Abstract

Title of Dissertation: Vision-based 3-D Tracking of Humans in Action

Dariu M. Gavrila, Doctor of Philosophy, 1996

Dissertation directed by: Professor Larry Davis
Department of Computer Science

The ability to recognize humans and their activities by vision is essential for future machines to interact intelligently and effortlessly with a human-inhabited environment. Some of the more promising applications are discussed.

A prototype vision system is presented for the tracking of whole-body movement using multiple cameras. 3-D body pose is recovered at each time instant based on occluding contours. The pose-recovery problem is formulated as a search problem and entails finding the pose parameters of a graphical human model whose synthesized appearance is most similar to the actual appearance of the real human in the multi-view images. Hermite deformable contours are proposed as a tool for the 2-D contour tracking problem.

The main contribution of this dissertation is that it demonstrates for the first time a set of techniques that allow accurate vision-based 3-D tracking of arbitrary whole-body movement without the use of markers.

Vision-based 3-D Tracking of Humans in Action

by

Dariu M. Gavrila

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1996

Advisory Committee:

Professor Larry Davis, Chairman/Advisor
Professor Rama Chellappa
Professor David Mount
Professor Alex Pentland
Professor Azriel Rosenfeld
Professor Hanan Samet

Dedication

To my parents, with gratitude.

Acknowledgements

First, I would like to thank my advisor Professor Larry Davis for his continued support throughout my Dissertation. I appreciated his confidence for allowing me to pursue a topic which seemed very challenging and interesting. I still owe him the analysis of the Argentine Tango.

It was a pleasure to have the opportunity to interact with Professor Alex Pentland and his fine research group during my stay at the M.I.T. Media Laboratory. Although the research done there (Chapter 5) remained in its early stages due to time limitations, the insights gained are long-lasting.

Furthermore, I would like to thank Prof. Azriel Rosenfeld for the meticulous proof reading of various technical reports and for his willingness and insistence to go through his extensive vision bibliographies to assure no reference was overlooked in this dissertation. I would also like to thank the other members of the Dissertation Committee, Professors Rama Chellappa, David Mount and Hanan Samet for their useful feedback.

A number of current and former graduate students have helped me through the past years with their good friendship. Greg Baratoff was always willing to listen to my practice talks and give me thorough comments on the many drafts I would put before him. By now, he also knows my furniture by hart, having helped me with my many moves from one place to another. Zoran Duric was always ready to translate his research experience in good advice and to discuss real-life issues in the many coffee and tea conversations we had, once I learned how to avoid discussing certain topics in European politics.

A pleasant working environment was facilitated by many people through the years. Among them were Mark, Yaser, Vasi, Ross, Hismail, