3D-to-2D Spatiotemporal Registration of Long Human Motion Sequences

Ph.D. Thesis Proposal
Submitted to School of Computing

by

Wang Ruixuan
(HT026434B)

under the guidance of

Asso. Professor Leow Wee Kheng

School of Computing
National University of Singapore
December 2004

Abstract

Computer-aided human motion analysis between a 3D reference motion and the motion in 2D video has many potential applications, such as sport and dance coaching, physical rehabilitation, smart surveillance, etc. Compared to the 3D reference motion consisting of a sequence of 3D postures, the human in the videos may move faster or slower, or have different limb rotations. That is, the 3D reference motion and the human motion in the 2D video may have difference in time and space. So the problem is to determine the temporal correspondence between the 3D and 2D sequences, and at the same time, to determine the spatial difference between the posture in the 2D image and the corresponding 3D posture in the reference motion.

In order to investigate methods of solving the proposed problem, two simplified problems are studied in the preliminary work: 3D-2D motion registration of articulated stick figure, and articulated body posture refinement. In the first simplified problem, each body joint position in each image of the 2D sequence is known. Dynamic programming technique is used to find the temporal correspondence between the 3D motion and the 2D sequence of stick figure. And Newton method for optimization is used to find the spatial difference between the corresponding 3D and 2D stick figures. In the second simplified problem, for each 2D image of human body, the body posture in the image is refined starting from an initial posture estimation. Nonparametric belief propagation (NBP) technique is investigated to solve the problem.

Although several algorithms have been investigated to solve the two simplified problems, test results show that the algorithms may not provide accurate results for the problems. So in the proposed continuing work, we would extend the algorithms developed in the preliminary work to first find the approximate temporal correspondence and spatial difference, and then refine the approximate solutions to obtain the detailed spatial difference between any corresponding 3D posture and 2D image and to obtain the accurate temporal correspondence between the 3D reference motion and 2D video sequences.

Contents

1 Introduction 1

1.1 Motivation . . . 1

1.2 Motion Data . . . 2

1.3 Outline of Thesis Proposal . . . 3

2 Problem Description 4

2.1 Input Characteristics . . . 4

2.2 Simplified Problem Definition . . . 5

3 Related work 7

3.1 3D Articulated Body Tracking . . . 7

3.1.1 Problem Description . . . 7

3.1.2 Tracking Algorithm . . . 7

3.1.3 Measurement of Posterior Distribution $P(B_t|\mathcal{Z}_t)$ . . . 9

3.1.4 Search Strategy . . . 10

3.2 3D Articulated Body Posture Estimation from Single Image . . . 12

3.2.1 Problem Description . . . 12

3.2.2 Posture Estimation Algorithms . . . 12

3.3 Video Sequence Alignment . . . 14

3.3.1 Problem Description . . . 14

3.3.2 Sequence Alignment Algorithms . . . 15

3.4 Motion Retargetting . . . 17

3.4.1 Problem Description . . . 17

3.4.2 Retargetting Algorithms . . . 18

3.5 Commercial Training Systems . . . 20

4 Approaches 22

4.1 Problem Definition: Variation 1 22

4.2 Problem Definition: Variation 2 22

4.3 Problem Definition: Variation 3 23

4.4 Main Ideas for Solving Problem Variation 3 24

4.4.1 Initialization 24

4.4.2 Approximate Solution 24

4.4.3 Solution Refinement 25

4.4.4 Handling Local Minima 26

5 Preliminary Work 28

5.1 3D-2D Motion Registration of Articulated Stick Figure 28

5.1.1 Determining Global Rotation and Translation 29

5.1.2 Determining Temporal Correspondence 34

5.2 3D Articulated Body Posture Refinement 36

5.2.1 Graphical Model for NBP 36

5.2.2 NBP for Single Image 38

5.2.3 Test Results 40

6 Proposed Continuing Work 44

6.1 Approximate Registration 44

6.2 Solution Refinement 44

6.2.1 Refinement of T and A Given C 44

6.2.2 Refinement of C 46

6.3 Proposed Schedule 46

7 Conclusion 47

Appendix 48

A Articulated Human Body Model 48

B Belief Propagation (BP) **49**

References **50**

List of Figures

1 Postures of two persons. 1

2 Trajectories in two video sequences. 16

3 Motion retargetting. 17

4 Capture motion using a motion capture system. 20

5 Stick figure of human body. 28

6 Registration error of rigid-body transformation. 31

7 Error of global rotation. 31

8 Error of global translation. 32

9 The mean error of Registration. 32

10 The mean error of global rotation. 33

11 The mean error of global translation. 33

12 Temporal correspondence between two sequences. 35

13 Tree-structured graphical model. 37

14 2D joint position error in articulated posture estimation. 41

15 3D joint position error in articulated posture estimation. 41

16 2D joint position error with respect to joint angle noise. 42

17 3D joint position error with respect to joint angle noise. 43

18 Comparison of silhouette and intensity feature. 45

19 Articulated human body model. 48

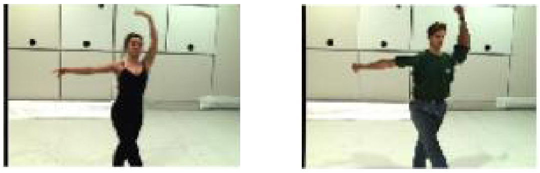
20 Examples of graphical models. 49

1 Introduction

1.1 Motivation

The analysis of human motion by a computer can be applied in many applications, such as sport and dance coaching, physical rehabilitation, smart surveillance, and vision-based intelligent human-computer interaction.

In sport and dance coaching, computer-aided motion analysis is a good way to help coaches assess the movements of athletes and dancers. In general, a coach analyzes a student's motion to determine the parts that require improvement, and then gives suggestions on how to correct and improve the motion. In such a case, a computer can record the motion of an expert and the student (Figure 1) and analyze the difference between them, it can help the coach to analyze the student's motion more precisely. Furthermore, when a novice wants to instruct himself at home without the presence of a coach, a computer can help the novice by indicating the difference between the expert's motion and the novice's motion. Computer-aided motion analysis mcan help novices understand and improve their motion.



(a) (b)

Figure 1: Postures of two persons. (a) An expert's standard posture. (b) A novice's posture that is slightly different from the standard posture.

In physical rehabilitation after injury, a patient requires to repetitively perform some action to help rehabilitate. A computer can help analyze the difference between a patient's motion and a standard motion, and help doctors diagnose the severity of the injury. It can also help the patient to rehabilitate effectively.

Smart surveillance is needed in many environments, such as airports, seaports, supermarkets, departmental stores, and ATMs. In such surveillance applications, one often needs to determine what a person is doing by analyzing the person's motion. By analyzing the input with respect to stored motion models, a computer can help to monitor people's behavior and raise an alarm when suspicious motion is detected.

In vision-based human-computer interaction applications, a computer needs to understand what a person is doing. By analyzing the person's motion, the computer can respond to and interact with the person. For example, a computer may communicate with a person by recognizing the person's hand gesture and reading his sign language. In computer games, the computer may recognize a person's actions and respond accordingly.

In vision-based motion analysis, much work has been done in analyzing human's simple motion, such as walking, running, jogging, etc. However, when human motion is complex, long and cannot be easily segmented into distinct segments, the problem of long-sequence human motion analysis arises. Up till now, long-sequence human motion analysis has not been well studied. If a computer can analyze the detailed difference between the input motion and stored motion models, it will be able to understand more complex human motion. This development will have many application values, for example, in the scenarios described above.

1.2 Motion Data

To analyze human motion, human motion data have to be recorded first. Motion data can be recorded by one or more cameras as 2D video sequences. It can also be captured by a motion capture system as a 3D motion sequence.

Only 2D motion information is recorded in a single 2D video. Depth information of complex 3D motion in, e.g., sports and dances, is lost when the 3D motion is projected onto the camera's 2D image plane. Moreover, some parts of the human body are occluded by other body parts. Thus, single video of human motion does not provide sufficient information for accurate and detailed motion analysis.

When the human motion is recorded by sufficiently many cameras placed at different viewing angles, the multiple-camera system can potentially overcome depth ambiguity and self-occlusion. For example, Gavrila and Davis [GD96] used four calibrated cameras to estimate the 3D motion of hand swaying and complex two-person Tango dance. Deutscher et al. [DBR00] used three calibrated cameras to estimate the 3D motion of walking with turning around.

3D human motion data can be directly captured using a 3D motion capture (MOCAP) system. Three basic kinds of MOCAP systems are available in the market: magnetic, optical, and electro-mechanical. Magnetic MOCAP systems capture body joints' position and rotation by a set of (e.g., 15) cabled magnetic sensors placed on body joints. Optical MOCAP systems use multiple (e.g., 4) cameras to track and estimate the motion of a number of (e.g., 30) reflective markers attached to body joints. Electro-mechanical MOCAP systems come with a suite that is worn by a person. They use potentiometers to measure the rotation of each body part by changes in voltages caused by the rotation of the rods connecting adjacent body joints. All the MOCAP systems require precise calibrations.

In comparison, a single-camera system is cheap but does not provide complete 3D motion information. A multiple-camera system can potentially provide complete 3D information if sufficiently many cameras are used and the cameras are properly placed and set up to overcome self-occlusion. A magnetic MOCAP system is cheap (about US$40,000), but it can only capture 3D motion information in a small range (about 5m) because of the cables and space-limited magnetic field. An optical MOCAP system can capture 3D motion in a large range (about 10m if enough cameras are provided) but tends to be expensive (e.g., about US$100,000) and requires constrained environment and precise multiple-camera calibration. An electro-mechanical MOCAP system is cheaper (e.g., about US$30,000) than an optical MOCAP system. At the same time, it has a very wide operational range (about 30m), and is suitable for capturing complex and long-sequence motion.

In consideration of the above observations, we shall adopt the following setup in our studies. The 3D motion model of an expert will be captured using an electro-mechanical MOCAP system to obtain complete 3D motion information. This is done only once, so the time and effort put into capturing accurate data is not a major issue. On the other hand, to be practically affordable to general users, the novice's input motion will be captured by one or more cameras, depending on how many the novice can afford. If one camera is used, it is comparatively easy to be calibrated but depth ambiguity and self-occlusion may occur. If multiple cameras are used, it can alleviate or avoid depth ambiguity and self-occlusion but requires additional camera calibrations.

1.3 Outline of Thesis Proposal

In the following sections, we first formulate the problem of human motion analysis in our context (Section 2). This provides a concise definition of what we propose to study, and clarifies the concepts used in the following discussion. Then, existing work related to the proposed work is discussed and compared in Section 3. Next, possible approaches of solving the proposed problem are presented (Section 4), followed by preliminary work done and results obtained (Section 5). Section 6 outlines the remaining work to be carried out to complete the proposed work, along with the schedule. Finally, Section 7 concludes the thesis proposal.

2 Problem Description

To clearly describe the problem, it is necessary to first describe the characteristics of the inputs to the problem (Section 2.1). Due to the complexity of the problem, a simplified problem definition is given in Section 2.2 to clarify the major research issues involved. The actual problem to be solved is discussed later as problem variations in Section 4.

2.1 Input Characteristics

There are two inputs to the problem: 3D motion model and 2D input video.

3D Motion Model

The 3D motion model includes two independent parts and an optional part:

1. H: human Body Model
   This includes the general shapes and sizes of the human body parts, joints that connect the body parts, and the constraints on the joint rotation angles.
2. M: 3D Reference Motion
   $M = \{\mathbf{p}_t, \boldsymbol{\theta}_t, \boldsymbol{\omega}_{it}\}$, where $\mathbf{p}_t$ is the position and $\boldsymbol{\theta}_t$ is the global rotation of the human body in the world coordinate system at discrete time t. $\boldsymbol{\omega}_{it}$ denotes the 3D angles of joint i at time t in the local (limb) coordinate system.
3. Optional Constraints
   These include the relationships between body parts, or between body parts and the environment. For example, at a certain time instant, the two hands touch each other and one foot touches the ground.

Using H and M, we can compute the model posture, position, and global orientation, denoted as B_t, at time t. The sequence of B_t is called the *reference motion*. The optional constraints are not considered in the current proposal.

2D Input Video

Input video m'_k recorded by the k^{th} camera consists of a sequence of image frames $I'_{kt'}$ over time t'. Each image $I'_{kt'}$ contains a person in a certain posture.

The inputs have the following characteristics:

1. Based on currently available hardware technologies, we can assume that the reference 3D motion is sampled at a higher rate than the input 2D motion. For example, our Gypsy4 MOCAP system captures 3D motion at 120Hz whereas the video camcorders capture at 25 frames per second. So, it is necessary to establish a temporal correspondence between 2D video time t' (i.e. frame number) and 3D motion time t. Let C denote the mapping function from t' to t. Note that C is not a linear function because of possible differences in speed and duration of movement between the reference motion and the input motion. For example, compared to the reference motion, the human in the input video may move faster or slower, or have different limb rotations. In general, C should satisfy the temporal ordering constraint, i.e., for any two temporally ordered postures in the input motion, the two corresponding postures in the reference motion have the same temporal order.

2. In general, the human body model H and the human in the input video m' can have different body size and limb lengths. To match them, it is necessary to adjust for differences in body size and limb lengths. Denote the adjustment function by G, and the adjustment of H by $G(H)$.

3. The human motion in the input video is similar to the 3D reference motion. But there can be large differences between them in terms of direction, speed, and duration of movement of the limbs. These differences can be represented by the global translation and rotation of the 3D human body denoted by T and the articulation of joints denoted by A.

4. Let P_k denote the projection of the 3D human model to the image plane of the k^{th} camera, which includes camera interior and exterior parameters. It can be assumed that the camera interior parameters are fixed and known.

2.2 Simplified Problem Definition

The problem of interest is to determine the temporal correspondence between t and t' and to compute the difference between input posture and reference posture at the corresponding time. Suppose we can extract feature points from the input images such that these feature points include the joints of the human body. Then, the problem can be defined as one of finding a spatial corresponding between the 3D joint positions and the possible 2D joint positions at the corresponding time. Let $\mathbf{X}_{it}$ denote the 3D position of joint i computed from $G(H)$ in M at time t. Let $\mathbf{x}_{kjt'}$ denote the 2D position of feature point j extracted from image $I'_{kt'}$ at time t'. Let g denote spatial the correspondence function from 3D joint i to 2D feature point j. Then, the problem of matching the 3D reference motion and the 2D input motion can be defined as the following spatiotemporal registration problem: determine the functions G, P, T, C and g

that minimize the error E:

$$E = \sum_{k} \sum_{t'} \sum_{i} \| P_k(T_{C(t')}(\mathbf{X}_{iC(t')})) - \mathbf{x}_{kg(i)t'} \|^2 \,. \tag{1}$$

From this problem definition, we can see that matching the 3D reference motion and the 2D input motion requires, at least, solving the spatial registration problem (i.e., finding P, T, and g), the temporal correspondence problem (i.e., finding C), and finding the correct body adjustment function G.

Note that this is only a simplified problem definition that is used to illustrate the nature of the spatiotemporal registration problem, i.e., finding G, P, T, C and g. The definition of the actual problem to be solved will be described in Section 4.

3 Related work

Some commercial products have been developed to determine the difference between two human motion sequences (Section 3.5). However, they require that two motion sequences to be compared are both 3D, which have to be captured by a 3D motion capture system. To the best of our knowledge, no research has been done on spatiotemporal registration between 3D human motion and 2D videos. Nevertheless, several research topics have close relationships with the proposed problem, including 3D articulated body tracking (Section 3.1), 3D articulated body posture estimation from single image (Section 3.2), video sequence alignment (Section 3.3) and motion retargetting (Section 3.4).

3.1 3D Articulated Body Tracking

3.1.1 Problem Description

The problem is to estimate 3D body posture of articulated objects (e.g., human body or hand) from single or multiple image sequences [SBF00, DBR00, ST01, ST03, SBR$^+$04, SMFW04]. Compared to the proposed problem, this problem does not make use of a reference motion M, but an articulated human body model H is often required. In addition, there is no temporal correspondence problem because the multiple image sequences are synchronized. Furthermore, prior knowledge and constraints Q are often used to obtain more accurate solutions. For example, the articulated body should not be self-intersecting, the range of joint angles are limited, the change of joint angles between adjacent frames is limited to a small range, etc. As a result, articulated body posture estimation can be viewed as finding adjustment function G, projection P_k, rigid transformation T_t, joint articulation A_t that minimize E, subject to the constraints Q:

$$E = \sum_k \sum_t \parallel f(P_k(T_t(A_t(G(H))))) - f'(I'_{kt}) \parallel^2 \tag{2}$$

where f and f' are feature extraction functions, e.g., edge detection functions. The recovered T_t and A_t give the position, orientation and posture of the input human at each time t.

3.1.2 Tracking Algorithm

In practice, 3D articulated body tracking is often formulated in the Bayesian framework [SBF00, DBR00, ST01, ST03]. In this framework, 3D articulated body tracking is to estimate the 3D body posture from each image based on the current and all previous images. Let the estimated 3D body posture at discrete time t be denoted B_t and its history $\mathcal{B}_t = (B_1, B_2, ..., B_t)$. Let the image features at time t be Z_t with history $\mathcal{Z}_t = (Z_1, Z_2, ..., Z_t)$. The problem is to obtain a good posterior $P(B_t|\mathcal{Z}_t)$ for estimating the 3D body posture at time t. In order to solve the problem, it is usually reasonable to assume that (1) B_t is conditionally independent of B_{t-2} and $\mathcal{Z}_{t-1}$ given B_{t-1} and (2) Z_t is conditionally independent of B_{t-1} and $\mathcal{Z}_{t-1}$ given B_t. Using the assumptions and Bayes Rule, it can be shown that [IB96]

$$P(B_t|\mathcal{Z}_t) = k_t P(Z_t|B_t)P(B_t|\mathcal{Z}_{t-1}) \tag{3}$$

where k_t is a normalization constant that does not depend on B_t, and $P(B_t|\mathcal{Z}_{t-1})$ is given by the equation

$$P(B_t|\mathcal{Z}_{t-1}) = \int P(B_t|B_{t-1})P(B_{t-1}|\mathcal{Z}_{t-1})dB_{t-1} \tag{4}$$

From Equation 3, we can see that the posterior probability distribution $P(B_t|\mathcal{Z}_t)$ can be obtained from the likelihood $P(Z_t|B_t)$, the state transition probability distribution $P(B_t|B_{t-1})$, and the previous posterior probability distribution $P(B_{t-1}|\mathcal{Z}_{t-1})$.

The posterior $P(B_t|\mathcal{Z}_t)$ is often multi-modal because of two reasons. First, $P(B_t|B_{t-1})$ may be non-linear because of complex body motion in the tracking problem. It usually makes $P(B_t|\mathcal{Z}_{t-1})$ be multi-modal. Second, the likelihood $P(Z_t|B_t)$ are often multi-modal due to self-occlusion, depth ambiguity, and clutter when tracking an articulated body.

The tracking algorithms have to account for the multi-modality of the posterior $P(B_t|\mathcal{Z}_t)$. Kalman filter and extended Kalman filter are not suitable for 3D body tracking because they assume the uni-modal distribution of $P(B_t|\mathcal{Z}_t)$. CONDENSATION [IB96] provides a general mechanism to deal with the non-Gaussian distribution of the posterior. The CONDENSATION algorithm has the following two main steps:

1. Predict probability distribution $P(B_t|\mathcal{Z}_{t-1})$ of new state B_t:

   Draw samples $\mathbf{s'}_t^{(n)}$ by sampling the previous posterior $P(B_{t-1}|\mathcal{Z}_{t-1})$ to represent $P(B_t|\mathcal{Z}_{t-1})$.

   Generate samples $\mathbf{s}_t^{(n)}$ by sampling from $P(B_t|B_{t-1} = \mathbf{s'}_t^{(n)})$.

2. Measure and update prediction:

   Set a weight to each sample $\mathbf{s}_t^{(n)}$ by measuring $P(Z_t|B_t = \mathbf{s}_t^{(n)})$.

Although the original CONDENSATION can deal with multi-modal posterior, it may not be suitable for dealing with the high-dimensional 3D articulated body tracking. When the dimension increases, the number of samples required to represent the complete posterior increases exponentially [FP03b]. Since articulated body often has high degree of freedom (e.g., about 30 DOF), the traditional CONDENSATION is not practical under this condition. As a result, extensions and variations of CONDENSATION are proposed to deal with 3D articulated body tracking [DBR00, ST01].

There are two main issues in the extended CONDENSATION algorithms. One is how to measure the posterior $P(B_t|\mathcal{Z}_t)$ (Section 3.1.3). Another is the search strategy to find the good estimations of state B_t (Section 3.1.4).

3.1.3 Measurement of Posterior Distribution $P(B_t|\mathcal{Z}_t)$

Since $P(B_t|\mathcal{Z}_t)$ is multi-modal and unknown in advance, $P(B_t|\mathcal{Z}_t)$ is often represented non-parametrically by a set of weighted samples. However, due to the high dimensionality (e.g., 30) of the body posture's state space, using a limited number of weighted samples is difficult to represent the complete posterior $P(B_t|\mathcal{Z}_t)$. Instead, some researchers use the weighted samples to represent just the significant peaks and points around the peaks of the posterior [CR99, DBR00, ST01].

Using this representation, one additional search step (Section 3.1.4) is required after predicting the prior $P(B_t|\mathcal{Z}_{t-1})$ from $P(B_t|B_{t-1})$ and previous posterior $P(B_{t-1}|\mathcal{Z}_{t-1})$. Firstly, $P(B_t|B_{t-1})$ can be derived from the human body dynamical model $B_t = \mathcal{F}(B_{t-1}) + \mathcal{G}w$ where $\mathcal{F}$ and $\mathcal{G}$ obtained in advance represent the deterministic and stochastic components of the dynamics and w is the independent random noise variable. Then, $P(B_t|\mathcal{Z}_{t-1})$ is predicted from $P(B_t|B_{t-1})$ and previous posterior $P(B_{t-1}|\mathcal{Z}_{t-1})$ (described in CONDENSATION). Finally, a search step tries to find the peaks of the likelihood function $P(Z_t|B_t)$ starting from the samples of $P(B_t|\mathcal{Z}_{t-1})$. In the remainder of this section, we describe the method of likelihood estimation. And in next section (3.1.4), we describe search methods in the search step.

The likelihood can be estimated by similarity measure between the body posture B_t projected into 2D image plane and the human figure in the input image. Many image features can be used to measure the similarity, such as edge [GD96, DBR00, WN99, ST01], intensity [BM98, SBF00, WN99, ST01], and silhouette [DF99, DBR00, ST01]:

1. Edge: It can be easily detected and partially invariant to viewpoint and lighting, but it cannot reflect changes due to rotation of a limb about its 3D symmetrical axis, which is along direction of the limb.

2. Intensity: It may be able to capture axial limb rotation, but it is sensitive to lighting, deformable clothing, or lack of image texture, in which cases it will decrease tracking performance and lead to tracking drift.

3. Silhouette: It provides strong global information if it can be obtained from the image, but it cannot reflect limb information if the limb is projected inside the body contour.

In practice, the above image features are often combined to measure the similarity [WN99, DBR00, SBF00, ST01].

3.1.4 Search Strategy

After the likelihood can be estimated for any body posture state, efficient search strategy is required to find the peaks of the likelihood function in the state space. Continuous local optimization methods (e.g. Newton method) are often used. Global optimization methods can also be used, which include multiple random start, sampling methods (including regular and stochastic sampling), smoothing method, simulated annealing, tabu search, etc. Furthermore, Nonparametric belief propagation (NBP) is a new search strategy that can search in a lower dimensional state space.

From the likelihood function, a cost function $c(B_t)$ can be easily obtained, e.g., $c(B_t) = \exp\{-P(Z_t|B_t)\}$. So finding the peaks of the likelihood function is equivalent to finding the minimum of the cost function $c(B_t)$. In addition, prior knowledge or constraints can be combined into the cost function. The constraints can be used to reduce the valid search region in the state space.

Continuous Local Optimization Method

Continuous local optimization methods [BM98, DCR01] incrementally update an existing state estimate, e.g., using the gradient direction to guide the search direction toward a local minimum. Local optimization methods can find local minima, but cannot guarantee global optimality.

Multiple Random Start

Multiple random start [CR99, ST01] first selects random starting points in the search space and then performs local optimizations from these points. The local minimum with the smallest cost is considered as the global minimum. This method is the simplest global optimization method but it cannot guarantee that the global minimum is found.

Sampling Methods

Regular sampling method evaluates the cost function at a predefined region of points in the state space, e.g., a local rectangular grid [GD96] around a point in the state space. From the sampled state points, it selects the state point which has the minimal cost. After that, a new round of sampling around the newly selected point can be performed iteratively.

Stochastic sampling method generates random sampling points according to some probability distribution encoding “good places to look”. The distribution can be simple, such as Gaussian distribution [KM96]. It is straightforward to sample from such distribution. The distribution can also be complex in which direct sampling may be difficult. In such a case, importance sampling is often used [SBF00, DBR00]. From the sample points, the stochastic sampling method selects the point that has the minimal cost value as the search result. This method can also be iterated.

Densely sampling the entire state space would guarantee a good solution but is infeasible in a space with more than two or three dimensions. In order to obtain good state estimation in high dimensional state space, some researchers try to balance continuous local optimization method and sampling methods. For example, in [CR99, ST01], they first sampled a set of initial state points using sampling methods. For each sampled state point, they used local optimization method to find a corresponding point where the cost function is a local minimum. Then, among the set of local minimum points, they select the point at which the cost function is minimum. The combination of random sampling and local optimization can be viewed as the extension of multiple random start method.

Smoothing Method

Smoothing method [MW97] tries to smooth the rugged surface of the cost function such that most or all local minimum disappear. Then, the remaining major features of the surface only show a single or only a few minima, in which case local optimization methods can be used to find these minima. After that, by adding more and more details, the approximations made by the smoothing are undone, and finally one ends up at the global minimum of the original cost function surface. This method can often find the global or at least a good local minimum.

Simulated Annealing

Simulated annealing [Neu03] simulates the process of metal heating and slow cooling which brings the metal a more uniformly crystalline state with global minimum energy. In the process, the role of temperature is to allow the configurations to reach higher energy states with a probability given by Blotzmann's exponential law, such that they can overcome energy barriers that would otherwise force them into local minima. This method can converge in a probabilistic sense but is often very slow.

Tabu Search

Tabu search [Neu03] is to 'forbid' new search points to be in the regions that have already been searched in the search space. This method can avoid being trapped in a local minimum and lead to exploring new regions, such that it has more probability of finding the global minimum. It is often used together with other search methods discussed above.

Nonparametric Belief Propagation

High dimensionality of the state space is a main cause of the difficulty in 3D human body tracking. It is helpful to solve the problem in lower dimensional subspaces. Nonparametric belief propagation (NBP) [SIFW03, Isa03, SMFW04] is a new idea to solve 3D articulated body tracking in low (i.e., 6) dimensional state subspaces. It represents articulated body and the relationships between body parts by a graphical model. Every body part is encoded by one node in the graph, and every edge connecting two nodes indicates that there are relationships between the two nodes. Instead of directly calculating the posterior $P(B_t|\mathcal{Z}_t)$, the method calculates the conditional marginal distribution of each graph node by propagating information between nodes. Since body part's state is a lower dimensional state, the 3D tracking problem has been transformed from estimating a single high dimensional body posture state to estimating a set of lower dimensional posture states of body parts.

The advantage of NBP is clear. In the lower dimensional state subspaces, a limited number of weighted samples can be used to represent the distribution of each body part configuration. The number of required samples increases *linearly* with respect to the number of body part. Furthermore, prior constraints may be more easily represented in NBP [SMFW04] compared to other algorithms.

However, it is difficult for NBP to deal with self-occlusion [SMFW04]. In NBP, each node (or body part) has its own likelihood function that is estimated by the similarity between samples and the corresponding image part. If the body part is partially or fully occluded by other body parts, the likelihood function cannot be correctly estimated by the similarity. Only if the tracked motion is known and simple such as walking, the observation function can be learned from the motion model to deal with self-occlusion [SBR+04]. But this is not general because such motion model is generally unknown in 3D human body tracking.

3.2 3D Articulated Body Posture Estimation from Single Image

3.2.1 Problem Description

The problem is to estimate 3D body posture of articulated objects (e.g. human body or hand) from single image [HLF99, Bra99, AS00, RS00a, RAS01, MM02, AS03, GS03, SVD03, AT04, EL04, AASK04]. Compared to the proposed problem, this problem does not require an explicit 3D motion model which includes both H and M, and there is no temporal correspondence problem because of single image.

3.2.2 Posture Estimation Algorithms

Since single image does not provide enough information for estimating the 3D body posture, a model should be provided in advance. This model may be simple, e.g., a large set of exemplars. It can also be trained, e.g., a non-linear mapping between image features and the 3D body posture.

Compared to 3D articulated body tracking, both exemplar-based and mapping function-based methods can avoid the need for explicit initialization and 3D body modelling and rendering. However, the methods are limited to recover a small set of body posture which has been stored or learned.

Exemplar-based method

Exemplar-based method [AS00, MM02, SVD03, AS03, AASK04] stores a set of exemplar images whose 3D posture is known, and estimates posture by searching for exemplars similar to each image. Since multiple body posture may have very similar corresponding images, this method often outputs multiple 3D body posture estimations for each image.

Because matching the image and each exemplar is often computationally expensive, researchers often save the computation by constructing an embedding [FL95, HS03, AASK04]. Embedding [TdSL00, RS00b] technique maps a point in the image space into another low-dimensional space, such that the similarity measurement between images can be efficiently computed in the embedded low space.

Exemplar-based method has its own advantage and disadvantage. It does not need to train a complex model. But it needs to store a large memory of exemplars. Since the exemplars record only a limited number of body posture information, it is not possible to obtain a good posture estimation if the body posture in the input image is different from those in the exemplars.

Mapping Function-based Methods

These methods learn a nonlinear mapping function that represents the relationships between body image features and 3D body posture in the corresponding images. During learning, a rich set of image features (e.g., silhouette [EL04], histogram of shape context [AT04]) are extracted from each training image as the input, and the output is the known 3D posture in the corresponding training image. During posture estimation, the features in the input image is extracted and then input to the mapping function, and the output is the corresponding body posture estimation.

Agarwal and Triggs [AT04] used 100-dimensional local shapes of a human image silhouette as the input vector, and 55-dimensional 3D full body posture features as the output. Given a set of labelled training examples, they used relevance vector machine [Tip00] to learn a nonlinear mapping function that consists of a set of weighted basis functions.

Rosales et al. [RS00a, RAS01] used Hu moment of the body image as the input vector, and 22 joint angles as the out vector. Given the training set, this method learned a set of forward mapping functions, each of which is a combination of sigmoidal and linear functions, using EM technique. Using their algorithm, a complex many-to-many mapping can be obtained which mainly consists of the combination of the learned mapping functions.

Recently, manifold is used in posture estimation. A manifold is a topological space that is locally Euclidean. If the input image comes from a known type of 3D motion model (e.g. walking), the 3D motion model can be represented as a nonlinear manifold in a high-dimensional space. By mapping the manifold into a lower dimensional space using embedding technique, and learning the two nonlinear mappings between the embedded manifold and both visual input (i.e., silhouette) space and 3D body configuration (i.e., body posture) space, 3D body posture can be estimated from each input image by the two mapping functions. Elgammal and Lee [EL04] used Generalized Radial Basis Function (GRBF) interpolation framework for the nonlinear mapping.

Mapping function-based methods can directly estimate body posture from single image. However, they also have shortcomings. Because the true distribution of the high-dimensional articulated body posture is very complex, such methods may only deal with a small set of body posture. Also, when using manifold for learning mappings, the methods [EL04] are limited to recovering the body posture which is similar to those in the 3D motion model.

3.3 Video Sequence Alignment

3.3.1 Problem Description

Video sequence alignment [Ste98, GP99, LRS00, CI00, CI01, CI02, CSI02, RGSM03] is to establish 2D image point correspondence both in time and in space between two video sequences. Suppose one video sequence is $I_t, t = 1, ..., l$ and another is $I_{t'}, t' = 1, ..., l'$, video sequence alignment can be viewed as finding spatial transformation T (between two camera image planes) and temporal correspondence function C that minimize E.

$$E = \frac{1}{l'} \sum_{t'=1}^{l'} \| f(T(I_{C(t')})) - f(I_{t'}) \|^2 \tag{5}$$

Compared to the proposed problem, video sequence alignment just need to find a constant spatial transformation T and a temporal correspondence C between two video sequences. It does not use 3D motion information. Therefore, it cannot analyze the detailed difference of two human motion in two video sequences.

In video sequence alignment in which the sequences capture the same motion, it is often assumed that the cameras are far from the object points so that object motion can be viewed as roughly planar. Under this assumption, spatial transformation T between two video sequences can be considered as a homography [Fau93]. However, it does not mean that the two video sequences cannot capture 3D motion. When the two cameras capture 3D motion, essential or fundamental matrix can be used to approximate the spatial transformation T [RGSM03].

In order to solve the problem, feature points are first extracted in both video sequences. T

and C are then determined using corresponding feature points.

3.3.2 Sequence Alignment Algorithms

Before sequence alignment, 2D feature points are often tracked in each video sequence. According to whether two video cameras record the same motion of the same person or the motion of two persons, two main kinds of methods are used respectively: linear temporal correspondence and dynamic time warping.

Linear Temporal Correspondence

When the two video cameras are fixed during recording the same motion, it is reasonable to assume that there is a linear temporal correspondence C between the two video sequences, i.e., $t = C(t') = st' + \triangle t'$, where s denotes the ratio of frame rates of the two cameras [Ste98, CSI02].

Trajectory correspondence is often used instead of point correspondence when solving video sequence alignment [CI02, CSI02, RGSM03]. In this case, each motion is considered to be composed of a set of feature point trajectories. Each feature point trajectory is a trajectory of an object point representing its location in each frame along the temporal sequence (Figure 2). The main idea of solving video sequence alignment can be summarized as [CI02]:

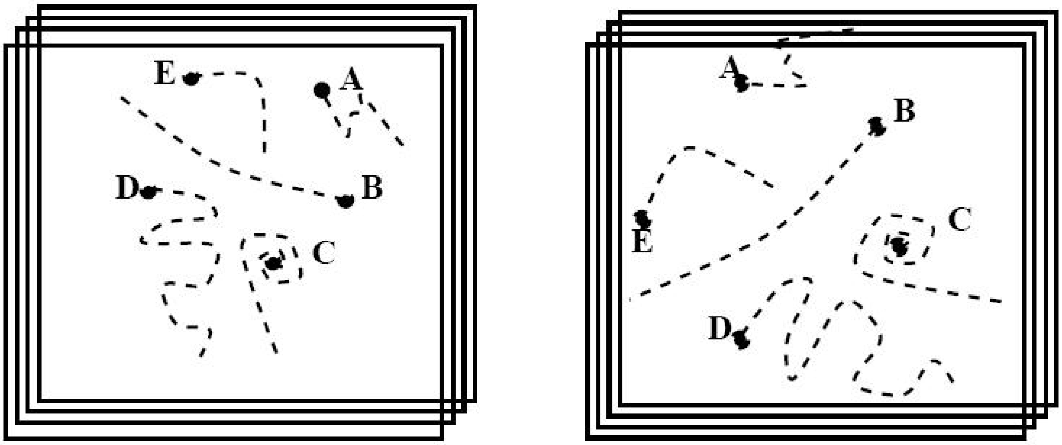
1. Randomly select a number (N) of pairs of possibly corresponding trajectories.

2. For each pair of trajectories, estimate a pair of T and C.

3. Finally, select the pair of T and C from the estimated N pairs which minimize the error E. Theoretically, all pairs of possibly corresponding trajectories between the two videos should be explored. In real implementation, hypothesize-and-test paradigm is used to eliminate exhaustive searching for possible matches. It often randomly selects part of the possibly corresponding trajectories as the hypotheses [CI02].

The main difficulty in the above method is to estimate T and C for each pair of trajectories. During the estimation of T and C, one often assumes that the s parameter in C is known [Ste98, CSI02] (e.g., between PAL and NTSC sequence, it is $s = 25/30 = 5/6$). So far, for C, one only needs to estimate the time offset $\triangle t'$ between two sequences. Stein's method [Ste98] exhaustively searched all possible real-value time offset $\triangle t'$. On the other, the method of Caspi et al. [CSI02] exhaustively searched all possible integer time offset $\triangle t'$. For each $\triangle t'$, they estimated T by minimizing the difference between one trajectory and the other transformed trajectory.

The above method can align two video sequences that capture the same dynamic scene by different types of sensors (e.g., light and infrared) or in slightly different view points or zooms. However, it often assumes that each moving object is rigid and can be viewed as one



(a) Sequence 1 (b) Sequence 2

Figure 2: Trajectories in two video sequences. (a) and (b) display the trajectories of the moving objects over time captured in two video sequences (from [CSI02]).

motion point. Therefore, it cannot align complex motion sequences such as 3D human motion. Moreover, since it assumes that two video sequences have a linear time offset, it may fail for videos with a dynamic time shift.

Dynamic Time Warping

Recently, Rao et al. [RGSM03] proposed a novel method to establish temporal correspondence between two videos. The videos may not capture the same dynamic scene, but they should capture similar motion such as two individuals doing the same motion. The authors assume that the point trajectories have been found and only need to align the two videos temporally. The temporal correspondence function is denoted by C, $t = C(t')$. They used the fundamental matrix $\mathbf{F}$ to approximate the spatial transformation T. That is, between corresponding points $(u(C(t')), v(C(t')))$ and $(u'(t'), v'(t'))$, the ideal difference $d(t')$ is:

$$d(t') = \begin{bmatrix} u(C(t')) \\ v(C(t')) \\ 1 \end{bmatrix}^T \mathbf{F} \begin{bmatrix} u'(t') \\ v'(t') \\ 1 \end{bmatrix} = 0 \tag{6}$$

Then, the difference e between the two trajectories can be defined as the mean-squared difference:

$$e = \frac{1}{l} \sum_{t=1}^{l} d^2(t') \tag{7}$$

Now the problem of temporal alignment is transformed to that of finding $\mathbf{F}$ and C that minimize the error e.

Noticing that $\mathbf{F}$ is used here only for measuring the difference between the two trajectories, Rao et al. did not directly solve $\mathbf{F}$ but indirectly obtain a difference value between the trajectories from $\mathbf{F}$. From Equation 6, using a set of corresponding points, they get $\mathbf{Mf} = 0$, where $\mathbf{M}$ is derived from the coordinates of corresponding points, and $\mathbf{f}$ is derived from the elements of $\mathbf{F}$. For a solution of $\mathbf{f}$ to exist, $\mathbf{M}$ must have a rank of at most eight. However, due to noise and alignment error, the rank of $\mathbf{M}$ may not be exactly eight. In this case, they use the 9^{th} singular value of $\mathbf{M}$ to measure the match of two trajectories. Under such a measurement, they use Dynamic Time Warping (DTW) to find the temporal correspondence function C. DTW is a well-known dynamic programming technique that matches a test sequence with a reference sequence if their time scales are not linearly aligned but when time ordering constraint holds [MRR80]. It finds an optimal match between the reference sequence and the input sequence by stretching and compressing sections of the reference sequence.

3.4 Motion Retargetting

3.4.1 Problem Description

Motion retargetting is to adapt a 3D motion from one character to another [Gle98, LS99]. In computer animation, in order to make use of the captured data of human motion, animators often need to adapt them to a different character. Here we only discuss articulated characters (Figure 3).

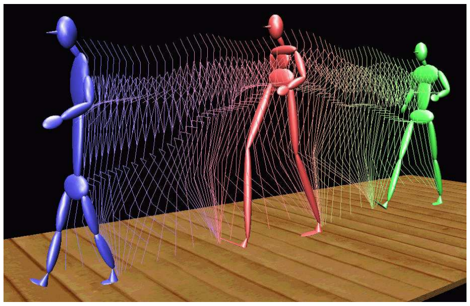


Figure 3: Motion retargetting. Walking motion is retargetted from the right character to the center and the left characters. The three characters have similar articulated structure but different limb and body lengths (from [LS99]).

During motion retargetting, some important properties or constraints Q should be preserved. One category of constraints, namely character constraints, describes the configuration of the articulated characters, such as the range of joint angles and the anatomical relationship among the joints. The second category of constraints, namely spatial constraints, describes the the specific configuration of the articulated characters in some time instants. For instance, the feet must touch the floor in some time instants in walking motion. Another category of constraints, namely dynamics constraints, describes that the retargetted motion should have no artifacts compared with the original motion. For instance, the retargetted walking should have no jerkiness.

Let $M = \{\mathbf{X}_{it}\}$ and $M' = \{\mathbf{X}'_{it}\}$ denote the source motion the retargetted motion respectively. The symbols $\mathbf{X}_{it}$ and $\mathbf{X}'_{it}$ denote the positions of joint i in the source motion and retargetted motion at time t. The source and target articulated characters have the same structure and their corresponding joints are known. Furthermore, the difference between M and M' is often measured in terms of features $\mathbf{f}$, such as joint angles, to remove the need to detect spatial transformation between the joint points. So, the difference E between motion M and M' can be denoted by:

$$E = \frac{1}{nl} \sum_{t=1}^{l} \sum_{i=0}^{n} \parallel \mathbf{f}(\mathbf{X}_{it}) - \mathbf{f}(\mathbf{X}'_{it}) \parallel^2 \tag{8}$$

Then, motion retargetting can be defined as finding the motion M' that minimizes the error E subject to the constraints Q.

In motion retargetting, the two articulated characters' body models are known, one only needs to adapt the motion from one body model to another body model under some constraints. In comparison, in our proposed problem, one articulated model is known, but the other articulated model corresponding to the body in image sequences is unknown. In order to register two human motion, one needs not only to find the second articulated body model, but also to retarget the reference motion to the second model.

3.4.2 Retargetting Algorithms

From the above problem description, we can see that the key difficulty is how to handle constraints Q such that the retargetted motion M' of the new character satisfies constraints Q and is similar to the source motion M. There are two major methods of handling constraints in motion retargetting [Gle01]: per-frame inverse kinematics plus filtering (PFIK+K), and space-time method.

PFIK+K Method

In motion retargetting, the handling of spatial constraints and character constraints is typically called "inverse kinematics". Inverse kinematics (IK) is a process for determining the configuration parameters (e.g., joint angles) of a character based on specifications of some posture features, such as end-effector (e.g., feet and hand) positions. The relationships between the end-effector positions and the configuration parameters are often non-linear and quite complex. This means that IK solvers often must rely on sophisticated methods. The typical IK solvers fall into two categories: analytic methods and numerical methods.

Analytic methods use closed form geometric equation to compute the configurations of a character's parameters directly. Given a set of spatial constraints (e.g., hands and feet positions), geometric equations can directly obtain the configurations (e.g., joint angles) of a character. Although analytic methods can provide guaranteed fast solutions [TGB00], they lack flexibility in how they choose solutions in under-constrained cases and in what types of problems they can handle.

Numerical methods construct a cost function that combines spatial constraints, character constraints, and the error between corresponding motion frames. They use optimization techniques (e.g., gradient method) to provide a more general, albeit computationally expensive, IK solution. Because the inverse kinematic equations are nonlinear, numerical methods often search iteratively for possible solutions using standard optimization techniques. Lee and Shin [Lee 1999] presented an IK solver that combines analytic and numerical methods. They employed analytic method for the limbs, where closed form solutions are available, and used numerical method to optimize the other configuration of of human body.

The PFIK+K method [LS99, Gle01] includes one initialization and one iterative process. The initialization is to directly generate the initial motion M' of the new character. For each joint in M', it has the same feature $\mathbf{f}$ (e.g., joint angle) as that in M. The iterative process includes two steps. The first step uses an inverse kinematics solver applied to each frame of the initial motion M' to handle spatial constraints and character constraints. This step can obtain the configuration of the character for each frame of the motion M'. Although the character configuration satisfies spatial constraints and character constraints for each frame, the motion M' consisting of the frames may have artifacts and visually unnatural motion because the first step does not consider the possible relationships among multiple frames. Such possible artifacts can be resolved in the second step.

The second step provides a global process that considers multiple frames together to remove artifacts. It is often a low-pass filter which removes the spikes and other discontinuities among multiple frames caused by the first step. Because the two steps are performed independently, one may undo the work done by the other. Therefore, they are often interleaved in an iterative process.

