Model-Based Human Pose Estimation with Spatio-Temporal Inferencing

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Youding Zhu, M.S.

* * * * *

The Ohio State University

2009

Dissertation Committee:

Professor Richard Parent, Adviser

Professor James W. Davis

Professor Raghu Machiraju

Dr. Kikuo Fujimura

Approved by

Adviser

Graduate Program in
Computer Science and
Engineering

ABSTRACT

This thesis presents a computational framework for human pose estimation from depth video sequences. The framework has a potential to achieve interesting applications such as robot motion retargeting, activity recognition, etc, wherever joint motion is an appropriate representation of the human motion. On the one hand, feature points that are informative for pose estimation are tracked with depth image analysis. Human poses are reconstructed from these feature points with kinematic constraints including joint limits and self-collision avoidance. On the other hand, human poses could be estimated based on local optimization using dense correspondences between 3D data and the articulated human model. Both could be unified with temporal motion prediction based on Bayesian information integration. We demonstrate our results for humanoid robot motion learning through a novel collision-free retargeting as well as for an example of the human pose estimation with environmental clutters. We show the computational results on a set of challenging motions where limbs interact with each other.

This work is dedicated to my parents, to my sister, and to my wife

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Richard Parent, for his support and encouragement. His insightful and constructive comments greatly improve the presentation and content of this dissertation. I could not have completed this dissertation without his technical and editorial advice.

I would like to thank my dissertation committee members, Prof. James W. Davis, and Prof. Raghu Machiraju for their valuable comments. I would like to thank Prof. Richard S. Denning for his valuable comments during the oral defense. I am thankful that in the midst of all their activity, they accepted to be members of the oral exam committee.

I would like to thank Dr. Kikuo Fujimura for his caring, patience, and providing me with financial support of my Ph.D study and dissertation research. His excellent guidance of my research is extremely appreciated.

I am indebted to many colleagues at Honda Research Institute USA for providing a stimulating environment during my internship study there. I would like to thank Dr. Victor Ng-Thow-Hing for his sharing of valuable experience about motion capture system. I would like to thank Dr. Jongwoo Lim for his sharing of camera calibration software tools. My special thanks go to Dr. Behzad Dariush for his advice, supervision, inspiring discussion and crucial contribution on this dissertation research.

Many thanks to friends and colleagues at OSU with whom I have had a wonderful time, Changshan Wu, Fang Ren, Jue Wang, Xia Liu, Lijie Xu, Ben Liu, and Qianglin Cai.

Lastly, I would like to thank my parents and elder sister. They were always supporting me and encouraging me with their best wishes. I would like to thank my wife, Zhusheng. She was always there and stood by me through the good and hard times.

VITA

1997	B.E., Survey Engineering, Tongji University, Shanghai, China
2000	M.E., GIS and Photogrammetry, Tongji University, Shanghai, China
Sept,2000-Apr, 2002	Graduate Research Associate,Center for Mapping, The Ohio State University.
July,2006-Sept, 2006	Research Intern,Honda Research Institute, Mountain View, CA.
June,2007-Sept, 2006	Research Intern,Honda Research Institute, Mountain View, CA.
Apr,2002- present	Graduate Research Associate,Dept. of Computer Science and Engineering, The Ohio State University.

PUBLICATIONS

Research Publications

Youding Zhu, Behzad Dariush, Kikuo Fujimura "Controlled human pose estimation from depth image streams". *IEEE Time-of-Flight Camera based Computer Vision Workshop*, 2008.

Behzad Dariush, Michael Gienger, Arjun Arumbakkam, Christian Goerick, Youding Zhu, Kikuo Fujimura "Online and markerless motion retargeting with kinematic constraints". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

Youding Zhu, Kikuo Fujimura "Constrained Optimization for Human Pose Estimation from Depth Sequences". *Asian Conference on Computer Vision*, 2007.

Youding Zhu, Kikuo Fujimura, Victor Ng-Thow-Hing "Estimating Pose Sequences from Depth Image Streams". *IEEE Humanoid*, 2005.

Youding Zhu, Kikuo Fujimura "Head Pose Estimation for Driver Monitoring", *IEEE Intelligent Vehicles Symposium*, 2004.

Youding Zhu, Kikuo Fujimura "3D Head Pose Estimation with Optical Flow and Depth Constraints", *Fourth International Conference on 3-D Digital Imaging and Modeling*, 211-216, 2003.

Youding Zhu, Kikuo Fujimura "Driver Face Tracking Using Gaussian Mixture Model (GMM)", *IEEE Intelligent Vehicles Symposium*, 2003.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in:

Computer Graphics	Prof. Richard Parent
AI/Computer Vision	Prof. James W. Davis
Digital Signal Processing	Dr. Kikuo Fujimura

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Markerless estimation of 3D human pose from visual observations is one of the most challenging problems in Computer Vision because of the complexity of the models which relate observation with pose. An effective solution to this problem has many applications in areas such as video coding, visual surveillance, human gesture recognition, biomechanics, video indexing and retrieval, character animation, and man-machine interaction [31, 91, 50].

One of the major difficulties in estimating 3D pose from visual input involves the recovery of the large number of degrees of freedom in movements which are often subject to kinematic constraints such as joint limit avoidance, and self penetration avoidance between two body segments. Such difficulties are compounded with insufficient temporal or spatial resolution, ambiguities in the projection of human motion onto the image plane, and when a certain configuration creates self occlusions. Other challenges include the effects of varying illumination and therefore appearance, variations of appearance due to the subject's attire, required camera configuration, and real time performance for certain applications.

There are two main approaches in estimating 3D human pose, categorized as model-based approaches and learning-based approaches. Model-based approaches rely on an explicitly known parametric human model, and recover pose either by inverting the kinematics from known image feature points on each body segment [6, 87], or by searching high dimensional configuration spaces which is typically formulated deterministically as a nonlinear optimization problem [62], or probabilistically as a maximum likelihood problem [73]. Methods based on optimization typically suffer from local minima and require good initialization. When an image sequence is available, temporal information is often used to track the human pose from a known initialization and an approximate dynamical model.

In contrast, learning-based approaches directly estimate body pose from observable image quantities and do not require initialization and an accurate 3D model [2, 52]. In example-based learning, inferring pose is typically formulated as a k-nearest neighbors search problem where the input is matched to a database of training examples whose 3D pose is known. Computational complexity of performing similarity search in high dimensional spaces and on very large data sets has limited the applicability of these approaches. Although faster approximate similarity search algorithms have been developed based on Locally-Sensitive Hashing [70], computation speed remains a challenge with learning-based approaches.

The sensing modality used in existing markerless methods may be classified into three categories: single passive camera systems, multiple passive camera systems, and time-of-flight (TOF) depth camera systems. The majority of previous work use a single passive camera based on either a model-based approach [73, 74, 79, 46, 76, 75] or a learning-based approach [47, 35, 8, 66, 67, 4, 51, 49, 70, 2, 77, 78]. Although

the notion of using a single passive camera to reconstruct pose is very attractive, the problem is very challenging and perhaps intractable for realistic applications. The depth information provided by multiple camera systems, including stereo-based systems, has been effective in making the problem more manageable [32, 63, 23, 86, 33, 43, 97]. Range data provides a valuable cue in resolving the depth ambiguity problem that exists when only a single passive camera is used. Nevertheless, multi-camera systems have their own limitations including calibration and portability if the setup involves multiple cameras mounted in a controlled environment. In stereo based methods, the depth resolution is often poor, particularly in regions with little texture.

The recently introduced time-of-flight based imaging devices have captured the attention of researchers and have the potential to address some of the aforementioned limitations with single and multi-camera passive systems [96, 43]. For example, the experimental data used in this thesis are captured by CSEM 3000 depth camera [83], which is based on a phase-measuring time-of-flight (TOF) principle [28]. LED source emits near-infrared (NIR) light intensity-modulated with a few tens of MHz. The emitted light is reflected by the scene objects and imaged by 3D-sensor. By measuring the phase delay, the distances from each pixel to scene object can be determined. A depth map is obtained with a resolution of 177(width)x144(height) and a field of view of 43.6° x 34.6°. An advantage of TOF cameras over stereo camera with a short baseline configuration is its relatively good depth resolution. For each pixel, the CSEM camera has an accuracy of higher than 1cm within the distance measurement range of 7 meters, while stereo camera often has a poor depth resolution for the regions void of texture. The TOF camera further provides the advantage of mobility

comparing with multiple camera configuration or stereo camera with a wide baseline configuration. As a result, a humanoid robot equipped with TOF camera can roam from room to room and capture 3D scene flexibly, while multiple camera configuration or stereo camera with a wide baseline configuration lacks such mobility. In addition, the nature of images captured by such devices facilitates segmentation of the human from the background clutter. Since this technology is relatively new, the use of TOF cameras for 3D reconstruction of human pose has only recently been explored [96, 43].

1.2 An Application of Real-Time Human Pose Estimation

Transferring motion from a human demonstrator to a humanoid robot is an important step toward developing robots that are easily programmable and that can replicate or learn from observed human motion [69, 57]. The so-called motion retargeting problem has been well studied and several off-line solutions exist based on optimization approaches that rely on pre-recorded human motion data collected from a marker-based motion capture system. The problem is often formulated and solved as a constrained non-linear optimization problem, where the objective is to minimize the error between the human motion and the target motion, subject to the kinematic constraints [88, 56]. Such approaches are often performed off-line, in static environments, using pre-recorded human motion obtained from marker based motion capture systems [59, 68].

In many human-robot interaction applications, the requirements of interactivity in dynamically changing environments as well as simplicity in sensing and instrumentation, have placed stringent demands on the retargeting procedure. One such

application requiring online and interactive performance involves imitative social interaction of robots with children with learning disabilities, such as autism [64]. These requirements include capturing human motions unobtrusively, without instrumenting them with markers. Also, human motion capture with multiple cameras in a special environment may neither be feasible nor practical. The imaging modality to reconstruct the human performer's pose must be simple and the underlying retargeting mechanism must cope with this.

For example, we recently performed online experiments to control the motion of the Honda humanoid robot ASIMO based on human gestures. The pipeline is illustrated in Figure 1.1. Such an interactive system has many applications, including whole body remote operation of ASIMO from observed human motion.

In this application, we use the CSEM depth camera to capture the human motion in real time. the pose estimation software localizes a set of key-points on the human body. These key-points are scaled to ASIMO's dimensions and commanded to the robot's motion control system. For this application, human pose estimation is an indispensable module for the interactive robot motion learning purpose. A successful human pose estimation will thus enable a robot to observe the human motion, and learn the human motion.

1.3 Challenges: Markerless Human Motion Capture from Image Streams

The general problem of human motion analysis involves sensing, digitizing, and recording the motion of objects in motion [58]. Many existing techniques are based on motion capture systems, which accurately record the 3D position of markers attached to the body. While such systems produce high quality human motions, they

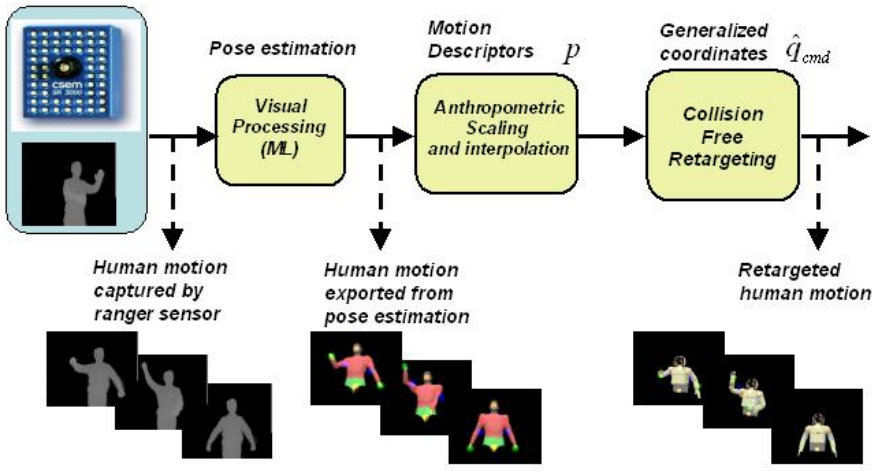


Figure 1.1: Motion Learning for ASIMO robot.

have several limitations. First, the process of calibrating cameras, instrumenting a human with markers, and analyzing the recorded motion is complex, time-consuming, and requires expertise from an operator who has been trained to use the system. In addition, for many applications, markers are obtrusive and may hinder the range of activities that can be reliably performed. Furthermore, such systems are not easily portable. Experiments must be confined to a laboratory equipped with the motion capture system. The aforementioned limitations have restricted the use of such system for applications which do not require interactive or real-time human motion processing. The need for a portable, easy to operate, and markerless system has prompted researchers in computer vision to actively pursue new human motion analysis and 3D pose estimation algorithms based on simple and markerless systems.

In this study, we propose to address the problem of interactive 3D human pose estimation from image streams, particularly depth image streams, without using markers. The reconstructed pose represents the configuration space (joint space) of all degrees

of freedom of an articulated 3D human model. The human model is biomechanically motivated and characterizes the degrees of freedom, range of motion, and skeletal structure observed in humans.

The process of reconstructing the configuration space of a kinematically constrained human model from 3D observations in pixel coordinates is characteristic of solving an estimation problem. An optimal estimate at interactive rates is very difficult, if not impossible for the following reasons. First, the motion of an articulated human model occupies a high dimensional configuration space having more than 30 degrees of freedom whose range of motion must be restricted due to limits in joint range of motion, and constraints due to self penetration of the bodies. In addition, self occlusions of body parts may result in local minima in the high dimensional space, and makes it difficult to search for the globally optimal solution. Finally, camera perspective projection can cause pose ambiguity where different human poses can have similar observed images, which results in many local minima in the high dimensional space, and makes it difficult to search for the globally optimal estimation. Although searching for an optimal solution in real time is challenging, any proposed solution should at the very least satisfy the following criteria: (1) be able to automatically initialize the pose estimation (2) be able to estimate the pose for long sequences accurately; (3) recover from tracking error or drift automatically; (4) be able to track through occlusions.

Certain applications present additional challenges. For example, in this study, we consider interactive applications where real-time processing at a relatively high frame rates is essential. Furthermore, the system must be portable and robust to dynamic backgrounds. To our knowledge, currently there is no 3D human pose estimation

system meeting all of the above criteria. The overall objective of this thesis is to explore a robust, real-time pose estimation systems without using markers.

1.4 Motivation: Interactive Human Pose Estimation from Depth Video Sequence

The objective of this research is to satisfy the requirements of real time performance, robustness to self occlusions and background clutter, automatic initialization, and system portability while avoiding the use of markers. It is extremely challenging, if not impossible, to rely on a single passive camera as the imaging modality. Real time depth images are, for all practical purposes, essential in meeting the stated objective. Although a variety of sensors provide range information in real time, the imaging device used in this research is based on a time of flight (TOF) camera [83]. An attribute of depth images obtained from a TOF camera is that the human body can be segmented from the background clutter once the working volume of the human performer is established. The segmentation of the human from the background is done with very little computational overhead. As will be shown in this thesis, with a segmented human, the pose tracking can be automatically initialized, incrementally, by detecting body limbs which are not self-occluded.

Depth images provide a valuable cue in tracking and reconstructing complex poses such as those which contain self-occlusions as in Figure 1.2(a). In practice, detecting and tracking self-occluded limbs cannot be resolved by using a single optical camera as in Figure 1.2(b). In particular, all methods based on silhouette information will fail to detect and track regions in the image where there are self occlusions. Although non-silhouette based methods have been proposed to detect and track self-occluded limbs,

their robustness very much depends on the illumination conditions, body texture, and perhaps extensive training in case of learning based methods.

Depth information is also critical in resolving depth ambiguity. Figure 1.3(a) illustrates two poses. In the left image, the right arm is held behind the body while in the right image, the right arm is held in front of the body. In this example, the silhouette of each image does not reliably reveal information as to whether the arm is in-front-of or behind the body. If depth information is available, as shown in Figure 1.3 (b), the distance from the right hand to the camera center can be computed once the right hand is detected.

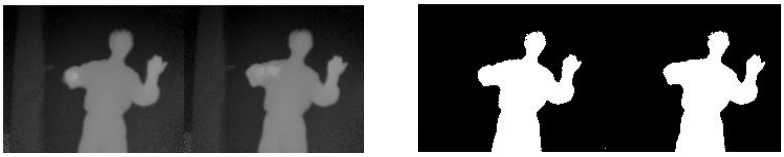


Figure 1.2: Advantage of depth information to resolve pose ambiguity for self occluded arm. (a) depth information. (b) different poses but same silhouettes from color images.

1.5 Robust Pose Estimation and Related Work

Recovering from tracking failures is one of the most challenging problems in robust human body tracking and pose estimation. Consider the example postures shown in Figure 1.4. When the arm is far from the camera, such as the throwing motion depicted in the right bottom image of Figure 1.4(a), the spatial resolution of the arm region may be insufficient for tracking based on local optimization based methods.

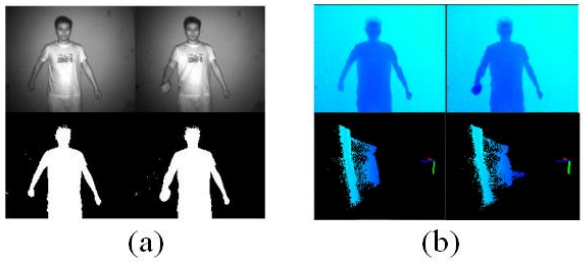


Figure 1.3: Advantage of depth information to resolve depth ambiguity for forward and backward right arm. (a) silhouettes from color images. (b) depth information at front and side views.

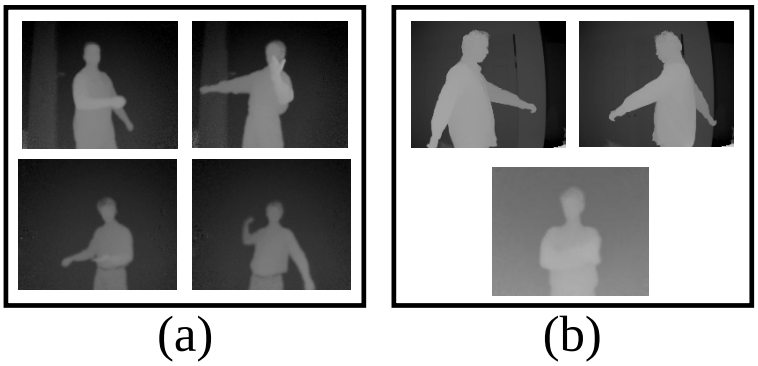


Figure 1.4: (a) Posture examples that are to be tracked; (b) Posture examples that are beyond the scope of this work

Although there are methods to estimate pose from low-level image analysis, the estimation results can be noisy, and even erroneous in certain cases where body parts become smaller or severely occluded.

In other example poses shown in Figure 1.4 (a), self occlusions between body parts can occur. For example, the lower arm can occlude the upper arm, or the head can be occluded by the arms. In many situations, self occlusions are intermittent. That is,

an occluded limb may re-appear in subsequent frames. A robust tracking algorithm must deal with intermittent occlusions to prevent tracking failures.

As described above, typical model-based pose estimation algorithms can result in tracking failures when body parts have low spatial resolution or when they become occluded. For cyclic motion patterns such as walking, it is possible to incorporate temporal prediction to improve the tracking results [44]. However, it is difficult to construct an effective temporal motion model for general motions. In many of the existing human tracking methods, tracking long sequences will result in tracking failure which cannot be easily recovered. Most methods do not describe solutions for recovering from tracking failure.

This thesis is to study the robustness of pose tracking from a depth video input. We define a pose tracking approach as ***robust*** if it has at least the following characteristics: (1) track the pose through temporary occlusions between body parts; (2) detect and track limbs that are reappearing from the previous occlusions; (3) automatically recover from tracking error when a set of specific poses are detected.

The algorithms presented in this dissertation fall into the category of model based approaches using depth image streams. The contribution of this research is the development of interactive, markerless pose estimation algorithms which are automatically initialized and robust in tracking and reconstruction of motions which exhibit intermittent self occlusions. In particular, we present algorithms for body part localization from depth image analysis (Section 1.5.2), pose estimation from low dimension motion descriptors with kinematic constraints (Section 1.5.3), pose estimation from constrained local optimization with dense correspondences (Section 1.5.4), and pose estimation with Bayesian information integration (Section 1.5.5).

1.5.1 Depth Scene Structure Assumption

Body part localization is a process in which head, torso, arms and legs are extracted from the rest of the image. As we see, analyzing the depth image so as to detect and label body parts reliably is a complex task considering the diversity of the human motions. To lower the burden for body part localization, we have made two kinds of assumptions: 1) assumption about background; 2) assumption about user motion. Assumption about background is pretty common, where we expect the depth distance between the observed subject and background to be larger than the depth sensor's resolution so that we can segment out the observed subject from background.

Assumption about user motion is to avoid the pose ambiguity and increase the pose detectability as well as the pose observability. To estimate the pose from the single view depth image, there is an inherent symmetric ambiguity as showed in the top row in Figure 1.4(b). In order to avoid such pose ambiguity, our algorithm assumes the user performs front-facing motion only (body twist angle up to 40degrees).

The pose detectability depends on three factors: (a) the distance $d_{\text{bodyparts}}$ between body parts; (b) the resolution r_{sensor} of the depth sensor ; (c) the depth slicing resolution r_{slice} in software implementation. We assume that the depth sensor has sufficient depth resolution to differentiate the overlapping body parts. Otherwise, our algorithm assumes the undetected limb keeps the same posture as the one estimated in the previous frame. For example, CSEM depth sensor can measure the distance with 1cm accurate within the range of 7m. Let us assume the $[d_{near}, d_{far}]$ be the range of interest, and d be the distance measurement. We generate the depth image I, which has a range between 0 and 255, from the measured distance based on

nonlinear mapping:

$$\mathrm{I} = \max(\min(0, \frac{d - d_{near}}{d_{far} - d_{near}}), 255) \tag{1.1}$$

During the depth slicing operation as described in Chapter 3, let depth slicing resolution be 7 depth interval value, d_{near} = 1m, and d_{far} = 6m, then, we are able to detect the body parts that have a minimal depth difference of:

$$d_{diff} = \frac{d_{far} - d_{near}}{255} \times 7 = 14\text{cm} \tag{1.2}$$

For certain poses, the detected limbs might be severely occluded or too small to have a good observation condition. We define the pose observability for a certain body part as the ratio: $\eta = \frac{PA_{\text{bodypart}}}{A_{\text{bodypart}}}$, where PA_{bodypart} is the body part projected area on the image, and A_{bodypart} is the body part surface area. When the observability of the body part is too low, our algorithm also assumes the limb keeps the same posture as the one estimated in the previous frame.

Our algorithm handles intermittent occlusion between body parts at the tracking stage. But the postures with low detectability and low observability are beyond the scope of this proposed method as showed in the bottom row in Figure 1.4(b).

1.5.2 Body Part Localization, Automatic Pose Initialization and Recovering from Pose Tracking Failure

An effective approach for recovering from pose tracking failure is based on the body part detection and tracking framework. In Gavrila and Davis [30], the subject is asked to wear tight-fitting clothes with contrasting colors on sleeves so that body part edge can be detected even when one body part occludes another.

Sigal et al [76, 75] are among the pioneers to integrate the bottom-up detection information during the pose tracking process to aid recovery from transient tracking

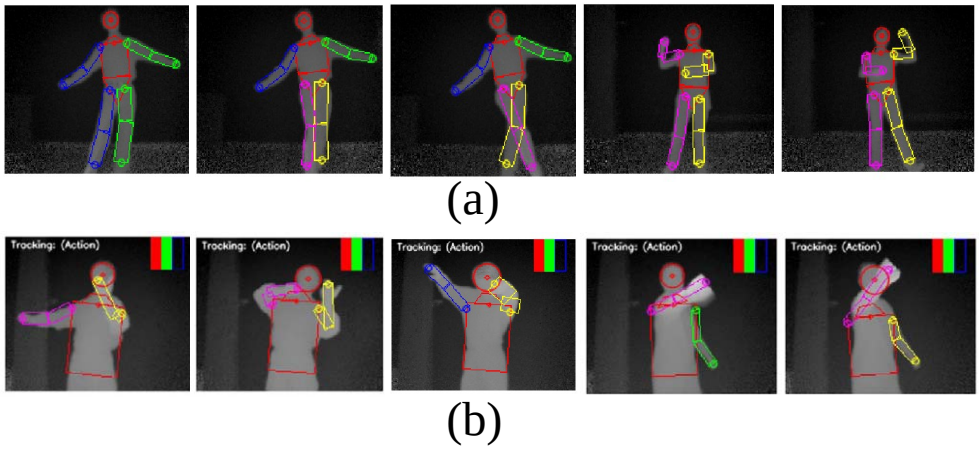


Figure 1.5: (a) Whole body part detection and limb labeling; (b) Upper body part detection and limb labeling

;

failures. But body part detectors used by Sigla et al are noisy, and possibly find spurious parts. Furthermore, they are not able to differentiate between left and right limbs. Hence they rely on the pose inference algorithm to assemble the body parts into sound poses. Although this allows them to use simpler body part detectors, it increases the computational burden, and is not optimal for interactive pose estimation. In general, both the diversity of visual appearance caused by clothing and background clutter make it difficult to localize the body part accurately and reliably from optical images [53, 46].

Other approaches will first apply background subtraction to extract a silhouette of the human figure, and find the body parts afterwards [34, 95]. However, such contour analysis based approaches can not detect the limbs that are located within the silhouettes.

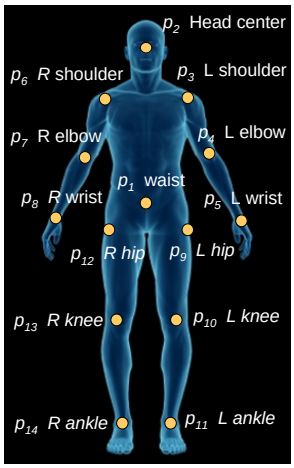


Figure 1.6: Key-points designated in detection and tracking

Our method to recover from tracking failure is based on the accurate detection of body parts from depth image as well as the reliable labeling of limbs as showed in Figure 1.5. Firstly, we use a deformable model to detect the head-neck-torso-waist. Secondly, skeleton analysis is used to detect open or looped limbs. The arms located within the silhouettes are further detected based on depth slicing operations. Thirdly, we take advantage of the depth information to infer the occlusion states of the body parts based on both temporal information and spatial context information. Finally, we represent the detected body parts with a natural 2D skeletal structure.

Because of this incremental body part detection, we can automatically initialize the pose tracker. Furthermore, from 2D skeletal structure, we can extract a set of low-dimensional key-point features (key-points) corresponding to prominent anatomical landmarks as in Figure 1.6, and use them to reconstruct the human model pose based on a novel constrained inverse kinematics formulation.

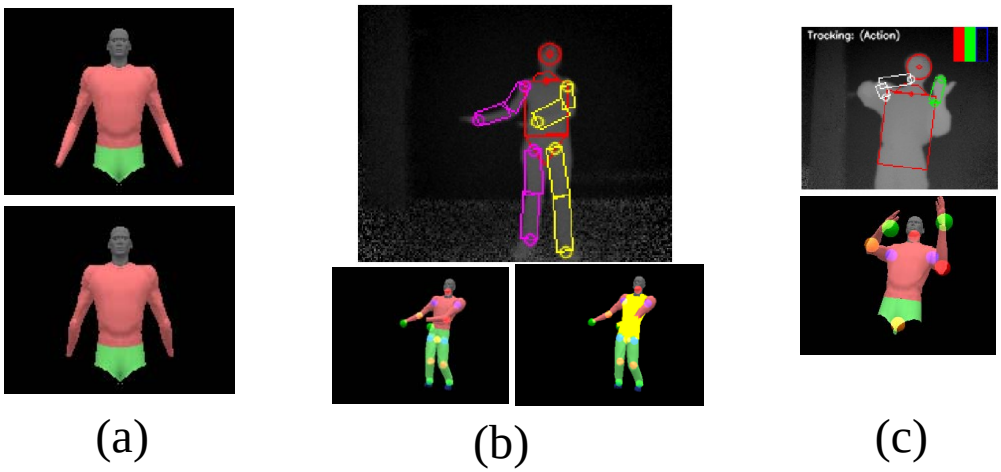


Figure 1.7: Pose estimation (a) using joint limits to avoid awkward posture; (b) using collision avoidance to avoid penetration; (c) feedback component from C-CLIK (right arm is not detected as showed with white color, and we use the predicted key-points to estimate the pose)

1.5.3 Pose Estimation from Low Dimensional Motion Descriptors with Constrained Closed Loop Inverse Kinematics

Given a set of key-points, the constrained closed loop inverse kinematics (C-CLIK) procedure, introduced in this thesis, produces human model pose variables subject to kinematic constraints such as joint limits and self penetration as in illustrated in Figure 1.7 (a,b). Without enforcing the kinematic constraints, the animated 3D human model may produce awkward or unrealistic motions. The C-CLIK framework allows the representation of the large number of human degrees of freedom involved in the execution of movement tasks to be expressed by a small number of key-points.

Reasonable estimates of human pose can be constructed from a small set of key-points, provided we have an appropriate human kinematic model. Furthermore, the C-CLIK algorithm is a Cartesian based tracking controller with an inherent prediction mechanism which can be exploited to estimate human model configurations even when there are missing observations. The prediction mechanism of C-CLIK leads to the feedback component. Feedback from the predicted locations of key-points are used to: a) estimate position of key-points which are not observed by the detector and b) resolve ambiguities which may be present during the key-point detection phase.

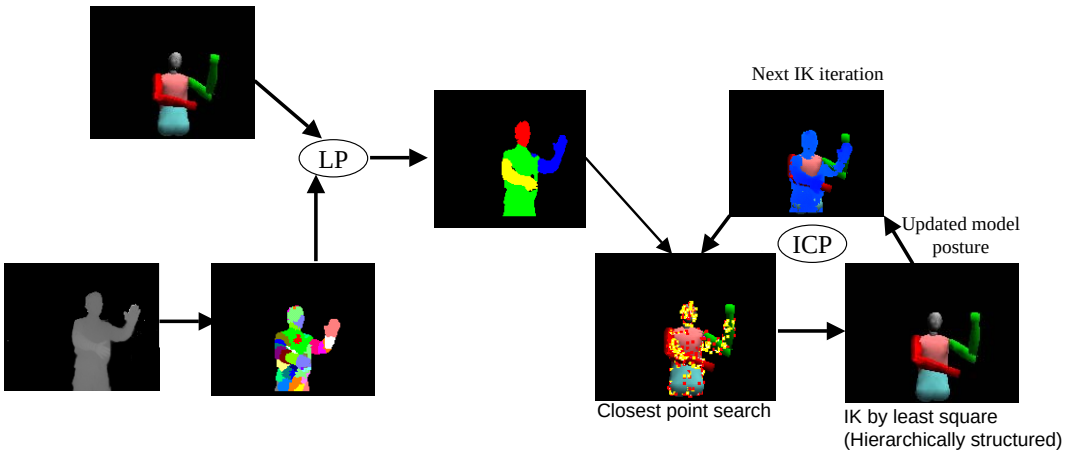


Figure 1.8: Constrained local pose optimization with LP-ICP

1.5.4 Constrained Local Optimization For Pose Estimation

Instead of specifically detecting key-points, other pose estimation approaches [33, 43, 97] use variants of iterative closest point (ICP) [7] to estimate the pose from a

set of dense correspondences between the human model vertices and depth data. A common issue with ICP approaches (or local optimization approaches) for human pose tracking is that the model may drift away from the data or get stuck in local minima.

Our work uses a constrained local optimization technique to estimate the pose of an articulated model from a stream of point clouds possibly from depth sensors. It is shown to overcome some issues of existing approaches for human pose tracking using similar types of data streams. The proposed approach (LP-ICP) is composed of two closely-coupled steps as illustrated in Figure 1.8: coarse human body part labeling and joint position estimation. In the first step, a number of constraints are extracted from notable image features such as the head and torso, and major parts of the human upper body are labeled using these constraints with linear programming (LP). The second step estimates joint positions optimally using dense correspondences between depth data and human model parts. A grid acceleration data structure is used to achieve pose estimation at a high frame rate without loss of accuracy.

We have made a comparative study of markerless pose tracking based on a commercial marker-based tracking system and shown that our joint positions have an accuracy about several centimeters.

