually locating each leg segment in the images. Section 4.1.1 compares the effectiveness of different distance measurements. Section 4.1.2 further investigates the effect of increasing the number of templates.

4.1.1 Spearman Correlation for F-measure and distance

Since the observation model for the RGB image is entirely based on 2D information, a good distance measurement is one that produces a lower distance for particles that have more overlap between the projected cylinder model and the hand labeled region for the

legs. The Spearman rank correlation [31] is used to evaluate the distance measurements defined in Equations 3.3, 3.7, and 3.8.

The Spearman rank correlation assesses how well the relationship between two variables can be described by a monotonic function. A perfect Spearman correlation of +1 or -1 occurs when each variable is a perfect monotone function of the other. A positive Spearman correlation indicates that as the value of one variable increases, the value of the other variable increases as well, whereas a negative Spearman correlation indicates the opposite. The two variables used for the Spearman correlation are F-measure (defined in the next paragraph) and the distance measurement of the observation model. The intuition is that if the projected particle overlaps well with the labeled region (i.e. a high F-measure value), then a good distance measurement should produce a corresponding low distance value. As a result, the Spearman correlation should be negative and as close as possible to -1.

The F-measure is a mean of precision and recall that measures how well a projected particle overlaps with the true location of the legs in the image. Let $P1_j$ be the set of pixels that corresponds to the projection of the cylinder model of the leg segment j of a particle. Let $P2_j$ be the set of pixels that corresponds to the labeled region corresponding to the leg segment j in the image. The recall, precision, and F-measure are defined as follows:

$$recall = \frac{|(P1_j) \cap (P2_j)|}{|P2_j|} \tag{4.1}$$

$$precision = \frac{|(P1_j) \cap (P2_j)|}{|P1_j|} \tag{4.2}$$

$$F = \frac{2(precision)(recall)}{(precision + recall)} \tag{4.3}$$

The result of the Spearman correlation for three subjects is shown in Table 4.1. Note that the Spearman correlation in each cell corresponds to the mean of the Spearman correlations over the 6 leg segments. 1000 pairs of (F-measure, distance) are generated for the Spearman calculation by randomly generating 1000 particles over 100 labeled frames. The result shows that the distance measurements for gradient and all four representations

Table 4.1: Average Spearman correlation for each observation: HoC in HSV (1), HoC in RGB (2), moments in HSV (3), moments in RGB (4), and HoG (5). The result is shown for 3 subjects, averaged over 100 labelled frames and 1000 randomly generated particles in each frame.

		1 template					30 templates				
Subject	Measure	1	2	3	4	5	1	2	3	4	5
1	Mean	-0.57	-0.58	-0.49	-0.48	-0.56	-0.61	-0.54	-0.45	-0.48	-0.51
	Std	0.24	0.20	0.25	0.25	0.23	0.21	0.18	0.19	0.25	0.20
2	Mean	-0.59	-0.51	-0.60	-0.47	-0.44	-0.60	-0.51	-0.53	-0.48	-0.46
	Std	0.21	0.21	0.23	0.20	0.25	0.19	0.20	0.21	0.21	0.23
3	Mean	-0.54	-0.51	-0.45	-0.44	-0.54	-0.48	-0.49	-0.45	-0.46	-0.47
	Std	0.27	0.23	0.24	0.25	0.21	0.23	0.21	0.22	0.19	0.24
Overall	Mean	-0.57	-0.54	-0.52	-0.46	-0.51	-0.56	-0.51	-0.47	-0.47	-0.48
	Std	0.24	0.21	0.24	0.23	0.23	0.21	0.20	0.21	0.22	0.22

of color are working as expected by having a negative Spearman correlation. However, the difference between the different representations of color is only modest, as the best representation (i.e. HoC in HSV) and the worst (i.e. moments in RGB) representation are still within one standard deviation of each other. Increasing the number of templates to 30 does not lead to consistent improvements in the results. The reason is that this strategy only improves the result for particles that have high F-measure, as increasing the number of templates is designed to produce a lower distance for such particles. Such strategy does not necessarily produce a higher distance for random particles that do not overlap well with labels. The subsequent subsection illustrates this issue.