Space-time Method

Compared to PFIK+K, space-time method [WK88, Coh92, Gle97, GL98, Gle98] does not consider the frames individually but the whole motion. The method uses motion-displacement $D = \{\mathbf{d}_i(t) = \mathbf{f}(\mathbf{X}_{it}) - \mathbf{f}(\mathbf{X}'_{it}) | i = 1, ..., n;\ t = 1, ..., l\}$ to represent the difference between motion M and M', where $\mathbf{f}$ denotes the feature of points $\mathbf{X}_{it}$ and $\mathbf{X}'_{it}$. Then, cubic B-splines are used to represent D. B-spline can be used as a low-pass filter to solve the dynamics constraints. For the spatial constraints and character constraints, space-time method represent them by a set of equations and inequalities. Then retargetting problem is transformed to minimizing $e = \sum_{i,t} \mathbf{d}_i^2(t)$ subject to the set of equations and inequalities. This is a standard constrained optimization problem. It can be directly solved by quadratic programming method [WK88, Coh92]. It can also be transformed to unconstrained optimization problem by constructing a cost function that combines e and the set of equations and inequalities. Gradient method [Gle98] can be used to solve such unconstrained optimization problem.

The power of the space-time constraints method is also its drawback. While the method provides tremendous opportunity to define constraints and objective functions that describe properties of the motion, these must be defined for the method to work. Defining such mathematical characterizations for motion properties is a challenging task. Also, the method requires solving a single mathematical problem for the entire motion. This leads to very large constrained optimization problems that are usually very difficult to solve.

3.5 Commercial Training Systems

There are commercial products that can measure the detailed difference between two human motion, e.g., 3D-Golf™ and 6D-Research™.

3D-Golf™ is a real-time golf swing analysis and training system. The golfer's swing is captured and the important characteristics are calculated and compared to a reference motion. It can measure all aspects of the swing in position and orientation. It also gives suggestions for swing correction and drills for practice.

Similarly, 6D-Research™ is designed for high-accuracy applications of motion measurement and biomechanics research. It can be tailored to many motion measurement applications. Some training centers such as the US Olympic Committee and the Australian Institute of Sport have used it for sport training.

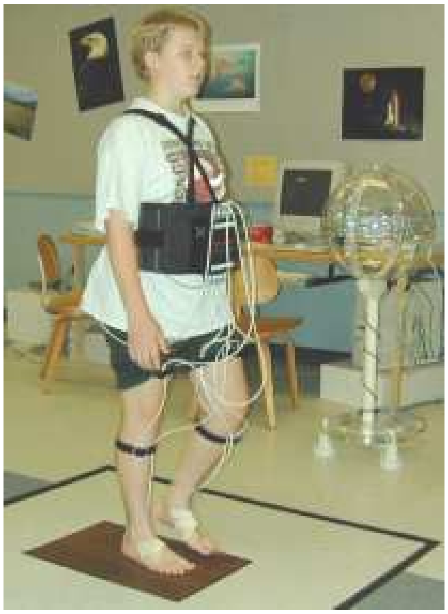


Figure 4: Motion capture

These systems use electromagnetic motion capture devices to obtain 3D motion information. Special hardware and software are required, and special devices including multiple electromagnetic sensors have to be attached onto the human body (Figure 4). Although they may be applicable to some kinds of motion and affordable to special training centers, such systems may not be applicable to complex sport coaching and self-coaching where the devices are not affordable or not permitted to be attached to human body.

4 Approaches

In the simplified problem defined in Section 2.2, 2D feature points have to be extracted from the input images in order to perform spatiotemporal registration. However, in practice, it is very difficult to extract possible joint points from the input images without a large number of false alarms. Moreover, some joint points will be missed due to occlusion of body parts. On the other hand, assuming only a single moving person and stationary cameras, it is much easier to remove the background to isolate the human body regions in the input images. Let $S_{kt'}$ denote the human body region in image $I'_{kt'}$ at time t'. In the following, we will discuss three possible approaches for solving the proposed problem by describing three variations of the problem definition.

4.1 Problem Definition: Variation 1

This variation defines the problem as a match between the projected model of B_t and the human body region $S_{kt'}$ in $I'_{kt'}$. It takes into account adjustment G of difference in body size, global rigid-body transformation T (rotation and translation) of human body, and projection P of 3D model to image plane. The problem is to determine the functions P, T, G and C that minimize the error E:

$$E = \sum_k \sum_{t'} \|P_k(T_{C(t')}(G(B_{C(t')}))) - S_{kt'}\|^2 \tag{9}$$

where $\|\cdot\|$ denotes an appropriate difference measure between the projected model and the input body region.

This problem defines a spatiotemporal registration between the 3D reference motion and the 2D input motion. But, it does not allow for the computation of the detailed difference in joint angles between the reference and the input motion. The difference between the reference motion and the input motion is given only by the global rigid-body transformation T and the temporal correspondence C.

4.2 Problem Definition: Variation 2

Given enough cameras, it is possible to recover 3D human motion from 2D input videos. Then, the recovered 3D human motion can be compared with the 3D reference motion. This problem formulation thus consists of two sub-problems.

1. Recovery of 3D Motion from Multiple 2D Videos

   This sub-problem has to take into account the differences in body size and limb lengths, where G adjusts for the difference between the human model H and the human body in the input videos. The motion recovery problem can be defined as one that matches the projection of an articulated human model with the human body region in the input image. Thus, this sub-problem is to determine the functions P, T', A' and G that minimize the error E_1:

$$E_1 = \sum_k \sum_{t'} \|P_k(T'_{t'}(A'_{t'}(G(H)))) - S_{kt'}\|^2 \tag{10}$$

where T' denotes rigid-body transformation and A' denotes articulation of joints.

2. Comparison of 3D Motion
Let $B'_{t'} = T'_{t'}(A'_{t'}(G(H)))$ denote the recovered 3D human posture at time t'. Then, this sub-problem is to determine the functions T, A and C that minimize the error E_2:

$$E_2 = \sum_{t'} \|T_{C(t')}(A_{C(t')}(G(B_{C(t')}))) - B'_{t'}\|^2 \,. \tag{11}$$

The difference between the reference motion and the input motion is given by T, A and C.

This approach has two separate minimization stages. The first stage has a very large search space because it needs to articulate a static 3D model H into the postures that will match the human regions in the images. Even if possible constraints like body joint range and motion smoothness are considered, reliably and accurately recovering B' is still a difficult and complex problem, which requires enough cameras. In the second stage, since $B'_{t'}$ and B_t are both 3D postures, T and A can be more easily computed if C is accurately determined.

In the case of single camera, the problem will become more complex and ill-posed. Since multiple postures may have the same 2D projection in the image, we cannot reconstruct a single posture $B'_{t'}$ but a set of possible postures $B'_{t'j}$ at each time t'. As a result, in the second stage, we have to determine the best matching posture among those in the set. Let $j(t')$ denote the index of the best posture in the set at time t'. Then the second subproblem is to determine the T, A, C and $j(t')$ that minimize E_2:

$$E_2 = \sum_{t'} \|T_{C(t')}(A_{C(t')}(G(B_{C(t')}))) - B'_{t'j(t')}\|^2 \,. \tag{12}$$

4.3 Problem Definition: Variation 3

This variation is an extension of problem variation 1 by including articulation A of 3D model. The problem can be defined as finding the functions P, T, A, G and C that minimize the error E:

$$E = \sum_k \sum_{t'} \|P_k(T_{C(t')}(A_{C(t')}(G(B_{C(t')})))) - S_{kt'}\|^2 \,. \tag{13}$$

The difference between the reference motion and the input motion is given by T, A and C.

In comparison, variation 1 cannot compute the detailed difference in joint angles between the reference motion and the input motion. Variation 2 can compute the detailed difference between the two motion. But it requires enough multiple cameras to recover the 3D motion from the input videos, and there is a large search space during recovery. In the case of single or insufficient cameras, it is difficult to recover the postures at each time, and the second stage becomes a more difficult sub-problem. Variation 3 can compute the detailed difference between the two motion in a unified step. It also has a smaller search space than problem variation 2. Compared to variation 1 and 2, variation 3 is more promising. We will focus on variation 3 and discuss the main ideas for solving variation 3 in Section 4.4.

4.4 Main Ideas for Solving Problem Variation 3

From Equation 13, we can see that the proposed problem is a high-dimensional optimization problem. It is infeasible to directly solve it. In practice, we try to solve the problem in the following stages: initialization, approximate solution, solution refinement.

4.4.1 Initialization

In the initialization stage, each camera projection function P_k and the body adjustment function G will be determined. Since these functions are constant over time in the input videos, they need to be determined only once.

Each P_k can be easily determined. First, corresponding feature points in some image frames of m'_k can be identified manually or automatically. Then, the projection function P_k for each camera k can be computed using these corresponding points.

G is manually computed at present. G is the adjustment between human body model H and the human body in input videos. In order to find G, it is necessary to know the corresponding image parts of each human body part in at least one input image. In practice, as most researchers do, we manually set G by comparing the human model and the human body in the input images. This procedure produces an adjusted human body model $G(H)$ which matches the size and limb lengths of the human body in the input videos.

4.4.2 Approximate Solution

In the second stage, approximate solutions would be determined by solving problem variation 1. The global transformation T at each time and the temporal correspondence C can be obtained in this stage.

First, for each input image, the global rigid-body transformation T between each 3D posture B_t and the human figure $S_{kt'}$ in the input image can be determined, by minimizing the error between the projection of the transformed 3D posture and the human figure in the input image. Multiple random search method described in Section 3.1.4 can be used to find the best T.

Then, Dynamic Time Warping (DTW) can be used to determine the approximate temporal correspondence C. DTW is a dynamic programming technique. In the DTW method, here we can use the minimum error between any pair of 3D posture and 2D input image obtained in the first step as the distance between the pair. At the same time, we use the temporal order constraint in the DTW, i.e., the corresponding two 3D postures should have the same time order as the two input images.

From the second step, we can obtain the approximate C. And for each pair of corresponding 3D posture and 2D input image, the T obtained in the first step is the approximate solution of global rigid-body transformation.

4.4.3 Solution Refinement

In the third stage, the optimal C, T and A will be determined. First, using the approximate solutions of C and T from the second stage, we can obtain the initial body posture estimations for each input image. Then, starting from the initial estimations, perform an iterative refinement to determine the T and A for every input image. The iterative refinement is the most difficult part in our proposed problem. It is actually an optimization process. Finally, C will be refined, and corresponding new T and A will be directly obtain based on the refined C.

We can make use of the reference motion to search for the T and A. Since the human body in the input video tries to perform the same motion as the reference motion, the input motion and the reference motion are similar at least in some corresponding frames. In these pairs of corresponding frames, the postures in the reference motion may provide good initial body posture estimations (and corresponding good T and A) to the input images when estimating body posture. These initial estimations can also serve as good initial estimations for neighboring input images by propagating information to the neighbors. In each propagation iteration, for every input image, new posture estimations are searched from previous iteration's posture estimations of the current image and the neighboring images. This approach reduces the size of search space for optimal solutions.

In each propagation iteration, many search methods can be used to find T and A for each input image, including continuous local optimization methods, sampling methods and NBP based methods (Section 3.1.4), etc. The preliminary result based on the NBP technique will be described in Section 5.2.

In order to help find the best body posture and corresponding T and A for each input image, prior constraints can be used to constrain and reduce the search region in the parameter space. For example, body parts cannot penetrate each other, body joints have limited ranges of variation, and the change of joint angles between adjacent input images is limited to a small range. These constraints can be added to the search algorithm.

Finally, C (and corresponding new T and A) will be refined. From the above discussion, we can see that given any temporal correspondence C, at least a pair of T and A can be obtained such that E has the same minimum. That means there are many global minima existing in the problem. In order to obtain the good solution of T, A and C, we can use one reasonable prior knowledge that the sum of difference in articulation between two motion should be as small as possible. This knowledge comes from the intuition of using as small modification as possible to change the input motion to the reference motion. That is,

$$E_A = \sum_{t'=1} \parallel A_{C(t')} \parallel^2 \tag{14}$$

should be minimized.

The minimization of E_A can start from the result of the above iterative refinement. First, we obtain the 3D posture of each input image by transformation T and A of the posture in the reference motion. Then, minimize E_A by dynamic programming technique which registers the reference motion and the recovered 3D posture sequence. From the minimization result, we can obtain the optimal T, A and C.

Note that the above approach is different from that in problem variation 2. In variation 2, 3D motion is recovered independent of the reference motion. In comparison, in the above approach, the reference motion information is used throughout the whole algorithms. It plays an important role especially in the third stage where the reference motion is used to help search for good posture estimations and corresponding T and A.

4.4.4 Handling Local Minima

Handling local minima during searching is an important issue for us to find the global minimum. There are three independent ways which are often used together to deal with this issue: (1) reducing the number of local minima, (2) finding good initial estimates, and (3) using global optimization methods.

The number of local minima can be reduced by carefully designing matching cost function. For example, Sminchisescu and Triggs used robust (Lorentzian and Leclerc) function to measure the error during matching [ST01], which can smooth the original error function. They also used a Gaussian kernel to smooth the extracted edges before using the edge feature for matching. In addition, they combined prior knowledge or constraints into the cost function to reduced search region such that the local minima outside the search region do not appear in the cost function. The constraints were represented mathematically and considered as part of the cost function.

Good initializations can help find the global minimum easily, without considering the effect of many other local minima. If good initial estimates can be obtained in advance which lie at the vicinity of global minimum, local optimization methods (e.g., gradient method, Newton method, Levenberg-Marquardt method) is enough for finding the global minimum. In our propose problem, at least for some input images, the good initial estimates can be obtained from the reference motion after the approximate C and T are obtained.

The third way is using global optimization methods during search. From Section 4.4.3, we can see that the difficult task is an optimization process in the continuous parameter space. The global optimization methods described in Section 3.1.4 can be used in our proposed problem. For example, multiple random start and sampling method can be combined to search for the global minimum [CR99, ST01], where sampling method provide good samples for the multiple random start method. During search, smoothing method can be used to smooth the cost function. Tabu method can record the history of searched region such that new search points are in the new region.

5 Preliminary Work

This section describes preliminary work done and results obtained. The objectives of the preliminary work are to investigate methods of solving simplified versions of the proposed problem. Solving these simplified problems helps us to understand the difficulties of the problems and the properties and behaviors of the methods.

Two types of problems are considered. Section 5.1 focuses on 3D-2D Motion Registration of Articulated Stick Figure. Section 5.2 discusses 3D articulated body posture refinement from single image.

5.1 3D-2D Motion Registration of Articulated Stick Figure

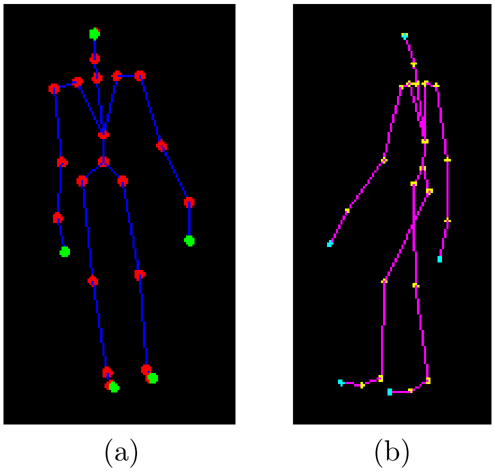


Figure 5: Stick figure. (a) is a 3D articulated stick figure of human model. (b) is one posture projection of the 3D stick figure model from certain viewpoint.

This section describes methods of performing motion registration of a 3D articulated stick figure of a human model (Figure 5 (a)) and its 2D projection (Figure 5 (b)) through a single perspective camera projection. That is, the methods solve the simplified problem defined in Section 2.2. In particular, since the 2D image is the projection of the 3D model, there is no need to adjust for difference in body size and limb lengths between the 3D model and the stick figure in the 2D image. That is, the body-size adjustment function G is a unity function.

For this kind of simplified problem, the correspondence between the joints of the 3D model and their 2D projections are known. The projection function P is also known. So, the purpose of solving this problem is to learn how to determine the global rotation and translation (Section 5.1.1) of the 3D model and the temporal correspondence (Section 5.1.2).

5.1.1 Determining Global Rotation and Translation

Objective

In this case, each 3D posture B_t of the 3D model at time t is projected to a corresponding 2D posture B'_t at the same time instance. So, the temporal correspondence C is a unity function $C(t) = t$.

Let $\mathbf{X}_i$ denote the homogeneous coordinates of a joint in the 3D model, and $\mathbf{x}_i$ the homogeneous coordinates of its 2D projection. The relationship between $\mathbf{X}_i$ and $\mathbf{x}_i$ under perspective projection $\mathbf{P}$ and 3D rotation $\mathbf{R}$ and translation $\mathbf{T}$ is given by:

$$\mathbf{x}_i = \mathbf{P}[\mathbf{R}\,\mathbf{T}]\,\mathbf{X}_i \tag{15}$$

where

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

and f is known. Equation 15 describes a nonlinear transformation, and $\mathbf{R}$ is a rotation matrix that is orthonormal. The objective is to determine the optimal $\mathbf{R}$ and $\mathbf{T}$ at each time instance t independently.

Method

The method I used to solve for $\mathbf{R}$ and $\mathbf{T}$ is taken from [FP03a]. This method first obtains the product $\mathbf{P}[\mathbf{R}\,\mathbf{T}]$ as a single unconstrained matrix $\mathbf{G}$ with 12 parameters. Then, it uses $\mathbf{G}$ to solve for the least-square solution of $\mathbf{R}$ and $\mathbf{T}$, which together have only 6 parameters.

The matrix $\mathbf{G}$ can be easily obtained by solving the following equation using linear least-square:

$$\mathbf{x}_i = \mathbf{G}\mathbf{X}_i \;\; i = 1, \ldots, n \tag{17}$$

where n is the number of joints.

After obtaining $\mathbf{G}$, $\mathbf{R}$ and $\mathbf{T}$ are determined using Newton method. Let $\mathbf{\Theta}$ denote the vector that contains the three rotation parameters and the three translation parameters. Let $\mathbf{e}_i$ denote the error between the corresponding points

$$\mathbf{e}_i = \mathbf{P}[\mathbf{R}\,\mathbf{T}]\,\mathbf{X}_i - \mathbf{x}_i \tag{18}$$

and e_{ix} and e_{iy} denote the x- and y-components of $\mathbf{e}_i$ in the normal 2D spatial coordinate system. That is, e_{ix} and e_{iy} are normalized by the third component of $\mathbf{e}_i$, which is represented in the homogeneous coordinate system.

Let $\mathbf{E}$ denote the error vector that contains e_{ix} and e_{iy}, for $i = 1, \ldots, n$. The Newton method solves for the optimal $\mathbf{\Theta}$ that minimizes $\|\mathbf{E}\|^2$ iteratively, i.e.,

$$\mathbf{\Theta}_{k+1} = \mathbf{\Theta}_k + \lambda_k \mathbf{q}_k \, . \tag{19}$$

The vector $\mathbf{q}_k$ is the search direction during optimization and is obtained by solving the equation

$$(\mathbf{J}_k^T \mathbf{J}_k + \mathbf{Q}_k)\mathbf{q}_k = -\mathbf{J}_k^T \mathbf{E} \tag{20}$$

where $\mathbf{J}_k$ is the Jacobian of $\mathbf{E}$, and $\mathbf{Q}_k$ is the sum of the product of every $\mathbf{E}$'s component and its Hessian matrix [GMW81]. The direction vector $\mathbf{q}_k$ is obtained using Modified Cholesky factorization of $\mathbf{J}_k^T \mathbf{J}_k + \mathbf{Q}_k$ [GMW81].

The parameter λ_k controls how much $\mathbf{q}_k$ affects the iterative update of $\mathbf{\Theta}_k$. In general, λ_k can be a constant or can vary over iterations. In the current implementation, λ_k is determined by searching linearly for a value that minimizes $\|\mathbf{E}\|^2$ at iteration k given $\mathbf{\Theta}_k$ and $\mathbf{q}_k$ [GMW81].

Test Results

In the test, a 3D motion sequence of 82 frames is used. The 3D model of the stick figure has 21 joints. Three performance measures are computed to assess the performance of the algorithm: (1) registration error E_R, (2) mean error in three rotation angles E_r, (3) error in position (i.e., translation) E_p:

$$\begin{aligned} E_R(t) &= \frac{1}{nh'}\|\mathbf{E}_t\|^2 && (21) \\ E_r(t) &= \frac{1}{3}\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\| && (22) \\ E_p(t) &= \frac{1}{h}\|\hat{\mathbf{p}}_t - \mathbf{p}_t\| && (23) \end{aligned}$$

where h' and h are the heights of 2D and 3D stick figures respectively, $\boldsymbol{\theta}_t$ and $\hat{\boldsymbol{\theta}}_t$ are the actual and recovered rotation angle vectors, and $\mathbf{p}_t$ and $\hat{\mathbf{p}}_t$ are the actual and recovered positions (i.e., translation vectors) of the 3D model.

In the test, the heights of 3D and 2D stick figures are 140 pixels and 280 pixels respectively. The random noise is added to each 2D joint position in each 2D input image. For example, when two random values sampled from $[-5\,\text{pixels}, 5\,\text{pixels}]$ are added to the two coordinates of one 2D joint, the random noise would be 1.8% (i.e., 5/280).

Figures 6−8 illustrate sample registration results. From the results, we can see that when there is no noise, rotation and translation can be correctly estimated and the registration error is almost zero. When 2D joint position noise is added (1.8% and 3.6%), the rotation estimation

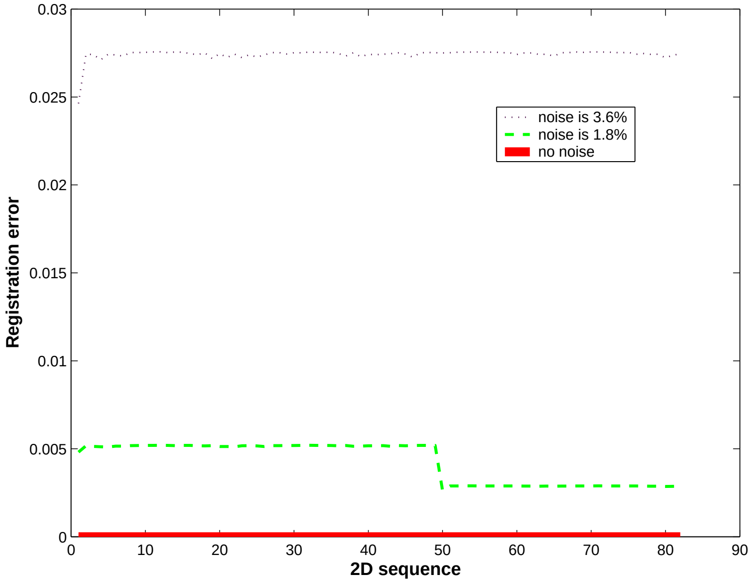


Figure 6: Registration error between each 2D figure and its corresponding 3D figure. When there is no noise, the error is almost zero. When the 2D joint position noise increases, the registration error also increases.

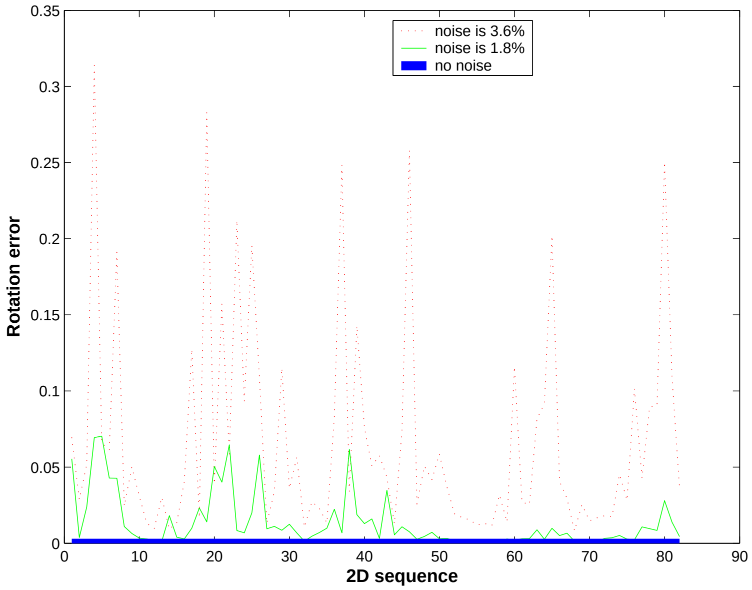


Figure 7: Rotation error between the actual and recovered rotation angles. The error is less than 1 degree even if the 2D joint position noise increases from 1.8% to 3.6% of the 2D stick figure height.

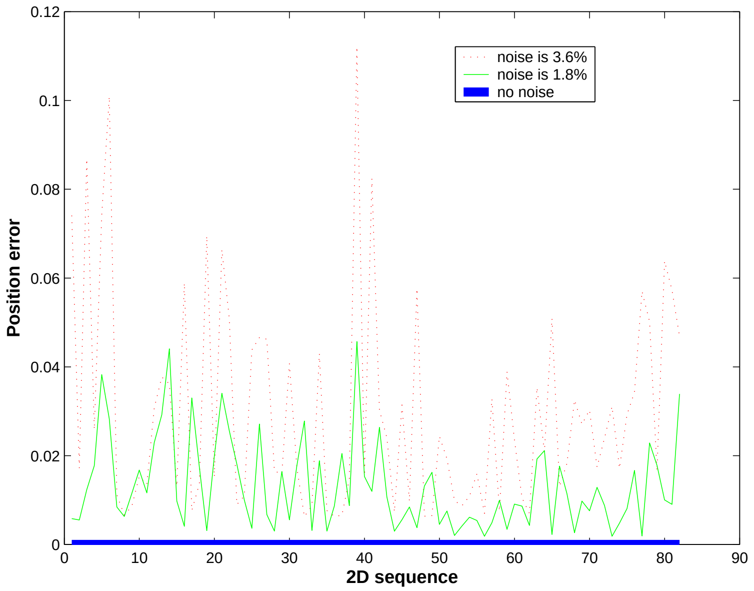


Figure 8: Position error between the actual and recovered 3D positions. The error has the trend of increasing when the 2D joint position noise increase.

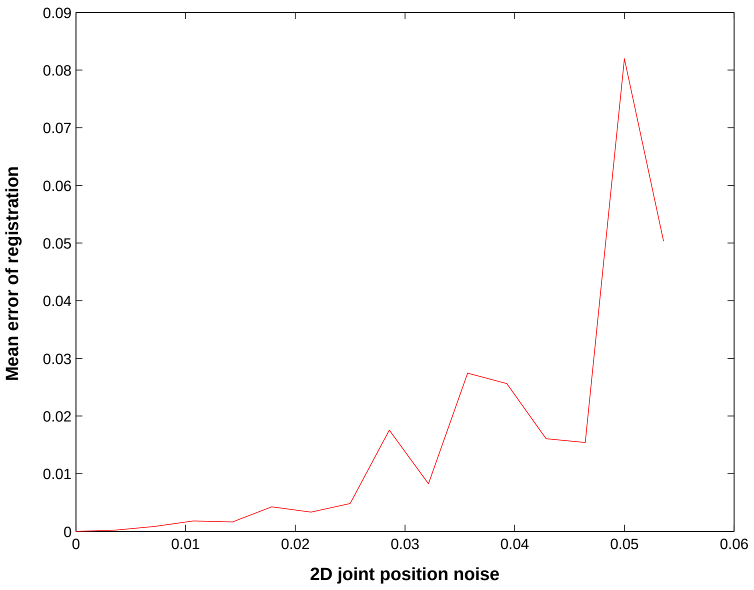


Figure 9: The mean error of Registration. The error increase slowly when the noise is less than 2.5% of the 2D stick figure height.

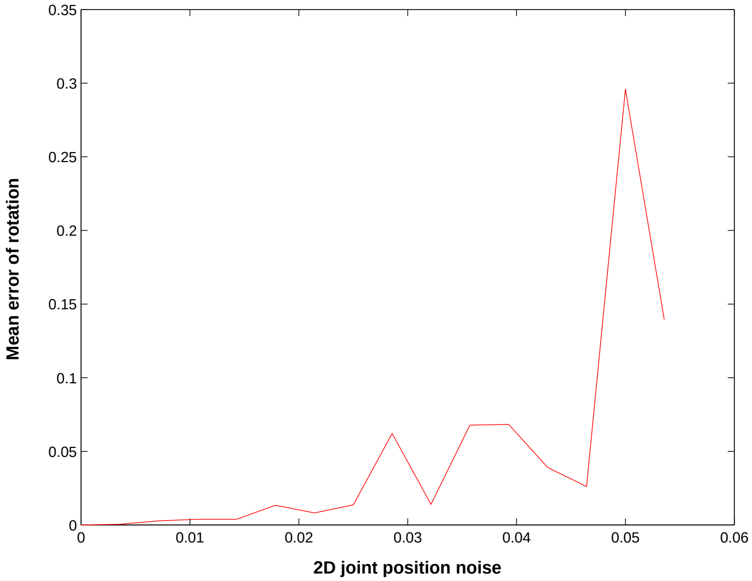


Figure 10: The mean error of rotation. The error increase slowly when the noise is less than 2.5% of the 2D stick figure height. But even the largest error is less than 1 degree.

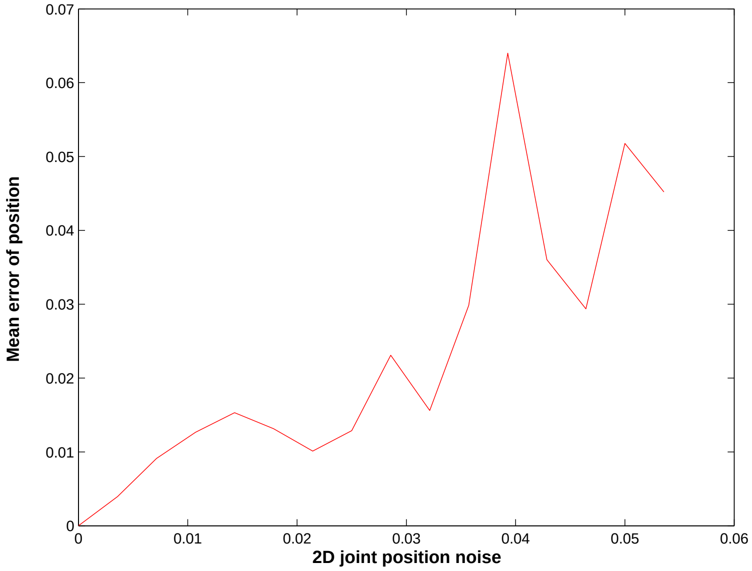


Figure 11: The mean error of position. Compared to the other two kinds of error, this error increase faster with respect to the noise. It is reasonable because the 2D joint position noise will directly affect the recovered 3D position.

error is less than 1 degree, and registration and position error increase with respect to the noise.