1.5.5 Bayesian Information Integration

A feature-based approach is robust and can recover from tracking failure when a body part is re-detected. However, its estimation accuracy depends solely on the localization accuracy of the feature points. The LP-ICP-based method can achieve

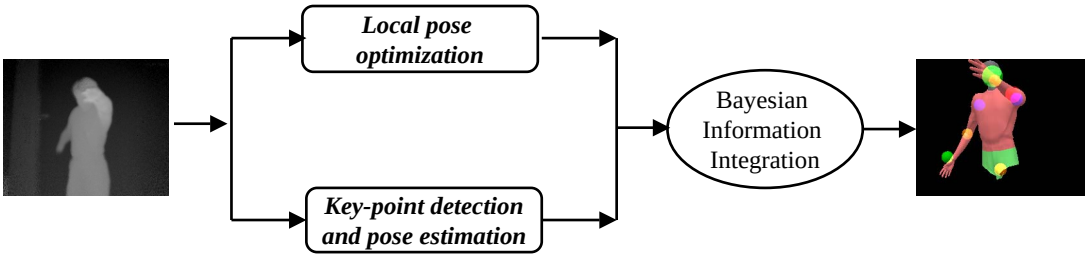


Figure 1.9: Bayesian information integration for pose estimation

a higher accuracy than feature-based method because it utilizes a set of dense correspondences. But, we have found that it is not able to recover from tracking failure once it is trapped in local minima of the energy function. The labeling of body parts depends on the estimated pose from the last frame, and it could result in the accumulation of pose tracking errors. As a result, the performance of ICP-based pose tracker is degraded after the tracking failure. We further propose a unified pose tracking based on the Bayesian inference framework to integrate the estimation results from the feature-based method and the estimation results from the local pose optimization method as in Figure 1.9.

1.6 Contributions

Our major contributions include:

1. The depth image analysis is developed to infer the body part occlusions with both spatial and temporal context. It is designated to enhance the robustness of the pose estimation so as to track longer sequences and recover from the tracking failure. A novel head-neck-torso-waist(or head-neck-torso for upper

body) template is designed, and enables us to detect those body parts from the depth image. Human limbs are detected based either on the distance transformed skeletons or depth slicing operations that are developed specifically for the depth image. Human limbs are labeled and tracked based on both spatial context and temporal information so as to result in a robust human body tracking method that satisfies the above robust pose tracking characteristics.

2. Human poses are reconstructed from a set of feature points with kinematic constraints. As a result, the animated graphical model satisfies the anthropometrical constraints of the human subject, such as joint limits. Without such constraints, one could possibly result in unnatural poses such as backward bending arms that are not visually appealing as well as physically incorrect.

3. Local optimization is developed to find the local minima for optimal pose estimation. The estimated motion is more accurate by its nature because we could use dense correspondences to estimate poses. Our local optimization for pose estimation is efficient for the interactive applications, and can track longer sequence compared to other similar approaches existing in the literature, although it has the major limitation that it cannot recover from tracking failure for the complicated motions with difficult poses.

4. The Bayesian framework is explored to integrate information from depth image analysis, model-based local optimization, and temporal prediction. This unified approach is optimal in terms of its tracking accuracy, error recovery ability, and ability to integrate with other pose estimation information, although the computational burden of the Bayesian-based method is a major concern for the

interactive applications with the current implementation. It remains as future work to have a parallel implementation to improve the computational speed.

5. A novel collision-free retargeting for interactive robot motion learning is proposed. This algorithm allows us to retarget the observed human motion from the depth sensor to the humanoid robot in real-time and safely (without self-collision between body parts).

Furthermore, the suggested work has following merits compared to existing approaches:

1. In contrast with learning-based methods, no training is necessary, except for the first few frames where the subject is requested to make t-pose postures. Moreover, our framework can be extended to incorporate learning-based estimation results through Bayesian inference.

2. Most of the suggested approaches in the literature can only achieve a low frame rate. The suggested work can achieve near real-time performance for interactive applications.

3. With the proposed pose estimation approach, we can demonstrate an interesting application, 'learn-from-demonstration', for a humanoid robot through collision-free retargeting as shown in Figure 1.1. To our knowledge, our proposed method is a novel motion retargeting method that enforces kinematic constraints such as joint limit avoidance and self penetration avoidance.

1.7 Dissertation Overview

The dissertation is organized as follows.

Chapter 2 reviews the existing work of human pose estimation from image sequences. Firstly, the latest research papers in 2D human motion estimation are discussed. This is followed by the detailed review of the existing work in 3D human motion estimation. Based on the input information differences, we can categorize the existing approaches for 3D human motion estimation into three systems: single camera system, multiple camera system, and depth camera system. Furthermore, the proposed approaches for each system can be classified as either model-based methods or learning-based methods. Limitations of the existing work are discussed based on estimation accuracy, ability to recover from tracking failure, and computation speed.

Chapter 3 describes a method to detect, label and track the body parts for an articulated 2D human figure using depth images. The major theme is not only to handle the transient self-occlusion between the body parts of the articulated human figure during the tracking, but also to robustly detect and track the reappearing limbs that are possibly missed in the previous frames. This results in a novel algorithm to label and inference the occlusion status of the body limbs based on both temporal tracking information and spatial context information. The experimental results and human feature point tracking accuracy are reported.

Chapter 4 reconstructs the 3D human motion from the trajectories of a set of feature points with the closed loop inverse kinematics. The algorithm we describe can handle missing feature points by predicting the missing feature points from the estimated joint motion. In particular, our algorithm can enforce both the joint limits constraint and self-collision avoidance (as in AppendixA) when reconstructing the motion. Without such constraints, the reconstructed results could have unnatural

poses such as backward bending arms and arm-torso penetration that are not visually appealing as well as physically incorrect.

Chapter 5 presents our previous work using a constrained local optimization technique to estimate the poses of an articulated 3D model from a stream of point clouds possibly from depth sensors. The approach is composed of two closely-coupled steps: coarse human body part labeling and joint position estimation. In the first step, a number of constraints are extracted from notable image features such as the head and torso, and major parts of the human upper body are labeled with these constraints. The second step estimates joint positions optimally using dense correspondences between the labeled depth data and human model parts. The proposed framework is shown to overcome some issues of existing similar ICP-based approaches for human pose tracking using similar types of data streams. Performance comparison with motion capture data is presented to demonstrate the accuracy of our approach.

Chapter 6 describes a unified 3D pose tracking approach based on the Bayesian inference framework to integrate the pose estimation results from the last two chapters. This results in a robust pose estimation method that is able to track the limbs through occlusions, recover from tracking failure whenever a limb is re-detected, track the pose with missing features, and improve the tracking accuracy with local optimization at the expense of the computational burden. Nonetheless, this demonstrates a potential pose estimation approach for parallel implementation, and more importantly, to be able to possibly integrate with other pose estimation modality such as pose estimation from machine learning methods.

Chapter 7 demonstrates the two interactive application scenarios for the proposed pose estimation method. The first is related to experiments of 'learn-from-demonstration' for the ASIMO robot, where ASIMO robot is able to mimic the observed human motion in an interactive speed. The second is related to the application of the pose estimation from low dimensional feature points.

While we have produced promising results, our techniques have the limitations. Chapter 8 summarizes the current work from the point of the view of the detectability and observability, and lists the future work.

Appendix A describes a novel impulse-based algorithm for the online motion retargeting that allow us to enforce self-collision avoidance constraint within the closed loop inverse kinematics framework.

CHAPTER 2

LITERATURE REVIEW

Much of previous work related to vision-based human motion estimation can be categorized into two groups: 2D human motion estimation from images, and 3D human motion estimation from images. Although in this thesis we are mostly interested in 3D human motion estimation, the discussion of 2D human motion estimation will make this literature review comprehensive. More importantly, 2D human motion estimation results are often used to enhance the 3D human motion estimation. Section 2.1 discusses the latest approaches for 2D human motion estimation and Section 2.2 reviews the latest papers on 3D human motion estimation. Section 2.3 summarizes this chapter.

2.1 Review: 2D Human Motion Estimation

In this thesis, we are interested in tracking a set of human feature points that are informative for pose estimation from depth image sequences. It is closely related to a popular topic in computer vision: how to estimate the 2D human motion from images and image sequences. As shown in Table 2.1, we will review the latest methods for 2D human motion estimation so as to understand the computational techniques for 2D human motion estimation and their limitations.

Approaches	Parametric Model	Method	Initialization	Error recovery	Occlusion handling
Ju96 [37]	cardboard person model	optical flow constraint and articulated constraint	manual	no	no
Cham99 [13]	scaled prismatic model	multiple hypothesis tracking	manual	no	implicit
Mori02 [51]	key points	matching with shape context descriptor	automatic	yes	implicit
Ramanan05 [61]	articulated figure	pictorial structure framework	automatic	yes	explicit
Wu03 [93]	articulated figure	dynamic markov network	manual	no	no
Shen04 [71]	articulated figure	dynamic markov network	manual	no	no

Table 2.1: Review of 2D human motion estimation approaches.

The cardboard person model [37] is a representative of the early method on this subject that models and estimates the body part motion field directly. The individual limb is described as a rigid planar patch whose image motion is defined as a eight-parameter model. To estimate these motion parameters, they utilize both the optical flow constraint arisen from brightness constancy and articulation constraint arisen from the articulated structure of the connected limbs. As is known, the optical flow constraint is sensitive to the cloth deformation. Therefore, the accumulation of tracking errors could result in the tracking-failure, which makes this approach questionable for tracking long sequences. This approach does not address the self-occlusion either. Yet the self-occlusion can arise frequently due to the interaction between limbs.

Instead of modeling and estimating the motion of each individual limb, the scaled-prismatic model is proposed to model the articulated figure based on its kinematic parameters [54], where two successive body parts are connected by a scaled prismatic joint with 2 degrees of freedom: rotation angle and length of link. With such a model, Cham et al [13]. propose the multiple hypothesis tracking with local optimization of probability density modes. In particular, let $p(x_t|Z_t)$ be the time-evolving probability distribution to be maintained. Based on Bayes rule, it can be defined to be: $p(x_t|Z_t) = kp(z_t|x_t)p(x_t|Z_{t-1})$. The authors propose to represent the $p(x_t|Z_t)$ as piecewise Gaussians with multiple modes. To update the distribution, priori distribution $p(x_t|Z_{t-1})$ is obtained from a kalman filter prediction step, and likelihood $p(z_t|x_t)$ is estimated as piecewise Gaussians with the modes obtained by locally optimizing the template registration with initial states sampled from $p(x_t|Z_{t-1})$. To avoid an exponential increase in modes, the posterior modes are pulled from the likelihood modes

modulated by the most compatible prior modes. Although the tracker has better performance compared to the single mode tracker in coping with the ambiguity arising from self-occlusion, the tracker is not guaranteed to handle the self-occlusion correctly since it does not model it explicitly.

Besides the above model-based approaches, there are approaches to locate the body parts from bottom-up image analysis. For example, Mori et al [51] locate a set of key points by matching the input image to the exemplar 2D views of the human body using the shape context descriptor. It starts with edge detection on the image using the boundary detector of Martin et al. [49], and a set of sample points are sampled from the detected edge pixels. This is followed by matching the sample points from the test image to the sample points from exemplar images, where the optimal correspondences between these two sets of points are found based on the shape contexts. Finally, the user-identified keypoints on the exemplar image are transferred to the test image by estimating the deformation function using the correspondences. The method is able to cope with occlusion using the labeled key point positions in the matched exemplar. But its performance depends on the number of exemplars, which could be unreasonably large considering the high degree of freedom in the human figure.

Based on the assumptions that certain canonical poses can be detected robustly, and limb appearances share the similarity across the sequence, it is proposed to learn the discriminative limb appearance models from the detected human figure with canonical poses, and then use them to predict the candidate limbs. Ramanan et al [60] obtain impressive results by assembling these limb candidates using the

efficient algorithms for tree-structured articulated figures [29]. This is one of a few methods that is able to cope with occlusion explicitly.

More than 20 DOFs are typically used to fully describe the motion of a 2D articulated human figure. This makes it too computationally intensive to track the motion with standard particle filters [26, 36] since the required number of particles increases exponentially in order to track human figure accurately. This is known as the curse of dimensionality. The recent approaches to address this is to use approximation inference on the graphical model including the belief propagation method [71] and the mean field method [93]. But the existing methods normally do not consider the self-occlusion scenarios, and it remains unclear how to approximate the likelihood function with self-occlusion [82].

2.2 Review: 3D Human Motion Estimation

As summarized in Table 2.2, there have been substantial advances on 3D human motion estimation, i.e., to estimate the joint motions of the 3D articulated human model. However, challenging problems remain due to the high number of degrees of freedom due to the dynamic range of poses during human activities, the diversity of visual appearance caused by clothing, visual ambiguities due to self-occlusion of non-rigid 3D object, and background clutter.

Based on the input information differences, we can classify the existing methods into three sub-categories: single camera systems, multiple camera systems, and depth camera systems. Furthermore, these approaches for each sub-category can be either model-based methods or learning-based methods.

Approaches	Model	Speed	Accuracy	Error Recovery
Model-based method				
Multiple Camera				
Gavrila96 [30]	tapered super-quadrics	n/a	n/a	n/a
Kakadiaris00 [40]	deformable model	n/a	n/a	n/a
Deutscher00 [25]	truncated cones	n/a	n/a	n/a
Cheung00 [16]	ellipsoids	16fps	n/a	n/a
Delamarre01 [22]	truncated cones	n/a	n/a	n/a
Carranza03 [11]	polygonal model	6.81sec/frame	n/a	grid Searching for fast motion
Sigal04 [75]	taped cylinder	n/a	n/a	body part detection
Kehl06 [41]	superellipsoids	1fps	n/a	n/a
Single Camera				
Yamamoto91 [94]	polygonal model	n/a	n/a	n/a
Bregler98 [9]	ellipsoids	n/a	n/a	n/a
Sidenbladh00 [73]	cylinders	5min/frame	n/a	n/a
Sminchisescu03 [79]	super-quadric ellipsoid	n/a	n/a	n/a
Lee04 [46]	truncated cones	8min/frame	n/a	n/a
Depth Camera				
Grest05 [33]	polygonal model	4fps	n/a	n/a
Knoop06 [43]	cylinders	10fps	n/a	n/a
Ziegler06 [97]	polygonal model	1fps	n/a	n/a
Learning-based method				
Multiple Camera				
Ren05 [63]	PSH+MoGraph	n/a	44cm	n/a
Single Camera				
Howe99 [35]	Mixture of Gaussian	n/a	n/a	n/a
Rosales01 [67]	Specialized maps	n/a	n/a	n/a
Mori02 [51]	Shape context matching	n/a	n/a	n/a
Shakhnarovich03 [70]	Parameter sensitive hashing	0.5fps with Matlab	n/a	n/a
Demirdjian05 [24]	PSH and multiple hypotheses	1fps	n/a	example-based
Agarwal06 [2]	Relevance Vector Machine	real-time	4.1°	n/a
Sminchisescu05 [77]	Bayesian mixtures of experts	n/a	n/a	n/a

Table 2.2: Review of 3D human motion estimation approaches.

2.2.1 Model-based Methods for Multiple Optical Camera Systems

In model-based methods, a kinematic representation of a 3D human model is used, and pose estimation is performed incrementally by minimizing the differences between the 3D model and observation. Model-based pose estimation is an ideal test bed for various advanced optimization algorithms. An early example of model-based pose estimation with multiple cameras is formulated by Gavrila and Davis [30]. They use a 22-DOF model (6-DOF for the translation and rotation of torso, and 4-DOF for each limb). Each body part is represented as *tapered super-quadratics*. The quadratic shape parameters for each body part are estimated during the initialization stage and pose parameters are estimated during the tracking stage. Both are realized by minimizing the undirected normalized chamfer distance between edge contours of synthesis image and edge contours of observed image. To speed up the minimization, they decompose the high dimension search space hierarchically and employ the best-first search in each subspace. The subject is indeed asked to wear tight-fitting clothes with contrasting colors on sleeves so that body part edge can be detected even when one body part occludes another. In addition,this optimization procedure is slow considering that one has to perform the image-synthesis, and chamfer distance computation for each intermediate model pose during the searching .

Kakadiaris and Metaxas [40] present a physically based framework, where a generalized force for each part is computed to update the translational and rotational parameters so as to minimize the discrepancy between the observed occluding contour and predicted model occluding contour. More specifically, an accurate shape model is estimated during the modeling phase, and joint motions (both joint angles and joint

velocities) are estimated subsequently by matching the contour points between synthesized model image and observed image. To cope with the occlusion, the authors suggested that it is helpful to identify the optimal camera views for each body part so as to mitigate the occluded and degenerated camera views. This is implemented as an ordering phase, where camera views are selected based on body part visibility and observability. Although the paper proposed the method to select the optimal camera view to handle the occlusion, it did not discuss the problem of recovering from tracking-failure. In a similar approach, Delamarre and Faugeras [22] convert the distance between model contours and the observed image contours to physical force, and estimate the pose with inverse dynamics on an articulated human model.

Instead of preforming deterministic searching in a high dimension space, Deutscher et al [25] present a stochastic search with annealed particle filtering. The main goal of the paper is to show that annealed particle filtering is able to track the motion with less particles than the standard condensation particle filtering approach. In their description, the particles are weighted by the edge and region measurements. Edge measurement models the edge gradient strength of the sampled silhouette pixels of the rendered model, and region measurement models the overlapping area between the rendered silhouette and the observed foreground. A procedure similar to simulated annealing is used to obtain a set of optimal particles for estimation which is more likely to avoid the local maxima. Firstly, this is a computationally expensive approach for pose estimation since simulated annealing requires iterations, and for each iteration one will have to perform the model back-projection in order to compute the edge and region measurements. Secondly, without incorporating the bottom-up detection information, it remains uncertain whether this approach is able to recover

from tracking-failure since particle filtering can only propagate in the neighboring influencing regions, but not jump between modes.

In a related work, Sigal et al [76, 75] propose the loose-limbed body model to estimate the pose possibly with multiple cameras. A loose-limbed body model is a probability graphical model that consists of nodes as body parts and edges connecting the body parts, collected from the neighboring time stamps $(t-1, t, t+1)$. The conditional probability distributions for the probability graphical model are learned from the training database. The optimal pose estimation is then performed by the approximation probability inference algorithm, a.k.a Non-Parametric Belief Propagation on the graphical model. Compared with the method proposed by Deutscher et al. [25], their method is able to integrate the bottom-up information obtained from limb and head detectors. The experimental results demonstrate better pose tracking performance than the results by Deutscher et al. [25]. In particular, their algorithm could have the potential to recover from the tracking-failure because the inference algorithm integrates bottom-up information, although it remains unclear what tracking performance the algorithm can achieve on the heterogeneous sequences that are different from the training sequences from which conditional probability distributions are learned.

With more cameras, Carranza et al [11] describe a system that estimates articulated body pose by minimizing the difference between the observed silhouettes and the projections of the model. Texture maps are generated during pose estimation to facilitate interactive rendering during playback. The authors observed that it is prone to locate the local minimum for certain problematic pose scenarios when it has fast moving body parts or one limb moves very close to the torso. To have a robust pose

tracker against these problems, they apply the grid searching to find the valid initial poses where the elbow and wrist must be projected into the image silhouettes of each camera view, and they are located outside of the torso. These valid poses are used as the initial poses for local optimization. They demonstrated impressive pose tracking results for the complex motions. But it is a computationally expensive procedure to use grid searching strategy, and it is not applicable for the interactive applications.

In contrast to these methods by projecting the model to image plane and comparing the camera projections of the 3D model with silhouettes or edges of camera images, Cheung et al [16] reconstruct the visual hull of an observed human based on a 3D voxel representation, and track each ellipsoid body part based on nearest neighbor segmentation between visual hull voxels and ellipsoid body parts. As described in their paper, the system consists of five cameras. Each camera is connected to a PC which locally extracts the silhouettes of the moving person in the image captured by the camera. The five silhouette images are sent to a host computer to perform 3D voxel-based reconstruction. Ellipsoids are then used to fit the reconstructed data. Six ellipsoidal shells (as analog to the head, torso, two arms and two legs of a human body) are used to fit the reconstructed data. Fitting is composed of two steps. In the first step, each voxel is assigned to the closest ellipsoid. In the second step, the ellipsoid parameters are updated from the assigned voxels based on moment analysis. Although this method is able to achieve 15 frames per second with multiple cameras, it has the major limitation that it cannot track the body parts correctly when they are close to each other.

In the related work, Kehl and Gool [41] present a pose estimation system from a visual hull reconstruction based on stochastic meta descent searching that combines

a stochastic sampling of model vertices and a gradient descent searching algorithm. The basic version of pose tracking is surface alignment, that is to optimally estimate the pose so to minimize the distance for a set of correspondences between a 3D model and the reconstructed visual hull. As the authors point out, this criterion alone is not robust when the body parts are close to each other, and the tracker could fail to follow the reappearing limbs afterwards. Their study is to augment the surface alignment distance with edge alignment distance. To compute the edge alignment distance, a set of occlusion points are sampled from 3D model, and they are matched with edge pixels from the images. In their implementation, they also find that it improves the tracker performance to utilize the reconstructed voxel color information when the correspondences between voxels and 3D model vertices are built based on nearest neighbor criterion. Finally their implementation of distributed pose estimation system could achieve 1 frame per second.

As seen, searching in a high dimension with possibly many local minima is difficult. As a result, most of model-based research works focus on incremental pose updating to take advantage of temporal coherency between frames. But such strategy could make it difficult to successfully track the disappearing-and-reappearing limbs when body parts are occluded and then reappear.

2.2.2 Model-based Methods for Single Optical Camera Systems

As opposed to the effort required to install multiple-camera systems, 3D pose estimation from monocular images is appealing for the convenience of the system setup. The main challenges for pose estimation from monocular images arise from

the high dimensionality of the parameter space, the kinematic singularity as analyzed by Morris and Rehg [54], possible forwards/backwards flipping caused by depth ambiguity [79], and self occlusion. As a result, the parameter space is multi-modal, and any trivial deterministic searching methods are suspectable to missing the globally optimal solution. Early approaches that are based on the alignment between projected model contour and image edge lines [65, 90] or based on constant brightness criteria [94, 9, 19] are sensitive to the image noise and doubtful to track long sequences.

Recently, a number of model-based stochastic methods have been introduced. Sidenbladh et al [73] employ particle filtering to track the walking motion based on the learned walking motion using linear dimensionality reduction from PCA. Their experimental results confirm that the learned walking motion dynamics could be useful to reduce the dimensionality of pose tracking, and predict the motion. But the parameter estimation still relies on the standard particle filter technique by propagating the samples across time, and assigning the weights to samples with the evaluated image likelihood. This has limitations. Firstly, it is not able to track a long sequence because of accumulated errors. Secondly, their method can not estimate the torso rotation accurately. Finally, it seems difficult to automatically initialize the 3D pose configuration for tracking.

Sminchisescu et al [79] elaborate two major difficult problems for pose tracking. One is that the likelihood distribution inevitably includes a lot of local maxima. The other one is pose ambiguity that rises from the twofold forwards/backwards flipping. To cope with the first difficulty, they propose a hybrid search algorithm that combines scaled-covariance sampling and robust continuous optimization subject to physical

constraints and model priors. Covariance-scaled sampling intends to provide a set of initial pose configurations for the subsequent continuous optimization. To handle the second difficulty, for each pose configuration sampled from scaled-covariance, they enumerate the possible skeletal kinematic trees so as to rapidly find the initial pose configurations that reflect the possible forwards/backwards flipping. Their experimental results demonstrate that their approach is more effective in avoiding the local minima than the conventional particle filter approach. However, their approach can not deal with the transient body part tracking failure, and is not able to recover from it because of its incremental pose-estimation nature.

Lee and Cohen [46] set an ambitious goal to estimate the human pose from a single image with background clutter. They build a generative human model to represent the human body shape, pose and clothing as a priori distribution. A human image synthesized from a candidate human model is compared with the real image to evaluate the likelihood function. Since the posterior distribution is a multiplication of the two, they propose to sample from the posterior distribution using Markov Chain Monte Carlo (MCMC) framework. The MCMC with random-walk sampling converges slowly. Their speed-up technique is to take advantage of the low-level detection results such as face, skin color, and limb segmentation to form the efficient MCMC proposal function for pose estimation from image. This approach can be applied to images with background clutter to perform simultaneous segmentation and pose estimation. However, it is not without drawbacks. Firstly, it requires that the priori distributions, such as pose configuration, human shape model, and clothing model are learned from an existing database. It is not a trivial task to learn these informative priori distributions unless applying some heuristics about the underlining database.

Secondly, it also requires the heuristics to compute the proposal distributions for various body joint positions. It remains unclear what performance the algorithm will have if these body joint positions are diffuse. For interacting arms, it is more difficult to distinguish the left and right joint proposal maps.

Although it is indeed very attractive to be able to estimate poses from a single optical camera, it is a much more challenging task than the pose estimation from multiple optical cameras, as demonstrated by the number of papers and systems proposed. In particular, a single projection from an optical camera makes it difficult to recover the body part depth.

2.2.3 Model-based Methods for Depth Camera Systems

Depth measurement from stereo cameras provides information to resolve the depth ambiguity as in a single camera. Grest et al [33] adapt the Iterative Closest Point (ICP) [7] algorithm to the pose estimation for an articulated human model. It starts with the analytical computation of a Jacobian matrix for non-linear optimization with Gaussian-Newton iteration that is expected to minimize the distances for a set of corresponding points between observed depth points and visible model vertices. In order to estimate the pose, their algorithm is similar to the standard ICP approach that consists of two steps. Firstly, for each observed point, the nearest visible model vertex is found. Secondly, the model pose is updated with a non-linear optimization technique. The remaining is about how to efficiently find the correspondences. In their implementation, they find the visible model vertices by rendering the model triangles with OpenGL, and these visible model vertices are sorted into an associative array based on their heights. For each observed point, the associative array is searched for

the point with the next height value. The searching could benefit from the sorted associative array for an early stop if the height distance is larger than the recorded minimal Euclidean distance. Their implementation could achieve 4 frames per second. But it remains questionable whether its is be able to track complex motion where body parts get close to each other.

Knoop et al [43] also use ICP to update pose parameters by incorporating multiple input data from different sensors such as 3D data from a time-of-flight depth camera and stereo camera, and hands/face tracking from color camera. As we know, the success of ICP method depends on the robustness to find the correspondences. Firstly, a set of correspondences between 3D model vertices and 3D data from the time-of-flight sensor or stereo camera is found based on nearest neighbor criterion, and a set of constraining equations are formed from the correspondences. Secondly, they utilize a 2D feature tracker to track hands and face. For each tracked feature, a constraining equation is formed. Both types of constraining equations are integrated together by assigning higher weights for constraining equations from feature tracking because correspondence from feature tracking is more accurate than correspondence from depth sensor. Their experimental results show that they can achieve more accurate pose tracking by integrating both types of correspondences than the pose tracking by only utilizing the individual type of correspondence. However, it becomes a challenging task to have a 2D hand/face tracker that works well for various complicated motion, and they do not elaborate on how the robustness of 2D feature tracker could affect their 3D pose estimation.

Ziegler et al [97] use the unscented Kalman filter (UKF) based on a set of correspondences between model vertices and observed stereo point cloud. It is a hybrid

method of ICP and UKF. A set of σ-points (a.k.a σ-poses) are sampled from the posterior distribution of pose state vector as required for unscented Kalman filter [38]. For each σ-pose, the visible model vertices are found by utilizing the OpenGL depth buffer, and further down-sampled by a random sampling step. The correspondences between visible model vertices and 3D depth points from sensors are built, and a measurement vector is formed from the correspondences. Thereafter, the UKF is utilized to updated the posterior distribution. In contrast to the particle filter based approach, which requires a large number of particles to work well since pose tracking has a high-dimensional space, their UKF implementation is able to track pose with only a set of 29 σ-poses. Nevertheless, it remains unknown whether it will be able to recover from tracking-failure as ICP iteration is sensitive to the outliers in the correspondences.

Although it is time consuming to perform brute force nearest neighbor searching, these methods are able to achieve accurate pose estimation with high frame rate after applying a certain nearest neighbor searching speedup technique. Consequently, at present, it is more promising to utilize the depth camera system for real-time or interactive applications than the other camera systems. An important limitation of the existing approaches is that they do not discuss tracking failure and how to recover from it. It is observed that a brute-force implementation of ICP could result in tracking failure because the accurate correspondences are difficult to obtain for the scenarios when body parts are close to each other.

2.2.4 Learning-based Methods for Pose Estimation

In learning-based methods, pose estimation is a process to predict the pose based on a learned regression between image observations and pose. Such an approach is attractive for a number of reasons. Firstly, it can be fully automatic without manual intervention, such as manual initialization. Secondly, it does not have the error accumulation that exists in the model-based methods.

An early approach is proposed by Howe et al [35]. They reconstruct the 3D pose from 2D joint tracking results. A set of motion snippets are derived from the training sequences with known 3D motions. A prior probability distribution for the snippets, defined by the Gaussian Mixture Model, is learned. They compute the likelihood based on the difference between observed 2D snippets (exported from the 2D tracker) and projected 2D snippets of the yet unknown 3D snippets. The optimal unknown 3D snippet is found to be the one that maximizes the posterior distribution. This method is able to reconstruct the 3D motion with a high accuracy if high quality 2D tracking results are available. But it is a nontrivial task to have a robust 2D tracker that works very well for complex motions with self-occlusions.

Rosales and Sclaroff [66] make the early efforts to learn the direct mapping from visual features to poses. They represent the human pose as the 2D body joint configuration. Firstly, the visual features (they use Hu-moments) are extracted from the body shape image obtained from background segmentation. During the training stage, they cluster the human body configuration into groups. For each cluster, a mapping is learned based on the multi-layer perceptron from visual features to body configurations. They further learn another mapping from body configurations

to visual features whose purpose is to synthesize the visual feature from body configuration. During the testing stage, given the extracted visual features, a set of possible poses are predicted by the learned mapping functions for all the clusters. From these predicted poses, the system synthesizes the visual features again. The most likely pose is found to be the predicted pose from the identified cluster whose visual feature distribution best matches the synthesized visual feature. It is possible to take advantage of temporal coherence of the motion with majority voting based on the fact that neighboring frames are likely to be from the same cluster. Promising results are reported for both frontal and non-frontal view pose estimation. However, the authors find that for complex motion the estimation could be error-prone because of the limited discriminative power of Hu-moments and inherent ambiguity of poses after projection to 2D.

Shakhnarovich et al [70] present a method that uses the parameter-sensitive hashing to rapidly retrieve similar examples from a large database of example pose images in sublinear time. Pose estimation is then reduced to the local regression on the retrieved examples. The set of hash functions are formed from a set of simple binary decision trees that use the local image features. During the training stage, both binary decision trees and the local image features are optimally found based on the Adaboost algorithm. However, this approach does require that the tracked person is segmented from the image, and carefully aligned afterwards. The estimated poses could be erroneous for testing examples that are not covered by the training set. Recently, there are two extensions [24, 86] based on this approach. Demirdjian et al [24] use the parameter-sensitive hashing to quickly search for a small set of pose examples that lie close to the modes of the likelihood function. These are combined

with the predicted modes from the temporal propagation. Then, all the hypotheses are refined to locate local modes based on ICP using the depth data. Therefore, their method is able to locate the major modes that reflect the landscape of the likelihood function. Finally, the likelihood function and temporal propagation are integrated together based on the Bayesian tracking framework. In the second extension, Taycher et al [86] use parameter-sensitive hashing to both build the transition matrix of the motion graph, and efficiently evaluate the observation potential, which allows them to formulate pose tracking as the grid filtering process that can be computed in real-time with a high performance computer.

In a related work with multiple cameras, Ren et al [63] are able to control the avatar to follow human dancing motion by finding the optimal yaw angle and body configuration from the motion graph by learning a set of hashing functions with Adaboost algorithm. They demonstrate the effectiveness of their method on the complex human motion of the swing dancing. Instead of using the edge direction histogram as feature vectors as in paper [70], they use the rectangular filters to compute the feature vectors from the silhouette images.

Instead of retrieving similar examples from the training database, Agarwal and Triggs [2] describe a learning-based method for recovering 3D human body pose from single image or monocular image sequences by directly estimating a regression function. Their first contribution is to define a histogram-of-shape-contexts descriptor, extracted from the segmented silhouette image, as the input vector for regression function. Then, they evaluate several different regression methods: ridge regression, Relevance Vector Machine regression, and Support Vector Machine regression over both linear and kernel bases. They also propose to include the prediction from a

second order linear autoregressive human body dynamics model into the regression input vector. This enables them to use the pose dynamics to disambiguate poses which have similar image silhouettes due to the perspective projection.

Sminchisescu et al [77] formulate the pose tracking as density propagation in a discriminative graphical model, where conditional probability is represented as Bayesian Mixture of Expert Models and learned from the observations encoding the appearance of image silhouettes of the training examples. Unlike the regression method proposed by Agarwal and Triggs [2], this approach allows them to model the multi-modal distributions and is more appropriate for pose recovery problem comparing with other single modal regression methods.

3D Pose estimation is a challenging problem for learning-based algorithms. All of the learning-based methods share the following similar limitations. Firstly, pose estimation with regressions in high dimension pose space has the dilemma caused by the curse of dimensionality. When the number of training examples are limited, the prediction capability on novel inputs is constrained, and error-prone. With increasing number of training examples in order to estimate the complex motions, the probability distribution of the pose space becomes multi-modal which is difficult to be learned in general. Secondly, most of current methods are sensitive to the input noise, which often is very difficult, if not impossible, to remove from test images that are obtained from background subtraction. Consequently, the estimation accuracy is not as high as the model-based approaches. Last but not the least, without domain knowledge, there is an inherent ambiguity to infer 3D pose from its 2d projection, such as silhouette representation.