4.1.2 Number of Templates and distance measurement

This subsection investigates the distance measurements of each observation as an effect of increasing the number of templates. Table 4.2 shows the average distance of 100 ground truth image regions against the initial 1 and 30 templates, averaged over 100 labeled frames and 6 leg segments. The results show that increasing the number of templates consistently

Table 4.2: Average distance of ground truth image region computed against the initial 1 and 30 templates for each observation: HoC in HSV (1), HoC in RGB (2), moments in HSV (3), moments in RGB (4), and HoG (5).

		1 template					30 templates				
Subject	Measure	1	2	3	4	5	1	2	3	4	5
1	Mean	0.49	0.39	1.17	0.26	0.17	0.32	0.32	0.64	0.23	0.15
	Std	0.044	0.054	0.16	0.04	0.03	0.13	0.089	0.3	0.049	0.03
2	Mean	0.31	0.35	0.61	0.26	0.23	0.29	0.29	0.58	0.25	0.18
	Std	0.67	0.077	0.17	0.044	0.04	0.039	0.047	0.19	0.091	0.03
3	Mean	0.62	0.38	0.96	0.31	0.18	0.47	0.31	0.71	0.26	0.16
	Std	0.061	0.05	0.17	0.045	0.035	0.13	0.058	0.21	0.07	0.03
Overall	Mean	0.47	0.38	0.92	0.28	0.19	0.36	0.31	0.64	0.24	0.16
	Std	0.057	0.060	0.17	0.043	0.035	0.10	0.065	0.23	0.070	0.030

reduces the distance of the ground truth for every observation. Table 4.3 shows the average distance of 1000 random particles for the same 100 frames against the initial 1 and 30 templates. Comparing the values between Table 4.2 and Table 4.3 under 1 template, the distance of random particles is consistently higher than the distance of ground truth by more than one standard deviation. By increasing the number of templates to 30, the distance of random particles remains about the same. Since increasing the number of templates reduces the distance of ground truth and has no effect on random particles, this strategy increases the difference of distance between the good particles and bad particles, which creates a better separation between the two sets of particles. As a result, such strategy will improve the tracking results in general.

4.2 GaitRITE Measurement

In the following experiments, we measure step width and step length errors of the mean prediction over 5000 particles against ground truth data obtained with a GaitRITE mat (an array of pressure sensors that measures the spatial location of the feet when they

Table 4.3: Average distance of random particles computed against the initial 1 and 30 templates for each observation: HoC in HSV (1), HoC in RGB (2), moments in HSV (3), moments in RGB (4), and HoG (5).

		1 template					30 template				
Subject	Measure	1	2	3	4	5	1	2	3	4	5
1	Mean	0.60	0.63	1.69	0.98	0.30	0.56	0.63	1.30	0.98	0.31
	Std	0.16	0.19	0.73	0.59	0.12	0.18	0.21	0.65	0.58	0.12
2	Mean	0.48	0.55	1.20	0.96	0.30	0.49	0.57	1.22	0.95	0.31
	Std	0.16	0.22	0.63	0.64	0.12	0.16	0.21	0.64	0.64	0.13
3	Mean	0.71	0.63	1.55	0.98	0.30	0.67	0.61	1.49	0.86	0.31
	Std	0.11	0.18	0.50	0.45	0.11	0.14	0.19	0.56	0.46	0.13
Overall	Mean	0.60	0.60	1.48	0.97	0.30	0.60	0.60	1.33	0.93	0.31
	Std	0.14	0.20	0.62	0.56	0.12	0.16	0.20	0.62	0.56	0.13

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 [28]. From a clinical perspective, physical therapists routinely use step length and width to assess gait recovery. 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 4.4.