Figure 9−11 respectively illustrate the three kinds of error with respect to the 2D joint position noise. For each kind of error and each range of random noise, the mean error of the whole 2D sequence is computed. From the figures, we can see that the errors have the increasing trend when the 2D joint position noise increases.

5.1.2 Determining Temporal Correspondence

Objective

In this case, each 2D posture $B'_{t'}$ at time t' in the 2D input sequence is the projection of a corresponding 3D posture B_t at time t in the 3D reference motion. Any pair of corresponding frames $B'_{t'}$ and B_t may be at different time t' and t, but any two 2D postures should have the same temporal order as the corresponding two 3D postures. So the temporal correspondence C between the two motion sequences is a nonlinear function $C(t') = t$. The objective is to find such a temporal correspondence C.

Method

The method to solve for C is based on Dynamic Time Warping (DTW) [MRR80, KP01]. DTW is a dynamic programming technique. First, the dynamic programming method is used to find a temporal correspondence between the two motion sequences. Then for each 2D posture in the 2D input motion, if there are multiple corresponding 3D postures, the most similar 3D posture is selected as the corresponding 3D posture.

In the method, we use $d(t', t)$ to measure the distance between any pair of 2D posture $B'_{t'}$ and 3D posture B_t, i.e.,

$$d(t', t) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{G}\,\mathbf{X}_{it} - \mathbf{x}'_{it'}\|^2 \tag{24}$$

where the matrix $\mathbf{G}$ is the product $\mathbf{P}[\mathbf{R}\ \mathbf{T}]$ obtained using the method described in Section 5.1.1.

Using this distance measurement, a dynamic programming technique is used to solve for C by minimizing the sum E of distance over time t', where

$$E = \sum_{t'=1}^{L'} d(t', C(t')) \tag{25}$$

and C satisfies the temporal order constraint. Suppose $C(1) = 1$ and $C(L') = L$, and let $D(t', t)$ denote the global minimum distance from frame pair $(1, 1)$ up to (t', t). Then, finding the minimization of E is equal to finding $D(L', L)$. Recursively, $D(L', L)$ can be found by the following formula:

$$D(t', t) = d(t', t) + \min\{D(t' - 1, t - 1), D(t' - 1, t), D(t', t - 1)\} \tag{26}$$

where $D(1, 1) = d(1, 1)$. The optimal C can be obtained by backward following the path from $D(L', L)$ to $D(1, 1)$ on which the global minimum distance is obtained.

In general, in the path from $D(L', L)$ to $D(1, 1)$, one 2D posture may correspond to multiple 3D postures. In this case, the most similar 3D posture is selected from the multiple ones as the corresponding posture.

Test Results

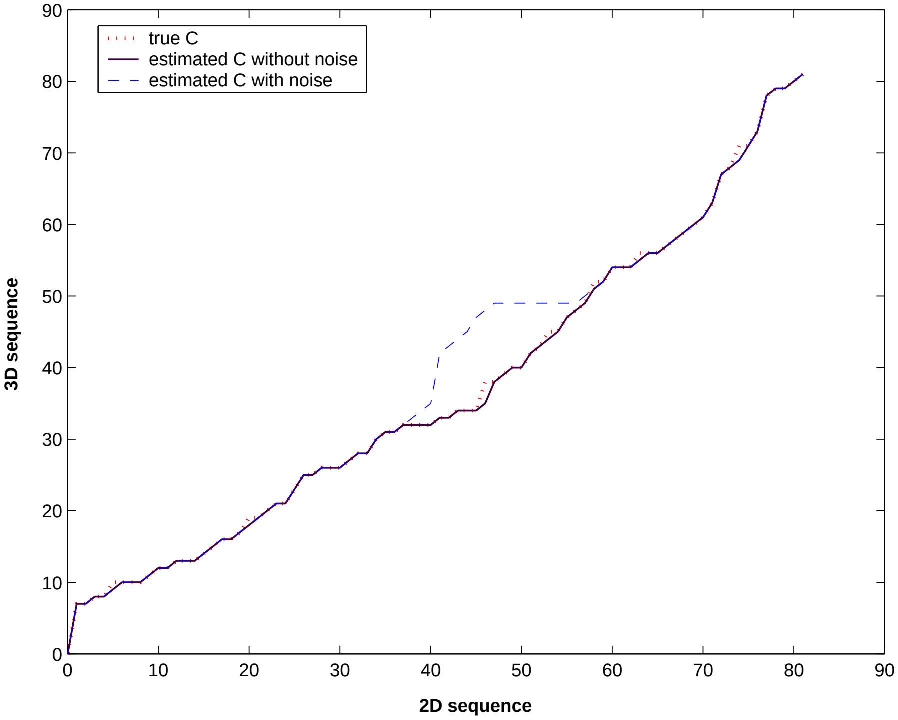


Figure 12: Temporal correspondence between 2D and 3D stick figure sequences.

In the test, both the 3D reference motion and the 2D input sequences consist of 82 frames but have nonlinear temporal correspondence. In the case that there is no joint position noise in the input images, we find our algorithm can almost recover the truth of temporal correspondence no matter how much the nonlinearity of C is. Figure 12 illustrates one example that the estimated C (the solid line) and the actual C (the dotted line) are almost the same.

In the case that there is joint position noise in the input images, we find that the described method may not find the optimal C. In our experiment, we add joint position noise to a small segment of the 2D input sequence (i.e., from frame 40 to 50) by replacing the segment with another segment (i.e., from frame 50 to 60). The experiment result shows that the estimated C (dashed line in Figure 12) is not the optimal. Although only the small segment of 2D sequence (i.e., from frame 40 to 50) is different from the 3D sequence, the small segment will cause its neighboring frames (i.e., from frame 50 to 60) not to find the correct time corresponding 3D frames. The possible reason is clear. The small segment of 2D input sequence (i.e., from frame 40 to 50) is very different from the corresponding 3D postures, but it may be similar to other 3D postures. In this case, even the method can provide the global minimum, the global minimum does not correspond to the optimal temporal correspondence. So the method needs to be improved and extended.

5.2 3D Articulated Body Posture Refinement

This section describes a method to estimate 3D articulated body posture from a single input image. That is, the methods solves the first sub-problem of problem variation 2 defined in Section 4.2. In particular, the input image is the projection of a 3D posture of the articulated human model (Appendix A), so the body adjustment function G is a unity function. The projection function P is also known to be orthographic.

In this problem, an initial body posture and initial global rotation and translation is assumed to be obtained in advance. So, the purpose of solving this problem is to learn how to refine the global rigid-transformation T (i.e. rotation and translation) and body joints articulation A given the initial estimations.

The method I used is based on Nonparametric Belief Propagation (NBP) technique [SIFW03, Isa03, HW04, SMFW04]. Instead of estimating whole body posture from the input image, belief propagation (Appendix B) can estimate the states of each body part by considering the relationships between any two adjacent body parts. For example, when the state of left upper arm has been close to correct state, the left lower arm will be limited to a small region in which the correct lower arm state can be more easily found.

To use NBP, a graphical model is designed to represent the body parts' states, the relationships between body parts, and relationships between body part state and image observations (Section 5.2.1). Using it, one NBP algorithm is designed (Section 5.2.2).

5.2.1 Graphical Model for NBP

In the original BP (Appendix B), it implicitly assumes that there is no self-occlusion between the observations $\mathbf{z}_i$ of different body parts' state $\mathbf{x}_i$. In practice, self-occlusion usually happens especially in human motion. In this case, The joint probability of whole body state $\mathcal{X}$ and corresponding image observation $\mathcal{Z}$ will become

$$p(\mathcal{X}, \mathcal{Z}) = \alpha_1 \prod_{(i,j)\in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in \mathcal{V}} \phi_i(\mathcal{X}, \mathbf{z}_i) \tag{27}$$

Then it is not a trivial problem using the original BP to calculate marginal distribution $p(\mathbf{x}_i|\mathcal{Z})$. Fortunately, in our motion registration problem, there are many input images to which good initial body posture estimations can be obtained from the 3D reference motion sequence, therefore we may use the following two formulas to calculate marginal distribution,

$$m_{ij}^n(\mathbf{x}_j) \approx \alpha_2 \int_{\mathbf{x}_i} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)\phi_i(\mathbf{x}_i, \tilde{\mathcal{X}}_{-i}^{n-1}, \mathbf{z}_i) \prod_{k\in \Gamma(i)\backslash j} m_{ki}^{n-1}(\mathbf{x}_i)d\mathbf{x}_i \tag{28}$$

$$\hat{p}^n(\mathbf{x}_j|\mathcal{Z}) \approx \alpha_3\phi_j(\mathbf{x}_j, \tilde{\mathcal{X}}_{-j}^{n-1}, \mathbf{z}_j) \prod_{i\in \Gamma(j)} m_{ij}^n(\mathbf{x}_j) \tag{29}$$

where $m_{ij}^n(\mathbf{x}_j)$ is the message propagated from node i to j in iteration n, and $\tilde{\mathcal{X}}_{-i}^{n-1}$ is the set of body parts estimations except the i^{th} body part which comes from the previous $(n-1)^{th}$ iteration. Using Equations 28 and 29, we can deal with self-occlusion when estimating body posture, although the convergence of Equation 29 need to be theoretically proved.

From Equations 27−29, we can see that designing a graphical model includes designing body part state variable $\mathbf{x}_i$, the relationships $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ between two body parts, and the relationships $\phi_i(\mathbf{x}_i, \tilde{\mathcal{X}}_{-i}^{n-1}, \mathbf{z}_i)$ between state $\mathbf{x}_i$ and corresponding observation $\mathbf{z}_i$.

State Variable $\mathbf{x}_i$

State variable $\mathbf{x}_i = (\mathbf{p}_i,\boldsymbol{\theta}_i)$ represent the i^{th} body part position $\mathbf{p}_i$ and orientation $\boldsymbol{\theta}_i$, and the corresponding observed variable $\mathbf{z}_i$ represent the image observation for the i^{th} body part. Every node in the tree-structured graphical model (Figure 13) represents a pair of $\mathbf{x}_i$ and $\mathbf{z}_i$. The relationship between $\mathbf{x}_i$ and $\mathbf{z}_i$ is represented by the observation function $\phi_i(\mathbf{x}_i, \tilde{\mathcal{X}}_{-i}^{n-1}, \mathbf{z}_i)$. In addition, due to the articulation property of human body, at least there is one relationship between any two adjacent body parts $\mathbf{x}_i$ and $\mathbf{x}_j$. This relationship (corresponding to the edges that connect nodes in Figure 13) is represented by the potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$.

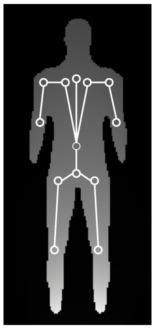


Figure 13: Tree-structured graphical model

Potential Functions $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$

Potential function $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ can represent any relationships between body part i and j, such as body part connection constraint and joint angle limits. Currently, we use it to represent just connection constraint between two adjacent body parts. Without loss of generality, suppose node i is the parent of node j, then using the state $\mathbf{x}_i$ of node i, one position of the j^{th} body part can be computed by a rigid transformation T. Using the connection constraint between the two body parts, there is

$$\psi_{ij}^n(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}(T(\mathbf{x}_i) - \mathbf{p}_j; 0, \Lambda_{ij}^n) \tag{30}$$

where $\psi_{ij}^n(\mathbf{x}_i, \mathbf{x}_j)$ represents the probability of $\mathbf{x}_j$ given $\mathbf{x}_i$, and $\Lambda_{i,j}^n$ is the variance matrix of the gaussian function $\mathcal{N}$ in the n^{th} iteration of NBP. Note that $\Lambda_{i,j}^n$ may be different in different iterations. Here $\Lambda_{i,j}^n$ is gradually decreasing with respect to iteration number n, which may have similar annealed simulation effect to that in annealed particle filter [DBR00].

Observation Functions $\phi_i(\mathbf{x}_i, \tilde{\mathcal{X}}_{-i}^{n-1}, \mathbf{z}_i)$

Observation function $\phi_i(\mathbf{x}_i, \tilde{\mathcal{X}}_{-i}^{n-1}, \mathbf{z}_i)$ measures the likelihood of $\mathbf{z}_i$ given $\mathbf{x}_i$. In order to measure the likelihood, each estimate of body part state $\mathbf{x}_i$ is required to be rendered and then projected together with $\tilde{\mathcal{X}}_{-i}^{n-1}$, and then compare the similarity between the projected image and the input image. Currently, we use edge and silhouette as the feature for the similarity measurement. Chamfer distance is used to measure the edge similarity, and overlapping rate of the projected image to the human body image region in the input image is used to measure the silhouette similarity. The relative weight between edge and silhouette similarity is experimentally determined.

5.2.2 NBP for Single Image

After designing potential functions and observation functions, NBP can be used to search for body part states by iteratively updating each message and each marginal distribution. In the NBP algorithm, each message $m_{ij}^n(\mathbf{x}_j)$ is represented by a set of K weighted samples,

$$m_{ij}^n(\mathbf{x}_j) = \{(\mathbf{s}_j^{(n,k)}, \omega_{ij}^{(n,k)}) | 1 \leq k \leq K\} \tag{31}$$

where $\mathbf{s}_j^{(n,k)}$ is the k^{th} sample of the j^{th} body part state in the n^{th} iteration and $\omega_{ij}^{(n,k)}$ is the weight of the sample. Correspondingly, the marginal distribution is also represented by a set of weighted samples,

$$\hat{p}^n(\mathbf{x}_j | \mathcal{Z}) = \{(\mathbf{s}_j^{(n,k)}, \pi_j^{(n,k)}) | 1 \leq k \leq K\} \tag{32}$$

where $\mathbf{s}_j^{(n,k)}$ is the same as that in Equation 31 and $\pi_j^{(n,k)}$ is the corresponding weight.

In each iteration, each message $m_{ij}^n(\mathbf{x}_j)$ and each marginal distribution $\hat{p}^n(\mathbf{x}_j | \mathcal{Z})$ are updated based on Equations 28 and 29. Since the messages and marginal distributions are nonparametric, the update is based on the Monte Carlo method. The update process is described in the following:

1. Use importance sampling to generate new samples $\mathbf{s}_j^{(n+1,k)}$ from related marginal distributions of previous iteration. The related marginal distributions include the neighbors' and its own marginal distributions of previous iteration. The new samples are to be weighted respectively in the following two steps to represent corresponding messages and marginal distributions.

2. Update messages. For each new sample $\mathbf{s}_j^{(n+1,k)}$ and each neighboring node $i \in \Gamma(j) = \{i|(i,j) \in \mathcal{E}\}$, calculate the weight $\omega_{ij}^{(n+1,k)}$, where

$$\omega_{i,j}^{(n+1,k)} = \sum_{k=1}^{K} [\psi_{ij}(\mathbf{s}_i^{(n,k)}, \mathbf{s}_j^{(n+1,k)}) \frac{\pi_i^{(n,k)}}{\omega_{ij}^{(n,k)}}] \tag{33}$$

   Equation 33 is the nonparametric version of Equation 34, which represent message (28) in terms of marginal distribution (29) [SMFW04], i.e.,

$$m_{ij}^n(\mathbf{x}_j) = \alpha_2 \int_{\mathbf{x}_i} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \frac{\hat{p}^{n-1}(\mathbf{x}_i|\mathcal{Z})}{m_{ji}^{n-1}(\mathbf{x}_i)} d\mathbf{x}_i \tag{34}$$

   The updated messages will be used to update marginal distributions.

3. Based on the updated messages, each marginal distribution is updated. For each sample $\mathbf{s}_j^{(n+1,k)}$, calculate the weight $\pi_j^{(n+1,k)}$, where

$$\pi_j^{(n+1,k)} = \phi_j(\mathbf{s}_j^{(n+1,k)}, \tilde{\mathcal{X}}_{-j}^n, \mathbf{z}_j) \prod_{l \in \Gamma(j)} \omega_{lj}^{(n+1,k)} \tag{35}$$

   then $\pi_j^{(n+1,k)}$ is re-weighted because we use importance sampling to generate sample $\mathbf{s}_j^{(n+1,k)}$. The updated marginal distributions will be used to update messages in the next iteration.

3D body posture can be estimated from the set of marginal distributions. The mean of $\hat{p}^n(\mathbf{x}_j|\mathcal{Z})$ or the sample with the maximum weight in $\hat{p}^n(\mathbf{x}_j|\mathcal{Z})$ can be used to represent the estimation of j^{th} body part state. However, due to the depth ambiguity in single video, we cannot assure that the estimation of each body part state is the truth.

There are differences between our algorithm and others' NBPs. Sudderth et al. [SMFW04] used Gaussian mixtures to represent messages and marginal distributions, and a complex Gibbs sampler is required to generate samples in each iteration. In our algorithm, like BPMC [HW04], we just use a set of weighted samples to represent messages and marginal distributions, and use importance sampling to generate samples. Compared to BPMC in which importance function comes from the same node's marginal distribution of previous iteration, and in which they re-weight messages by the importance function, our importance function comes from multiple marginal distributions, i.e. both the neighboring marginal distributions and the same node's marginal distribution of previous iteration. And we re-weight the marginal distributions, not messages, by the importance function, and we believe that it is more reasonable from the importance sampling theory. Also, BPMC is used for rigid object whereas we deal with articulated body. Furthermore, compared to other algorithms [SMFW04, HW04], we make use of body posture estimation of previous iteration to deal with self-occlusion, and embed the annealing idea into the algorithm by modifying potential functions in each iteration.

5.2.3 Test Results

In the test, a 3D reference sequence of 80 frames are used. In order to test the accuracy of our method to estimate body posture from each input image of the 2D input sequence, we should know the ground truth of body posture for each input image. Currently, we use the projection of the 3D reference sequence as the input sequence, such that we can know the true posture for each input image. Then we modify the 3D reference sequence by adding random noise to each joint angle of each posture. The modified posture at time t in the modified reference sequence is used as the initial posture for the input image at time t in the input sequence. In our NBP algorithm, 150 weighted samples are used to represent each message and each marginal distribution. 6 iterations are repeated twice.

Two performance measures are computed to assess the performance of our algorithm: 2D joint position error E_{2D} and 3D joint position error E_{3D}, i.e.,

$$E_{3D}(t) = \frac{1}{nh} \sum_{n}^{i=1} \|\hat{\mathbf{p}}_{3ti} - \mathbf{p}_{3ti}\| \tag{36}$$

$$E_{2D}(t) = \frac{1}{nh} \sum_{n}^{i=1} \|\hat{\mathbf{p}}_{2t} - \mathbf{p}_{2t}\| \tag{37}$$

where $\hat{\mathbf{p}}_{3it}$ and $\mathbf{p}_{3it}$ are the estimated and true 3D position of the i^{th} joint at time t, and $\hat{\mathbf{p}}_{2it}$ and $\mathbf{p}_{2it}$ are the estimated and true 2D position without the depth value. h is the articulated body height and it is 195cm in the test.

In the first experiment, we test how much the error can be reduced from initial posture to estimated posture over time. Each initial posture is generated by adding a random noise to each joint angle of the true posture. The random noise is sampled from $[-30^o, 30^o]$.

Figure 14 illustrates the 2D joint position error of initial posture and estimated posture. From the figure, we can see that the estimated posture can reduce more than half error (i.e., about from 9% to 4%) of the initial posture for most input images. But when there is severe self-occlusion in input images, the error may not reduce so much, such as the error in the 5^{th} input image of the video sequence.

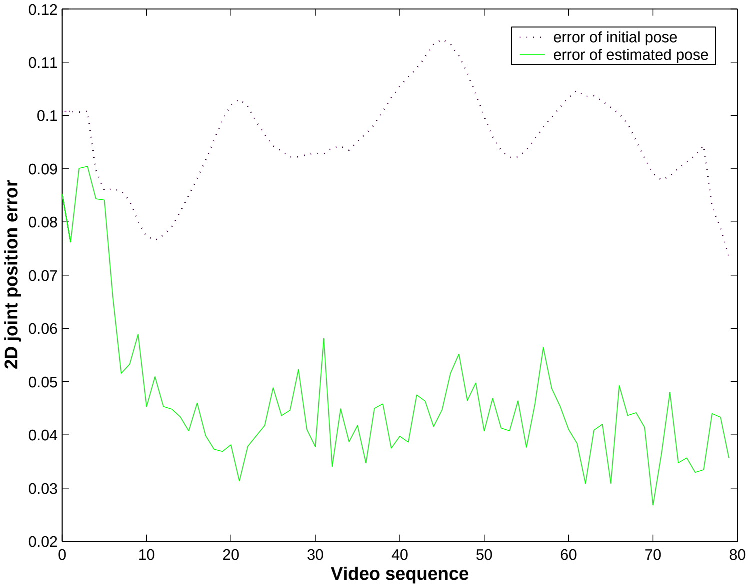


Figure 14: 2D joint position error. The error of each estimated posture is much less than the initial posture for most input images.

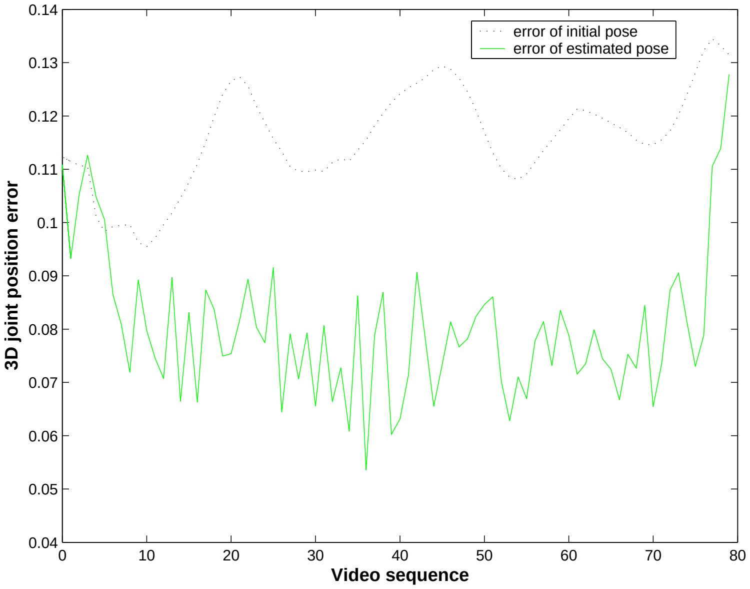


Figure 15: 3D joint position error. The error of estimated posture is less than the initial posture for most input images. But due to depth ambiguity, the depth information cannot be accurately estimated such that the 3D joint position error decreases less compared to the 2D joint position error.

Figure 15 illustrates the 3D joint position error of initial posture and estimated posture. Compared to 2D joint position error, 3D joint position error decreases less from initial posture to estimated posture, and sometimes it may increase from initial posture to estimated posture (e.g., for the 4^{th} input image). One possible reason of less error decreasing is that we use single image such that depth information of each joint cannot be accurately estimated. Depth ambiguity may be resolved when using multiple video sequences recording human motion from different viewpoints, or using the estimations of neighboring input images to constrain the current estimations.

In the second experiment, we test how the error change with respect to the random noise of each joint angle. The random noise is increased from 0 to $[-30^o, 30^o]$. For each range of random noise, the mean error is obtained by summing over a small segment of sequence (i.e., from frame 10 to 30).

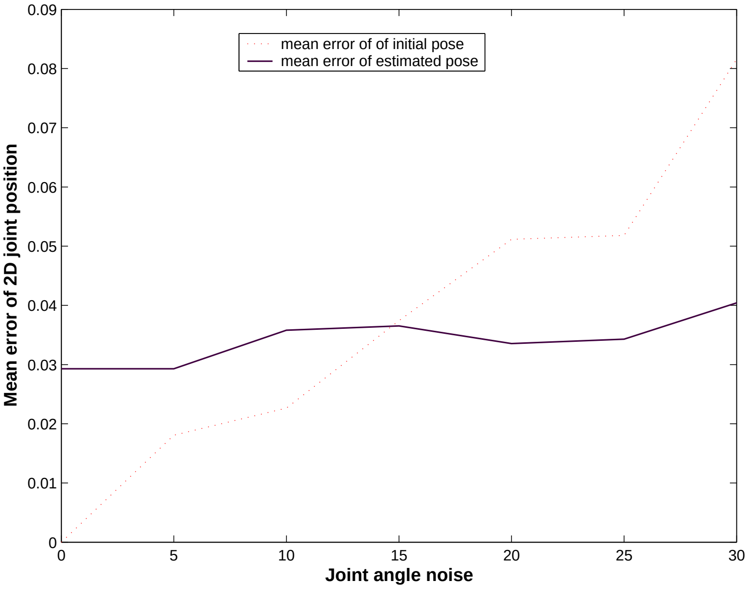


Figure 16: 2D joint position error with respect to joint angle noise. The error of estimated posture change little when the joint angle noise increases. But the algorithm cannot get accurate posture estimation even if the initial posture is accurate.

Figures 16−17 illustrate the 2D and 3D joint position (mean) errors of initial posture and estimated posture. It shows that when the joint angle noise is increased, the errors of initial posture increase while the errors of estimated posture increase little.

However, from the figures 14−17, we can also see that even the best estimated postures are not accurate enough, whatever the initial posture is accurate or not. In addition to the severe self-occlusion and depth ambiguity, another possible reason is that we just use silhouette and edges to match, in which case the two kinds of features may not be discriminative enough. We believe that intensity information will help to match if the intensity between body parts

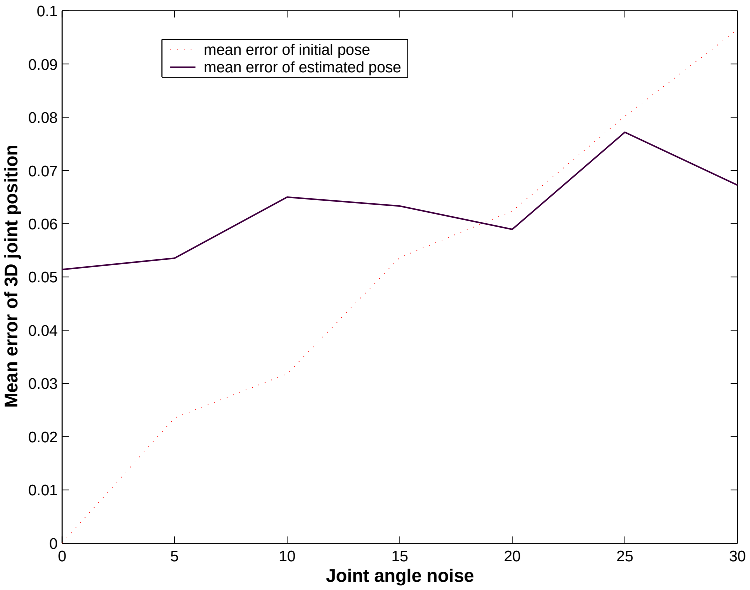


Figure 17: 3D joint position error with respect to joint angle noise. The error of estimated posture change a little when the joint angle noise increase. But the algorithm cannot get accurate posture estimation even if the initial posture is accurate.

are not the same, and we plan to include the intensity feature into matching in the proposed continuing work. In addition, we have not used such constraints as joint angle limits and body parts non-penetration in our algorithm. Using these constraints may help to obtain better estimation results by reducing the search region in the state space.

6 Proposed Continuing Work

Based on the discussion in Section 4 and 5, we propose to do the following work:

1. *Approximate registration between a 3D reference motion and 2D input videos*:
   The task is to find the approximate temporal correspondence C between the 3D and the 2D sequences, and to find the approximate global-rigid body transformation T between any pair of corresponding frames based on the approximate C.

2. *Solution refinement of C, T and articulation A*:
   Starting from the approximate C and T, the goal is to determine the accurate C and corresponding T and A based on the accurate C.

6.1 Approximate Registration

Approximate registration between a 3D reference motion and 2D input videos is to determine approximate solutions of C and T. The approximate solutions will be used as the initial estimation for later solution refinement. The method I have proposed in Section 4.4.2 can be used to perform the approximate 3D-to-2D registration of human motion sequences. But this method needs to be tested and refined.

6.2 Solution Refinement

After getting approximate C and T, the accurate C, T and articulation A will be determined. The idea to solve the task has been described in Section 4.4.3. In the idea, first based on the approximate C, T and A are refined for each pair of corresponding 3D frame and 2D input image. The refined T and A can provide the 3D body posture estimation for each input image. Then based on the estimated 3D posture sequence and the 3D reference motion sequence, the accurate C (and corresponding T and A) can be determined using dynamic programming technique. In the following, I will discuss the two steps respectively.

6.2.1 Refinement of T and A Given C

Given C, the task is how to find the accurate T and A for each pair of corresponding 3D frame and 2D input image. For each input image, considering the corresponding 3D posture in the reference motion as the initial estimation, the task is equivalent to estimating 3D articulated body posture from each input image. In the preliminary work (Section 5.2.3), I have used one NBP algorithm to estimate body posture from single input image. But the posture estimation results of the algorithm is not accurate enough. Based on the analysis of test results in Section 5.2.3, I propose to perform the refinement task by adding the following ideas:

1. Using more information during matching:
   Currently just edge and silhouette information are used during matching. But intensity information is one more important information in general if not all body parts have the same intensity (Figure 18). In addition to edge and silhouette features, I propose to use intensity feature to measure the similarity between the estimated posture and the input image during matching.

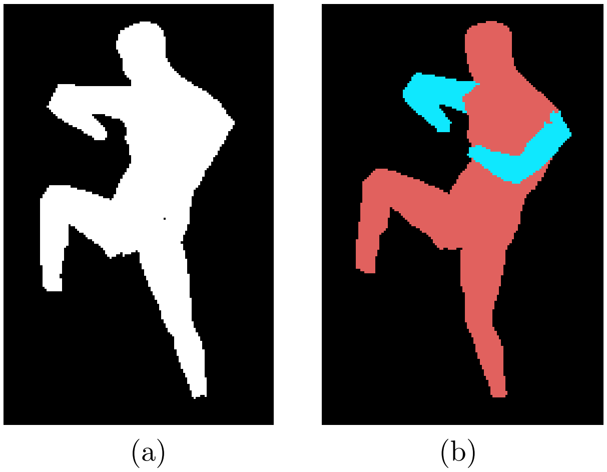


Figure 18: Feature for matching. (a) is the silhouette of a body image, and (b) is the intensity of the body image. Compared to silhouette, intensity can provide more information about where the left arm is in the image.

2. Using neighboring information during posture estimation:
   When estimating posture from each input image, currently only initial posture estimation coming from the corresponding 3D frame is used to search for better posture. However, since the body postures in two neighboring input images often change quite a little, the posture estimations of each input image can provide initial posture estimations for its neighboring images. So, in each search iteration, for every input image, new posture estimations can be searched from both the neighboring information (i.e., 3D reference posture and estimated 3D posture) and the previous iteration's posture estimations of the input image. This idea can reduce the size of space to search for optimal solutions.

3. Reducing search space by prior knowledge or constraints:
   Prior knowledge or constraints can be used to reduce the search space during posture estimation. We propose to add the constraints to the search process. They include joint angle limits, body parts non-penetration, etc. For the example in the NBP algorithm,

given the estimation of each body part state, the body joint angles can be easily computed. By checking whether each joint angle satisfy the range of limits, it can decide which body part state is valid or not.

4. Simultaneous registration to multiple 2D inputs:
   So far, I suppose that the 2D input sequence is single. Since it is expected that the proposed algorithm can deal with both single and multiple sequences, I propose to generalize the algorithm such that it can deal with multiple 2D input sequences. In the case of multiple 2D sequences, the parameters of multiple cameras can be determined in our initialization stage (Section 4.4.1). Then the input image features from the same time instant in the multiple 2D sequences can be extracted to match with the correspondingly projected 3D body posture. For the example in the NBP algorithm, the matching process is performed in the observation function, in which one estimation of body posture is projected to match with the input images coming from multiple 2D sequences.

6.2.2 Refinement of C

In this step, C (and corresponding new T and A) will be refined. From the approximate C and the correspondingly refined T and A obtained above, the 3D body posture of each input image can be obtained. Using the estimated 3D posture sequence, we can register between the estimated and reference motion sequence to find the final C and corresponding T and A. Dynamic programming technique proposed in Sections 4.4.3 and 5.1.2 can be used. We propose to improve the dynamic programming technique used in our preliminary work to robustly deal with two 3D sequence registration.

6.3 Proposed Schedule

The approximate research schedule is as the following:

Tasks	**Time**
Approximate Registration	December 2004 to February 2005
Refinement of Temporal Correspondence C	March to April 2005
Refinement of Articulation A and Global Rigid-body Transformation T	May to December 2005
Thesis Writing	January to May, 2006

7 Conclusion

From the above analysis, it can be shown that spatiotemporal registration between a 3D reference motion and 2D input videos is a challenging problem. It needs to find the temporal correspondence between the 3D motion and the 2D input video sequences, and find the articulation and global rigid-body transformation for each pair of 3D posture and 2D input image.

Although we have provided some algorithms to solve one simplified problem and one subproblem, initial test results show that the algorithms should be improved and extended in order to obtain accurate results. Dynamic programming technique can be used to deal with temporal correspondence C, but it may not find the accurate C when there is large differences in postures between some corresponding frames. One NBP algorithm has been developed to estimate 3D body posture from single image, given one initial posture estimation. It can obtain approximate articulation A and global-rigid body transformation T for each pair of 3D posture and 2D input image. But it needs to be improved and extended to get more accurate results. We propose to extend these algorithms to solve our proposed problem.

Appendix

A Articulated Human Body Model

Human body consists of body joints and body parts connected by joints. Adjacent joints are connected by bones. Each body part consists of single or multiple bones and the flesh attached on the bone(s). A standard mesh model is used to represent the body shape (Figure 19 (a)(b)), and each vertex in the mesh is attached to related body part (Figure 19 (c)). For each body part's size, supposing there is a fixed rate between width and thickness, we can use two parameters (length and width) to represent the size of each body part. Currently, the parameters are obtained by manually matching body parts between the standard body model and the human body in the input video.

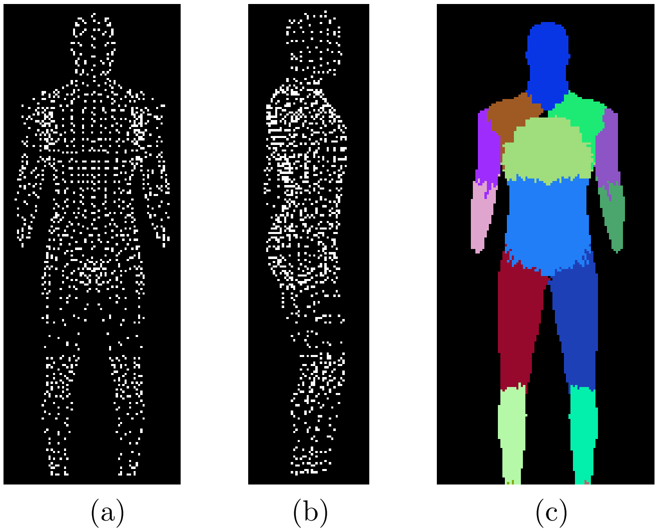


Figure 19: Human body model. The vertexes in the body mesh model are displayed from front view (a) and side view (b). Each vertex and triangle in the model is assigned to one specific body part (c).

Given the human body model, human body posture can be represented by a set of joint angles and the global body position and orientation. Different joints may have different degree of freedom and therefore different number of joint angles. For example, shoulder joint has three DOF but elbow joint has two. The global 3D orientation and 3D position parameters are assigned to the body center (virtual root) joint. Each body part's orientation and position can be directly calculated by forward kinematics.

B Belief Propagation (BP)

Belief propagation is an inference algorithm for graphical model. An undirected graph $\mathcal{G}$ consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ (Figure 20). Each node $i \in \mathcal{V}$ is associated with an state variable $\mathbf{x}_i$ and a local observation $\mathbf{z}_i$. Denote $\mathcal{X} = \{\mathbf{x}_i | i \in \mathcal{V}\}$ and $\mathcal{Z} = \{\mathbf{z}_i | i \in \mathcal{V}\}$ as the sets of all state and observed variables. The objective is to infer $\mathcal{X}$ from $\mathcal{Z}$. If the graphical model is pairwise MRFs, which means the largest clique size in the graph is two, the probability density function can be factorized as

$$p(\mathcal{X}, \mathcal{Z}) = \alpha_1 \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathcal{V}} \phi_i(\mathbf{x}_i, \mathbf{z}_i) \tag{38}$$

where $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is potential function and $\phi_i(\mathbf{x}_i, \mathbf{z}_i)$ is observation function.

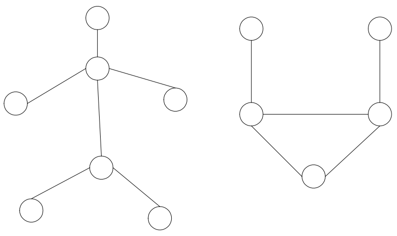


Figure 20: Graphical models

Instead of directly calculating $p(\mathcal{X}, \mathcal{Z})$, people calculate the conditional marginal distribution $p(\mathbf{x}_i|\mathcal{Z})$ of each node. If the graph is acyclic or tree-structured, $p(\mathbf{x}_i|\mathcal{Z})$ can be obtained by belief propagation (BP) [YFW02]. BP is an iteratively local message passing process. Define the neighborhood of node $i \in \mathcal{E}$ as $\Gamma(i) = \{k|(i,k) \in \mathcal{E}\}$. Message $m_{ij}(\mathbf{x}_j)$ can be viewed as the information propagated from node i to neighboring node j, and is computed iteratively using the update algorithm:

$$m_{ij}^{n}(\mathbf{x}_j) = \alpha_2 \int_{\mathbf{x}_i} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j)\phi_i(\mathbf{x}_i, \mathbf{z}_i) \prod_{k\in\Gamma(i)\backslash j} m_{ki}^{n-1}(\mathbf{x}_i) d\mathbf{x}_i \tag{39}$$

where n denotes the n^{th} iteration, and $\Gamma(i)\backslash j$ denotes the neighbor of i except j. At each iteration, each node can get an approximation $\hat{p}^n(\mathbf{x}_j|\mathbf{Z})$ to the marginal distribution $p(\mathbf{x}_j|\mathbf{Z})$ by combining the incoming messages with the local observation:

$$\hat{p}^n(\mathbf{x}_j|\mathcal{Z}) = \alpha_3 \phi_j(\mathbf{x}_j, \mathbf{z}_j) \prod_{i\in\Gamma(j)} m_{ij}^{n}(\mathbf{x}_j) \tag{40}$$

For tree-structured graphs, $\hat{p}^n(\mathbf{x}_j|\mathcal{Z})$, also called belief, will converge to the true marginal distribution $p(\mathbf{x}_j|\mathcal{Z})$. However, for graphs with continuous state variable $\mathbf{x}_i$, exact inference using integration (Equation 39) is often infeasible. As a result, messages (and marginal distributions) can be represented nonparametrically by a set of weighted particles or a set of kernel densities [SIFW03, Isa03, HW04]. Correspondingly, several message update methods in the nonparametric belief propagation are designed [SIFW03, Isa03, HW04]. NBP can be viewed as an extension of particle filter and can be used in more general vision problems that graphical model can describe.

References

[AASK04] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *CVPR*, 2004.

[AS00] V. Athitsos and S. Sclaroff. Inferring body pose without tracking body parts. In *CVPR*, 2000.

[AS03] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *CVPR*, 2003.

[AT04] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *CVPR*, 2004.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998.

[Bra99] M. Brand. Shadow puppetry. In *ICCV*, 1999.

[CI00] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *CVPR*, 2000.

[CI01] Y. Capsi and M. Irani. Alignment of non-overlapping sequences. In *ICCV*, 2001.

[CI02] Y. Capsi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Trans. on Pattern Analysis and Medical Intelligence*, 24(11):1409–1424, 2002.

[Coh92] M. F. Cohen. Interactive spacetime control for animation. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):293–302, 1992.

[CR99] T.J. Cham and J.M. Rehg. A multiple hypothesis approach to figure tracking. In *CVPR*, 1999.

[CSI02] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. In *VAMODS workshop with ECCV*, 2002.

[DBR00] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000.

[DCR01] D.E. Difranco, T.J. Cham, and J.M. Rehg. Recovery of 3-D figure motion from 2-D correspondences. In *CVPR*, 2001.

[DF99] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *ICCV*, 1999.

[EL04] A. Elgammal and C.S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *CVPR*, 2004.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision.* The MIT Press, Cambridge, Massachusetts. USA, 1993.

[FL95] C. Faloutsos and K.I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD*, pages 163–174, 1995.

[FP03a] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, 2003.

[FP03b] D.A. Forsyth and J. Ponce. Tracking with non-linear dynamic models. One chapter excluded from "Computer Vision: A Modern Approach", 2003.

[GD96] D.M. Gavrila and L.S. Davis. 3D model-based tracking of humans in action: A multi-view approach. In *CVPR*, 1996.

[GL98] M. Gleicher and P. Litwinowicz. Constraint-based motion adaptation. *J. Visual. Comput. Animation*, 9:65–94, 1998.

[Gle97] M. Gleicher. Motion editing with spacetime constraints. *Proceedings 1997 Symposium on Interactive 3D Graphics*, pages 139–148, 1997.

[Gle98] M. Gleicher. Retargeting motion to new characters. In *ACM SIGGRAPH*, 1998.

[Gle01] M. Gleicher. Comparing constraint-based motion editing methods. *Graphical Models*, 63:107–134, 2001.

[GMW81] P. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, A Subsidiary of Harcourt Brace Jovanovich, Publishers, 1981.

[GP99] M. Giese and T. Poggio. Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. *In IEEE Workshop on Multi-view Modeling and Analysis of Visual Scene*, 1999.

[GS03] T. D. Kristen Grauman and G. Shakhnarovich. Inferring 3D structure with a statistical image-based shape model. In *ICCV*, 2003.

[HLF99] N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *NIPS*, 1999.

[HS03] G.R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *PAMI*, 25(5):530–549, 2003.

[HW04] G. Hua and Y. Wu. Multi-scale visual tracking by sequential belief propagation. In *CVPR*, 2004.

[IB96] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV*, 1996.

[Isa03] M. Isard. Pampas: Real-valued graphical models for computer vision. In *CVPR*, 2003.

[KM96] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. In *CVPR*, 1996.

[KP01] E.J. Keogh and M.J. Pazzani. Derivative dynamic time warping. *Department of Information and Computer Science, University of California, Irvine*, 2001.

[LRS00] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. on Paatern Analysis and Machine Intelligenece*, 22:758–767, August 2000.

[LS99] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH*, 1999.

[MM02] G. Mori and J. Malik. Estimating human body configurations using shape context matching. *ECCV*, 2002.

[MRR80] C. Myers, L. Rabinier, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(6):623–635, 1980.

[MW97] J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM J. Optimization*, pages 814–836, 1997.

[Neu03] A. Neumaier. *Complete search in continuous global optimization and constraint satisfaction*, November 2003.

[RAS01] R. Rosales, V. Athitsos, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *ICCV*, 2001.

[RGSM03] C. Rao, A. Gritai, M. Shah, and T.S. Mahmood. View-invariant alignment and matching of video sequences. *In ICCV*, 2003.

[RS00a] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of human body pose from a single image. In *Workshop on Human Motion*, pages 19–24, 2000.

[RS00b] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[SBF00] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *ECCV*, 2000.

[SBR$^+$04] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, 2004.

[SIFW03] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *CVPR*, 2003.

[SMFW04] E.B. Sudderth, M.I. Mandel, W.T. Freeman, and A.S. Willsky. Visual hand tracking using nonparametric belief propagation. In *IEEE CVPR Workshop on Generative Model based Vision*, 2004.

[ST01] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body trakcing. In *CVPR*, 2001.

[ST03] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *CVPR*, 2003.

[Ste98] G.P. Stein. Tracking from multiple view points: Self-calibration of space and time. *In DARPA IU Workshop*, pages 521–527, 1998.

[SVD03] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

[TdSL00] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[TGB00] D. Tolani, A. Goswanmi, and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62:353–358, 2000.

[Tip00] M. Tipping. The relevance vector machine. In *Neural Information Processing Systems*, 2000.

[WK88] A. Witkin and M. Kass. Spacetime constraints. *Computer Graphics (SIGGRAPH 88 Proceedings)*, 22:159–168, 1988.

[WN99] S. Wachter and H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[YFW02] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical report, MERL, 2002.