2.3 Summary

The problem to estimate 3D human pose from image sequences has attracted computer vision researchers since the 90s. As we can see from the existing methods, the main research themes include: pose estimation accuracy, pose estimation robustness due to self-occlusion between body parts, pose estimation ambiguity due to projection, and pose estimation efficiency.

The existing methods are not without limitations. 3D pose estimation with model-based approaches can achieve high estimation accuracy. But they often require manual initialization. Error accumulation or numerical drift can occur, and eventually cause the failure of the tracker for the inputs from long sequences. 3D pose estimation with learning-based approaches provides a unified framework for automatic initialization and tracking. But they often obtain error-prone results on the novel inputs. They also have lower pose estimation accuracy than the model-based approaches.

Not many papers discuss how to recover from tracking error. Sigal et al [76, 75] integrate the bottom-up detection information during the pose tracking process. It generates the pose hypotheses from the detected body parts to recover from tracking failure. This approach does require multiple cameras in order to localize the 3D body parts. Demirdjian et al [24] generate the pose hypotheses by querying the pose database with parameter-sensitive-hashing algorithm. When parameter-sensitive-hashing algorithm provides the sound pose hypothesis, it is possible to recover from tracking failure.

As we can see from the above review of the existing methods, in terms of computational speed and estimation accuracy, pose estimation from depth camera is more

promising for interactive or real-time applications than others. This thesis explores the robust pose estimation from depth camera.

In addition, we observe that for Human Computer Interaction applications, the proposed method promises to be more robust against tracking failure, i.e. recovering from tracking failure whenever possible.

CHAPTER 3

DEPTH IMAGE ANALYSIS: BODY PART DETECTION, LABELING AND TRACKING

This chapter presents a method to detect, label and track the body parts for an articulated 2D human figure using depth images. In order to have a robust pose tracker, one of the crucial processing steps is to localize each visible limb. On the one hand, a visible limb can disappear suddenly due to various reasons. For example, an arm could go to the back of torso and is not visible during a motion. Or an arm could get close to, and merge to, the torso so that depth resolution is not high enough to detect the limb. Also it is possible that a visible limb could be occluded temporarily by another limb. On the other hand, a missing limb can reappear later at an arbitrary time. To cope with these uncertainties, we propose to infer self-occlusion status of each limb using both temporal information and spatial context information so as to not only handle the transient self-occlusion between limbs of the articulated human figure during the tracking, but also robustly detect and track the reappearing arms.

Section 3.1 presents the general background on Bayesian theory for image-based detection, labeling, and tracking. Section 3.2 presents a novel algorithm to perform the robust tracking of human body parts from depth images. The experimental results are presented in Section 3.3, and Section 3.4 summarizes this chapter.

3.1 Bayesian Technique for Image Based Detection, Labeling and Tracking

Bayesian technique is a mathematical foundation for image based detection, labeling, and tracking. Image based detection and labeling is essentially based on the Bayesian decision theory by quantifying the tradeoffs between various classification decisions using the probability and the costs associated with such decisions [27]. Image-based tracking is essentially based on the parameter estimation, either maximal likelihood estimation or Bayesian estimation, to determine the optimal parameters based on the observations.

3.1.1 Bayesian Decision Theory for Image Based Detection

The image based detection problem is to decide if the image includes an instance of the object (hypothesis ω_1) or if the image has only background (hypothesis ω_0). Let λ_{ij} be the loss incurred for deciding ω_i when the true state is ω_j, let $p(I|\omega_i)$ be the conditional probability distribution of observing image I when the true state is ω_i, and let the $p(\omega_i)$ be the priori probability of state ω_i, then, the decision rule can be simplified to be: decide ω_1 if

$$\frac{p(I|\omega_1)}{p(I|\omega_0)} > \frac{(\lambda_{10} - \lambda_{00})}{(\lambda_{01} - \lambda_{11})} \frac{p(\omega_0)}{p(\omega_1)} \tag{3.1}$$

Furthermore, if we assume the uniform priori, the decision rule is simplified to be: decide ω_1 if

$$\frac{p(I|\omega_1)}{p(I|\omega_0)} > threshold \tag{3.2}$$

We shall use the above detection method to detect the head-neck-torso-waist from the image.

3.1.2 Bayesian Decision Theory for Image Based Labeling

In a similar manner, we can pose the image based labeling as a decision problem. Given a detected object O, let ω_i be the hypothesis that the object belongs to ith category, and let $p(O|\omega_i)$ be the conditional probability distribution of observing object O when the true label is ω_i. Then, we can have the following labeling rule: decide ω_1 if

$$\frac{p(O|\omega_1)}{p(O|\omega_0)} > threshold \tag{3.3}$$

We shall use such labeling method to decide the limb label for the detected limb pixels.

3.1.3 Parameter Estimation for Image Based Tracking

Let x_t denote the state of modeled object at time t, z_t denote the set of image observations at time t, then, image based tracking is formulated as online Bayesian parameter estimation:

$$p(x_t|z_1, z_2, \cdots, z_t) = \kappa_t p(z_t|x_t) p(x_t|z_1, z_2, \cdots, z_{t-1}) \tag{3.4}$$

where

$$p(x_t|z_1, z_2, \cdots, z_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|z_1, z_2, \cdots, z_{t-1}) \tag{3.5}$$

$p(x_t|x_{t-1})$ is the temporal dynamics of the object and κ_t is a normalization constant that does not depend on x_t.

When there is no temporal dynamics we can take advantage of or we want to estimate the parameters from one image frame, it is possible that we estimate the object state x_t directly based on:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}}\ p(z_t|x_t) p(x_t) \tag{3.6}$$

We use this to obtain the optimal estimation of the articulated limb parameters.

3.2 Depth Image Analysis Techniques for Tracking Human Body Parts

Robustly tracking human body parts is a challenging task for computer vision researchers. In fact, the existing model-based 2D human motion tracking approaches seldom discuss the recovery from the tracking failure. The approaches based on bottom-up image analysis are promising. For example, Mori et al [51] propose to locate the human body key points by matching the input image with exemplars using shape context descriptors. But its performance depends on the number of exemplars, which could be too large to be retrieved efficiently. Ramanan et al [60] propose to learn the discriminative limb appearance models by detecting the certain canonical poses across the image sequence, and further use them to predict the candidate limbs. But this method is not appropriate for online motion estimation. Other methods that are based on the incremental pose tracking [79, 33] between successive frames did not attempt to detect the body parts. Consequently they could not track such reappearing arm robustly because their tracking results depend on the initial values. Our proposed approach is also based on bottom-up depth image analysis. In particular, it has two effective components to cope with self-occlusion. Firstly, we take advantage of the depth information to infer the occlusion states of the body parts based on both temporal information and spatial context information; Secondly, we track the body parts in a hierarchical order based on their visibility. It is observed that head-neck-torso-waist is almost always visible. So, we find head-torso-waist first, and try to locate and track the limbs afterwards. Finally, our pose estimation with constrained closed loop inverse kinematics algorithm introduced in Chapter 4 is able to cope with

the missing body parts, and has a good tracking convergence property when the missing body parts are detected again.

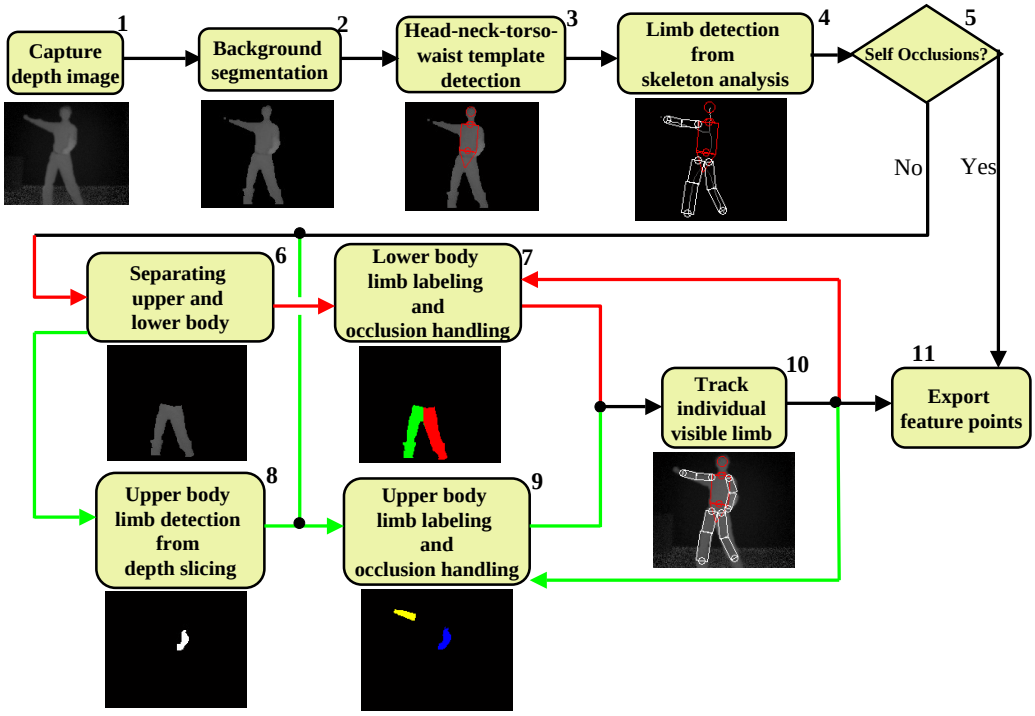


Figure 3.1: System overview: depth-image based body part detection, labeling and tracking

3.2.1 System Overview of Human Body Part Detection, Labeling and Tracking

The approach to detect and track the body parts from depth image is composed of several sequential steps from body part detection to body part labeling and tracking. Figure 3.1 lists these steps, which are briefly described below as an overview:

1. Capture the depth image of a subject

2. Segment the subject from the background, and generate the foreground image
3. Detect the head-neck-torso-waist template from the foreground image
4. Detect the limbs by analyzing the skeleton of the foreground image
5. Decide whether we find a T-pose frame from the previous step. If we find a T-pose frame, we shall export the feature points without performing the following steps
6. Generate the lower body image and the upper body image based on the detected head-neck-torso-waist template
7. Label the pixels in lower body image as left leg or right leg, and infer the occlusion status of each leg
8. Perform depth slicing to detect upper body limbs that are in front of the torso
9. Label the detected upper body limbs as left arm or right arm, and infer the occlusion status of each arm
10. Track each visible limb
11. Export the feature points for further processing

3.2.2 Background Segmentation

The background segmentation adopted here is different from the adaptive background segmentation techniques [80, 39] where the color distribution of each background pixel is modeled as a Gaussian mixture model. One major limitation of such an approach is that it requires the capture of a few background images offline in order

to learn the background model in advance. On the contrary, we take advantages of the depth information to perform background segmentation. Firstly, we set the depth volume of interest [0, Zmax), and any depth pixel outside of the depth volume is set to be background [Zmax,∞].

Secondly, for each depth pixel (i, j) inside the depth volume, let $[x_{ij}, y_{ij}, z_{ij}]$ be its coordinate in camera coordinate system. We estimate the normalized normal $\hat{n}_{ij} = [nx_{ij}, ny_{ij}, nz_{ij}]$ as:

$$n_{ij} = \begin{bmatrix} x_{i+1j} - x_{i-1j} \\ y_{i+1j} - y_{i-1j} \\ z_{i+1j} - z_{i-1j} \end{bmatrix} \times \begin{bmatrix} x_{ij+1} - x_{ij-1} \\ y_{ij+1} - y_{ij-1} \\ z_{ij+1} - z_{ij-1} \end{bmatrix} \tag{3.7}$$

and

$$\hat{n}_{ij} = \begin{bmatrix} nx_{ij} \\ ny_{ij} \\ nz_{ij} \end{bmatrix} = \frac{n_{ij}}{\|n_{ij}\|} \tag{3.8}$$

We set a depth pixel to be background if its normal deviates from up vector $u = [0, 1, 0]$ more than a certain angle:

$$ny_{ij} \geq \text{thr}_{\text{ang}} \tag{3.9}$$

Finally, we remove the noise pixels in the background by one iteration of morphological open operations, and fill the small holes in the foreground.

We have tested our background segmentation method on a large number test videos captured with CSEM SR3000 depth camera [83]. In addition to the fact that our described background segmentation does not need to capture the background frames, it obtains better feet-floor segmentation than adaptive background segmentation does.

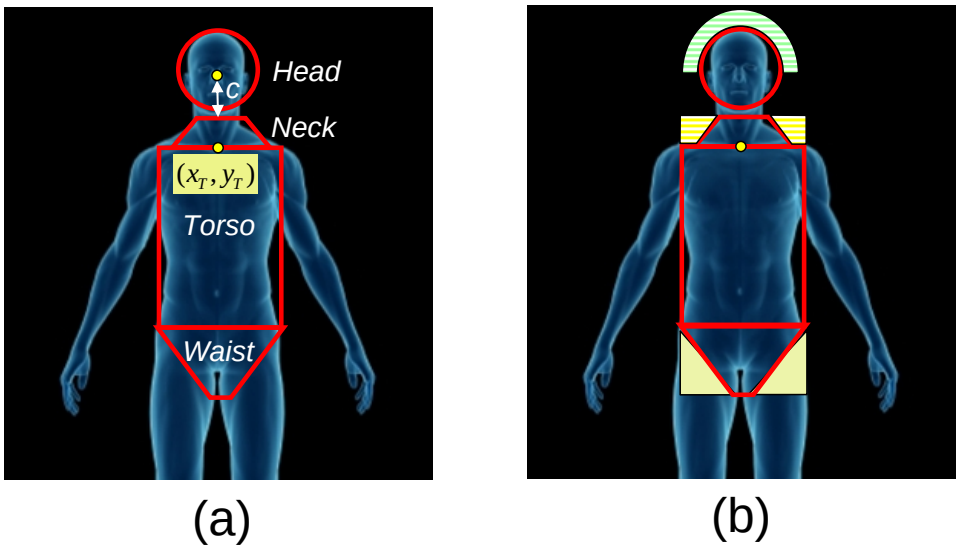


Figure 3.2: (a) Head-Neck-Trunk deformable template, where we have lumped the torso and waist into the trunk template. (b) its region-based observation model.

3.2.3 Head, Neck, and Trunk Detection

We make use of a deformable template which we refer to as *HNT Template* which consists of the head, neck, and trunk. The trunk is further decomposed into torso and waist. They are represented by a circle, trapezoid, rectangle, and another trapezoid, respectively, as shown in Figure 3.2(a). The head circle template is parameterized by $H = \{x_{H0}, y_{H0}, r_0\}$, where r_o represents the radius of the head circle template and (x_{H0}, y_{H0}) are the head center coordinates. The torso is represented as a rectangular box with parameters $T = \{x_T, y_T, w_T, h_T, \alpha\}$, where w_T and h_T represent the width and height of the torso box, respectively, α describes the inclination angle of the torso in the image plane relative to the upright posture, and (x_T, y_T) are the image coordinates at the midpoint of the top edge in the torso box. The neck template is represented as a trapezoid, rigidly attached to the torso box as shown in Figure 3.2(a).

The neck trapezoid is parameterized by $N = \{x_T, y_T, w_{N_1}, w_{N_2}, h_N, \alpha\}$, where w_{N_1} and w_{N_2} correspond to the width of the upper and lower trapezoid edges, and h_N is the height of neck. The waist trapezoid, connected to the torso box with a rotation joint, is parameterized by $W = \{x_W, y_W, w_{W_1}, w_{W_2}, h_W, \beta\}$. The position of waist joint (x_W, y_W) is located at the midpoint of the bottom edge in the torso box. w_{W_1}, w_{W_2} and h_W are the top width, bottom width, and height of waist trapezoid. β describes the rotation angle of waist relative to torso. We obtain relative edge lengths of the template from anthropometric studies reported in the biomechanics literature [92], which reports body segment measurements as a fraction of the total body height.

To detect and localize the defined *HNT Template*, let $L = \{H, N, T, W\}$ denote a configuration of the *HNT Template*, that localizes the head circle, neck trapezoid, torso rectangle, and waist trapezoid. Let θ be a set of distribution parameters used to define the *HNT Template*,

$$\theta = \{\lambda_1, \cdots, \lambda_7, (\mu_1, \sigma_1), \cdots, (\mu_4, \sigma_4)\} \tag{3.10}$$

As will be described shortly, λ corresponds to parameters in a function describing the likelihood of detecting the *HNT Template* and (μ, σ) are the mean and standard deviations in the associated prior distribution functions. These parameters are learned by collecting training examples from image sequences and distribution functions given below. Our algorithm takes a depth image I as input and produces an optimal pose configuration L. To evaluate output L, we compute $P(I|L, \theta)$, that is, the likelihood that image I would have been generated, given configuration L and assumption θ. By Bayes' rule, we have

$$P(L|I, \theta) \propto P(I|L, \theta)\ P(L|\theta) \tag{3.11}$$

where $P(L|\theta)$ is the prior probability of the HNT configuration. Assuming the image likelihood functions for the parameters are independent, we obtain

$$P(I|L,\theta) = P(I|H)P(I|N)P(I|T)P(I|W) \tag{3.12}$$

The prior distribution over H, N, T, and W includes:

$$P(L|\theta) = P(r_0|\theta)\ P(w_T|\theta)\ P(h_T|\theta)\ P(c|\theta) \tag{3.13}$$

where c is the distance from the head center to top edge midpoint of the neck trapezoid. Then the output configuration L is either accepted or rejected based on the following criterion imposed on the likelihood function,

$$L(H,N,T,W) \ \text{ is} \begin{cases} \text{accepted if } \log(P(L|I,\theta)) > thr \\ \text{rejected otherwise} \end{cases} \tag{3.14}$$

where the threshold thr is determined empirically during training by computing the likelihood function L for several hundred frames and observing trunk detection results.

For the head likelihood function, distribution function $P(I|H) = e^{-\lambda_1 N_{10H} - \lambda_2 N_{01H}}$ is used, where N_{10H} and N_{01H} represent the numbers of false negative (background pixels in the head circle) and false positive (foreground pixels in the buffered head boundary represented by a green striped region above the head in Figure 3.2(b)), respectively. Similarly, for neck likelihood function, we use distribution function $P(I|N) = e^{-\lambda_3 N_{10N} - \lambda_4 N_{01N}}$, where N_{10N} is the number of background pixels in the neck trapezoid, and N_{01N} is the number of foreground pixels in the buffered neck boundary (yellow striped region on the right and left side of the neck template in Figure 3.2(b)). For the torso likelihood function, we use distribution function $P(I|T) = e^{-\lambda_5 N_{10T}}$, where N_{10T} is the number of background pixels in the torso box. Note that false positive pixels are not considered since arms frequently occlude the torso box. For the

waist likelihood function, we use distribution function $P(I|W) = e^{-\lambda_6 N_{10W} - \lambda_7 N_{01W}}$, where N_{10W} is the number of background pixels in the waist trapezoid, and N_{01W} is the number of background pixels in the buffered waist boundary (blue striped region on the right and left side of the waist template in Figure 3.2(b)). Finally, without loss of generality, prior distribution functions are assumed to be Gaussian (η) with mean μ and standard deviation σ, (i.e. $(\eta(\mu, \sigma))$.

$$P(r_0|\theta) = \eta(\mu_1, \sigma_1)$$
$$P(w_T|\theta) = \eta(\mu_2, \sigma_2)$$
$$P(h_T|\theta) = \eta(\mu_3, \sigma_3)$$
$$P(c|\theta) = \eta(\mu_4, \sigma_4)$$

We initialize the *HNT* configuration with the following steps that allows us to localize the initial optimal *HNT*. The center of gravity (COG) is found, and the crotch point (crotch line afterward), defined as the lowest waist point (waist line) is located. We scan a few oriented line segments that pass the COG and are located inside the foreground. Torso width and orientation are estimated from the oriented line segment having the minimal length. COG and torso orientation together define the major torso axis. Shoulder line is then localized by scanning a line segment, perpendicular to the major axis, upwards starting from COG until finding the one whose length is shorter than 90 percenter of the measured torso width. With these overall trunk information, the neck template is initialized at the middle of shoulder line, and its orientation is defined based on torso major axis. The head template is initialized based on the the relative distance to the neck. The torso and waist template are initialized based on their relative heights to the trunk height.

During the tracking, we generate the *HNT* hypotheses both from the above described method and locally sampled *HNT* configurations based on the tracked configuration from the last frame. Figure 3.3 illustrates detected and localized *HNT* template results, and shows the effectiveness of the proposed algorithm in deforming the template and matching with the image observations.

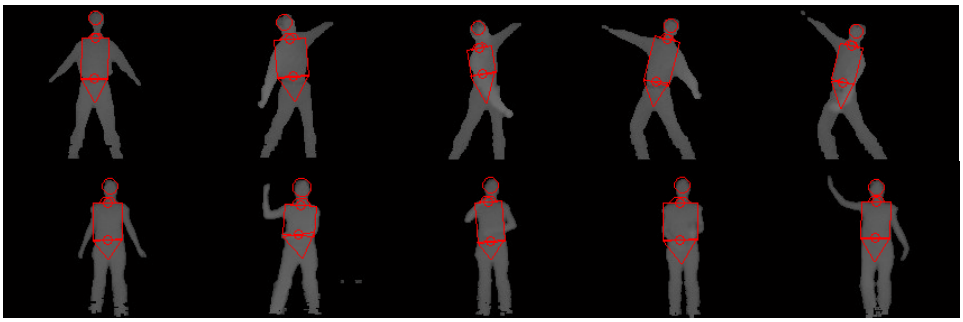


Figure 3.3: Detected and localized *HNT Template*.

3.2.4 Limb Detection from Skeleton Analysis

Once the head, neck, and trunk are detected, limbs (two arms and two legs) are to be detected. There is a class of poses which we refer to as *non-occluding configuration* (NOC for short), where there are no self occlusions and a simple and robust detection algorithm exists. In a NOC configuration, the skeleton image and its distance transformed image are sufficient to detect and localize all limbs. An example of two configurations where there are no self occlusions is illustrated in Figure 3.4. From foreground image I, the skeleton image I_S is generated. For NOC, I_S contains five major end-points, corresponding to two feet, two hands, and the head. Then, distance transform is performed on I_S to generate I_D, as shown in Figure 3.4(c). The

start point p_S is chosen to be the point that is nearest to the center of gravity of I (red dot in Figure 3.4(c)).

Refereing to Figure 3.4(a), the foreground image I with the detected HNT is illustrated for two different non-occluding configurations. The skeleton image I_S is shown in Figure 3.4(b). In principle, the skeleton image of a non-occluding configuration should contain five end-points, corresponding to two feet, two hands, and the head. In reality, certain poses may present artifacts or ambiguities in the skeleton image, where it is difficult to discern if an observed end-point is in fact an actual end-point. To address this issue, we process the skeleton image further and create distance transformed image I_D of the skeleton image as shown in Figure 3.4(c). End-points must have distance values larger than a minimum value which is determined from anthropometric data. This constraint is used to eliminate spurious artifacts (wrong candidates for limbs). The head end-point is actually localized from the center of the head template. The identified end-points are depicted in Figure 3.4(d).

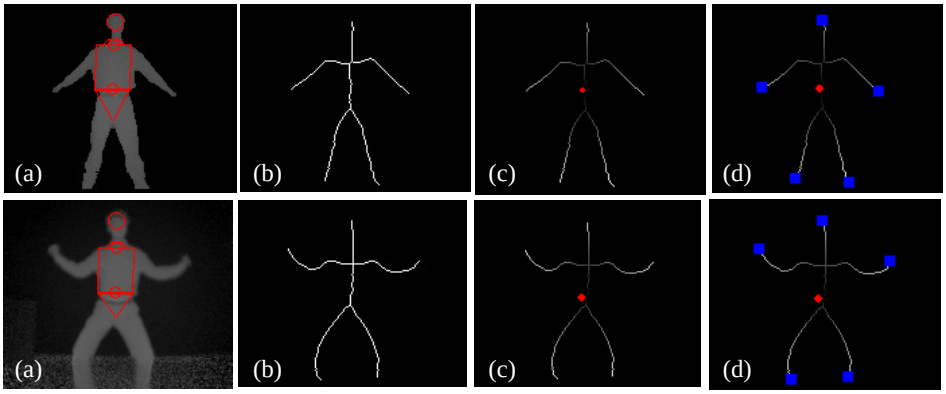


Figure 3.4: Procedure for detecting a configuration with no self occlusions (Non-occluded configuration) (a) Foreground image and detected head-neck-torso-waist template; (b) Skeleton image of foreground image; (c) Distance transformed skeleton image; (d) Detected end-points

3.2.5 Test for Self Occlusions

The next step is to determine whether the current pose has self occlusions or not. Two conditions should be satisfied for a given configuration to be categorized as non-occluded. First, as we described above, the procedure depicted in Figure 3.4 must produce the four visible end-points corresponding to the hands and feet [1]. Furthermore, we require that the distance from the hand end-points to the entry point of the torso template, and the distance from the feet end-points to the entry point of the waist template be greater than a minimum distance. This minimum distance is obtained from anthropometric data of arm and leg lengths. If the above conditions are satisfied, we have succeeded in labeling this configuration as a non-occluded configuration and the key-points can be localized as described below.

The left and right wrist key-points and the left and right ankle key-points correspond to adjusted coordinates of the four corresponding end-points, where the adjustments reflect the displacement distance from the end-point to the actual key-point. The head center key-point is also localized as the center of the head template. The shoulder and hip key-points correspond to the entry points to the torso and waist templates, respectively, along the distance transformed image. The elbow and knee key-points are localized from the intersection points of the upper arm and lower arm, and upper leg and lower leg, respectively. Finally, the waist key-point is simply localized at the origin of the waist template. In summary, the skeleton analysis is sufficient to localize all key-points when there are no self occlusions (non-occluded configuration) and no further processing is required as shown in Figure 3.1.

[1] One exception is when there is loop closure at the end-points, such as when the hand are over the head and touch each other. There are no self occlusions in this case, yet the hand end-point may not be detected as described in the previous section.

3.2.6 Separating Upper and Lower Body

Human body configurations frequently exhibit self occlusions which present challenges that cannot be addressed by skeleton analysis alone. If there are self occlusions, further processing is required to detect limbs and the associated key-points. Considering limited interaction between upper and lower body limbs, we treat the detection, tracking, and labeling of the upper-body and lower body limbs independently. Therefore, the first step in handling self occluded configurations is to separate the upper and lower body limbs, as illustrated in Figure 3.1.

In subsequent sections, we present how to deal with self occlusions and how to exploit temporal information and spatial context information to resolve ambiguities and provide a solution to the labeling and data association problem [5].

3.2.7 Limb Detection, Labeling and Tracking

We now describe how to identify limbs and label them for general configurations (including occlusions). We discuss the upper-body first, while the lower body is identified by using a similar (but simpler) algorithm as described next. In the following we use I_{Arms} to represent the image region corresponding to arms. For an occluded configuration, three steps are taken in identifying I_{Arms} : (1) open arm detection on skeleton; (2) loop detection on skeleton; and (3) depth slicing. These steps correspond to cases that the arm is found beside the trunk, the arm is forming a loop, and the arm is found in front of the trunk.

Let us start with open arm detection. This step is similar to the one in Section 3.2.4. A segment $p_i q_i$ is detected where p_i is the end-point and q_i is the entry-point that is connected to the trunk. If the length of segment $p_i q_i$ is longer than a

threshold and entry-point q_i is within the shoulder region of the trunk, an arm is recognized as an arm (See examples in Figure 3.5). At this present, the label of the detected arm remains undefined since it is not obvious which arm p_iq_i corresponds to.

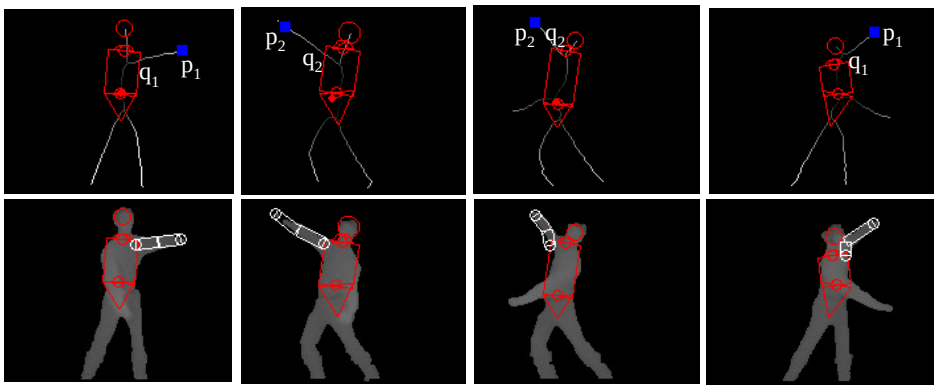


Figure 3.5: Open arm detection from I_D.

Loop detection is done by using distance transformed skeleton image I_D as illustrated in Figure 3.6. A loop is detected when a local maximum non-terminal pixel is found in I_D. In Figure 3.6, q_i represents the endpoint of the arc that is directly connected to the shoulder area of the trunk, while s_i represents the other endpoint of the arc. We see that arc $q_ir_is_i$ corresponds to an arm and that point r_i closely approximates the elbow of the corresponding limb, where r_i is the extreme point in the x axis. After detecting the loop, its label (left arm or right arm) is determined based on which side of the body the loop is located. After the first two steps, the number of detected arms may be still less than two. In such a case, we use depth slicing to detect possible arm regions which occlude the torso. A slicing plane is moved along

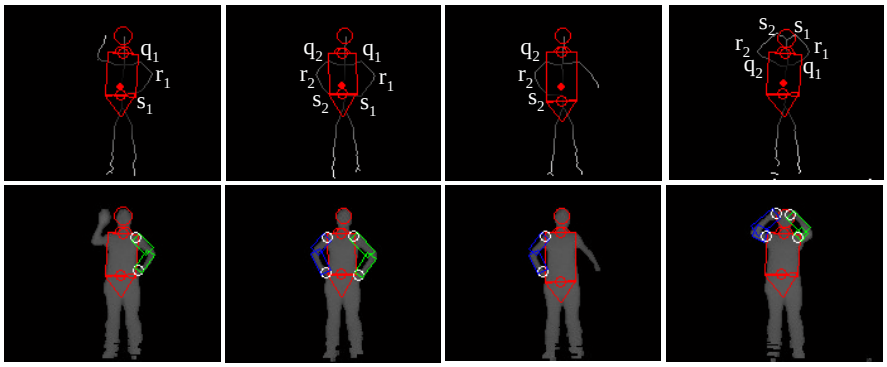


Figure 3.6: Looped arm detection, labeling from I_D.

the depth axis from the camera toward the human subject. Connected components are extracted for regions whose depth values are closer to the camera than the slicing plane. The slicing continues until connected blobs becomes too large to be an arm. A few examples from this operation are shown in Figure 3.7. After detecting region

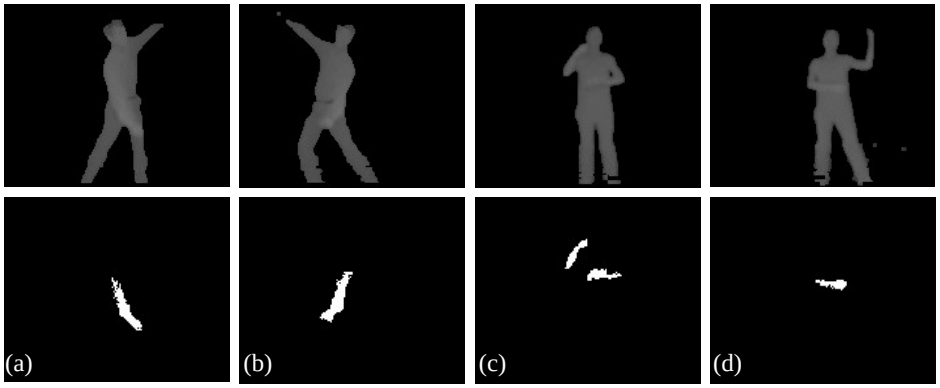


Figure 3.7: Arm blob detection examples from depth slicing

I_{Arms}, the remaining task is to assign labels to the region. We integrate both temporal

tracking and spatial context with a two-slice Bayesian network to perform labeling and occlusion inference.

Let's first define some variables and notation used in the remainder of the paper. For brevity, we use the notation LA, RA, LL, and RL associated with variables in the left arm, right arm, left leg, and right leg, respectively. Let R_1 and R_2 be two regions that are detected by Step 1 (i.e., candidates for open-arms), let R_3 and R_4 be regions detected by Step 2 (i.e., loop detection), and let R_5 and R_6 be regions detected by depth slicing. We have at most two regions from R_i (i= 1, $\cdots$, 6). From the previous discussion, we know that correct labeling can be easily assigned to depth pixels belonging to R_3 and R_4 so we discuss the case for R_1, R_2, R_5 and R_6, namely, we have I_{Arms} as the intersection of these regions,

$$I_{\text{Arms}} = \{I_x | x \in R_1 \cup R_2 \cup R_5 \cup R_6\}, \tag{3.15}$$

where I_x is the depth value in the image at the location of pixel x. Our task is to assign for each pixel x in I_{Arms} a label L_x, where $L_x \in \{\text{LA, RA}\}$.