The mean and standard deviation of the errors in step length and step width are summarized in Tables 4.5 and 4.6 for different camera setups. For the Kinect, the results

Table 4.4: 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

Table 4.5: Mean and standard deviation of step length error in cm. Cue 1, Cue 2, and Combine correspond to the Kinect likelihoods defined in Equations 3.20, 3.21, and 3.19, respectively. Binocular is the binocular RGB camera setup. RGB 1 and RGB 2 are the monocular RGB camera setups.

Subject	Measure	Cue 1	Cue 2	Combine	Binocular	RGB 1	RGB 2
1	Mean	4.67	7.11	2.73	21.30	46.12	21.49
	Std	1.66	3.11	2.31	9.95	17.06	11.25
2	Mean	3.88	15.46	3.87	10.66	24.45	25.27
	Std	4.69	6.98	2.27	8.67	10.75	13.16
3	Mean	4.60	4.77	3.48	16.84	17.83	17.33
	Std	2.28	4.23	2.02	10.36	9.92	13.29
Overall	Mean	4.38	9.13	3.36	16.27	29.57	21.36
	Std	2.88	4.77	2.20	9.66	12.58	12.57

suggest that the distance metric computed in 3D space (cue 1, as defined in 3.13) has lower error in step length, while the one computed in 2D image space by projection (cue 2, as defined in 3.14) has lower error in step width. More importantly, the result of the combined function is generally close to, and in 5 out of 6 cases better than, the best results of the two separate cues for each subject.

Note that the RGB setups have significantly higher error than the Kinect setup for both measures. 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

Table 4.6: Mean and standard deviation of step width error in cm. Cue 1, Cue 2, and Combine correspond to the Kinect likelihoods defined in Equations 3.20, 3.21, and 3.19, respectively. Binocular is the binocular RGB camera setup. RGB 1 and RGB 2 are the monocular RGB camera setups.

Subject	Measure	Cue 1	Cue 2	Combine	Binocular	RGB 1	RGB 2
1	Mean	8.71	2.37	2.87	6.51	11.03	3.53
	Std	1.07	1.91	2.50	4.55	3.44	2.48
2	Mean	7.00	3.45	3.04	6.28	3.68	4.83
	Std	2.01	2.43	2.52	6.00	2.88	2.99
3	Mean	4.56	2.36	1.75	6.76	10.73	9.12
	Std	2.48	1.22	1.17	4.44	7.50	7.46
Overall	Mean	6.76	2.73	2.55	6.52	8.48	5.83
	Std	1.85	1.84	2.06	5.00	4.61	4.31

motions in uncontrolled environments change the gradient and color information of the tracking targets in comparison to the reference templates, even though the effect is partially corrected by increasing the number of templates and using the best color representation. These two factors significantly contribute to the poor results for the RGB cameras. Finally, the tracking results with 2 cameras are consistently better than the tracking results with 1 camera.

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 [53] 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 4.5 and 4.6, the error of our tracker with the combined likelihood for the depth sensor is slightly larger than the reported variability in their study.

The tracker with the depth sensor successfully tracks the legs over the entire sequence for each of the 3 subjects. As demonstrated in Figure 4.1, tracking is successful even when there is significant missing data, shown as black patches in the color-coded depth image. In frames 4 to 6 when a significant portion of the points are missing, the prediction of the feet

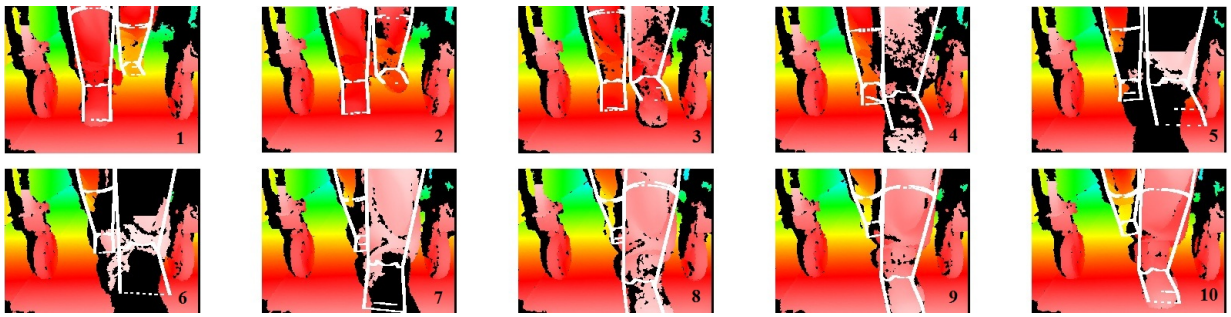

Figure 4.1: The problem of missing data in tracking with a depth sensor. Although tracking of the right foot temporaily fails in frames 4 to 6 when this problem becomes severe, recovery immediately follows when the points become visible again starting in frame 7.

goes off-track slightly. However, tracking recovers quickly when the points are observable again in the last 4 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 4.2, the tracker is able to infer the location of the partially occluded leg segments in the first 4 frames. In the next 5 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 that 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 4.2 and 4.1 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 the tracker with the depth sensor is robust against moderate missing data, partial occlusion, and deformable tracking targets.

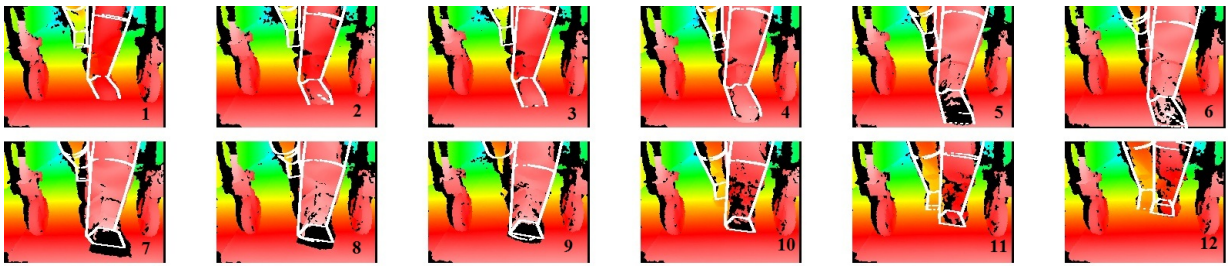

Figure 4.2: The problem of occlusion in tracking with a depth sensor. In frames 5 to 9, the left foot is totally occluded by the right leg, and the tracker mistakenly predicts the foot is in the air instead of on the ground. Recovery immediately follows when the left foot becomes visible again starting in frame 10.

4.3 Running Time

The software is implemented in Matlab, where a portion of the code is written in C++. This C++ code is called from Matlab through mex files. The current running time to process one frame with 5000 particles, ranges from 14 to 19 seconds for the single Kinect observation, and ranges from 13 to 16 seconds for the single RGB observation. Binocular RGB setup approximately doubles the computation time of the single RGB observation. Computation is done on a modern 2.5 Ghz computer, with parallelized computation of particles over 2 cores using Matlab's parallelization facility. A significant speed-up could be achieved by: parallel computation of particles in C++ instead of Matlab, rasterization through GPU OpenGL instead of the 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.

At the moment, off-line computation is sufficient because the video would be processed after its capture by kinesiologists. In the future, it would be desirable to estimate pose in real-time for real-time feedback or navigation assistance.

4.4 Summary

In the experiments, three different camera setups are evaluated. The Kinect setup produces the best result overall, but may temporarily break down when occlusion or missing observation become severe. To solve this issue, a better dynamic model is needed to place particles at better locations apriori. The binocular setup is able to partially correct for depth error, while the monocular setup produces the worst results overall. Strategies to improve RGB observations are also evaluated. The result suggests that different representations of color seem to make only a small difference, while increasing the number of templates consistently improves the results for all observations.