Each articulated limb is represented by a 2-dimensional (projected) human model with parameters $\Phi = \{(x_0, y_0), \alpha, \beta, l_1, l_2, w_1\}$, where (x_0, y_0) is the shoulder location, α is the shoulder joint rotation angle, β is the elbow joint rotation angle, l_1 is the upper arm length, l_2 is the lower arm length, and w_1 is the width of the limb. Let us use X_{LA} and X_{RA} to represent the configuration representing articulated left and right arm, respectively. Two auxiliary variables are used in our computation. The first one is *occlusion state*, O_{LA} for the left arm and O_{RA} for the right arm. These are decided from the resulting labeling image L_{Arms}. Secondly, let us use H_{LA} and H_{RA} to be the histogram of depth values for pixels representing the left and right arms, respectively. This is used to characterize the left and right arm during tracking.

Figure 3.8 contains the diagram for the Bayesian network for arm labeling and occlusion inference. The flow of inference is as follows. Using pose information from time slice $t-1$ stored in X_{LA}^{t-1} (or X_{RA}^{t-1}), a prediction of pose for time slice t denoted as $\tilde{X}_{LA}^{t}$ ($\tilde{X}_{RA}^{t}$) is generated. Based on $\tilde{X}_{LA}^{t}$ ($\tilde{X}_{RA}^{t}$) and O_{LA}^{t-1} (O_{RA}^{t-1}), each pixel x is associated with a probability $P(L_x)$ representing how likely x belongs to left (right) arm at time t, given I_{Arm} for time slice t. Pixel x is assigned an appropriate label by using $P(L_x)$. Then, X_{LA}^{t} (X_{RA}^{t}) is optimally computed by using the set of pixels whose labels are LA (RA). Finally, $P(L_x)$ is used to determine O^t.

Let us use X_{HNT} to represent the Head, Neck, Trunk configuration as determined by the localization method as described in Section 3.2.3. The location of the torso in X_{HNT} determines the location of shoulder with Gaussian noise. The conditional probability of label assignment is defined based on the occlusion states from the last frame as well as on tracked arm locations and detected arm locations:

$$P(L_x^t = \mathrm{LA}|O_{\mathrm{LA}}^{t-1}, O_{\mathrm{RA}}^{t-1}, X_{\mathrm{LA}}^t, H_{\mathrm{LA}}^t, X_{\mathrm{RA}}^t, H_{\mathrm{RA}}^t, R_1^t, R_2^t, R_5^t, R_6^t) =$$

$$\begin{cases} P_1(L_x^t = \mathrm{LA}|X_{\mathrm{LA}}^t, H_{\mathrm{LA}}^t, X_{\mathrm{RA}}^t, H_{\mathrm{RA}}^t) & \text{if } O_{\mathrm{LA}}^{t-1} = 0, O_{\mathrm{RA}}^{t-1} = 0 \\ P_2(L_x^t = \mathrm{LA}|R_1^t, R_2^t, R_5^t, R_6^t) & \text{if } O_{\mathrm{LA}}^{t-1} = 1, O_{\mathrm{RA}}^{t-1} = 1 \\ P_3(L_x^t = \mathrm{LA}|X_{\mathrm{LA}}^t, H_{\mathrm{LA}}^t, R_1^t, R_2^t, R_5^t, R_6^t) & \text{if } O_{\mathrm{LA}}^{t-1} = 0, O_{\mathrm{RA}}^{t-1} = 1 \\ P_4(L_x^t = \mathrm{LA}|X_{\mathrm{RA}}^t, H_{\mathrm{RA}}^t, R_1^t, R_2^t, R_5^t, R_6^t) & \text{if } O_{\mathrm{LA}}^{t-1} = 1, O_{\mathrm{RA}}^{t-1} = 0 \end{cases} \tag{3.16}$$

$P_1(L_x^t = \mathrm{LA}|\cdot)$ is determined by temporal tracking information based on geometric distance and depth histograms as:

$$P_1(L_x^t = \mathrm{LA}|X_{\mathrm{LA}}^t, H_{\mathrm{LA}}^t, X_{\mathrm{RA}}^t, H_{\mathrm{RA}}^t) = \frac{e^{-\gamma d_{\mathrm{LA}}(x)} H_{\mathrm{LA}}(I_x)}{e^{-\gamma d_{\mathrm{LA}}(x)} H_{\mathrm{LA}}(I_x) + e^{-\gamma d_{\mathrm{RA}}(x)} H_{\mathrm{RA}}(I_x)} \tag{3.17}$$

where $d_{\mathrm{LA}}(x)$ is the distance from x to the left arm:

$$d_{\mathrm{LA}(x)} = \begin{cases} 0 & \text{if } x \text{ is inside left arm} \\ d(x, \mathrm{LA}) & \text{otherwise} \end{cases} \tag{3.18}$$

where $d(x, \mathrm{LA})$ is the minimal distance from x to edges of the left arm. $d_{\mathrm{RA}}(x)$ is defined similarly. In short, a pixel x has a high probability of belonging to LA, if x is sufficiently close to where LA was in the previous frame. In case two arms are overlapping in the image, x has a high probability of belonging to LA if it has a depth value that is close to one of depth values represented by the left arm in the previous frame.

$P_2(L_x^t = \mathrm{LA}|\cdot)$ is determined by spatial context information when there is no temporal tracking information. This case occurs when the arms are behind the torso, or when they are close to the torso.

Assuming $x \in R_i^t$, we define the fraction of number of non-zero pixels in the overlapping regions of the detected arm (R_i^t) and left half torso ($R_{X_{\mathrm{LHT}}^t}$) with respect to the number of non-zero pixels in the overlapping regions of the detected arm and the torso $R_{X_{\mathrm{T}}^t}$, by

$$f_{x,\mathrm{Left}}^t = \frac{\#\{R_i^t \cap R_{X_{\mathrm{LHT}}^t}\}}{\#\{R_i^t \cap R_{X_{\mathrm{T}}^t}\}}, \tag{3.19}$$

where the notation $\#\{.\}$ represents the number of nonzero pixels in the region defined inside the parenthesis.

The spatial distribution of left arm and right arm given the torso position is given as:

$$P_2(L_x^t = \mathrm{LA}|\cdot) = \frac{f_{x,\mathrm{Left}}^t}{f_{x,\mathrm{Left}}^t + f_{x,\mathrm{Right}}^t} \tag{3.20}$$

$P_3(L_x^t = \mathrm{LA}|\cdot)$ and $P_4(L_x^t = \mathrm{LA}|\cdot)$ are determined by hybrid temporal and spatial information. Assuming that a tracked arm does not move fast between successive frames, the conditional probability is defined based on the overlapping area between enlarged tracked arm and detected arms. Let $R_{X_{LA}^t}$ be the enlarged left arm region, and $x \in R_i^t$. The conditional probability of labeling, when the right arm is occluded

and the left arm is tracked in the last frame, is given as:

$$P_3(L_x^t = \text{LA}|\cdot) = \frac{\#\{R_i^t \cap R_{X_{LA}^t}\}}{\#\{R_i^t\}} \tag{3.21}$$

$P_4(L_x^t = \text{LA}|\cdot)$ is defined similarly. The conditional probability for visible states of the arms is given as:

$$P(O_{\text{LA}}^t = 0|L_x^t, x \in I_{\text{Arms}}^t) = \frac{\sum_x \mathbf{1}_{(L_x^t=\text{LA})}}{A_{\text{LA}}} \tag{3.22}$$

and

$$P(O_{\text{RA}}^t = 0|L_x^t, x \in I_{\text{Arms}}^t) = \frac{\sum_x \mathbf{1}_{(L_x^t=\text{RA})}}{A_{\text{RA}}} \tag{3.23}$$

where $\mathbf{1}_{(condition)}$ is the indicator function with $\mathbf{1}_{(condition)} = 1$ if *condition* is true and $\mathbf{1}_{(condition)} = 0$ if *condition* is false. The parameters A_{LA} and A_{RA} are the projection of predicted left arm and right arm areas, respectively. Finally, based on labeling

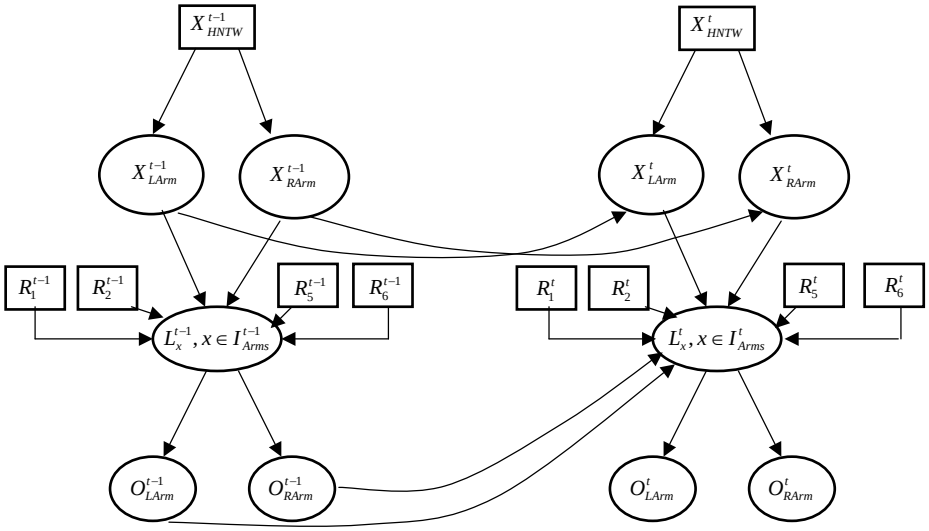


Figure 3.8: Arm labeling and occlusion inference with two-slice Bayesian network.

and occlusion states of upper-body limbs, a local optimization is performed for each visible arm in order to report the optimal estimation of the limb pose.

$$\hat{X}^t_{\mathrm{LA}} = \arg\min_{\alpha,\beta,l_1,l_2}\{E_{\mathrm{LA}(\alpha,\beta,l_1,l_2)}\} \tag{3.24}$$

and

$$E_{\mathrm{LA}}(\alpha,\beta,l_1,l_2) = N_{01A} + N_{10A} + w * |l_1 - l_2| \tag{3.25}$$

where N_{01A} is the number of pixels located inside the left arm template X_{LA} whose labels are not LA, N_{10A} is the number of pixels located outside the left arm template X_{LA} whose labels are LA, and w is a weight that is selected based on experiments.

Labeling for the lower body limbs is done in a manner similar to the one for the upper-body. See the corresponding two-slice Bayesian network in Figure 3.9. The location of the waist from the observed X_{HNT} determines the location of left and right pelvis with Gaussian noise. The conditional probability of label assignment is defined as:

$$P(L^t_x = \mathrm{LL}|X^t_{\mathrm{LL}}, H^t_{\mathrm{LL}}, X^t_{\mathrm{RL}}, H^t_{\mathrm{RL}}) = \frac{e^{-\gamma d_{\mathrm{LL}}(x)} H_{\mathrm{LL}}(I_x)}{e^{-\gamma d_{\mathrm{LL}}(x)} H_{\mathrm{LL}}(I_x) + e^{-\gamma d_{\mathrm{RL}}(x)} H_{\mathrm{RL}}(I_x)} \tag{3.26}$$

where $d_{\mathrm{LL}}(x)$ is the distance from x to left leg location:

$$d_{\mathrm{LL}(x)} = \begin{cases} 0 & \text{if } x \text{ is inside left leg} \\ d(x,\mathrm{LL}) & \text{otherwise} \end{cases} \tag{3.27}$$

where $d(x,\mathrm{LL})$ is the minimal distance from x to edges of left leg. $d_{\mathrm{RL}}(x)$ is defined similarly. The rest of the process is the same as that for the upper-body. In case of the lower-body labeling, the formulation is simpler than that of the upper-body due to the fact that legs do not occlude the trunk for poses that we are considering.

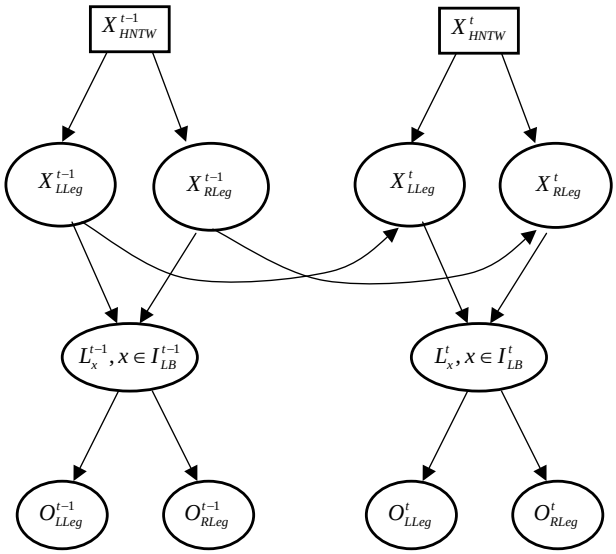


Figure 3.9: Leg labeling and occlusion inference with two-slice Bayesian network.

3.3 Experimental Results

Experiments were performed using a single time-of-flight range sensor [83]. The human performer was asked to perform various actions such as T-pose motion, crossing leg motion, side walking motion, and dancing motion etc. Figure 3.10, 3.11 and 3.12 show snapshots of the tracked whole body articulated figures. Interestingly, we also could apply the similar computation to only estimate the upper body motion as shown in Figure 3.13, 3.14, and 3.15.

As shown, the implemented feature tracker is able to track the feature points through the transient occlusions between left and right leg as in Figure 3.11, as well as the transient occlusions between left and right arm as in Figure 3.15.

The arm detection is able to detect the reappearing arms as shown in the last two images of Figure 3.14 and among other demo videos. We are able to make a correct arm labeling decision based on the spatial and temporal context information as shown in the Figures.

Various occlusion scenarios are shown in the figures as well. As we can see, our head-neck-torso detection is robust when arm occludes head as in Figure 3.13, 3.14, and 3.15. Left and right arm labeling is robust when they interact with each other as in Figure 3.15.

To evaluate the feature point tracking accuracy, we manually locate the feature positions (left of Figure 3.16) on the depth image, and read the 3D positions from camera data. We call this as manual trajectory, and assume it as the ground truth. During the feature tracking, we export the feature positions from 2D feature tracking results (right of Figure 3.16), and read the 3D positions from camera data. We call this as raw trajectory. We evaluate the 2D tracking accuracy by comparing raw trajectory with manual trajectory. Table 3.1 summarizes the feature point tracking errors for each feature point obtained from the KF2 sequence. As shown, the limb end feature points (hand or ankle) have the highest error because (1) the depth sensor is noisy at the object boundary; (2) the segmentation around the foot-floor contact is noisy due to the fact there is not enough contrast to differentiate feet from floor. Even though there exists a bias between the manually localized and observed feature points, our feature tracking has an overall accuracy around 6cm.

3.4 Summary

In this chapter, we have proposed a novel 2D articulated figure tracking from depth image sequences. Firstly, we have a robust procedure to detect the T-pose frames. Secondly, both temporal tracking information and spatial context information are utilized to label the limb pixels for non-T-pose frames. Occlusion state of each individual limb is further inferred from the labeling results, and visible limbs are tracked between successive frames.

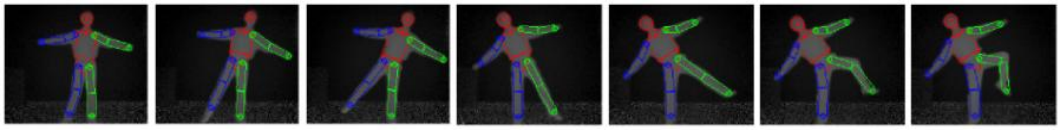


Figure 3.10: Whole body articulated figure tracking for T-pose motion. The t-pose detection is robust for various cases

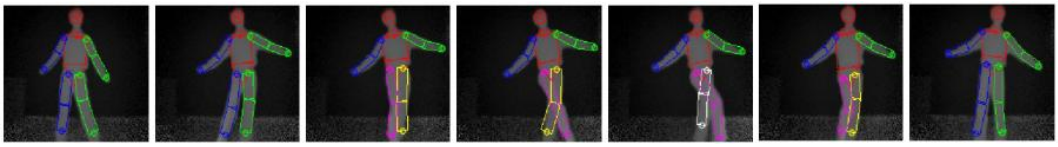


Figure 3.11: Whole body articulated figure tracking for crossing leg motion. The occlusion inference for left and right leg results in the correct tracking of the leg through transient occlusions

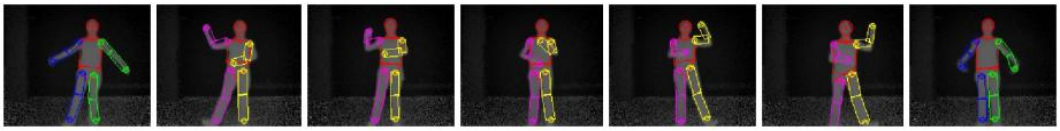


Figure 3.12: Whole body articulated figure tracking for dancing motion. Depth slicing is able to detect the limbs in front of the torso

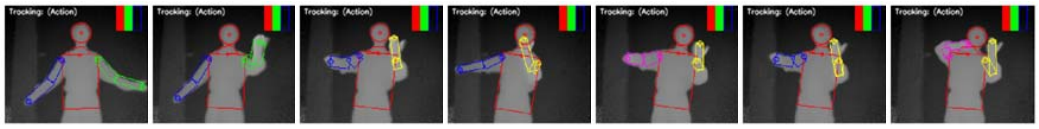


Figure 3.13: Upper body articulated figure tracking for violin-playing motion. The head-neck-torso detection is robust even when head is partially occluded.

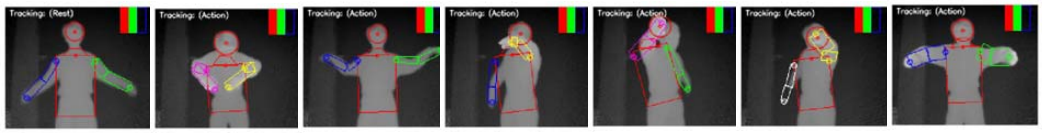


Figure 3.14: Upper body articulated figure tracking for swimming motion. We are able detect and label the reappearing limbs as shown in the last two images.

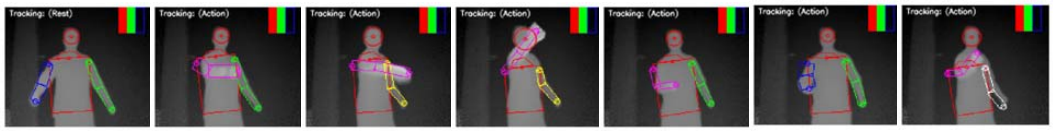


Figure 3.15: Upper body articulated figure tracking for frisbee-throwing motion. We are able to label and track the interactive left and right arm correctly based on temporal and spatial information

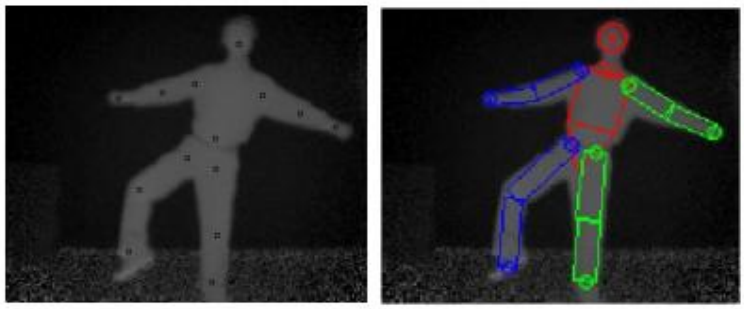


Figure 3.16: Left: feature points from manual localization; Right: feature points from tracking.

FeaturePoint	Error (in millimeter)		
	X(mean,std)	Y(mean,std)	Z(mean,std)
RightHand	(-28,43)	(-22,33)	(-109,51)
RightElbow	(-43,46)	(6,36)	(-28,11)
RightShoulder	(-32,47)	(54,31)	(36,41)
Waist	(7,30)	(16,39)	(11,29)
LeftHand	(20,38)	(-22,44)	(-92,44)
LeftElbow	(90,48)	(10,42)	(-21,15)
LeftShoulder	(22,43)	(25,33)	(8,26)
Head	(1,36)	(9,17)	(1,28)
RightPelvis	(-12,45)	(58,33)	(14,29)
RightKnee	(-1,25)	(35,83)	(-4,25)
RightAnkle	(16,45)	(-42,72)	(-6,71)
LeftPelvis	(9,50)	(69,39)	(12,37)
LeftKnee	(21,29)	(63,71)	(5,24)
LeftAnkle	(-21,48)	(-26,49)	(-4,41)
Overall	(4,52)	(17,59)	(-13,55)

Table 3.1: Raw trajectory position error for KF2 motion sequence.

CHAPTER 4

POSE ESTIMATION WITH LOW-DIMENSION FEATURE POINTS

In this chapter, we describe a novel algorithm to reconstruct human motion from a set of feature points possibly observed from depth image sequences. Section 4.1 presents the overall procedure to perform pose estimation from a set of feature points with constrained closed loop inverse kinematics (C-CLIK). Section 4.2 shortly describes the set of feature points to be used in this chapter. Section 4.3 presents the closed loop inverse kinematics to estimate the poses as joint motions from the feature point trajectories. Section 4.4 presents a method to enforce the joint limits during the pose estimation. Appendix A presents a method to enforce the collision avoidance between body parts. The experimental results are presented in Section 4.5, and Section 4.6 summarizes this chapter.

4.1 Overview of Pose Estimation with Low-dimension Feature Points

Figure 4.1 illustrates the different modules in the computational pipeline. Our algorithm reconstructs human pose from a possible k features, corresponding to 3D positions of prominent anatomical landmarks on the body. Without loss of generality, we consider fourteen ($k = 14$) such whole body features as illustrated in Figure 4.2.

The depth images are used as input to a visual processing module which detects m ($m = 0 \cdots k$) whole body features, denoted by p_{det}, at approximately 6-12 frames per second. Note that the number of detected features at each frame may be fewer than fourteen(i.e. $m < k = 14$) due to occlusions or unreliable observations. For numerical stability in subsequent modules, the detected features are re-sampled to a higher rate (usually 100 HZ) and represented by the vector $\bar{p}_{det}$.

Among the fourteen whole body features, those features which are undetected may be estimated using feedback from the prediction mechanism in a pose estimation module (feedback path 1 in Figure 4.1). If $m < k$, the detected features are augmented with $(k - m)$ predicted features (p) obtained from forward kinematics computations of the reconstructed pose. The augmented feature vector, denoted by p_d, represents the $k = 14$ desired features used as input to a pose estimation module. The recovered pose, parameterized by the vector q, describes the motion of the $n = 25$ degree of freedom whole body model.

The predicted features could also be fed-back to resolve ambiguities in case multiple candidates for a given feature are detected (false positives detected) or if a given feature is missing or intermittently occluded. This scenario corresponds to feedback path 2 in Figure 4.1.

For an n degree of freedom human model, the configuration space, or joint space, described here by the vector $q = [q_1, \cdots, q_n]^T$, fully characterizes the motion of the human model. The mapping between configuration space velocities and Cartesian space velocities is obtained by considering the differential kinematics relating the two spaces,

$$\dot{p}_i = J_i(q)\,\dot{q} \tag{4.1}$$

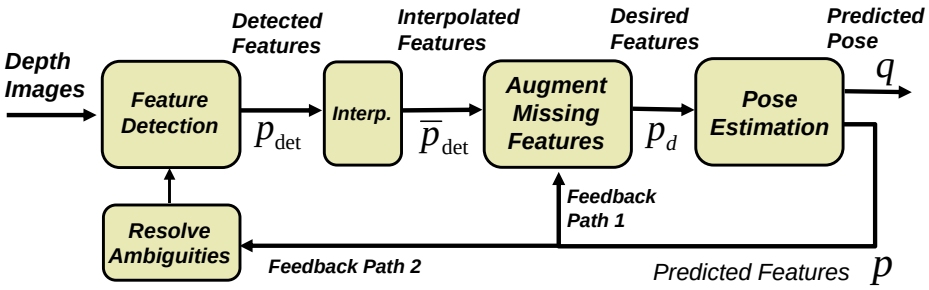


Figure 4.1: System diagram of the entire pipeline.

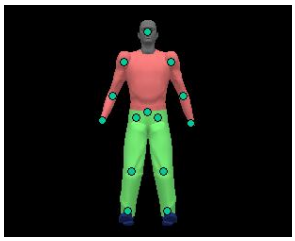


Figure 4.2: Whole body features used in experiments.

where $J_i \in \Re^{3\times n}$ is the Jacobian of the i_{th} feature [20] and $\dot{p}_i$ is the velocity of p_i.

4.2 Feature Detection

The feature detection follows the proposed algorithm as described in the last chapter. For each tracked frame of the 2D articulated figure, we determine a set of feature points by reading 3D positions from depth camera as shown in Figure 4.3.

4.3 Cartesian Tracking Control: CLIK

Cartesian tracking control refers to a control policy that produces the joint variables (q) such that the Cartesian errors between the estimated features and the desired

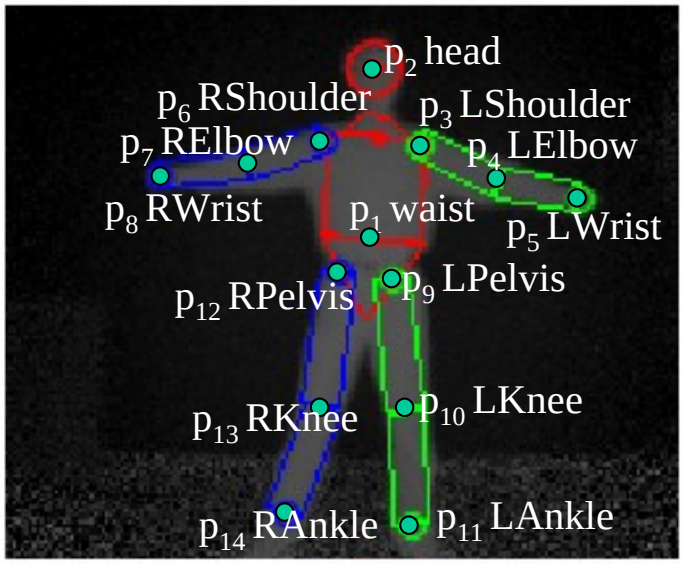


Figure 4.3: Whole body features points labeled along with the tracked 2D articulated figure

(from observations) features are minimized. The tracking performance is very much subject to the human model kinematic constraints as well as the execution of multiple and often conflicting feature tracking requirements. We employ a tracking control approach based on a Cartesian space kinematic control method known as closed loop inverse kinematics (CLIK). The basis for the solution of the CLIK algorithm is the inversion of the differential kinematics relating Cartesian variables and joint variables as described by Equation 4.1. For simplicity, we momentarily drop the superscript i with reference to the i_{th} feature.

Let the desired variables be denoted by a subscript d. The joint velocities may be computed by inverting Equation (4.1) and adding a feedback error term to correct

for numerical drift.

$$\dot{q} = J^*(\dot{p}_d + K\ e) \tag{4.2}$$

where J^* denotes the regularized right pseudo-inverse of J weighted by the positive definite matrix W_1,

$$J^* = W_1^{-1} J^T\ (J W_1^{-1} J^T + \lambda^2 I)^{-1} \tag{4.3}$$

The parameter $\lambda > 0$ is a damping term, and I is an identity matrix. The vector $\dot{p}_d$ corresponds to the desired feature velocity. The matrix K is a diagonal 3×3 positive definite gain matrix, and e is a vector that expresses the position error between the observed and computed features. The position error is simply defined as $e = p_d - p$, where p_d and p correspond to the observed and computed feature positions, respectively.

The formulation above considers estimation of human pose from a single feature. Multiple features can be handled in two ways, namely by *augmentation* or *prioritization*. These methods are described in detail in robot motion control literature [72]. In this thesis, we consider feature augmentation which refers to the concatenation of the individual spatial velocities and the associated Jacobian matrix and feedback gain matrix.

Let i $(i = 1 \cdots k)$ be the index of the i_{th} feature $\dot{p}_i$ and the associated Jacobian J_i. We form a $3k \times 1$ augmented spatial velocity vector $\dot{p}$ and a $3k \times n$ augmented Jacobian matrix J as follows,

$$\dot{p} = \begin{bmatrix} \dot{p}_1^T & \cdots & \dot{p}_i^T & \cdots & \dot{p}_k^T \end{bmatrix}^T \tag{4.4}$$

$$J = \begin{bmatrix} J_1^T & \cdots & J_i^T & \cdots & J_k^T \end{bmatrix}^T \tag{4.5}$$

Likewise, $\dot{p}_d$ in the augmented space is the concatenation of the individual feature velocity vectors. The solution of tracking control algorithm in the augmented system follows exactly the same way as that previously described by Equation (4.2). The tracking error rate for each element of a feature can be controlled by the augmented feedback gain matrix K, which represents a $3k \times 3k$ diagonal matrix in the augmented space. The trajectory tracking error convergence rate depends on the eigenvalues of the feedback gain matrix in Equation (4.2); the larger the eigenvalues, the faster the convergence. In practice, such systems are implemented as discrete time approximation of the continuous time system; therefore, it is reasonable to predict that an upper bound exists on the eigenvalues, depending on the sampling time. A particular feature or its individual components can be more tightly tracked by increasing the eigenvalue of K associated with that direction. By modulating the elements of K, we can effectively encode the relative level of confidence we have in our observations. Measurements with higher confidence will be assigned higher feedback gain values.

4.4 Joint Limit Avoidance Constraints

Chan and Dubey [14] developed a joint limit avoidance algorithm based on a Weighted Least-Norm (WLN) solution. The WLN solution considers a candidate joint limit function, denoted by $H(q)$, that has higher values when the joints near their limits and tends to infinity at the joint limits. One such candidate function is given by

$$H(q) = \frac{1}{4} \sum_{i=1}^{n} \frac{(q_{i,max} - q_{i,min})^2}{(q_{i,max} - q_i)(q_i - q_{i,min})}$$

where q_i represents the generalized coordinates of the i_{th} degree of freedom, and $q_{i,min}$ and $q_{i,max}$ are the lower and upper joint limits, respectively. The gradient of H,

denoted as ∇H, represents the joint limit gradient function, an $n \times 1$ vector whose entries point in the direction of the fastest rate of increase of H.

$$\nabla H = \frac{\partial H}{\partial q} = \left[\frac{\partial H}{\partial q_1} \ , \cdots , \ \frac{\partial H}{\partial q_n}\right] \tag{4.6}$$

The element associated with joint i is given by

$$\frac{\partial H(q)}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2 \ (2q_i - q_{i,max} - q_{i,min})}{4(q_{i,max} - q_i)^2 \ (q_i - q_{i,min})^2}$$

The gradient $\frac{\partial H(q)}{\partial q_i}$ is equal to zero if the joint is at the middle of its range and goes to infinity at either limit. As described in [14], we define the joint limit gradient weighting matrix, denoted by W_{JL}, by an $n \times n$ diagonal matrix with diagonal elements w_{JLi} $(i = 1 \cdots n)$. The scalars w_{JLi} are defined by

$$\mathrm{w}_{JLi} \ = \ \begin{cases} 1 + |\frac{\partial H}{\partial q_i}| & if \quad \Delta|\partial H/\partial q_i| \geq 0 \\ 1 & if \quad \Delta|\partial H/\partial q_i| < 0 \end{cases} \tag{4.7}$$

The term $\Delta|\partial H/\partial q_i|$ represents the change in the magnitude of the joint limit gradient function. A positive value indicates the joint is moving toward its limit while a negative value indicates the joint is moving away from its limit. When a joint moves toward its limit, the associated weighting factor described by the first condition in Equation 4.7, becomes very large causing the motion to slow down. When the joint nearly reaches its limit, the weighting factor is near infinity and the corresponding joint virtually stops. If the joint is moving away from the limit, there is no need to restrict or penalize the motions. In this scenario, the second condition in Equation (4.7) allows the joint to move freely.

4.5 Results

In this section, we report results of key-point detection and human pose reconstruction for several upper-body and whole body image sequences.

Experiments were performed using a single time-of-flight (TOF) range sensor [83]. The human performer was instructed to face the camera and perform various complex actions. The current implementation works well for body twists up to 40 degree rotation on either side of a front facing posture. Large twists and severe interaction between upper and lower body limbs are a challenge in the current implementation, but can be accommodated in future versions. To initialize the tracking, the performer assumes a configuration which does not result in self occlusions (for example, a T-pose) for about 1-2 seconds.

As will be described later, the proposed algorithm performs nearly twice as fast when considering the upper-body versus whole-body pose estimation. Upper-body motions were considered for requirements of online interactive performance for certain applications, such as human to humanoid robot motion retargeting [21]. Figures 4.4- 4.9 illustrate snapshots of limb detection and pose reconstruction results for an upper-body sequence corresponding to motions of violin playing, orchestra conductor, cello playing, swimming, frisbee throwing, and Taiji dance, respectively. Whole-body tracking and reconstruction results are shown in Figures 4.10, 4.11, and 4.12.

Rows 1 and 3 in each Figure illustrate the limb detection and tracking results superimposed onto the depth image. The reconstructed human model pose is shown in rows 2 and 4. The limb detection templates are color coded to determine limb labels (right and left), type of analysis (self-occluded or non-occluded), and whether there is sufficient confidence in the detection results. In particular, the following color coding is used: green (left arm/leg detected from skeleton analysis), yellow (left arm/leg detected from depth slicing), blue (right arm/leg detected from skeleton analysis), pink (right arm/leg detected from depth slicing).

Furthermore, when the detector cannot reliably detect a limb templates or the *HNT Template*, the associated templates are color coded with white. An arm/leg that is color coded with white is treated as an occluded limb. When the *HNT Template* is not detected, the entire frame is skipped. When the *HNT Template* is detected, the template is shown by a red color. Finally, a cyan color arm/leg template indicates the presence of severe self occlusions in which case, the predicted arm position is used.

An example of low-confidence detection result (white color coding) is illustrated in the last frame (bottom right corner) of Figure 4.4. In this case, the algorithm relies on feedback from the predicted key-points which are shown on the reconstructed kinematic model. Yet another instance can be found in Figure 4.5 (row 3, 2^{nd} and 3^{rd} frame from left). For the whole body sequence, Figure 4.11 shows three frames where a leg is not reliably detected and shown in white.

The sequence in Figure 4.10 does not contain self-occlusions. The key-points are detected simply by performing skeleton analysis. In the presence of self-occlusions, the proposed algorithm is able to track limb segments through the intermittent occlusions between left and right leg as shown in Figure 4.11, as well as intermittent occlusions between left and right arm as shown in Figure 4.8. The arm detection is able to detect the re-appearing arms as shown in the last two images of Figure 4.7. We are able to make a correct arm labeling decision based on the spatial and temporal context information.

Various occlusion scenarios are shown in the Figures as well. As we can see, the *HNT Template* detection is robust when the arm occludes the head as shown in Figure 4.4, 4.7, and 4.8. The left and right arm labeling is robust when the arms interact with each other as in Figure 4.8.

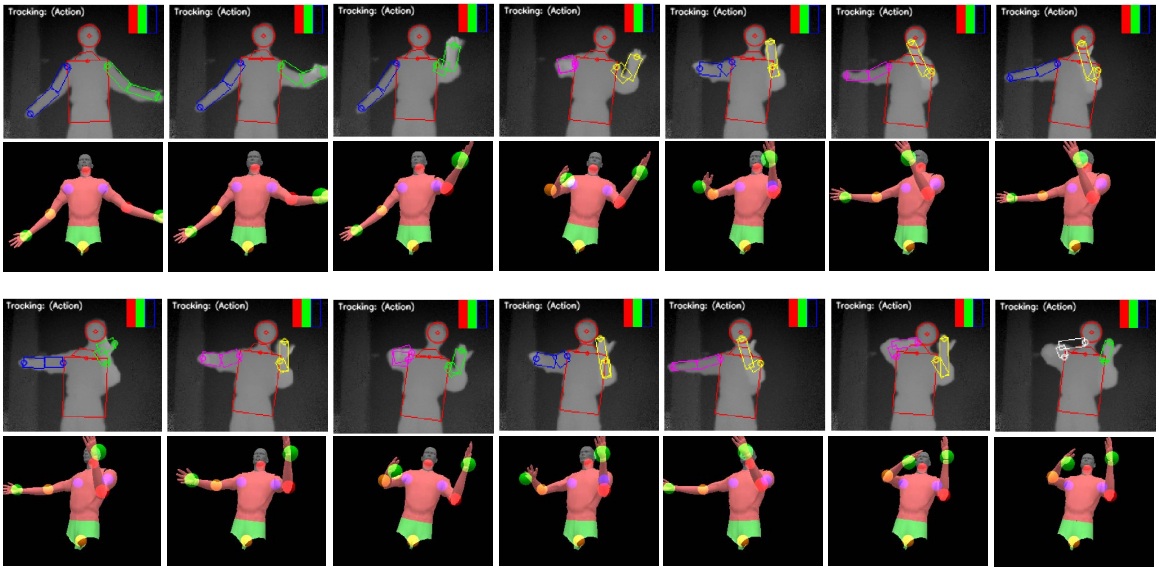


Figure 4.4: Violin Playing Action (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

We have confirmed that the constrained closed loop inverse kinematics module enforces the joint limit and self-penetration avoidance constraint, as described in sections 4.4 and A.1, respectively. These constraints ensure that the reconstructed pose is kinematically realistic.

4.5.1 Error in Key-point Localization

To evaluate the model pose tracking accuracy, we manually locate the feature positions (left of Figure 4.13) on the depth image, and read the 3D positions from camera data. We call this as manual trajectory, and assume it as the ground truth. During the feature tracking, we export the feature positions from 3D model pose tracking results (right of Figure 4.13). We call this as model trajectory. We evaluate the 3D pose tracking accuracy by comparing model trajectory with manual trajectory. Table 4.1 summarizes the model pose tracking errors for each feature point obtained

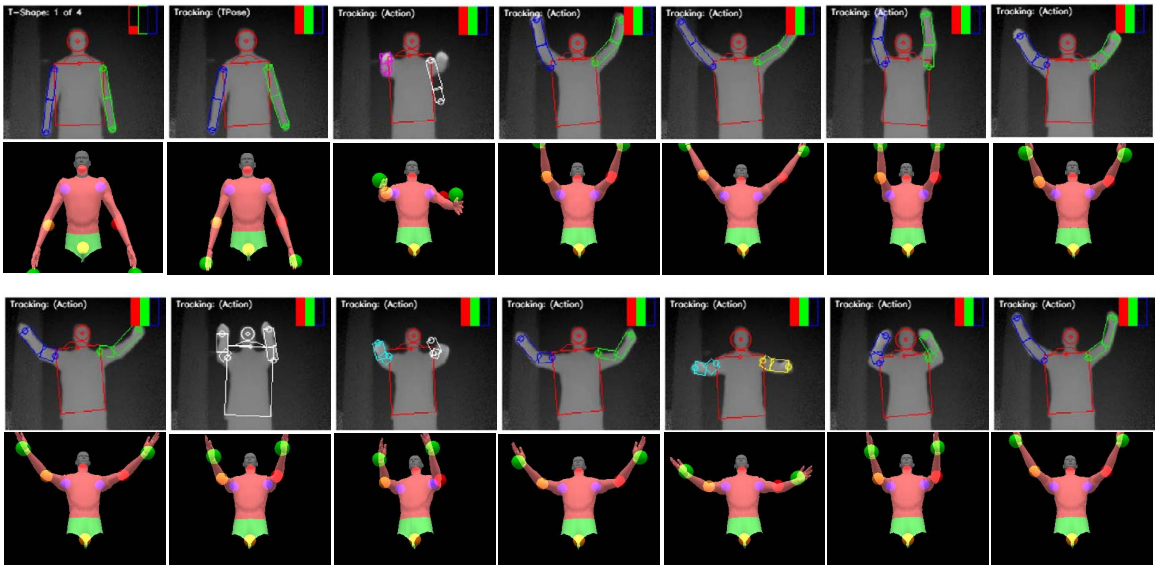


Figure 4.5: Orchestra Conductor (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

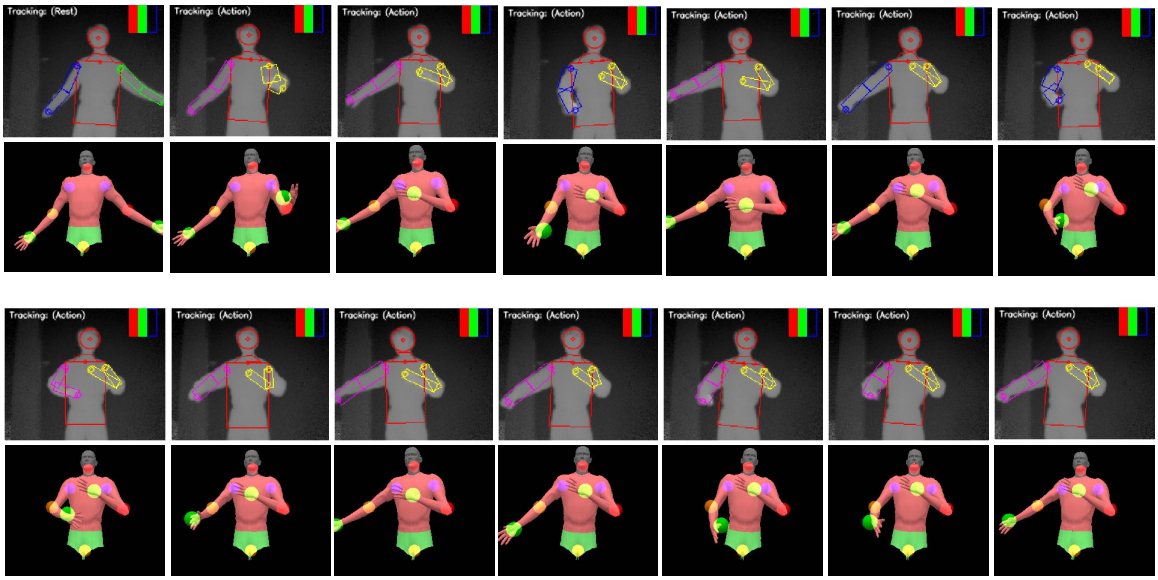


Figure 4.6: Cello playing action (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

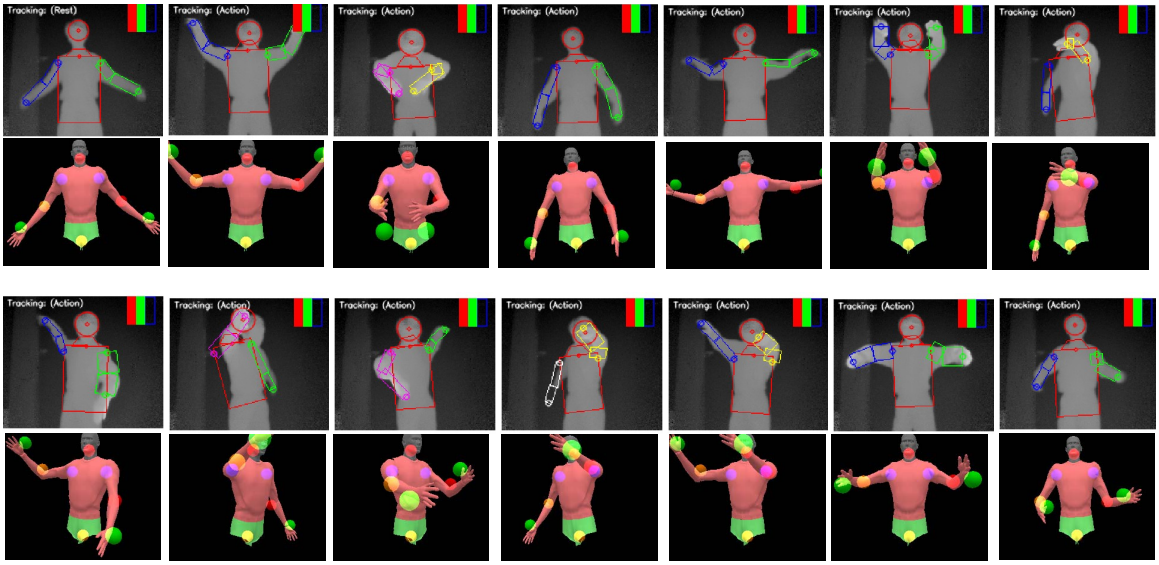


Figure 4.7: Swimming action (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

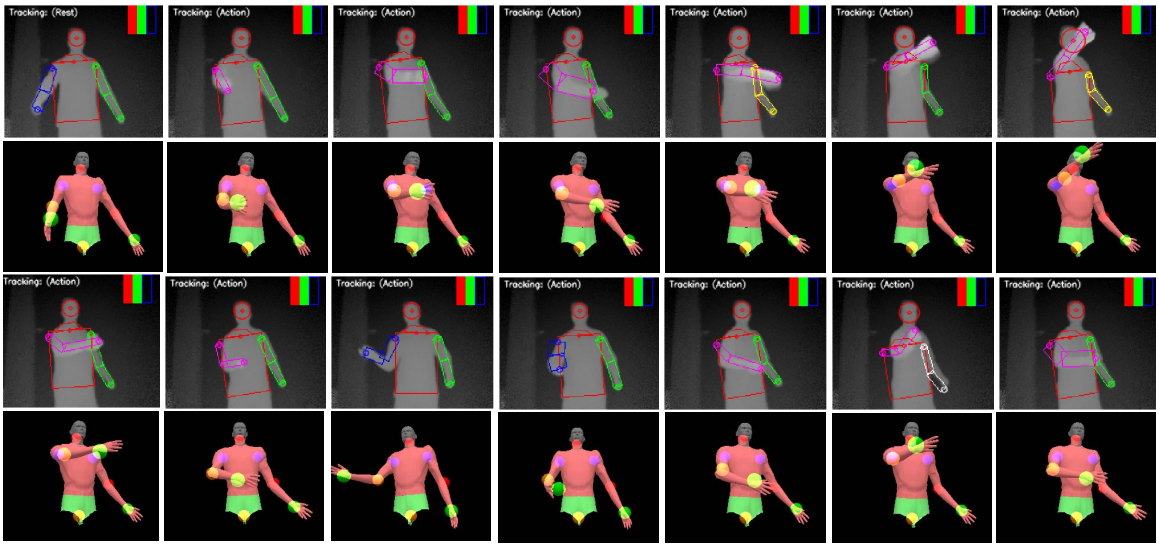


Figure 4.8: Frisbee throwing action (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

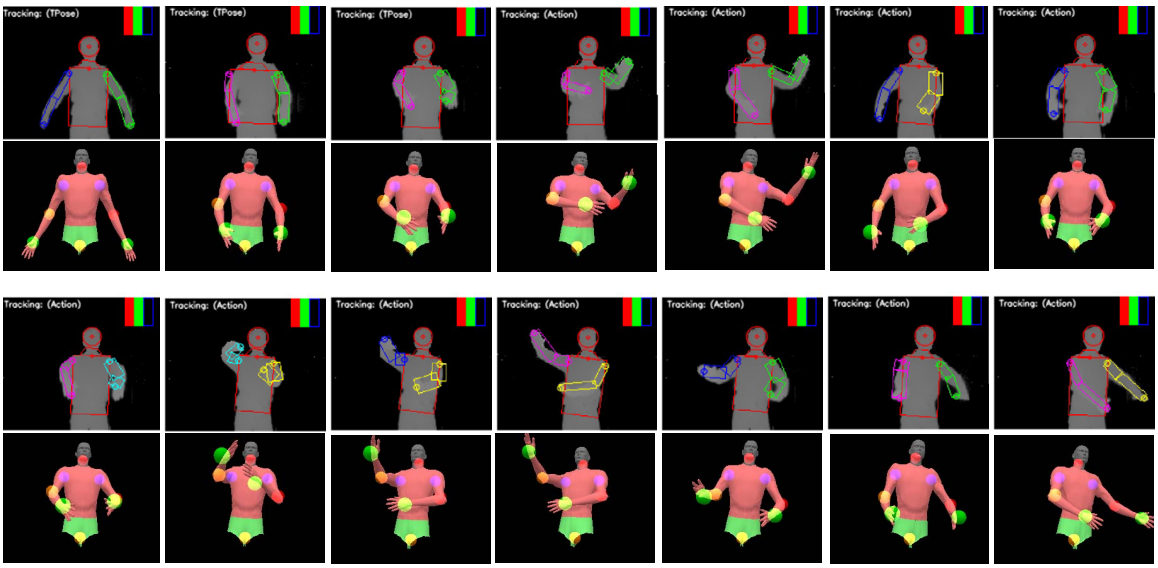


Figure 4.9: Taiji dance (upper-body). Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

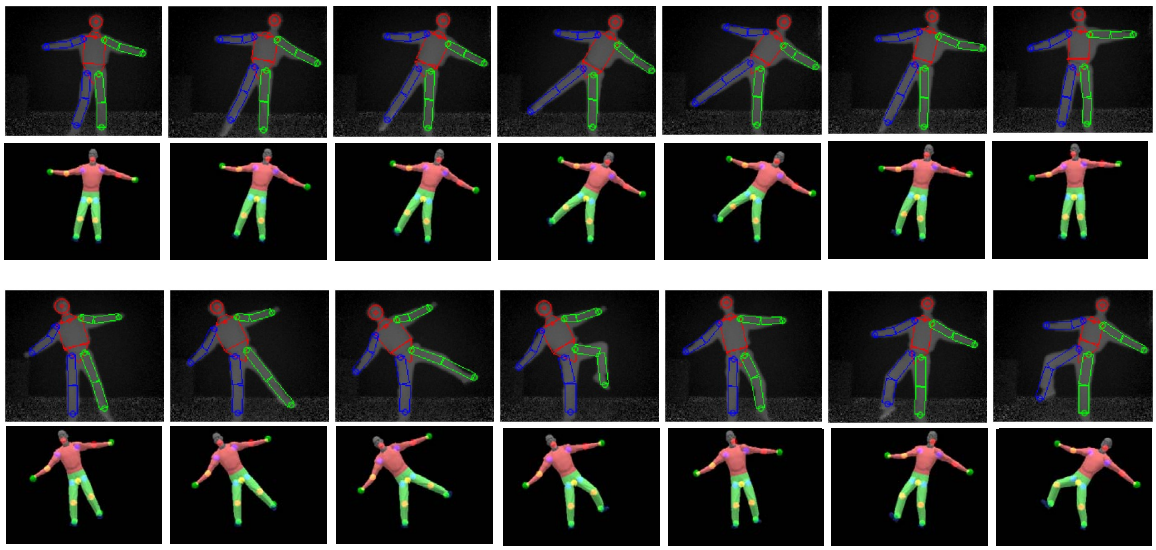


Figure 4.10: Whole body motion without self occlusions. Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

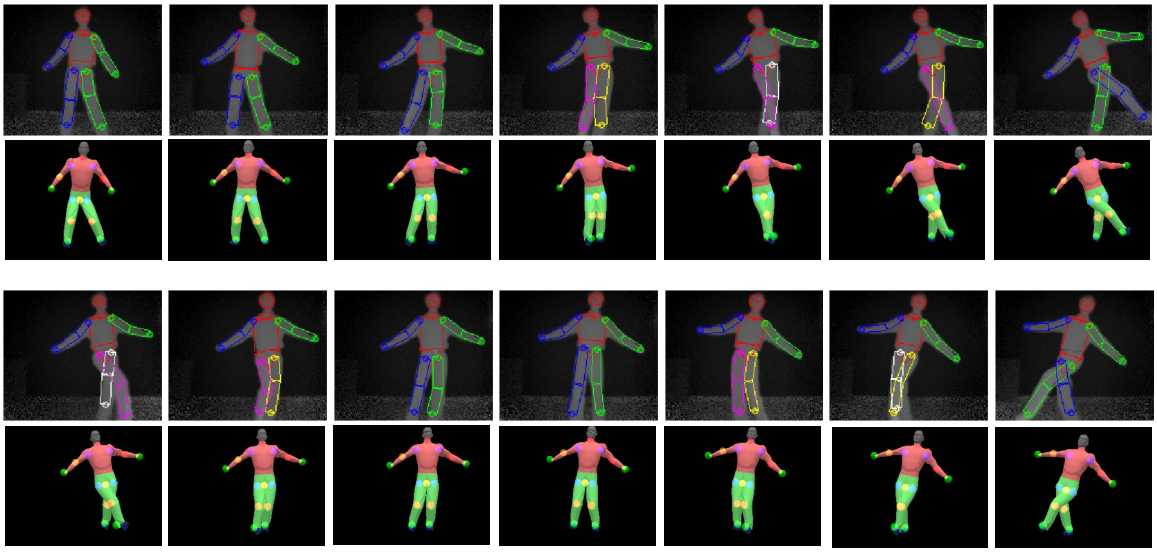


Figure 4.11: Whole body motion with self occlusions during leg crossing. Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

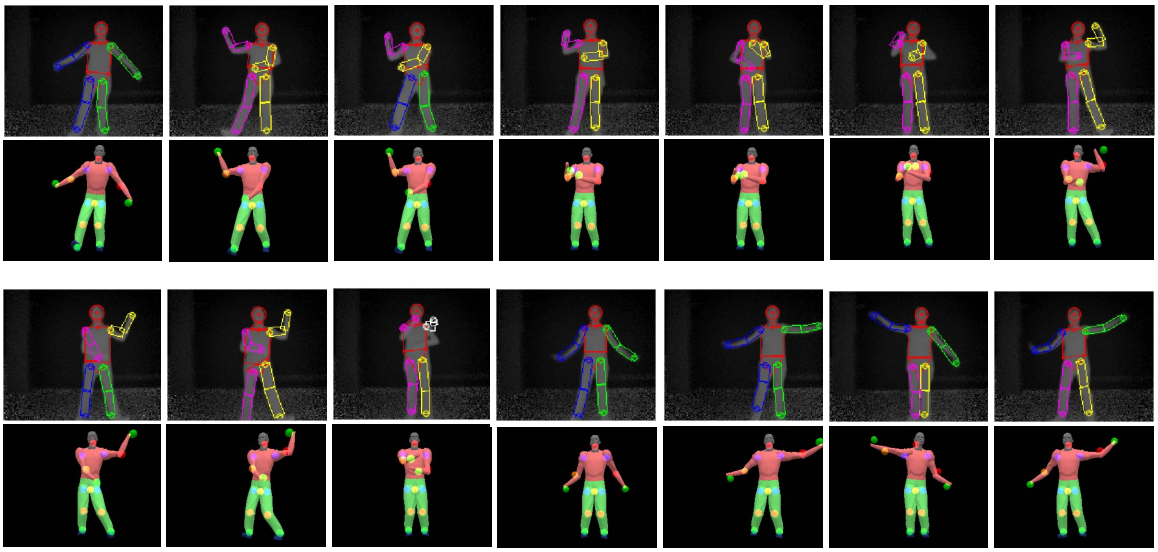


Figure 4.12: Whole body motion with self occlusions during a dancing sequence. Rows 1 and 3: depth image sequence with the detected features. Rows 2 and 4: corresponding reconstructed pose.

from the KF2 sequence. Even though there exists a bias between the model and observed person skeletal body sizes, our model pose tracking has an overall accuracy around 9cm.

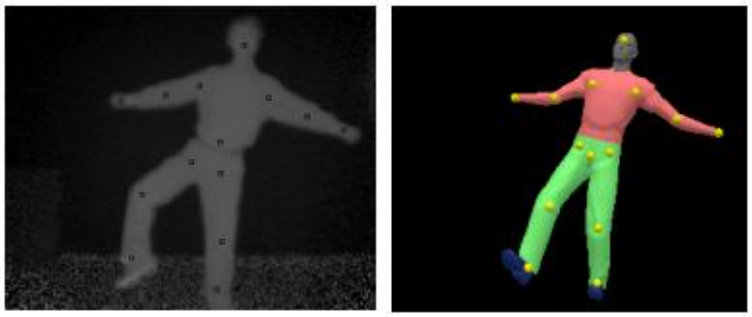


Figure 4.13: Left: feature points from manual localization; Right: feature points from tracked model.

Figure 4.14- 4.16 depicts the mean error ($err = p_{man} - p$), where p_{man} is assumed to be the manually determined ground truth values. In particular, the error in the X direction illustrates the error is largely due to a bias which is a result of imprecise limb-dimensions of the model. The standard deviation may be a better indicator of the the actual error due to the algorithm provided a more appropriate human model is available.

4.5.2 Computational Performance

Figure 4.17 illustrates the frame rate versus frame number for the whole body leg crossing sequence shown in Figure 4.11. The experiment was performed on a standard 2.13 GHz Laptop PC. The frame rates for this sequence range from 1.3 to 9.3 frames per second (fps), depending on the the existence and degree of self occlusions. The

FeaturePoint	Error (in millimeter)		
	X(mean,std)	Y(mean,std)	Z(mean,std)
RightHand	(-135,36)	(21,61)	(-143,43)
RightElbow	(-46,40)	(13,36)	(85,18)
RightShoulder	(-86,20)	(1,47)	(-11,25)
Waist	(30,40)	(-71,25)	(37,18)
LeftHand	(108,28)	(20,60)	(-141,51)
LeftElbow	(58,34)	(-6,47)	(58,23)
LeftShoulder	(74,39)	(2,35)	(-10,17)
Head	(7,31)	(22,25)	(41,26)
RightPelvis	(-19,72)	(133,34)	(22,23)
RightKnee	(-52,57)	(43,121)	(51,38)
RightAnkle	(-60,65)	(29,98)	(155,56)
LeftPelvis	(-10,79)	(146,39)	(6,23)
LeftKnee	(33,52)	(64,106)	(62,32)
LeftAnkle	(-17,52)	(56,67)	(161,34)
Overall	(-8,80)	(34,84)	(27,93)

Table 4.1: Model trajectory position error for KF2 motion sequence.

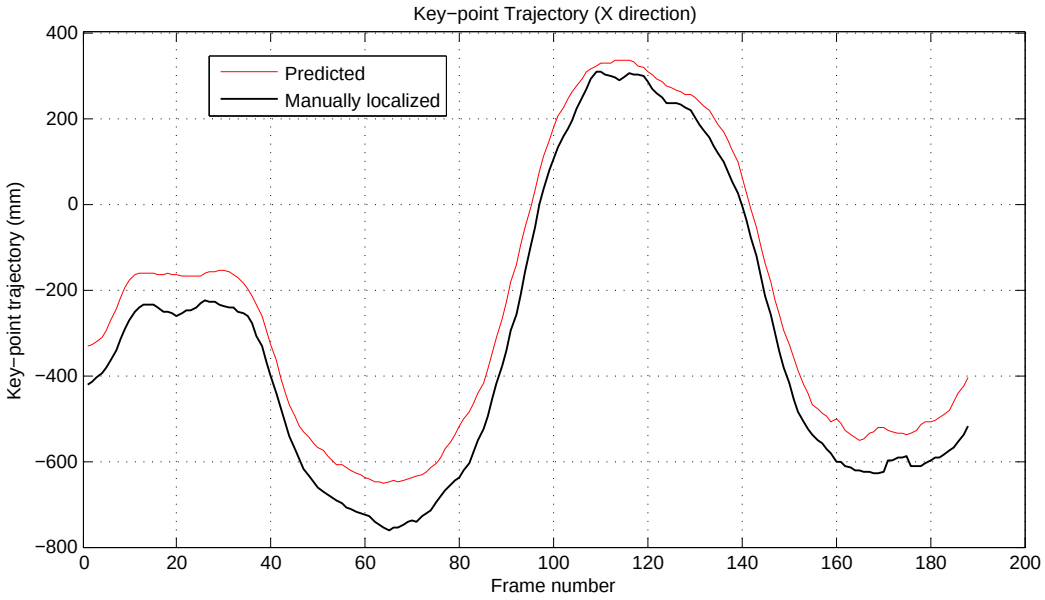


Figure 4.14: Mean error in X-direction of predicted versus manually localized key-point trajectory

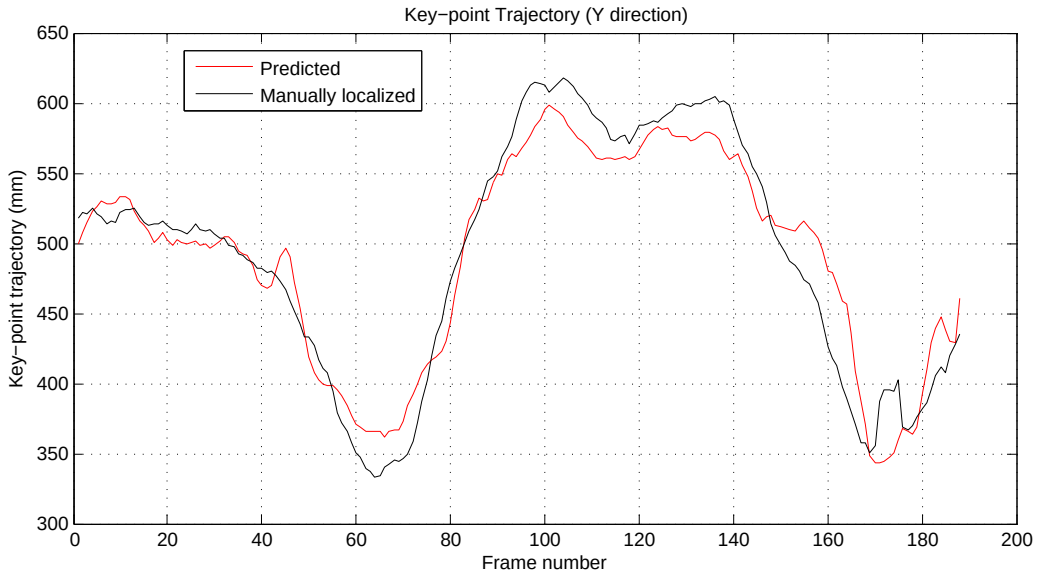


Figure 4.15: Mean error in Y-direction of predicted versus manually localized key-point trajectory

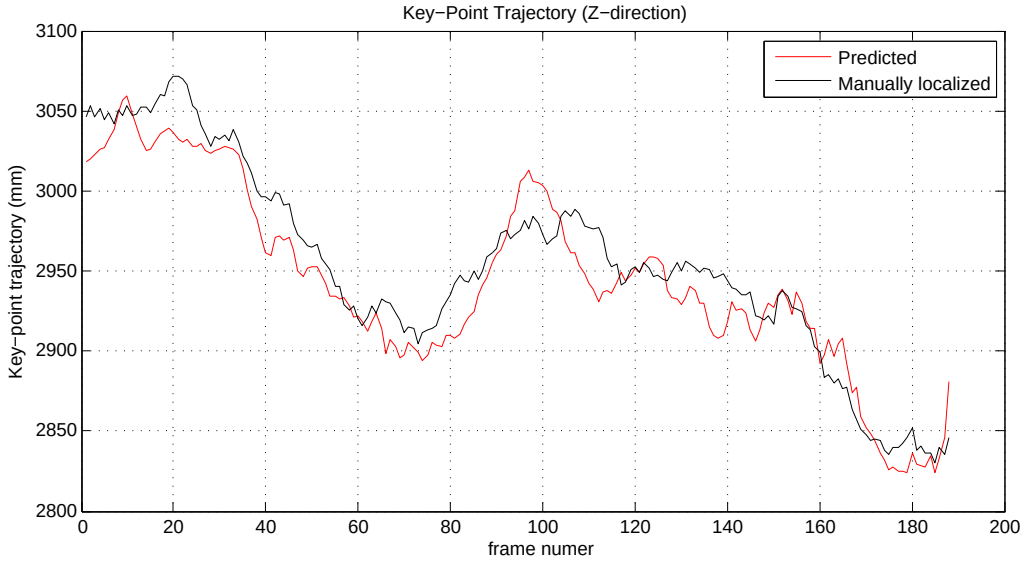


Figure 4.16: Mean error in Z-direction of predicted versus manually localized key-point trajectory

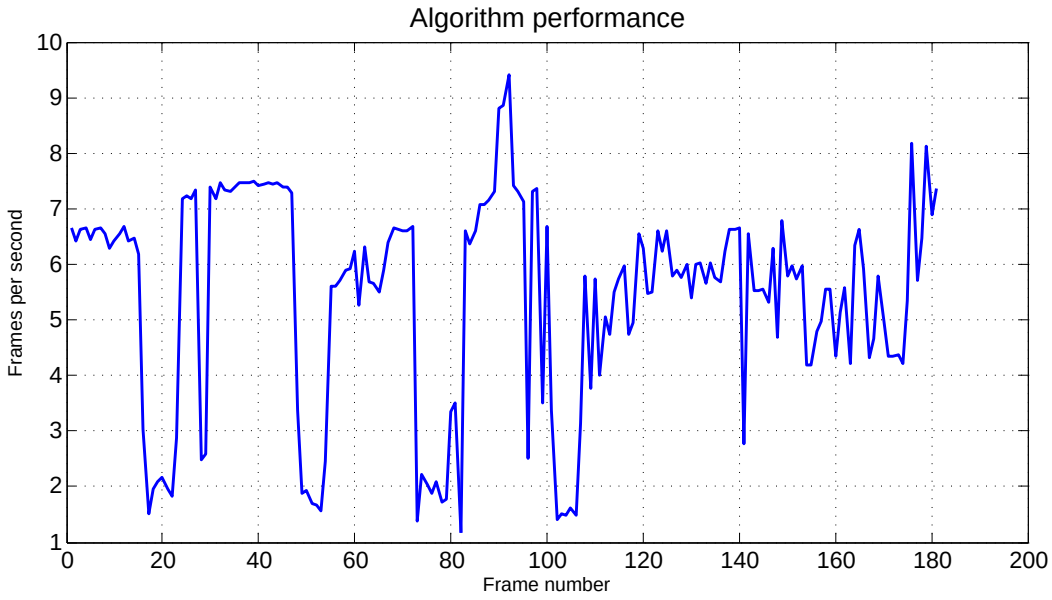


Figure 4.17: Algorithm performance for the crossing leg sequence

algorithm is most efficient when key-points are detected using skeleton analysis in a non-occluding configuration. Conversely, in the presence of self occlusions, more operations are needed to handle occluding configuration. For typical motions, the average performance is approximately 5 fps for whole body pose estimation and 10 fps for the upper-body pose estimation on the aforementioned laptop PC. Even without code optimization and using a standard processor, we can achieve whole body pose reconstruction at interactive rates. We have used the result of our pose estimation algorithm in an application involving real time transfer of human motion to the Honda humanoid robot ASIMO [21].