Chapter 5

Conclusion

In this thesis, three different camera systems (monocular RGB, binocular RGB, single depth sensor) are investigated for 3D limb tracking under a moving platform. All setups follow a particle filtering framework with a constant velocity dynamic model, and their tracking accuracy is examined. The experiments show that the tracker with the depth sensor successfully tracks the 3D poses of users over the entire video sequence and produces fairly accurate results as measured by the GAITRite mat. Although the tracker temporarily fails when there is significant amount of occlusion or missing data, it is able to recover immediately from such unfavorable conditions when the observation improves. On the other hand, the trackers with the RGB cameras produce results with higher error. Strategies for improving the RGB observations are also investigated. The results show that there is a gradual improvement by increasing the number of templates, and a modest difference between the different representations of color.

The system with the depth sensor 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.

5.1 Future Work

5.1.1 Single RGB Camera

There are a number of possible extensions for tracking with a single RGB camera. With the lack of depth information, the dynamic model becomes more important to place the particles at the right location in the first place before evaluating the likelihood. The motion model can be improved by using physics-based models that better respect the laws of physics and therefore produce gaits that better resemble human motion [13, 12]. Another direction is to learn the dynamics from data on a lower dimensional space even if the pose space is high dimensional [69].

As the tracking problem involves a moving camera with dynamic scene, it is important to handle the fact that the background and foreground appearance are constantly changing. Background subtraction for a dynamic background is difficult but not impossible and has been investigated by a number of researchers. In addition, there is a need for more robust methods to handle target appearance changes. Wavelets [34], which are robust against illumination and amplitude changes, may be a feasible appearance model in addition to the color and gradient information used in this thesis. Other popular tracking features include object contour [21, 16], symmetry, texture, edges, etc. Learning a dynamic appearance model over time [34] may further improve the robustness of tracking as the appearance of the target changes over time.

5.1.2 Multiple RGB Cameras

Once background subtraction is possible, shape-from-silhouettes with multiple cameras can be implemented to resolve depth ambiguity. One practical concern is that the accuracy of shape-from-silhouettes depends on the number of cameras used and the relative angle of each camera. Some engineering of the walker may be needed to properly install multiple cameras. For example, the base size of the walker needs to be extended to accomodate multiple cameras placed in different angles.

5.1.3 Depth Camera

For the depth camera Kinect, an alternative approach to the top-down particle filtering method is the point classification approach that classifies each foreground pixel according to the body part that it belongs to. Once classification is done, locating the joint position and recovering the state vector can easily be done. One way to classify each pixel is to treat this problem as a classification problem in machine learning. For example, Shotton [61] learns a decision forest and Anguelov [6] learns an MRF from data.

Another alternative approach is to use Kalman filtering instead of particle filtering. While Kalman filtering is able to represent only a Gaussian distribution, it may be suitable for tracking with the Kinect as there is no depth ambiguity with the data returned by the camera. The main advantage of Kalman filtering over particle filtering is speed, since the underlying distribution is modeled by a Gaussian instead of a number of particles.

The appearance model used in this thesis consists of cylinders. This is a rough estimation of limb shape, and is thus not accurate when the subject wears baggy pants or dresses that deforms over time due to folding of cloth. A data driven approach that builds a 3D mesh model of the legs before tracking will be more accurate than fixed cylinders.

5.1.4 Combining Cameras

Combining both RGB and depth cameras for tracking is an interesting avenue for future work. Tracking with either camera provides at most 3 pieces of information in each pixel (i.e. RGB values for RGB camera, XYZ coordinates for depth camera). Tracking with both cameras thus provide information in 6D, and the added information can potentially improve tracking accuracy.

Finally, it is demonstrated in the thesis that the depth camera outperforms RGB cameras under room lighting and minor sunlight. However, the signal of the Kinect will be washed out under direct sunlight. In that case, it is desirable to continue tracking with the RGB cameras. Similarly, the signal of RGB cameras is unreliable in a dark environment, and it is desirable to continue tracking with the Kinect under such condition. Hence, there

is a need for a framework that detects lighting conditions and automatically switches between the cameras or combines their signals in a non-detrimental fashion when tracking in a dynamic environment.

Appendix A

APPENDICES

A.1 Calibration of Cameras

The cameras are calibrated according to the calibration software [2] to obtain the intrinsic parameters (i.e. focal length and camera center). First, a number of photos need to be taken with a checkerboard placed in different positions visible to the camera. These photos are fed into the software to extract grid corners for each photo. Based on the grid corners of each photo and the known grid size, a non-linear optimization procedure using gradient descent is used to obtain the intrinsic parameters by minimizing the total reprojection error over all the parameters.

To obtain the extrinsic parameters (i.e. translation and rotation matrix) between two cameras, the same software is used. The intrinsic parameters of each camera are first obtained according to the previous paragraph. Based on the intrinsic parameters and the grid corners of the images in both cameras, The extrinsic parameters are computed by the same optimization procedure to minimize the reprojection errors on both cameras for all calibration grid locations. Cameras are calibrated once at the beginning of experiments, assuming the cameras are stably installed and the parameters remain the same throughout the experiments.

A.2 3D Model Rasterization

This section describes how a 3D model of the leg is rasterized in a 2D image. First, the 3D cylinder is represented using planar rectangular polygons that circumscribe the surface of the 3D model. Polygons that are not visible to the camera are detected using the backface culling algorithm and are removed to avoid further processing. Afterwards, each 3D end point of the polygons associated with pixel (i, j), denoted as $[x_{i,j}, y_{i,j}, z_{i,j}]$, is projected onto the image plane according to the following formula:

$$[i, j] = [\frac{(f_x)(x_{i,j})}{z_{i,j}} + x_0, \frac{(f_y)(y_{i,j})}{z_{i,j}} + y_0] \tag{A.1}$$

Here, f_x and f_y are the focal length, x_0 and y_0 are the camera centers, in the x and y axis respectively. The projected polygons represented by 2D end points are then clipped at the boundary of the image plane, triangulated, and rasterized using the Z-buffer scan conversion algorithm, which computes the depth value along with each rasterized pixel. The algorithms for backface culling, polygon clipping, polygon triangulation, and Z-buffer scan conversion are described in the book [26].

A.3 Cylinder Intersection Detection

This section describes the algorithm for detecting the intersection between two cylinders in 3D space. Since a 3D cylinder is represented using planar rectangular polygons, we can easily extract the faces and edges of the cylinder. The idea is that if there is at least one intersection between an edge in one cylinder model and a face in the other cylinder model, then there is an intersection between the two cylinders. As a result, the algorithm searches for an edge-face intersection over all the edges and faces of both cylinders and reports positive when such incidence is detected.

The edge-face intersection detection procedure goes as follows. First, a significant speed up can be achieved by using a trivial test that verifies whether the two points defined by the edge lie on different half spaces divided by the plane defined by the face. If both points

lie on the same half space, then the edge does not intersect the face and we can avoid further processing. In practice, this is the case for the majority of edge-face pairs. On the other hand, if the two points lie on different half space, the intersection point between the plane and the edge is computed [8]. Finally, we test whether the intersection point is enclosed by the face using the crossing number algorithm [39]. If the point is enclosed by the face, then the edge intersects the face.

References

[1] Open kinect. http://openkinect.org/wiki. 33

[2] Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib doc/, 2001. 54

[3] Kinect from Microsoft. http://www.xbox.com/en-US/kinect, 2011. 5

[4] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. In *IEEE Transactions on Pattern Analysis and Machine*, pages 44–58, 2006. 9, 10

[5] M. Alwan, A. Ledoux, G. Wasson, P. Sheth, and C. Huang. Basic walker-assisted gait characteristics derived from forces and moments exerted on the walkers handles: Results on normal subjects. *Medical Engineering and Physics*, 29:380–389, 2007. 2

[6] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and AY Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 169–176, 2005. 20, 52

[7] V. Athitsos and S. Sclaroff. An appearance-based framework for 3D hand shape classification and camera viewpoint estimation. In *International Conference on Automatic Face and Gesture Recognition*, pages 45–50, 2002. 12

[8] D. Badouel. *Graphics Gems*. Academic Press Inc., 1990. 56

[9] A. Balan, L. Sigal, M. J. Black, J. Davis, and H. Haussecker. Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 8

[10] D. Beymer and K. Konolige. A real-time tracking of multiple people using stereo. In *IEEE Frame Rate Workshop*, 1999. 18

[11] R. Brand. Shadow puppetry. In *International Conference on Computer Vision (ICCV)*, pages 1237–1244, 1999. 10

[12] M.A. Brubaker and D.J. Fleet. The kneed walker for human pose tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 15, 51

[13] M.A. Brubaker, D.J. Fleet, and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. In *International Journal of Computer Vision (IJCV)*, pages 140–155, 2010. 6, 15, 51

[14] H. Chen, P. Belhumeour, and D. Jacobs. In search of illumination invariants. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on TOF Computer Vision*, volume 1, pages 254–261, 2000. 28

[15] L. Chen and N. D. Georganas. An efficient and robust algorithm for 3D mesh segmentation. In *Multimedia Tools Application*, pages 109–125, 2006. 20

[16] Yi-Ru Chen, Cheng-Ming Huang, and Li-Chen Fu. Upper body tracking for human-machine interaction with a moving camera. In *Intelligent Robots and Systems (IROS09)*, pages 1917 –1922, October 2009. 51

[17] G. K. M. Cheung, T. Kanade, J. Bouquet, and M. Holler. A real-time system for robust 3D voxel reconstruction of human motions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 714–720, 2000. 17

[18] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–84, 2003. 17

[19] K. Choo and D.J. Fleet. People tracking using hybrid Monte Carlo filtering. In *International Conference on Computer Vision (ICCV)*, pages 321–328, 2001. 25

[20] I. Cohen and G Mdioni. Detection and tracking of objects in airborne video imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Interpretation of Visual Motion*, 1998. 6

[21] Larry Davis, Vasanth Philomin, and Ramani Duraiswami. Tracking humans from a moving platform. In *International Conference on Pattern Recognition (ICPR)*, pages 4171–4178, 2000. 51

[22] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2133, 2000. 25

[23] M. Dimitrijevic, V. Lepetit, and P. Fua. Human body pose detection using Bayesian spatio-temporal templates. In *Computer Vision and Image Understanding*, pages 127–139, 2006. 12

[24] E.Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *International Conference on Computer Vision (ICCV)*, pages 67–74, 2003. 7

[25] A. Elgammal and C. S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 681–688, 2004. 10

[26] J.D. Foley, A. Van Dam, S.K. Fiener, J.F. Hughes, and R.L. Phillips. *Introduction to Computer Graphics*. Addison-Wesley, 1994. 55

[27] 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. 19

[28] A. Gabell and U. Nayak. The effect of age on variability in gait. In *Journal of Gerontology*, volume 1, pages 662–666, 1984. 44

[29] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time motion capture using a single time-of-flight camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 755–762, 2010. 22

[30] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. In *German Association for Pattern Recognition (DAGM)*, pages 285–292, 2005. 19

[31] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics, 5th ed*, page 338. Macmillan. 41

[32] N.R. Howe. Silhouette lookup for automatic pose tracking. In *Workshop on Articulated and Non-Rigid Motion*, page 15, 2004. 10, 12

[33] M. Isard and A. Blake. CONDENSATION conditional density propagation for visual tracking. In *International Journal of Computer Vision (IJCV)*, pages 5–28, 1998. 25

[34] A.D. Jepson, D.J. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 10, pages 1296–1311, 2003. 8, 51

[35] B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, 2004. 6, 7

[36] R. Kehl, M. Bray, and L. VanGool. Full body tracking from multiple views using stochastic sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 129–136, 2005. 17

[37] 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)*, pages 74–79, 2005. 19

[38] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision (ICCV)*, pages 508–515, 2001. 18

[39] D. Lee and F. Preparata. Computational geometry: a survey. In *IEEE Transactions on Computers*, 1984. 56

[40] M. Lee and R. Nevatia. Human pose tracking using multi-level structured models. In *European Conference on Computer Vision (ECCV)*, pages 368–381, 2006. 13, 14

[41] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. In *IEEE Transactions on Pattern Analysis and Machine*, pages 1465–1479, 2006. 8, 21

[42] J. Malik, S. Belongie, T. Leung, and J.Shi. Contour and texture analysis for image segmentation. In *International Journal of Computer Vision (IJCV)*, pages 7–27, 2001. 13

[43] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. In *International Journal of Computer Vision (IJCV)*, pages 199–223, 2003. 22

[44] I. Mikic, M. Triverdi, E. Hunter, and P. Cosman. Articulated body posture estimation from multicamera voxel data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 455–460, 2001. 17

[45] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras. A target dependent colorspace for robust tracking. In *18th International Conference on Pattern Recognition*, pages 43–46, 2006. 8

[46] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision (ECCV)*, pages 666–680, 2002. 11

[47] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–333, 2004. 9, 13

[48] 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. 19

[49] S. Ng, A. Fakih, A. Fourney, P. Poupart, and J. Zelek. Towards a mobility diagnostic tool: Tracking rollator users leg pose with a monocular vision system. In *International Conference of IEEE Engineering in Medicine and Biology Society (EMBC)*, volume 1, pages 662–666, 2009. 23, 28, 31

[50] N.Jojic, M.Turk, and T.Huang. Tracking self-occluding articulated objects in dense disparity maps. In *International Conference on Computer Vision (ICCV)*, pages 123–130, 1999. 22

[51] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. In *Image and Vision Computing*, pages 99–110, 2003. 7

[52] Eng-Jon Ong, Antonio S, Micilotta, Richard Bowden, and Adrian Hilton. Viewpoint invariant exemplar-based 3D human tracking. In *Computer Vision and Image Understanding*, pages 178–189, 2006. 12

[53] T.M. Owings and M.D. Grabiner. Variability of step kinematics in young and older adults. *Gait Posture*, 9:20–26, 2004. 46

[54] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man, Cybernetics*, pages 3099–3104, 2004. 6

[55] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 3108–3113, 2010. 20, 22

[56] R. Rosales, V. Athitsos, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *International Conference on Computer Vision (ICCV)*, pages 378–387, 2001. 10

[57] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *Workshop on Human Motion*, pages 19–24, 2000. 10

[58] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *International Journal of Computer Vision (IJCV)*, pages 7–42, 2002. 18

[59] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *International Conference on Computer Vision (ICCV)*, pages 750–759, 2003. 9, 11

[60] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *International Conference on Computer Vision (ICCV)*, pages 1219–1225, 2009. 7

[61] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 21, 52

[62] M. Siddiqui and G. Medioni. Human pose estimation from a single view point, real-time range sensor. In *CVCG at IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 10, 22

[63] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 390–397, 2005. 12

[64] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular human tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 69–76, 2003. 9, 14

[65] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *International Conference on Computer Vision (ICCV)*, pages 1063–1070, 2003. 14

[66] H. Stern and B. Efros. Adaptive color space switching for face tracking in multi-colored lighting environments. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 249–254, 2002. 8

[67] M. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval of Image and Video Databases III*, volume 2, pages 381–392, 1995. 30, 31

[68] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *Computer Vision and Image Understanding*, pages 1677–1684, 2000. 12

[69] R. Urtasun, D.J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 238–245, 2006. 15, 51

[70] R. van der Merwe, A. Doucet, J. F. G. de Freitas, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems (NIPS)*, pages 584–590, 2000. 25

[71] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. In *Computer Vision and Image Understanding (CVIU)*, pages 174–192, 1999. 25

[72] J. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *Neural Information Processing Systems (NIPS)*, 2005. 15, 16

[73] H. Zhou and H. Hu. Human motion tracking for rehabilitation: a survey. In *Biomedical Signal Processing Control*, 2007. 1

[74] Y. Zhu, B. Dariush, and K. Fujimura. Controlled human pose estimation from depth image streams. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on TOF Computer Vision*, pages 1–8, 2008. 19