4.6 Summary

In this chapter, we have applied the closed loop inverse kinematics to reconstruct the human motion from a set of feature points. The algorithm we described can handle missing feature points by predicting the missing feature points from the estimated joint motion. In particular, our algorithm can enforce both the joint limits constraint and self-collision avoidance (described in Appendix A) when reconstructing the motion. Without such constraints, one could resulting unnatural pose such as backward bending arms and arm-torso penetration.

CHAPTER 5

CONSTRAINED LOCAL OPTIMIZATION FOR ARTICULATED MODEL POSE ESTIMATION

In this chapter, we present our previous work using a constrained local optimization technique to estimate the pose of an articulated model from a stream of point clouds possibly from depth sensors. The approach is composed of two closely-coupled steps: coarse human body part labeling and joint position estimation. In the first step, a number of constraints are extracted from notable image features such as the head and torso, and major parts of the human upper body are labeled with these constraints. The second step estimates joint positions optimally using dense correspondences between depth data and human model parts. The proposed framework is shown to overcome some issues of existing approaches for human pose tracking using similar types of data streams. Performance comparison with motion capture data is presented to demonstrate the accuracy of our approach.

The rest of the chapter is organized as follows. After a brief review of related work in section 5.1, our algorithm is described in section 5.2. Experimental results are presented in section 5.3, and section 5.4 summarizes this chapter.

5.1 Related Work on Pose Estimation from Point Clouds

Depth measurement provides necessary information to resolve depth ambiguity which is an issue when using a single color image [79]. Grest et al [33] adapt an Iterative Closest Point (ICP) approach to the articulated human model, where pose parameters are updated using inverse kinematics using dense correspondences between sampled depth observations and model vertices that are found based on the nearest neighbor association. Knoop et al [43] also use ICP to update pose parameters by incorporating multiple input data from different sensors such as stereo depth, hands/face tracking from color camera, etc. Ziegler et al [97] use an unscented kalman filter based on a set of correspondences between model vertices and observed stereo point cloud. As we can see, ICP is often used as a method of choice when a 3D model is to be fitted to 3-dimensional data. A common issue with ICP approaches for human pose tracking is that the model may drift away from the data or get stuck in local minima. An initial configuration is critical for ICP to converge correctly. Our framework also uses the idea of closest point correspondence as a part of the solution, but it is less susceptible to the problem of local minima due to the coarse body part identification. We also use a grid acceleration data structure so as to achieve pose estimation at a high frame rate without loss of accuracy.

5.2 Constrained Optimization for Human Pose Estimation from Depth Sequences

Our algorithm takes a depth image sequence representing human motion and outputs pose vectors of the upper body. Depth data is usually obtained by using stereo cameras, structured light sensors, or time-of-flight sensors. If other image

modality, e.g., a color image sequence corresponding to the depth sequence, is also available, our framework allows such data to be integrated to strengthen the result.

The algorithm consists of two major modules (Figure 5.1), namely (i) coarse body part labeling and (ii) model fitting. In the first module, the region within the given image corresponding to the human body is partitioned into small homogeneous segments. Such segments are formed so that within each segment, depth of each pixel is similar and areas of these segments are of a small and similar size. Each segment is then assigned a body part label (e.g., head, left arm) by using a label assignment framework. At this point, coarse body part identification within each image is completed and passed to the second module. In the second module, a polygonal human upper body model attaching to the underlying kinematic skeleton structure is fitted to the depth observation using ICP for each body part.

5.2.1 Body Constraints

A few body constraints are extracted from depth images.

1. Head and torso constraints: The head and torso are tracked by specialized modules. The head is tracked based on circle fitting with predicted head contour points from depth, while the torso is tracked based on box fitting, where a box with 5 degrees of freedom (x, y, height, width, and orientation) is positioned so as to minimize the number of background pixels within the box.

2. Depth constraint: For certain frames, an arm shape is clearly separable in depth when it is in front of the torso.

3. Color constraint (optional input): The skin color, if a color stream is available, can be used to constrain the hand position. We limit skin blob searching to be

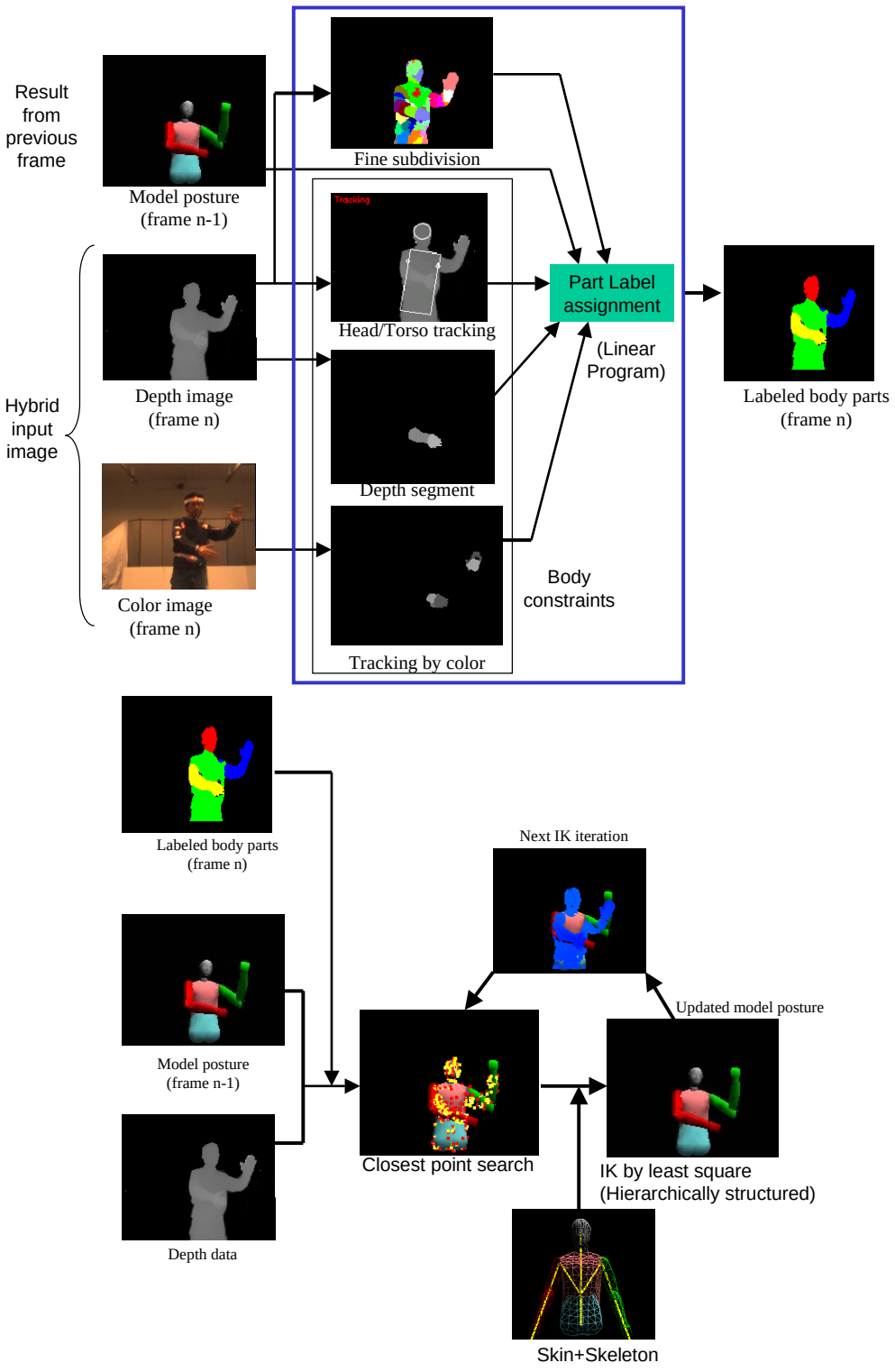


Figure 5.1: Flow of the Algorithm. The top half of the figure illustrates Step 1, in which coarse body part labeling is determined. The bottom half illustrates the process of determining joint positions by fitting.

within a certain area only so as to avoid the face blob area. We note that this is the supplementary information that does not have to be available in order for the method to work. All the tracked results reported in this work do not use this constraint information.

These cues are further used by body part labeling as described Section 5.2.2.

5.2.2 Linear Programming for Coarse Body Part Labeling

In the next step, we form an adjacent graph G where a node represents a small cluster of pixels within the human body (see Figure 5.1, top) and an edge represents adjacency relationship between two pixel clusters. A recursive subdivision strategy is taken to partition the image into pixel clusters based on homogeneity of depth values and spatial positions. Starting from the root node representing all pixels, the k-mean clustering (with k=2) is used to subdivide clusters further until each cluster has the property that it is sufficiently small in size and that it has small depth variance. All the leaf nodes form segment set S to be labeled. Segments $s_i \in S$ $(i = 1, 2, \cdots, N)$ are to be labeled as major body parts $p_1, p_2, \cdots, p_M$.

This is a labeling problem and formulated in the following optimization problem. A segment s_i in S is to be assigned a label by a function f. Each s_i has an estimate of its likelihood of having labeling $f(s_i)$. This comes from a heuristic in pose estimation. For example, if s is near the bottom of the image, its likelihood of being a part of the head is low. For this purpose, a non-negative cost function $c(s, f(s))$ is introduced to represent the likelihood. Furthermore, we consider two neighboring segments s_i and s_j to be related so that we would like s_i and s_j to have the same label. Each edge e in graph G has a nonnegative weight w_e indicating the strength of the relation.

table A_{ij}	Head (p_1)	Torso (p_2)	LeftArm (p_3)	RightArm (p_4)	... (p_M)
segment:s_1	A_{11}	A_{12}	A_{13}	A_{14}	(A_{1M})
segment:s_2	A_{21}	A_{22}	A_{23}	A_{24}	(A_{2M})
...					
segment:s_N	A_{N1}	A_{N2}	A_{N3}	A_{N4}	(A_{NM})

Constraints:
$A_{ij} = 1$ ifs_i belongs to p_j
$A_{ij} = 0$ otherwise
$\sum_{j=1}^{M} A_{ij} = 1$
Nearby pixels have similar label

Figure 5.2: Labeling problem.

Moreover, certain pairs of labels are more similar than others. Thus, we impose a distance $d()$ on the label set. Larger distance values indicate less similarity. The total cost of a labeling f is given by:

$$Q(f) = \sum_{s \in S} c(s, f(s)) + \sum w_e d(f(s_i), f(s_j))$$

For our problem, the following table (Figure 5.2) is to be completed, where binary variable A_{ij} indicates if segment s_i belongs to body part p_j. For $c(i, j)$, the Euclidean distance from segment to the model body part (from the previous frame) is used. More specifically, the Euclidean distance from a segment s_i to a model body part p_j is estimated as the sum of squared distances from a number of sampled pixels to their nearest model vertices in the model body part.

Since each segment belongs to only one body part, $\sum_{j=1}^{M} A_{ij} = 1$ hold. In addition to this constraint, a number of related constraints are considered.

1. Neighboring segments should have a similar label;

2. Head and torso constraint;

3. Depth slicing constraint;

4. Color constraint;

It turns out that this is an instance of the Uniform Labeling Problem, which can be expressed as the following integer programming by introducing auxiliary variables Z_e for an edge e to express the distance between the labels and we use Z_{ej} to express the absolute value $|A_{pj} - A_{qj}|$. Following Kleinberg and Tardos [42], we can rewrite our optimization problem as follows:

$$\min\{\sum_{i=1}^{N}\sum_{j=1}^{M} c(i,j)A_{ij} + \sum_{e \epsilon E} w_e Z_e\} \tag{5.1}$$

subject to

$$\sum_{j=1}^{M} A_{ij} = 1, i = 1, 2, 3, \cdots, N \tag{5.2}$$

$$Z_e = \frac{1}{2}\sum_{j=1}^{M} Z_{ej}, e \in E \tag{5.3}$$

$$Z_{ej} \geq A_{pj} - A_{qj}, e = (p,q); j = 1, 2, \cdots, M \tag{5.4}$$

$$Z_{ej} \geq A_{qj} - A_{pj}, e = (p,q); j = 1, 2, \cdots, M \tag{5.5}$$

$$A_{ij} \in \{0,1\}, i = 1, 2, \cdots, N; j = 1, 2, \cdots, M \tag{5.6}$$

Here, terms involving Z_e and Z_{ej} come from constraint (1). The weight w_e is given by $w_e = e^{-\alpha d_e}$, where d_e is depth difference between two adjacent segments and $\alpha = 20.0$ is selected based on experiments. In our study, we let $M = 4$ (head, torso, left arm, right arm).

For constraint (2), if the segment s_i is outside the tracked head circle, an additional constraint $A_{i,1} = 0$ is added. If segment s_i is outside of the tracked torso box, constraint $A_{i,2} = 0$ is added. To apply constraint (3) for the detected arm segments, constraint:$A_{i,3} + A_{i,4} = 1$ is added (as it is not clear whether it is the right or left arm). Finally, if there are tracked hand positions based on skin color information, we can add constraint: $A_{i,3} + A_{i,4} = 1$.

In general, solving an integer program optimally is NP-hard. However, we can relax the above problem to linear programming with $A_{ij} \geq 0$, and this can be solved efficiently by using a publicly available library, e.g., [1]. Kleinberg and Tardos [42] describe a method for rounding fractional solutions so that the expected objective function $Q(f)$ is within a factor of 2 from the optimal solution. In our experiments we find that this relaxed linear programming always returns an integer solution. (See an observation due to Anguelov et al [3].) The top of Figure 5.1 shows one example of this body part labeling result.

5.2.3 Model Fitting as Local Optimization for Pose Estimation

The human body model is represented as a hierarchy of joint link models with a skin mesh attached to it as in Lewis et al [48]. For the upper body model used in this thesis, a skin mesh and hierarchical skeleton structure are illustrated as in the bottom of Figure 5.1.

Given a set of 3D data point $P = \{p_1, p_2, \cdots, p_m\}$ as targets and their corresponding model vertices $V = \{v_1, v_2, \cdots, v_m\}$, the model pose vector q is estimated as

$$\hat{q} = argmin_q ||P - V(q)||_2 \tag{5.7}$$

where $q = (\theta_0, \cdots, \theta_n)^T$ is the pose parameter vector and v_i's are visible vertices of the polygonal model. To solve this minimization problem efficiently and robustly, we use a variant of inverse kinematics (known as damped least square), which is inspired by the well-known ICP algorithm [7]. The formulation (see Figure 5.7) minimizes

$$||J\Delta q - \Delta E||^2 + \lambda||\Delta q||^2$$

where J is the Jacobian.

The inverse kinematics with damped least square [10] has the benefit of avoiding singularities, thus making the process numerical stable. We use λ=0.1 based on our experiments. For articulated body pose estimation, the algorithm depends on the accuracy of finding correspondence pairs of data point and model vertices. Most recent works apply the nearest-neighbor searching between two point clouds: one contains all the observed 3D points from depth or other sensor; the other contains all the model vertices. Since iteration may be attracted to local minima as the algorithm is inspired by the well-known ICP algorithm [7], we apply the aforementioned body part labeling to limit the nearest neighbor searching between a subset of observed 3D points and a subset of visible model vertices for each body part. Thus, this not only speeds up the nearest neighbor searching, but also more importantly, it achieves robust pose estimation even for long sequences containing large motions between two consecutive frames. In our implementation, the OpenGL depth buffer is utilized to decide model vertex visibility.

In order to get faster computation, we use a grid based spatial index data structure to speed up the nearest neighbor searching between point clouds. We partition the working volume into a set of 3D grid. Because only scene profiles are used, we partition the xy plane of the working volume. Depth points and visible model vertices are

indexed into corresponding grids. To perform the nearest neighbor searching for a model vertex, we first find the grid where it is located. The nearest neighbor depth point in this grid is found afterward. Then, we recursively propagate the nearest neighbor depth point searching to the neighboring grids until the minimal distance from grid corners to the model vertex is greater than the current minimal distance from the model vertex to the depth points. As illustrated in Figure 5.8, the method contributes to speed-up by a factor of 6.

When capturing a pose sequence, the subject is initially requested to take an open-arm posture (so-called "T-pose"), in which his arms and torso do not overlap. At this initialization stage, body dimensions are measured and further used to scale the kinematic skeleton and polygonal human body model.

5.3 Experimental Results

The proposed pose estimation algorithm has been tested on many sequences collected from a few human subjects. Depth sequences and color sequences have been obtained by using a calibrated hybrid pair of stereo (CSEM SR3000 for depth and Sony DFWV500 for color) in a synchronous fashion. Furthermore, a motion capture system by PhaseSpace Inc. with 8 camera units has also been run synchronously with the hybrid camera system, recording coordinates of the eight major joints of the subject as the ground-truth reference. The subject wears markers for a motion capture purpose only and these markers are not used for the main algorithm. Test motion sequences include a complete semaphore flag signaling motion (A to Z), simple exercise movements, and TaiChi movements. The total number of frames collected

is 4800 and each test sequence is about 400 frames long. All sequences have been tracked successfully at the frame rate of 5∼9Hz on a 3.00GHz HP desktop.

Figure 5.3 contains tracked frames taken from full-length sequences, where the subject performs (a) a TaiChi motion and (b) an exercise motion, respectively. Pose estimation precision has also been compared against joint position data captured by a marker-based motion capture system. Figure 5.4 contains errors in various joint positions for the TaiChi motion sequence. As seen here, the overall tracking error is approximately 5cm (where the subject stands 1.5m to 2m from the camera). Similar tracking results have been obtained for the other sequences, such as out-of-plane-rotation (Figure 5.6) which is usually difficult to capture with single-color-camera-based pose estimation methods.

To compare tracking stability, we tested an ICP-based approach using exactly the same sequences. The ICP and our method have similar performance (in terms of the amount of error from the ground-truth), when tracking runs successfully. However, a significant difference is that for some frames (where our method processes successfully), the ICP based tracking fails and never recovers (as shown in Figure 5.5). Our method functions for all cases where the ICP method works. The ICP-based method is slower, because it has to do more iteration for convergence. This illustrates the advantages of the part labeling step in our framework.

At this point, let us compare our approach with a few other approaches using depth sequences. Ziegler et al [97] use depth sequences obtained by four stereo cameras. Point correspondences are based on spatial proximity which may result in wrong correspondences when body parts are close to each other. Our method is less susceptible to such a problem due to the use of inverse kinematics. Demirdjian et al [24] use

efficient example-based matching to improve the tracking of a set of likelihood modes. Large errors can still occur when the test example is not close to training examples. Our coarse labeling step has a similar function and finds the likelihood mode with constraints from bottom-up observations. Grest et al [33] introduce an ICP method for articulated body pose estimation, while they do not address the robustness of ICP-based pose estimation. Knoop et al [43] utilize skin color segmentation (therefore face and hand feature trackers) to improve the ICP-based pose estimation. It is not clear how to handle temporarily invisible face or hands. Some other approaches use multiple sensors to obtain more surface data. Cheung et al [15] use visual hull while Anguelov et al [3] use 3D range scan data to reconstruct human skeletal structures. Accurate pose estimation might be obtained using these methods since body parts are visible in multiple views.

5.4 Summary

We have presented an accurate method to estimate human pose from depth sequences. Our method consists of two major components that cooperate to estimate and track human motion. The first module is body component identification which has been solved by reducing it to linear programming. The second module is model fitting by using inverse kinematics based on dense correspondences between the depth data and the human kinematic model. The result of the second component, in turn, is used to initiate the first component for the next frame. The algorithm tracks human upper-body movements over several minutes of pose sequences at a speed of a few Hz using a laptop PC (up to 10Hz when a desktop with 3GHz is used). We have also made a comparative study of markerless pose tracking based on a commercial

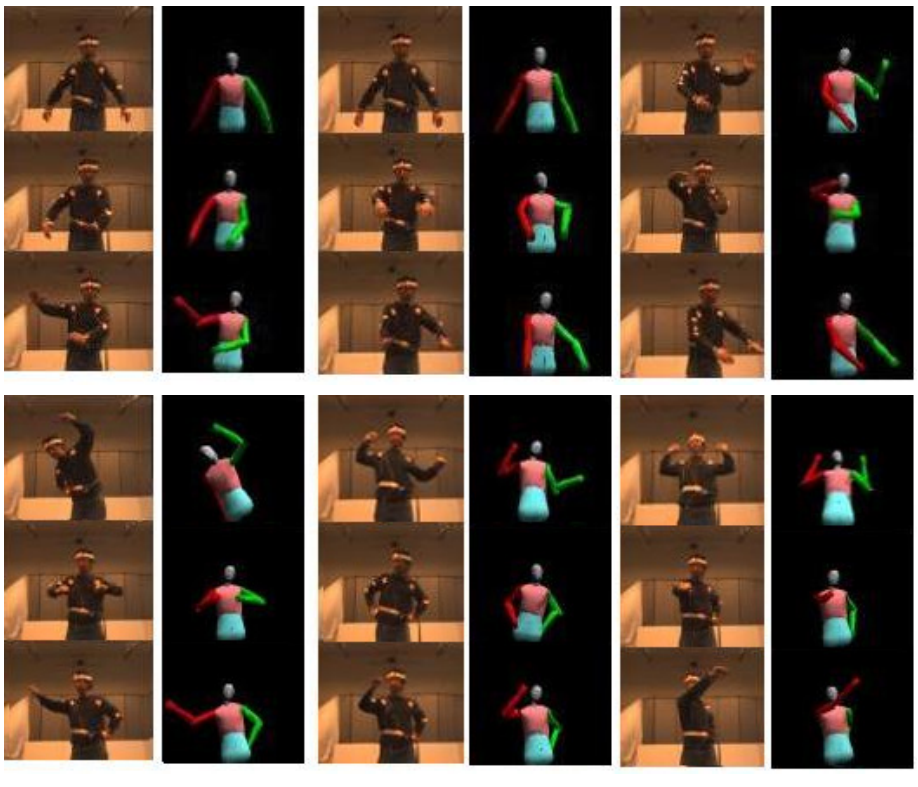


Figure 5.3: Snapshots of the algorithm output (a) TaiChi sequence, (b) Simple exercise sequence

Model Joints	error (in millimeter)		
	$\Delta X(\mu, \sigma)$	$\Delta Y(\mu, \sigma)$	$\Delta Z(\mu, \sigma)$
Right Hand	(-15, 49)	(-39, 58)	(23,44)
Right Elbow	(-23, 34)	(-70, 42)	(-48,59)
Right Shoulder	(21, 57)	(-43,19)	(1,25)
Waist	(-24, 26)	(-12, 15)	(-19,14)
Left Hand	(16, 61)	(-6, 86)	(44,45)
Left Elbow	(30, 35)	(-74, 39)	(71,66)
Left Shoulder	(-23, 53)	(-36, 30)	(27,30)
Head	(-15, 26)	(-18, 15)	(-22,15)
Overall	(-4, 49)	(-37, 50)	(22,52)

Figure 5.4: Accuracy table.

TaiChi	Color Image	Our Method	ICP only
Frame:178			

Figure 5.5: Stability comparison between our method and standard ICP

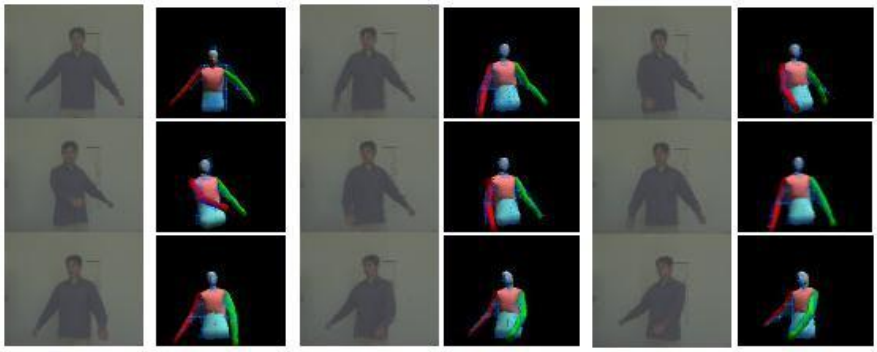


Figure 5.6: Examples of out-of-plane rotation up to 50 degree

model fitting(P:3D point,V:model vertex)

1. Form

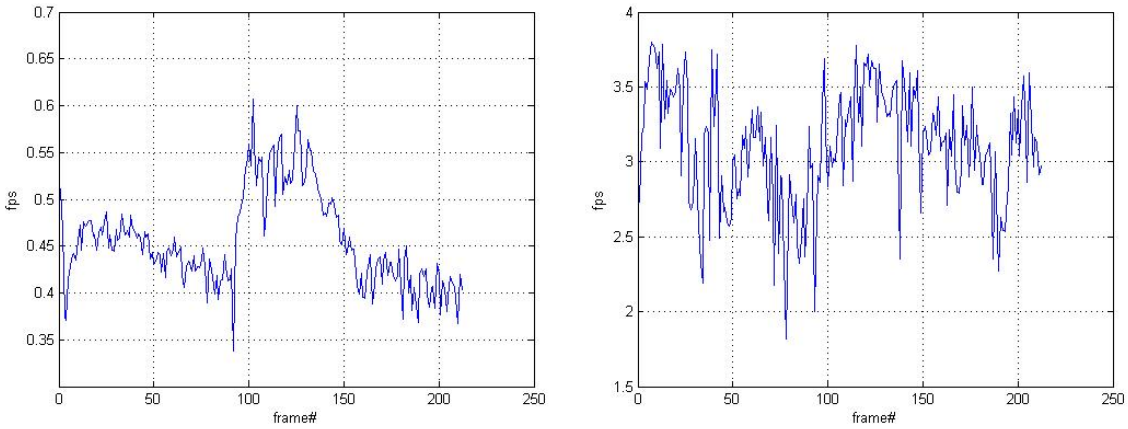
$$\Delta e_i = \begin{bmatrix} p_i^x - v_i^x \\ p_i^y - v_i^y \\ p_i^z - v_i^z \end{bmatrix}, \Delta E = \begin{bmatrix} \Delta e_1 \\ \Delta e_2 \\ ... \\ \Delta e_m \end{bmatrix}$$

2. Solve $J\Delta q = \Delta E$ by damped least square where J is Jacobian of model vertices

$$\Delta q = (J^T J + \lambda I)^{-1} J^T \Delta E$$
$$q = q + \Delta q$$

3. Repeat until Δq is sufficiently small

Figure 5.7: Model fitting procedure



Time distribution without Grid Acceleration

Time distribution with Grid Acceleration

Figure 5.8: Performance comparison with and without grid acceleration (on a 2.13GHz IBM Laptop)

marker-based tracking system and shown that our joint positions have an accuracy about 5 centimeters.

CHAPTER 6

MODEL-BASED HUMAN POSE ESTIMATION: A UNIFIED APPROACH

The objective of this thesis is to explore the problem of robust human pose estimation. A feature-based approach as described in Chapter 3 and 4 is robust and can recover from tracking failure when body part is re-detected. However, its estimation accuracy depends solely on the localization accuracy of the feature points. The ICP-based method with body part labeling as described in Chapter 5 has better human motion tracking performance than the previous ICP methods, and it can achieve a higher accuracy than feature-based method because it utilizes a set of dense correspondences. But, we have found that it is not able to recover from tracking failure once it is trapped in local minima of the energy function. The labeling of body parts depends on the estimated pose from the last frame, and it could result in the accumulation of pose tracking errors. As a result, the performance of ICP-based pose tracker is degraded after the tracking failure.

In this chapter, a unified pose tracking based on the Bayesian inference framework is proposed to integrate the estimation results from the feature-based method and the estimation results from the local pose optimization method. In Section 6.1, the overall procedure for pose tracking with Bayesian inference is presented. In Section 6.2,

we present the method to generate the pose hypotheses from the observed feature points from depth image analysis through the constrained inverse kinematics. In Section 6.3, we present the procedure to perform temporal prediction, density sampling from the Gaussian mixture distribution, and local pose optimization to generate more pose hypotheses. In Section 6.4, we present the tracking error measurement function and observation likelihood approximation. In Section 6.5, we approximate the posterior distribution,and define the optimal pose estimation. Section 6.6 includes the experimental results on Bayesian pose estimation, and Section 6.7 summarizes this chapter.

6.1 Pose Tracking with Bayesian Inference

Let q_t be the model pose parameters at time t, and $p(q_t|I_1, I_2, \cdots, I_t)$ be the probability distribution of pose parameters given all observed images $\{I_1, I_2, \cdots, I_t\}$, then Bayesian tracking is formulated as:

$$\begin{aligned} p(q_t|I_1, I_2, \cdots, I_t) &\propto p(I_t|q_t)p(q_t|I_1, I_2, \cdots, I_{t-1}) \\ &= p(I_t|q_t) \int_{q_{t-1}} p(q_t|x_{t-1})p(q_{t-1}|I_1, I_2, \cdots, I_{t-1})dq_{t-1} \end{aligned} \tag{6.1}$$

Assuming we can approximate the observation distribution as:

$$p(I_t|q_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \tag{6.2}$$

Let human dynamics have Gaussian noise $N(0, W)$, the temporal propagation is given by:

$$p(q_t|I_1, I_2, \cdots, I_{t-1}) = \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; f(\mu_j^{t-1}), \Lambda_j^{t-1} + W) \tag{6.3}$$

where $f(\mu_j^{t-1})$ is any appropriate pose dynamic process.

Using the above Bayesian tracking equation, we can represent the posterior as:

$$p(q_t|I_1, I_2, \cdots, I_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; f(\mu_j^{t-1}), \Lambda_j^{t-1} + W) \quad (6.4)$$

As we can see, this will increase the Gaussian components for the posterior distribution exponentially along the updating of time. Instead, we should approximate this with M component Gaussian distribution:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{j=1}^{M} \pi_j^t N(q_t; \hat{\mu}_j^t, \hat{\Lambda}_j^t) \quad (6.5)$$

Since we represent the posterior distribution as a sum of Gaussian, there are available methods to perform density approximation. In the simplest way, we can keep the dominant modes in the posterior distribution. Researchers [79, 13] also suggest to pick modes from likelihood function and combine them with compatible ones from the predicted priors. Some authors [24] also pick the modes from likelihood function and re-weight with predicted prior. If there are a large number of modes, we can sample the posterior and perform kernel density estimation as in paper [81].

The detailed illustration of this Bayesian inference framework to pose estimation is shown in Figure 6.1, where we are able to integrate three sources of information: the low-level detection, the local pose optimization and the temporal prediction information.

6.2 Constrained Inverse Kinematics

Let q_0 be the initial model pose, V be the set of model marker points, P be the set of observed feature points from sensor. Let $\hat{q} = ConstaintIK(q_0, V, P)$ denote the constrained inverse kinematics as:

$$\hat{q} = q_0 + sJ^*(P - V) \quad (6.6)$$

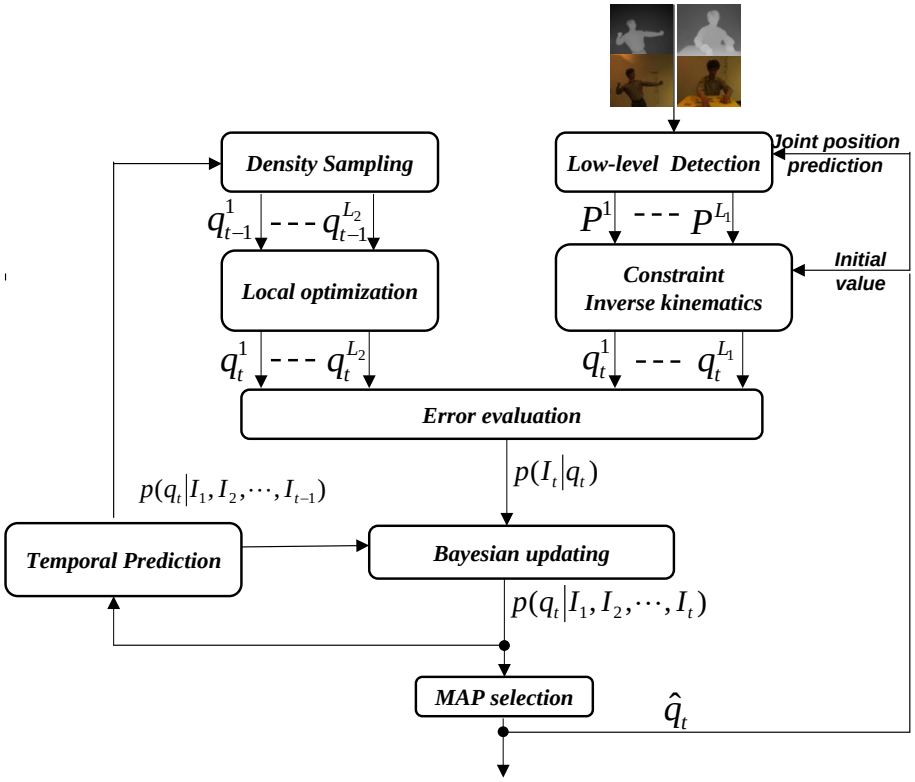


Figure 6.1: Robust pose estimation with Bayesian tracking framework.

$$J^* = W_1^{-1} J^T (J W_1^{-1} J^T + W_2)^{-1} \tag{6.7}$$

where s is a scalar to adjust the step size of inverse kinematics, W_1 and W_2 are defined as for singularity avoidance and joint limit avoidance.

In this study, the model marker points include the set of model vertices as shown in Figure 4.2. Their corresponding feature points as in Table 6.1 are detected through low-level analysis. At any frame, we might only detect a subset of these because of occlusion or detection-miss. Moreover, it is difficult to accurately localize elbow points for certain poses, and we can only obtain approximate elbow positions. As is known, the inverse kinematics based method depends on the starting pose values as well. Let

$\hat{q}_{t-1}$ be the optimal pose estimation of the last frame, let q^0_{t-1} be the resting pose, we would use the constrained inverse kinematics to generate following three sets of joint hypotheses ($L_1 = 3$) as in Table 6.2.

Feature point	Description
ws_t	waist point
ls_t	left shoulder point
rs_t	right shoulder point
hd_t	head point
le_t	left elbow point
re_t	right elbow point
lh_t	left wrist point
rh_t	right wrist point
lp_t	left pelvis point
lk_t	left knee point
la_t	left ankle point
rp_t	right pelvis point
rk_t	right knee point
ra_t	right ankle point

Table 6.1: Feature point description

As designed, q^1_t will generate the pose to use both the optimal estimation and all feature points. However, q^2_t will generate the pose from the starting pose so that this hypotheses will keep our estimation robust against the possibly erroneous estimations from the last frame. q^3_t will generate the pose without using the elbow feature points so that it will be robust against the errors in elbow detection.

Pose hypothesis	Starting pose	Feature points
q_t^1	$\hat{q}_{t-1}$	ws_t,ls_t,rs_t,hd_t,le_t,re_t,lh_t,rh_t, lp_t,lk_t,la_t,rp_t,rk_t,ra_t
q_t^2	q_{t-1}^0	ws_t,ls_t,rs_t,hd_t,le_t,re_t,lh_t,rh_t, lp_t,lk_t,la_t,rp_t,rk_t,ra_t
q_t^3	$\hat{q}_{t-1}$	ws_t,ls_t,rs_t,hd_t,lh_t,rh_t, lp_t,lk_t,la_t,rp_t,rk_t,ra_t

Table 6.2: Pose hypotheses from low-level detection

6.3 Temporal Prediction, Density Sampling and Local Pose Optimization

Since the motion to be tracked in this study is general and is of high uncertainty, a common approach is to model the human pose temporal dynamics as zero velocity with a Gaussian noise $N(0, W)$. Therefore, we can approximate the temporal prior as:

$$p(q_t|I_1, I_2, \cdots, I_{t-1}) = \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \tag{6.8}$$

Density sampling can be performed based on the posterior distribution as defined by Equation 6.5. As this is a standard Gaussian mixture distribution, density sampling can be performed by following steps:

(step1): sampling the component $j \sim \{\pi_1^{t-1}, \cdots, \pi_M^{t-1}\}$;

(step2): samping the Gaussian distribution $q_{t-1}^j \sim N(q_{t-1}; \hat{\mu}_j^{t-1}, \hat{\Lambda}_j^{t-1})$;

(step3): repeating the above steps until we obtain L_2 samples.

Let q_{t-1}^i be one of samples from density sampling, Vs denote a set of sampled model vertices that is visible from camera, Ps denote the set of 3D depth points that

is closest to Vs, and q_t^i denote the pose from local pose optimization:

$$q_t^i = ConstraintIK(q_{t-1}^i, Vs, Ps) \tag{6.9}$$

We obtain the visible model vertices Vs from depth buffer technique of OpenGL rendering of model. Closest point set Ps is obtained through grid acceleration data structure as in the last Chapter.

6.4 Tracking Error Evaluation

To evaluate the tracking quality, we use a tracking error measurement function that is based on the sum of the distances from sampled depth points to their corresponding closest model vertices. Without loss of generality, we denote Ps as the set of sampled depth point, and denote Vs as the set of visible model vertices that are closest to the Ps. Then, the tracking error measurement function can be defined as:

$$d^2(Ps, Vs(q_t)) = \sum_j \|Ps_j - Vs_j(q_t)\|^2 \tag{6.10}$$

With this tracking error measurement function, we can approximate the observation distribution as:

$$p(I_t|q_t) \propto exp\{-d^2(Ps, Vs(q_t))\} \tag{6.11}$$

We can further approximate the observation distribution by keeping only a few modes from the local optimization and constrained inverse kinematics on feature points. Let $\{\mu_k^t, k = 1, \cdots, k = K\}$ denote the set of modes, we can approximate the observation distribution as:

$$p(I_t|q_t) \approx \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \tag{6.12}$$

where, w_k^t can be estimated as:

$$\tilde{w}_k^t \approx exp\{-d^2(Ps, Vs(\mu_k^t))\}$$

$$w_k^t = \frac{\tilde{w}_k^t}{\sum_{k=1}^{K} \tilde{w}_k^t} \tag{6.13}$$

Λ_k^t can be estimated as:

$$\Lambda_k^t \approx (J_{Vs}^T J_{Vs})^{-1} \tag{6.14}$$

6.5 Bayesian Updating and MAP Selection

Given the observation distribution $p(I_t|q_t)$ as Equation 6.12, and the temporal priori $p(q_t|I_1, I_2, \cdots, I_{t-1})$ as Equation 6.8, we obtain the posterior distribution as:

$$p(q_t|I_1, I_2, \cdots, I_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \tag{6.15}$$

In order to avoid the exponential increasing of Gaussian components, without loss of generality, we first approximate it by the first M dominant observation modes as:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{k=1}^{M} \hat{w}_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \tag{6.16}$$

and then re-weight them with temporal prior:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{j=1}^{M} \pi_j^t N(q_t; \mu_j^t, \Lambda_j^t) \tag{6.17}$$

where, the weights π_j^t can be estimated as:

$$\tilde{w}_j^t = \hat{w}_k^t \sum_{j=1}^{M} \pi_j^{t-1} N(\mu_k^t; \mu_j^{t-1}, \Lambda_j^{t-1} + W)$$

$$\pi_j^t = \frac{\tilde{w}_j^t}{\sum_{j=1}^{M} \tilde{w}_j^t} \tag{6.18}$$

At any frame, the optimal pose estimation is exported as the mode in the posterior distribution $p(q_t|I_1, I_2, \cdots, I_t)$.

6.6 Performance Analysis and Experimental Results

The Bayesian pose tracking algorithm is implemented and tested on a set of whole body sequences as from the previous chapter. We summarize and compare its performance with ICP method and feature-based method as in Table 6.3. ICP method utilizes the general correspondences to estimate the pose, which does not require the tracking of feature points. Nevertheless, ICP method could result in tracking failure for the transient occlusions, and not be able to recover from it. Furthermore, ICP method could not be integrated with other information flexibly. The feature-based method is able to track through the transient occlusion, and recover from the tracking failure when the body parts are detected again. But it is not be able to update the pose without the feature points, and cannot integrate with other information either. As seen, Bayesian-based method is able to take advantage of both ICP and feature-based methods. It is able to track through transient occlusions, recover from tracking failure whenever body part are detected again, and update the pose by performing local optimization without features. But more importantly, Bayesian-based method has the potential to integrate other information flexibly whenever available, for example, the pose prediction from machine learning approach. Furthermore, the Bayesian-based method could achieve a higher accuracy for joint trajectories than feature-based method because it could take advantage of ICP to refine the alignment between 3D model and point clouds, as shown in Table 6.4. The obvious limitation for Bayesian-based method is the much higher computational burden, which makes it difficult to be used for interactive applications with current implementation.

Methods	Tracking through occlusion	Error-recovery	Tracking with missing features	Integration with other information	Speed
ICP	No	No	Yes	No	5∼9Hz
Feature-based method	Yes	Yes	No	No	3∼6Hz
Bayesian-based method	Yes	Yes	Yes	Yes	0.1Hz

Table 6.3: Comparison between various human pose estimation approaches

Methods	X trajectory accuracy	Y trajectory accuracy	Z trajectory accuracy
Feature-based method	80mm	84mm	93mm
Bayesian-based method	73mm	78mm	87mm

Table 6.4: A comparison of overall trajectory accuracy between feature-based method and Bayesian-based method

6.7 Summary and Future Work

We have presented a Bayesian method to integrate the pose estimation results from feature-based method and local optimization-based method. This demonstrates a potential approach to integrate the pose estimation results from different modalities to improve the robustness and accuracy of pose estimation. The computational burden of the Bayesian-based method is a major concern for the interactive applications with the current implementation, and it remains as future work to have a parallel implementation to improve the computational speed.

CHAPTER 7

APPLICATION SCENARIOS

In this dissertation, several novel algorithms have been presented for human pose estimation from depth image streams. The methods presented are computationally fast and perform at interactive rates. The recovery of 3D human pose from images has many applications in areas such as video coding, visual surveillance, human gesture recognition, biomechanics, video indexing and retrieval, character animation, and man-machine interaction. This chapter highlights two interactive applications considered in this dissertation. The first application, described in Section 7.1, considers the problem of transferring motion from a human demonstrator to a humanoid robot, an important step toward developing robots that are easily programmable and that can replicate or learn from observed human motion. The second application scenario, as described in Section 7.2, is related to the pose estimation from low dimensional feature points which could be obtained possibly from feature point tracking.

7.1 Online Human to Humanoid Motion Retargeting

Future robots promise to become an integral part of our everyday lives, serving as caretakers for the elderly and disabled, providing assistance in homes and offices, and assisting in surgery and physical therapy. For this to happen, programming must

become simpler, and movements more natural and human-like. In response to this challenge, there has been a growing interest in using captured human motion data as examples to simplify the process of programming or learning complex robot motions [57]. Captured human motion has been used to develop algorithms for 'learning from demonstration', a form of learning whereby a robot learns a task by watching the task being performed by a human [69]. One goal of 'learning from demonstration' has been to replace the time-consuming manual programming of a robot by an automatic programming process, solely driven by showing the robot the task by an expert teacher. Captured human motion has also been used in computer animation to transfer motion of one articulated figure to another figure with a similar structure [85, 84].

Transferring motion from a source system (typically a human) to a target system (typically an avatar or a humanoid robot) is referred to as motion retargeting. This problem has been well studied and several off-line solutions exist based on optimization approaches that rely on pre-recorded human motion data collected from a marker-based motion capture system. From the perspective of applications involving human robot interaction and gesture based tele-operation, there is a growing interest in online motion transfer, particularly without attaching markers to the human demonstrator.

In this section, we revisit the interactive human pose estimation described in Chapter 1.2 for use as an intelligent, gesture based interface for online transfer of human motion to the Honda humanoid robot, ASIMO . The ultimate goal is to interactively teach ASIMO new skills from human observations obtained using a markerless

vision system. Nevertheless, a solution to this problem is useful for other interactive applications such as interaction with virtual objects in a game or virtual reality environments. To the best of our knowledge, there is very few successful systems demonstrating online and marker-less gesture based user interfaces, particularly in applications involving retargeting motion to humanoid robots.

Figure 7.1 illustrates an overview of the online motion retargeting framework. The first step involves visual detection and tracking of a set of 3D anatomical landmarks (or key-points) in the upper-body from depth image observations. Since computation speed is important in this application, we utilize the key-point detection algorithm described in Chapter 3. The detected key-points, registered to a human model, correspond to 3D position vectors at the waist joint, two shoulder joints, two elbow joints, two wrist joints, and the head center (Figure 7.2). The output of the key-point detection module is represented by the vector p_d, where the subscript d denotes detected key-point.

The detected key-points are subsequently low pass filtered and normalized (limb lengths re-scaled) to our humanoid robot model, ASIMO, which has different dimensions, physical parameters, geometry, and degrees of freedom than the human model. Furthermore, the filtered and scaled key-points are up-sampled to a higher rate (100 HZ) to achieve numerical stability and good tracking within the retargeting module. The resulting vector, denoted by the vector p_r represents the reference motion of a key-point mapped to the robot. We will refer to these robot motion variables as task descriptors because they correspond to Cartesian space (or task space) positions and orientations in our proposed task space retargeting framework. Taking the reference

task descriptors as input, the retargeting module outputs kinematically constrained robot joint variables which are issued as commands to the robot.

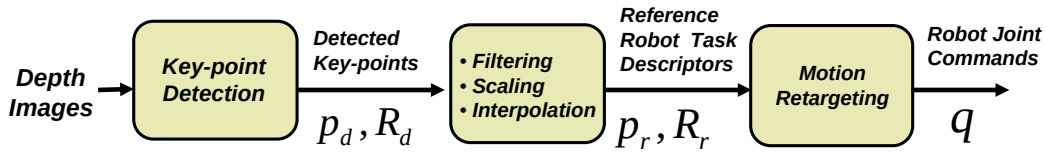


Figure 7.1: System diagram of the entire pipeline.

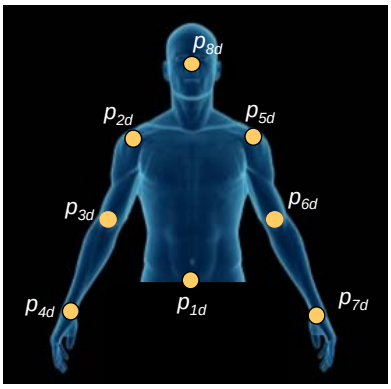
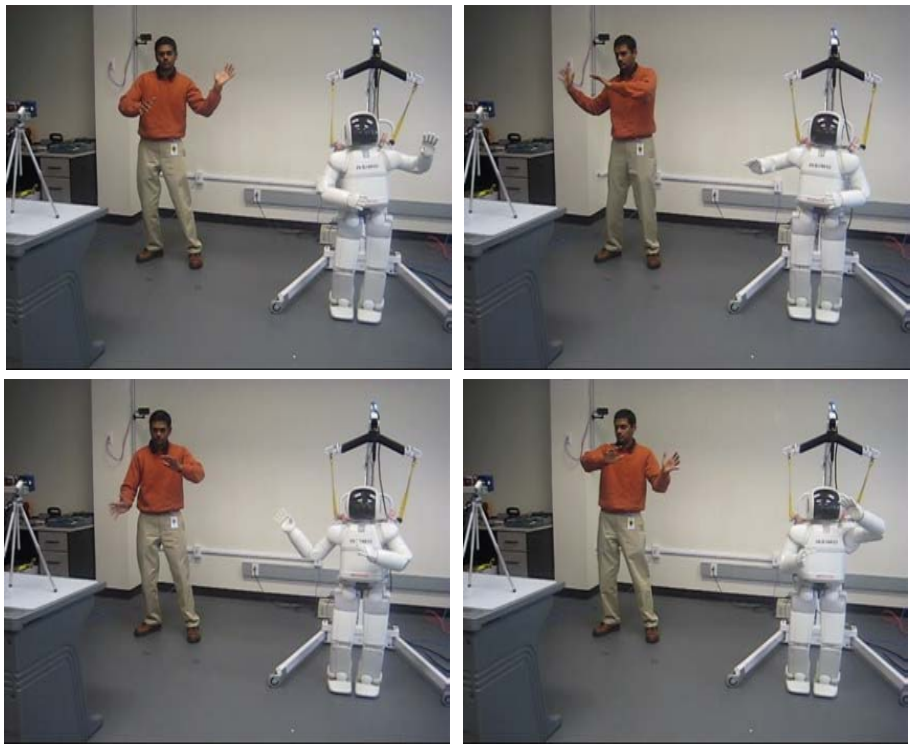


Figure 7.2: Key feature points (key-points) representing position descriptors used in the experiments.

Several experiments were performed on the Honda humanoid robot, ASIMO, using a single time-of-flight range image sensor [83]. The pipeline is illustrated in Figure 1.1. The visual processing module in the pipeline represents the marker-less motion capture methods proposed in this thesis. The intermediate anthropometric scaling and interpolation module will export the feature-points at 100 Hz sampling rate with

the normalized skeletal link sizes that match the ASIMO skeletal link sizes. Collision free retargeting module represents the proposed online retargeting method as in Appendix A that will transfer the observed human motion to the ASIMO model.

Figure 7.3 illustrates snapshots of the online retargeting from two different demonstrators performing a Taiji motion. ASIMO replicates the motion fairly well while enforcing all kinematic constraints including joint limits and self-collision avoidance.



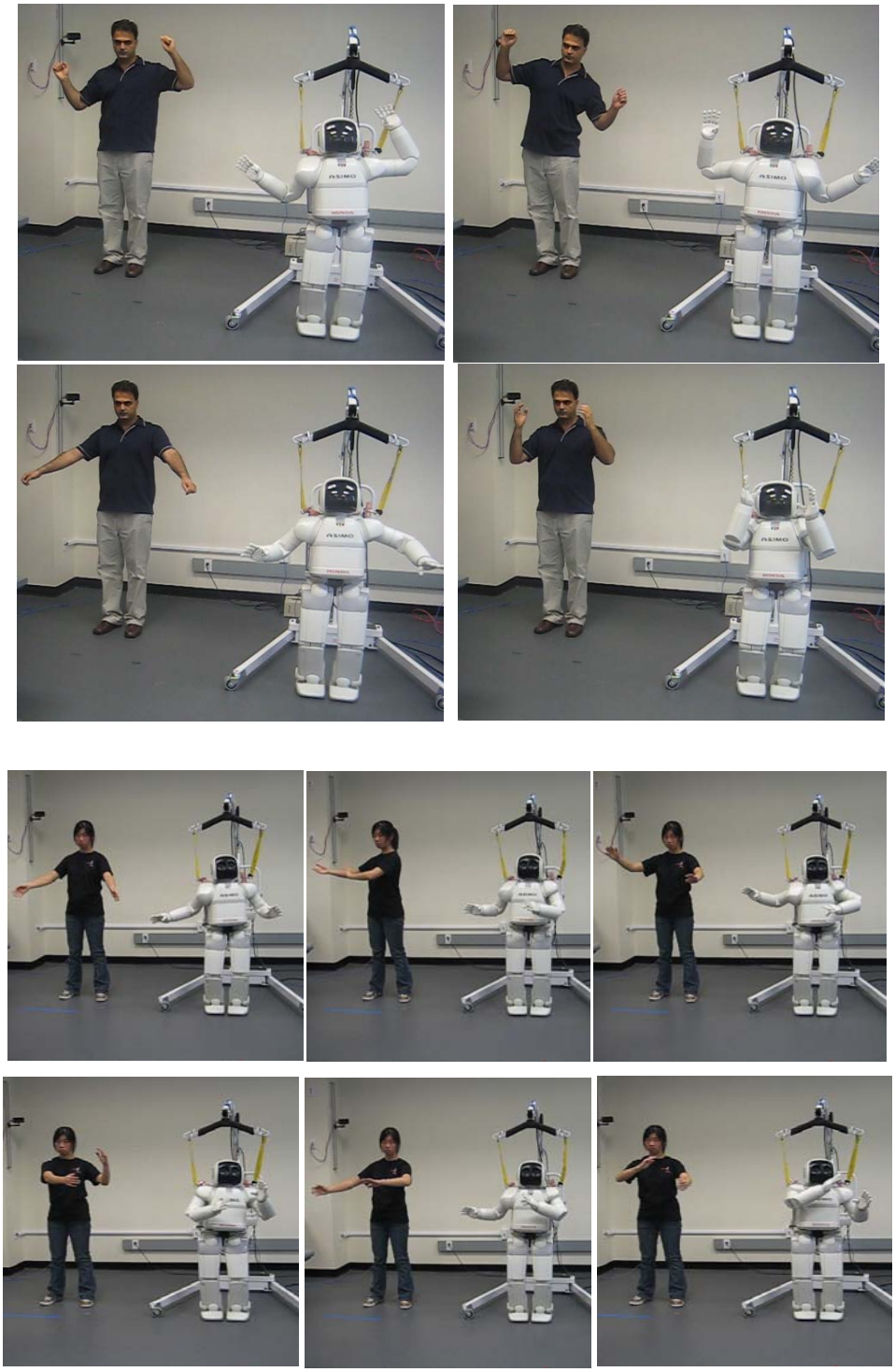


Figure 7.3: Snapshots from online motion retargeting to Asimo

7.2 Pose Estimation with Environmental Clutters

In this section, we would demonstrate that the proposed tracking framework is also appropriate for the pose estimation problem with environmental clutters. By taking advantage of color information from hybrid camera where we calibrate the depth sensor with a color camera, we are able to adapt the dynamic programming based energy-minimization approach for the tree-structured pictorial structure model as described in [29] to perform the feature point tracking by defining the upper body deformable template to have a tree structure.

Let the tree structure be represented as graph with vertices V and edges E, let $L_t(x_i, x_j)$ be pairwise constraint that encodes the relative position between neighboring body parts i and j, and $L_t(I, x_i)$ be body part constraint that encodes the appearance of the body part i. We can define the likelihood function at frame t as:

$$L_t(x) = \sum_{(v_i,v_j)\in E} L_t(x_i, x_j) + \sum_{v_i \in V} L_t(I, x_i) \tag{7.1}$$

Taking advantage of the tree structure, the following four steps are used to find an optimal solution that minimizes the likelihood function [29]:

1. Leaf node v_j with parent node v_i:

$$Q_j(x_i) = \min_{x_i}(L_t(x_i, x_j) + L_t(I, x_j)) \tag{7.2}$$

2. Intermediate node v_j with parent node v_i and child node v_k:

$$Q_j(x_i) = \min_{x_i}(L_t(x_i, x_j) + L_t(I, x_j) + \sum_{v_k \in C_j} Q_k(x_j)) \tag{7.3}$$

3. Root node v_j with parent node v_k:

$$x_j^* = \operatorname*{argmin}_{x_j}(L_t(I, x_j) + \sum_{v_k \in C_j} Q_k(x_j)) \tag{7.4}$$

4. Trace back from the root node to leaf nodes to find the optimal locations

Using the final optimal likelihood $L(x^*)$, we can detect the body parts if $L(x^*)$ is greater than a certain threshold.

In this experiment, Figure 7.4 shows the tree structure of the human upper body. The tree has its root at the torso(TS), and it has child nodes at head(HD),left shoulder(LS), and right shoulder(RS). Left shoulder has the child node at left elbow(LE), and left elbow has the child node at left wrist(LH). Similarly, right shoulder has the child node at right elbow(RE), and right elbow has the child node at right wrist(RH).

Without the loss of generality, we model the pairwise constraint $L_t(x_i, x_j)$ for each tree edge as the negative logarithm of the gaussian distribution:

$$L_t(x_i, x_j) = -log(\frac{1}{\sqrt{2\pi}\sigma_{ij}} exp^{-\frac{(\|x_i - x_j\| - d_{ij})^2}{2\sigma_{ij}^2}}) \tag{7.5}$$

We model the appearance for head as the length of ray radius surrounding the head circle:

$$L_t(I, HD) = -log(\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_{hd}} exp^{-\frac{(r_j - r_0)^2}{2\sigma_{hd}^2}}) \tag{7.6}$$

where r_j is the length of jth ray shooting from head center to the closest edge pixels along the ray. r_0 is the radius of head template circle. Similarly, the appearance for the left shoulder or right shoulder is the length of ray radius along the sampled the shoulder curve point:

$$L_t(I, LS) = -log(\prod_{j=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_{ls}} exp^{-\frac{(r_j - r_0^j)^2}{2\sigma_{ls}^2}}) \tag{7.7}$$

where r_0^j is the length of jth ray from shoulder point to the shoulder template curve as shown in Figure 7.4. r_j is the length of the jth ray from shoulder point to the closest edge pixel along the ray. The appearance for the left or right wrist use the

skin color distribution:

$$L_t(I, LH) = -log(\prod_{j=1}^{M} p_{skin}(r_j, g_j, b_j)) \tag{7.8}$$

where $p_{skin}(r_j, g_j, b_j)$ is any trained skin color distribution, and (r_j, g_j, b_j) is the sampled jth pixel around the hand. We did not try to model the appearance for elbow, since the pairwise constraint enables us to locate the optimal elbow. We learn the parameters for the pairwise constraint and appearance by collecting a few training image, and manually labeling the joint positions.

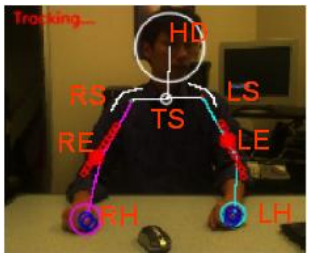


Figure 7.4: An illustration of tree-structured deformable model.

To avoid searching over the entire image, we further constrain the searching domain for each feature point as much as possible based on the image analysis to speed up the feature detection. We constrain the head positions to be located within certain distance to the last head position. With the trained skin color distribution, we can detect the skin blobs through color segmentation and connected component analysis. For example, Figure 7.5 shows the segmented hand blobs after removing the face skin blob. We constrain the hand positions to be within these detected hand blobs. We constrain the elbow position to two possible regions. The first possible elbow region is located within a certain distance to the last elbow position, which is shown as the

large red dot in the Figure 7.5. The second possible elbow region is located along the extended shoulder curve, which is shown as the cross red line in the Figure 7.5.

We export the feature points as the locations of the optimal solution after running the dynamic programming on the constrained domain for each feature point.

Figure 7.5: An illustration of skin blob detection through color segmentation.

Let the set of detected feature points be:

$$\{hd_t, ls_t, rs_t, le_t, re_t, lh_t, rh_t\}$$

We can estimate the pose by applying the online motion retargeting algorithm as described in Chaper 4. Figure 7.6 shows the pose estimation results for two examples sequences. The first sequence shows the pose estimation where the subject manipulates small objects on a desk. The second sequence shows the tracking of variable number of hands.

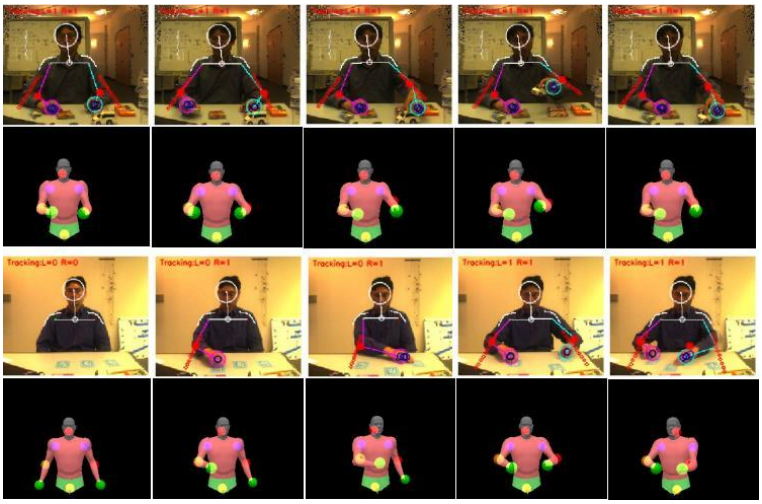


Figure 7.6: Experimental results on pose estimation with environmental clutters.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Markerless human pose estimation is an important problem in computer vision with various applications including, but not limit to, Human Computer Interaction, surveillance, biomechanics, and character animation. Researchers in Computer Vision are actively pursuing markerless human pose estimation systems that are portable, easy to operate, and have real time performance. However, the complexity of the problem makes it difficult. The major challenges are associated with (1) the recovery of the large number of degrees of freedom in movements which are often subject to kinematic constraints such as joint limit avoidance, and self penetration avoidance between two body segments; (2) ambiguities caused by self occlusions as well as the projection of human motion onto the image plane; (3) variations of appearance due to the subject's attire and illumination.

A large number of approaches have been proposed for markerless human pose estimation. We can categorize them into either model-based methods or learning-based methods. The sensing modality used in existing markerless methods may be classified into single passive camera systems, multiple passive camera systems, and time-of-flight (TOF) depth camera systems. The existing methods are not without

limitations. 3D pose estimation with model based approaches can achieve high estimation accuracy. But they often require manual initialization. Error accumulation or numerical drift can occur, and eventually cause the failure of the tracker for the inputs from long sequences. 3D pose estimation with learning-based approaches provides a unified framework for automatic initialization and tracking. But they often obtain error-prone results on the novel inputs. They also have lower pose estimation accuracy than the model-based approaches. Furthermore, not many papers discuss how to recover from tracking error. We observed that an effective solution for the interactive markerless human pose estimation problem should at the very least satisfy the following criteria: (1) be able to automatically initialize the pose estimation (2) be able to estimate the pose for long sequences accurately; (3) recover from tracking error or drift automatically; (4) be able to track through occlusions. To our knowledge, currently there is no 3D human pose estimation system meeting all of the above criteria.

This thesis presents a computational framework for human pose estimation from depth video sequences. On the one hand, feature points that are informative for pose estimation are tracked with depth image analysis. Human poses are reconstructed from these feature points with kinematic constraints including joint limits and self-collision avoidance. On the other hand, human poses could be estimated based on local optimization using dense correspondences between 3D data and the articulated human model. Both could be unified with temporal motion prediction based on Bayesian information integration. Comparing with the existing algorithms for pose estimation from depth streams, the proposed pose estimation algorithms are of great

advantages in terms of robustness, efficiency, accuracy and information integration as shown in Table 8.1.

Methods	Tracking through occlusion	Error-recovery	Tracking accuracy	Integration with other information	Speed
Grest05	No	No	N/A	No	4Hz
Knoop06	No	No	N/A	No	10Hz
Ziegler06	No	No	N/A	No	1Hz
Feature-based method (Chapter 3, 4)	**Yes**	**Yes**	**9cm**	No	3∼6Hz
LP-ICP method (Chapter 5)	No	No	**5cm**	No	5∼9Hz
Bayesian-based method (Chapter 6)	**Yes**	**Yes**	**8cm**	**Yes**	0.1Hz

Table 8.1: Advantages of proposed human pose estimation algorithms over other existing approaches using depth streams

In Chapter 3 and Chapter 4 we have explored the human pose estimation methods from depth image sequences that are capable of detecting, labeling, and tracking body parts, reconstructing the human pose with kinematic constraints. In contrast with the existing markerless motion capture systems, our robust pose tracking method has the following characteristics: (1) track the pose through temporary occlusions between body parts; (2) detect and track limbs that are re-appearing from the previous occlusions; (3) automatically recover from tracking error if a set of specific poses are detected; Chapter 3 presents a method to detect, label and track the body parts for an articulated 2D human figure using depth images. The algorithm is novel to be able

to label and inference the occlusion status of the body limbs based on both temporal tracking information and spatial context information. Chapter 4 and Appendix A describes the constrained closed loop inverse kinematics algorithm to reconstruct the 3D human motion from the trajectories of a set of feature points. This algorithm can enforce both the joint limits constraint and self-collision avoidance in online fashion during pose estimation.

In Chapter 5, local optimization is developed to find the local minima for optimal pose estimation. The estimated motion is more accurate by its nature because we could use dense correspondences to estimate poses.

A feature-based approach is robust and can recover from tracking failure when a body part is re-detected. However, its estimation accuracy depends solely on the feature points. The local optimization based method can achieve a higher accuracy than feature-based method because it utilizes a set of dense correspondences. But, we have found that it is not able to recover from tracking failure once it is trapped in local minima of the energy function. In Chapter 6, we further propose a unified pose tracking based on the Bayesian inference framework to integrate the estimation results from the feature-based method and the estimation results from the local pose optimization method.

Our contributions from this research work, which are listed in Section 8.1, extend the state of the art for interactive human pose estimation in a significant way. Current research work that is being carried out and possible future work are described in Section 8.2.

8.1 Contributions

The main contributions of this thesis include:

1. The depth image analysis is developed to infer the body part occlusions with both spatial and temporal context. It is designated to enhance the robustness of the pose estimation so as to track longer sequences and recover from the tracking failure. A novel head-neck-torso-waist(or head-neck-torso for upper body) template is designed, and enables us to detect those body parts from the depth image. Human limbs are detected based either on the distance transformed skeletons or depth slicing operations that are developed specifically for the depth image. Human limbs are labeled and tracked based on both spatial context and temporal information so as to result in a robust human body tracking method that satisfies the above robust pose tracking characteristics.

2. Human poses are reconstructed from a set of feature points with kinematic constraints. As a result, the animated graphical model satisfies the anthropometrical constraints of the human subject, such as joint limits. Without such constraints, one could possibly result in unnatural poses such as backward bending arms that are not visually appealing as well as physically incorrect.

3. Local optimization is developed to find the local minima for optimal pose estimation. The estimated motion is more accurate by its nature because we could use dense correspondences to estimate poses. Our local optimization for pose estimation is efficient for the interactive applications, and can track longer sequence compared to other similar approaches existing in the literature, although

it has the major limitation that it cannot recover from tracking failure for the complicated motions with difficult poses.

4. The Bayesian framework is explored to integrate information from depth image analysis, model-based local optimization, and temporal prediction. This unified approach is optimal in terms of its tracking accuracy, error recovery ability, and ability to integrate with other pose estimation information, although the computational burden of the Bayesian-based method is a major concern for the interactive applications with the current implementation. It remains as future work to have a parallel implementation to improve the computational speed.

5. A novel collision-free retargeting for interactive robot motion learning is proposed. This algorithm allows us to retarget the observed human motion from the depth sensor to the humanoid robot in real-time and safely (without self-collision between body parts).

8.2 Future Research

While we have presented promising initial results for the human pose estimation from depth sequences, there is scope for improvements. Firstly, there are components to be improved in the current pose estimation method. Secondly, we are striving to define the range of motions that can be handled by our algorithm based on detectability and observability. Thirdly, human pose estimation with descriptive motion constraints is discussed.

8.2.1 Accurate Occlusion Inference between Body Parts

Currently, for the feature-based pose estimation method as described in Chapter 3, we define the occlusion state of upper or lower limb as a binary variable. That is, the whole upper arm(leg) or the whole lower arm(leg) is either visible or not. Although this simplifies the occlusion inference process and speeds up the computation, it can not represent the partial observable limb that exists in real world scenario. This could reduce the tracking accuracy since the partially observed limb is skipped for tracking. An ideal representation would be to represent each limb occlusion state with an occlusion map, where the regions in the occlusion map correspond to the subparts in the upper or lower limb. For such an occlusion map representation, we will need a more differentiable appearance model for each limb subpart (or occlusion region). In contrast to the current depth histogram based appearance model, in the future we could possibly implement it with kernel density [17, 18] for each limb subpart.

8.2.2 Whole Body Pose Estimation with Interactions between Arms and Legs

Furthermore, for the feature-based whole body human pose estimation, we assume that the upper body is separable from the lower body, i.e. there is no interaction between upper body and lower body limbs. This is a valid assumption for normal walking or standing motions, and indeed we are taking advantage of it so as to speed up the limb labeling computation because we only need to differentiate the left arm from the right arm and the left leg from the right leg. However, for more complex motions such as gymnastic exercises, where upper body limbs are likely to interact

with lower body limbs, it remains as future work to extend our current implementation to incorporate such motions.

8.2.3 Constrained Local Optimization for Articulated Model Pose Estimation

Constrained local optimization for articulated model pose estimation as described in Chapter 5) is an accurate approach for pose estimation from depth sequences. One of the major limitations is that the labeling of body parts depends on the estimated pose from the last frame, and it could cause the accumulation of pose tracking error. As a result, the performance of ICP-based pose tracker is degraded after the tracking failure.

Our initial intuition is to use Bayesian inference framework to integrate the estimation results from the feature-based method and the estimation results from the local pose optimization method as described in Chapter 6. Although the pose estimation results from the unified approach are encouraging, the enormous computation burden makes it questionable for the interactive applications.

However, instead of performing the computational expensive Bayesian inference, another viable approach could be to supply the skeletal analysis results as the additional constraints during the linear programming based for coarse body part labeling.

8.2.4 Pose Ambiguity, Detectability, and Observability

We have explored the pose estimation techniques from the depth sequences. Considering the complexity of the human motion in the high dimensional space, they are not without limitations. Here we would make a summary on the poses that could be robustly reconstructed by our algorithm.

1. Pose Ambiguity: Camera perspective projection can cause pose ambiguity where different human poses can have similar observed images. This is an inherent ambiguity when estimating the poses from a single camera. Our algorithm assumes that the observed subject faces the camera.

2. Pose Detectability: Both the feature-based and the constrained local optimization methods would have a better pose estimation performance if the limbs can be detected robustly. This depends on two factors: (a) the distance between body parts $d_{\text{bodyparts}}$, (b) the resolution of the depth sensor r_{sensor}. We define the pose detectability as the ratio: $\lambda = \frac{d_{\text{bodyparts}}}{r_{\text{sensor}}}$. Our algorithm cannot robustly detect limbs from depth images when the detectability ratio λ is smaller than one.

3. Pose Observability: For certain pose, the detected limbs might be severely occluded or too small to be observed. We define the pose observability for a certain body part as the ratio: $\eta = \frac{PA_{\text{bodypart}}}{A_{\text{bodypart}}}$, where PA_{bodypart} is the body part projected area on the image, and A_{bodypart} is the body part surface area. Our algorithm cannot robustly identify the feature points or correspondences for such poses, and consequently cannot robustly reconstruct the poses, where the observability of the body part is too low.

Pose ambiguity might be partially resolved by the incremental torso orientation estimation. Pose detectability partially depends on the hardware, and the undetected limbs might be predicted from the estimated motion history. We can further verify the predicted limb poses by back-projecting to image plane and comparing with the

detected scene edge information. Our future work would like to analyze the observed sequences, find the optimal threshold for the pose observability.

8.2.5 Human Pose Estimation from Depth Video Sequence with Descriptive Motion Constraints

Human pose estimation involves the recovery of a large number of degrees of freedom of an observed articulated human model. In this thesis, we study the approaches to reconstruct the human pose from low-dimensional motion descriptors with human kinematics constraints including joint limit constraint and self-collision avoidance constraint. For some human-computer interaction applications, it is possible that a set of behaviors, normally represented as a set of descriptive motion clips, can be defined beforehand. The existing methods to enforce the constraints arising from the descriptive motions often are based on the matching methods using either the dynamic time warping [45] or the k-nearest neighbor searching on motion graph [63]. Jehee Lee et al [45] described a single camera vision-based avatar control interface by matching the observed visual feature sequences with the captured visual feature sequences using dynamic time warping. It uses the simple global silhouette features and actually has to introduce a three second delay in order to reduce the pose ambiguity. Using the local features from multiple cameras, Ren et al [63] were able to reduce the delay to be less than 1 second for the more complicated dancing motions. The system described by Chai et al [12] estimated the optimal behavior motions using both the optical marker positions computed from a stereo camera and the local matched joint motions from motion graph. This method reconstructs the poses from low dimensional inputs, and is closely related to our approach. However, it relies on markers to obtain the motion

of major human landmarks, and it incorporates the descriptive motion constraints using Levenberg-Marquardt programming method without considering the joint limits and self-collision avoidance.

Instead of performing such time-consuming motion matching, it is of great advantage to incorporate the descriptive motion constraints, in addition to the current kinematic constraints, into our constraint closed loop inverse kinematics formulation that is able to performing the online pose reconstruction. One viable approach is to learn the underlying low dimensional joint motion space by performing PCA analysis using the descriptive motion clips, and estimate the joint motion based on the reduced joint motion space representation from PCA analysis.

8.3 Summary

For the past two decades, markerless human motion capture has been an active research field motivated by various demanding applications, such as surveillance application, control application, and analysis application. In this thesis, we have explored the problem of robust human pose estimation with the consideration of the exciting application for humanoid robot motion retargeting.

Considering the complexity of the human motion, it could result in the poses with low detectability and observability, the robust estimation for such pose is remained as future work. One possible solution is to improve the observability by using more depth cameras viewing from difference viewpoints or to increase the detectability possibly by using the color information.

APPENDIX A

APPENDIX: ONLINE MOTION RETARGETING WITH SELF-COLLISION AVOIDANCE CONSTRAINT

Online motion retargeting is an important topic about transferring motions from one model to another. In animation field, people would like to reuse the exiting motion from the motion capture database so as to save time and cost. In robotics, people would like to drive the robot directly with the human motion instead of tedious manual programming. Both tasks require a motion retargeting module. During our study, we have invented an online motion retargeting method that allows us to enforce self-collision avoidance constraint. Such constraint is extremely important when one try to transfer the existing motion to a robot. A primitive online motion retargeting algorithm could result in self-collision between body parts of a robot. This could damage the expensive robot hardware, and should be avoided. Similarly, For animation application, retargeting results with self-collision are not visually appealing. Our algorithm is a useful tool to improve the visual quality of the retargeted animation.

A.1 Online Motion Retargeting with Self-collision Avoidance Constraint

Let us consider two unconnected rigid bodies, i.e. bodies which do not share a joint, as shown in Figure A.1. In general, Body A and body B may both be in

motion. However, for the simplicity of presentation and without loss of generality, suppose body A is moving toward a stationary body B. Let p_a and p_b represent the coordinates of the shortest distance $d(d \geq 0)$ between the two bodies, described in the base reference frame. Hereafter, we refer to p_a and p_b as collision points. The coordinates p_a and p_b can be obtained using a standard collision detection software. In this work, we use the SWIFT++ library [89].

Let $\hat{n}_a = \frac{p_b - p_a}{|p_b - p_a|}$ be the unit normal vector and $\vec{d} = d\ \hat{n}_a$ the vector from p_a to p_b. Consider a 3D virtual surface surrounding body A, shown by a dashed line in Figure A.1. For every point on body A, its associated virtual surface point is located by the vector $\vec{d_c} = d_c\ \hat{n}$, where d_c is the critical distance, and $\hat{n}$ is the unit normal vector at the surface point. Let p_{vs_a} be the coordinates of a point on the virtual surface of A defined by

$$p_{vs_a} = p_a + d_c \hat{n}_a. \tag{A.1}$$

We define the region between the actual surface of body A and its virtual surface as the critical zone. If body B is stationary, we can redirect the motion at p_a to prevent collision in the critical zone. For now, we consider that the redirection is invoked when $d < d_c$. Later in this section, we will use a blending approach to adjust the initiation of the redirection. In our CLIK control framework, one way to control (or redirect) the motion of p_a is by modifying the trajectory of the desired task descriptor p_r. Let us specify a redirected motion of p_a by p'_a and its associated velocity by $\dot{p}'_a$. The question is, how should we specify the magnitude and direction of $\dot{p}'_a$ to redirect the collision point to prevent the two bodies from penetrating deeper into the critical zone. There is no unique solution. The most straightforward, and perhaps conservative solution is to redirect the collision point in a direction opposite

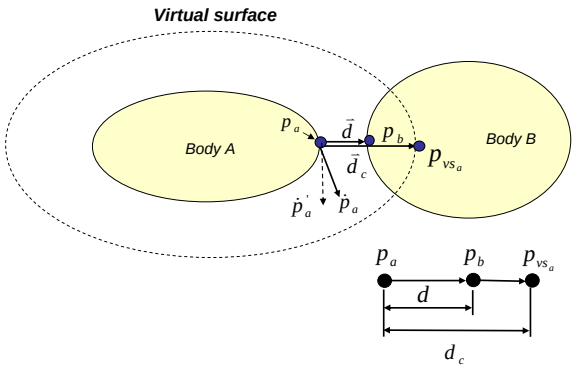


Figure A.1: Body A moving towards a fixed body B

to the unit normal vector $\hat{n}_a$. A more effective strategy is to redirect the collision point so that it slides along a direction which is tangent to the surface at the collision point, as shown in Figure A.1.

$$\dot{p}'_a = \dot{p}_a - < \dot{p}_a, \hat{n}_a > \hat{n}_a. \tag{A.2}$$

In theory, the above redirection vector will guide the collision point motion along the virtual surface boundary, producing a more natural motion toward the target.

To find the mapping between $\dot{p}'_a$ and $\dot{p}_r$, consider first the computation of the equivalent joint velocities which will redirect the collision point velocities along $\dot{p}'_a$. We compute the redirected joint velocity vector using the following mapping,

$$\dot{q}' = J_a^* \; \dot{p}'_a + SJ^*(\dot{p}_r + K \; e), \tag{A.3}$$

where $J_a = \partial p_a / \partial q$ is the Jacobian at the collision point and J_a^* is its weighted Damped Least Squares inverse. The matrix $S = diag(s_1 \cdots s_n)$ is a diagonal selection matrix where $s_i = 1$ when the i_{th} column of J_a has all zero entries and $s_i = 0$ elsewhere. The term $J^*(\dot{p}_r + K \; e)$ is simply the joint velocity solution obtained from

Equation 4.2. The physical interpretation of Equation A.3 is as follows. The first term determines the joint velocities needed to redirect the collision point velocities along $\dot{p}'_a$. Any zero column of J_a (all zero entries) implies that the associated degree of freedom does not contribute to the motion of the collision point. The second term in Equation A.3 is the orthogonal complement of the first term which computes the entries for those joint velocities which do not affect the motion of the collision point(s). Intuitively, it would seem more appropriate to formulate Equation A.3 using a two priority inverse kinematics strategy similar to the control of redundant manipulators [55]. In such a strategy, the first priority term corresponds to satisfying self collision avoidance by redirection (as in first term in Equation A.3). Utilizing redundancy, the second priority term can be constructed to satisfy the requirements for tracking the task descriptors. In practice, this approach leads to jerky behaviors due to numerical instability to arrive at a prioritized solution when multiple colliding pairs enter and exit the critical zone. When there are multiple collision pairs, there is insufficient degrees of freedom to perform the secondary tasks. Numerical instability can also arise since collision points may be discontinuous.

Based on the collision free joint velocity commands computed from Equation A.3, a redesigned position task descriptor trajectory may be computed as follows

$$\dot{p}'_r = J\ \dot{q}'. \tag{A.4}$$

The closed loop inverse kinematics equation with the modified parameters is given by

$$\dot{q} = J^*(\dot{p}'_r + K'\ e'), \tag{A.5}$$

where $e' = p'_r - p'$ and K' is an adaptively changing diagonal feedback gain matrix whose values decrease as the distance d decreases. Note that p'_r at the current time t may be computed by a first order numerical integration.

The instantaneous redirection $\dot{p}_a \rightarrow \dot{p}'_a$, as described above, produces a discontinuous first derivative of p_a at the boundary $d = d_c$. The discontinuity at $\dot{p}_a$ results in a discontinuity in $\dot{p}_r$, as given by the solution in Equation A.4. To preserve first order continuity, we may blend the solutions of $\dot{p}'_r$ before and after redirection occurs. A blended solution to Equation A.4 is given by

$$\dot{p}'_r = (1 - b)\ \dot{p}_r + b\ J\ \dot{q}', \tag{A.6}$$

where b is a suitable blending function such as,

$$b(d) = \frac{e^{-\alpha(d/d_c - \delta})}{1 + e^{-\alpha(d/d_c - \delta)}}, \tag{A.7}$$

where α and δ are scalar parameters used to modulate the blending rate and shift of the blending function, respectively. Figure A.2 shows the plot of b in relation to the ratio d/d_c for $\alpha = 15$. The blending function is plotted for $\delta = .5$ and $\delta = 1.0$. The parameter δ may be used to shift the distance d where blending is initiated and terminated. In the case $\delta = .5$, when $d > d_c$ the function $b(d) \approx 0$, implies that the second term in Equation A.6 is effectively zero so that there is no redirection of the original task descriptor velocity (i.e. $\dot{p}'_r = \dot{p}_r$). At the other extreme, when $d = 0$, the function $b(d) = 1$, implies that the first term in Equation A.6 is zero and the reference trajectory is altered in order to redirect the collision points along the tangent surface. To be more conservative, we may chose $\delta = 1.0$ in the blending function. In this way, blending initiates even before the collision points reach their critical distance. The case when body A is stationary and body B is in motion is the dual of the problem

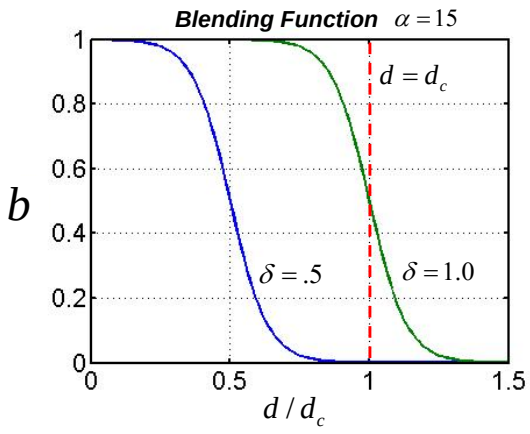


Figure A.2: Blending function at two values of δ

considered above. When both body A and body B are in motion, we can specify the redirection vectors at the collision points p_a and p_b and use task augmentation to control both critical points.

Figure A.3 illustrates snapshots of simulated retargeting results of a fast dancing motion with a full body twisting. The human data were obtained from the CMU motion capture database. These simulated results are generated using the humanoid robot ASIMO's model and geometry. The top row illustrates the results without invoking the collision avoidance algorithm. The colliding body segments, detected using the SWIFT++ collision detection software, are highlighted in yellow. The bottom row illustrates the results of the same motion when the collision avoidance was used.

For the dancing sequence, Figures A.4 and A.5 show the minimum distance between collision points on the left hand and torso segment pairs, and left hand and right hand collision pairs, respectively. The minimum distances are plotted with and

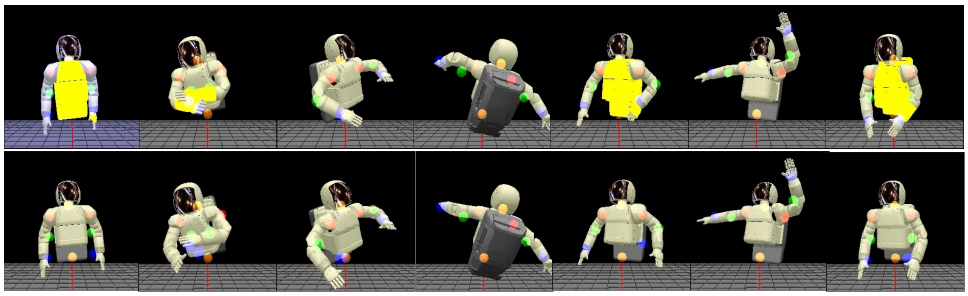


Figure A.3: Snapshots of simulated dancing motion with and without collision avoidance.

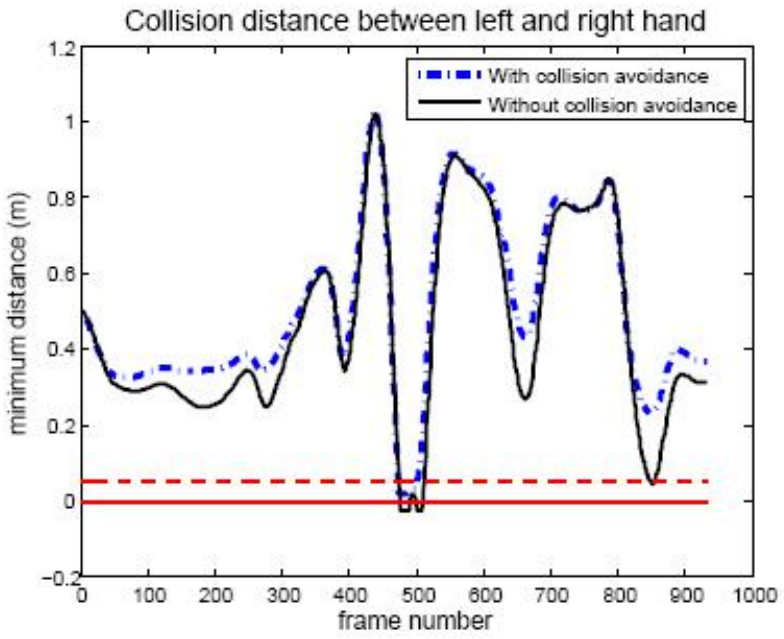


Figure A.4: Minimum distance between left and right hand collision points for a dancing motion. Critical zone is set at .05 meters, and depicted by the dashed line.

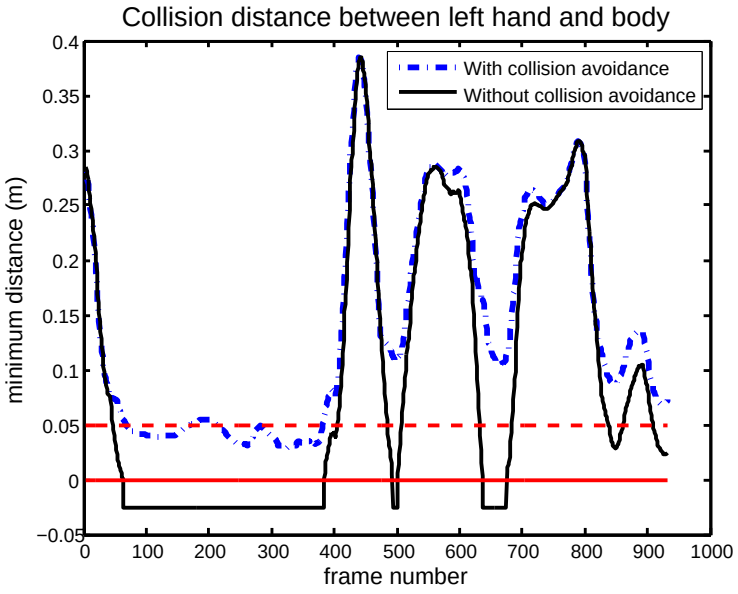


Figure A.5: Minimum distance between left hand and torso collision points for a dancing motion. Critical zone is set at .05 meters, and depicted by the dashed line.

Figure A.6: Snapshots of simulated Taiji motion with and without collision avoidance.

without using the collision avoidance algorithm. Without collision avoidance, the collision points attached to the left hand and torso segment penetrate the collision zone and eventually collide between frames 470 and 505 as shown in Figure A.4. Note that a negative distance implies collision and penetration of the two bodies. Penetration distance is clamped when penetration of the two bodies is beyond $-2.5\ cm$. When collision avoidance is turned on, contact between the two segments does not occur. The blending parameter was set at $\delta = .5$ such that blending is initiated at the critical distance of $d_c = 5.0\ cm$; therefore, collision points are not fully redirected at the virtual surface. Redirection is gradual, and penetration into the critical zone occurs. However, the two bodies do not collide. Figure A.5 shows more dramatic contact between the left hand segment and the torso segment. The collision avoidance algorithm can successfully avoid penetration.

We also performed simulated experiments based on detected features obtained from our markerless, single time of flight camera system. Figure A.6(a) and A.6(b) illustrate two snapshots of a Taiji motion sequence and the corresponding collision detection and avoidance results. In each snapshot, the depth image and the reconstructed human pose are shown in the upper left and upper right image, respectively. The bottom left image illustrates collision between the hand and the torso segment when collision avoidance is turned off. The lower right image illustrates that no collision is detected when collision avoidance is invoked. In all experiments performed using the CMU motion database or motions obtained from time of flight camera system, the retargeted motion with and without collision avoidance are visually very similar.

BIBLIOGRAPHY

[1] Lp solve reference guide. http://lpsolve.sourceforge.net/5/5.

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006.

[3] D. Anguelov, D. Koller, H. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3d range data. *Proc. of Uncertainty in Artificial Intelligence Conference*, pages 18–26, 2004.

[4] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. *ICCV*, 2:432–439, 2003.

[5] Y. Bar-Shalom and T. Fortmann. Tracking and data association. *Academic Press*, 1988.

[6] C. Barron and I. A. Kakadiaris. Estimating anthropometry and pose from a single image. *Computer Vision and Pattern Recognition*, 1:669–676, 2000.

[7] P. Besl and N. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992.

[8] M. Brand. Shadow puppetry. *ICCV*, 2:1237–1244, 1999.

[9] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *CVPR*, pages 8–15, 1998.

[10] Samuel R. Buss. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools 10*, 4:37–49.

[11] J. Carranza, C. Theobalt, M. A. Magnor, and H.P. Seidel. Free-viewpoint video of human actors. *SIGGRAPH*, pages 569–577, 2003.

[12] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. on Graphics*, 24(3):686–696, 2005.

[13] Tat-Jen Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. *CVPR*, 2:239–245, 1999.

[14] T. F. Chan and R. V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Trans. on Robotics and Automation*, 11(2), 1995.

[15] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture. *Comptuer Vision and Pattern Recognition*, 1:77–84, 2003.

[16] G. K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. *CVPR*, pages 714–720, 2000.

[17] D. Comaniciu, V. Ramesh, and P. Meer. real-time tracking of non-rigid objects using mean shift. *CVPR*, pages 142–151, 2000.

[18] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–577, 2003.

[19] M. M. Covell, A. Rahimi, M. Harville, and T. J. Darrell. Articulated-pose estimation using brightness- and depth-constancy constraints. *CVPR*, pages 438–445, 2000.

[20] J. J. Craig. *Introduction to robotics, mechanics and control*. Addison-Wesley, 2nd edition, 1989.

[21] B. Dariush, M. Gienger, A. Arumbakkam, C. Goerick, Y. Zhu, and K. Fujimura. Online and markerless motion retargeting with kinematic constraints. In *Int. Conf. Intelligent Robots and Systems(IROS)*, pages 191–198, Nice, France, 2008.

[22] Q. Delamarre and O. Faugeras. 3d articulated models and multiview tracking with physical forces. *CVIU*, pages 716–721, 2001.

[23] D. Demirdjian, L. Taycher, G. Shakhnarovich, K. Grauman, and T. Darrell. Avoiding the 'streetlight effect': Tracking by exploring likelihood modes. *ICCV*, 1:357–364, 2005.

[24] D. Demirdjian, L. Taycher, G. Shakhnarovich, K. Grauman, and T. Darrell. Avoiding the 'streetlight effect': Tracking by exploring likelihood modes. *ICCV*, 1:357–364, 2005.

[25] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. *CVPR*, 2, 2000.

[26] A. Doucet, N. de Freitas, and N. Gordon. Sequential monte carlo methods in practice. *New York: Springer-Verlag*, 2001.

[27] R. O. Duda, P. F. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2000.

[28] T. Oggier et al. An all-solid-state optical range camera for 3d real-time imaging with sub-centimeter depth resolution (swissrangertm). *SPIE*, 2003.

[29] Pedro F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[30] D. Gavrila and L. Davis. 3-d model based tracking of humans in action:a multi-view approach. *CVPR*, pages 73–80, 1996.

[31] D.M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[32] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. *ICCV*, 1:641– 647, 2003.

[33] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. *DAGM2005*, 2005.

[34] I. Hariatoglu, M. Flickner, and D. Beymer. Ghost3d: Detecting body posture and parts using stereo. *Workshop on Motion and Video Computing*, 2002.

[35] N. H. Howe, M. E. Leventon, and W.T.Freeman. Bayesian reconstruction of 3d human motion from single-camera video. *MERL technical report*, 1999.

[36] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditinal density. *ECCV*, pages 343–356, 1996.

[37] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboar people: A parameterized model of articulated image motoin. *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[38] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. *SPIE AeroSense Symposium*, 1997.

[39] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Proceeding of 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[40] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, 2000.

[41] R. Kehl and L.V.Gool. Markerless tracking of complex human motions from multiple views. *CVIU*, 104(2-3):190–209, 2006.

[42] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric partitioning and markov random fields. *Proc. IEEE Symposium on the Foundations of Computer Science*, page 14–23, 1999.

[43] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. *ICRA*, pages 1686–1691, 2006.

[44] Xiangyang Lan and Daniel P. Huttenlocher. A unified spatio-temporal articulated model for tracking. *CVPR*, 1:722–729, 2004.

[45] Jehee Lee, Jinxiang Chai, Paul Reitsma, Jessica Hodgins, and Nancy Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. on Graphics*, 21(3):491–500, 2002.

[46] M. W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. *CVPR*, 2:334–341, 2004.

[47] M. E. Leventon and W. T. Freeman. Bayesian estimation of 3-d human motion from an image sequence. *MERL technical report*, 1998.

[48] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformations: A unified approach to shape interpolation and skeleton-driven deoformation. *SIGGRAPH*, pages 165–172, 2000.

[49] D. Martin, C. Fowlkes, and J. Malik. Learning to find brightness and texture boundaries in natural images. *NIPS*, 2002.

[50] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2,3):90–126, 2006.

[51] G. Mori and J. Malik. Estimating human body configurations using shape context matching. *ECCV*, 3:666–680, 2002.

[52] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006.

[53] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. *CVPR*, pages 326–333, 2004.

[54] D. Morris and J. Rehg. Singularity analysis for articulated object tracking. *CVPR*, pages 189–196, 1998.

[55] Y. Nakamura. Advanced robotics, redundancy and optimization. *Adisson-Wesley*, 1991.

[56] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and Katsushi Ikeuchi. Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *Int. J. Robotics Research*, pages 829–844, 2007.

[57] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi. Imitating human dance motions through motion structure analysis. In *Intl. Conference on Intelligent Robots and Systems (IROS)*, pages 2539–2544, Lausanne, Switzerland, 2002.

[58] Rick Parent. Computer animation: Algorithms and techniques. *Morgan Kaufmann, San Francisco*, 2001.

[59] Nancy Pollard, Jessica K Hodgins, M.J. Riley, and Chris Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, May 2002.

[60] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. *CVPR*, pages 467–475, 2003.

[61] D. Ramanan and D. A. Forsyth. Strike a pose: Tracking people by finding stylized poses. *CVPR*, pages 271–278, 2005.

[62] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617, 1995.

[63] L. Ren, G. Shakhnarovich, J.K. Hodgins, H.Pfister, and P. Viola. Learning silhouette features for control of human motion. *ACM Transactions on Graphics*, 24(4):1303–1331, 2005.

[64] B. Robins, K. Dautenhahn, R. Boekhorst, and A. Billard. Robotic assistants in therapy and education of children with autism: can a small humanoid robot encourage social interaction skills? *Universal Access in the Information Society (UAIS)*, 4(2):105–120, 2005.

[65] K. Rohr. Incremental recognition of pedestrians from image sequences. *CVPR*, pages 8–13, 1993.

[66] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. *CVPR*, 2:721–727, 2000.

[67] R. Rosales and S. Sclaroff. Learning body pose via specialized maps. *NIPS*, 14:1263–1270, 2001.

[68] A. Safonova, N. Pollard, and J. Hodgins. Optimizing human motion for the control of a humanoid robot. In *Int. Symp. on Adaptive Motion of Animals and Machines (AMAAM2003)*, Kyoto, Japan, 2003.

[69] S. Schaal. Learning from demonstration. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, chapter 9, pages 1040–1046. MIT Press, 1997.

[70] G. Shakhnarovich, P.Viola, and T.Darrell. Fast pose estimation with parameter sensitive hashing. *ICCV*, 2:750–757, 2003.

[71] Chunhua Shen, Anton van den hengel, Anthony Dick, and Michael J. Brooks. 2d articulated tracking with dynamic bayesian networks. *Proceedings of the Fourth International Conference on Computer and Information Technology (CIT04)*, pages 130–136, 2004.

[72] B. Siciliano and J. Slotine. A general framework for managing multiple tasks in highly redundantrobotic systems. In *International conference on Advanced Robotics*, volume 2, pages 1211–1216, Pisa, Italy, 1991.

[73] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. *ECCV*, pages 702–718, 2000.

[74] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *ECCV*, pages 784–800, 2002.

[75] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. I. Isard. Tracking loose-limbed people. *CVPR*, 1:421–428, 2004.

[76] L. Sigal, M. I. Isard, B. H. Sigelman, and M. J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *Advances in Neural Infomation Procesing Systems 16*, pages 1539–1546, 2003.

[77] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *CVPR*, 1:390–397, 2005.

[78] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Learning joint top-down and bottom-up processes for 3d visual inference. *CVPR*, 2:1743–1752, 2006.

[79] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. *CVPR*, 1:18–20, 2003.

[80] C. Stauffer and W.E.L.Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 2:23–25, 1999.

[81] E. B. Sudderth, A. I. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. *CVPR*, 1:605–612, 2003.

[82] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. *CVPR workshop on generative model based vision*, page 189, 2004.

[83] SwissRanger. online time-of-flight camera information from http://www.mesa-imaging.ch/prodviews.php.

[84] S. Tak and H. Ko. A physically-based motion retargeting filter. *ACM Trans. on Graphics*, 24(1):98–117, 2005.

[85] S. Tak, O. Song, and H. Ko. Motion balance filtering. *Comput. Graph. Forum. (Eurograhics 2000)*, 19(3):437–446, 2000.

[86] Leonid Taycher, Gregory Shakhnarovich, David Demirdjian, and Trevor Darrell. Conditional random people: Tracking humans with crfs and grid filters. *CVPR*, 1:222– 229, 2006.

[87] Camillo J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Comptuer Vision and Image Understanding*, 80(3):349–363, 2000.

[88] A. Ude, C. G. Atkeson, and M. Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47:93–108, 2004.

[89] UNC Chapell Hill: Swift++ Library. Speedy walking via improved feature testing for non-convex objects. Internet page. http://www.cs.unc.edu/ geom/SWIFT++/.

[90] S. Wachter and H. H. Nagel. Tracking persons in monocular image sequences. *CVIU*, 74(3):73–80, 1999.

[91] L. Wang, W. Hu, and T. Tan. Recent development in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[92] David A. Winter. Biomechanics and motor control of human movement. *Wiley*, 1990.

[93] Y. Wu, G. Hua, and T. Yu. Tracking articulated body by dynamic markov network. *ICCV*, pages 1094–1101, 2003.

[94] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. *CVPR*, pages 664–665, 1991.

[95] Liang Zhao. Dressed human modeling, detection, and parts localization. *doctoral dissertation, tech. report CMU-RI-TR-01-19,Robotics Institute, Carnegie Mellon University*, pages 94–101, 2001.

[96] Y. Zhu and K. Fujimura. Constrained optimization for human pose estimation from depth sequences. *ACCV*, 2007.

[97] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. *CVPR*, 1:774–781, 2006.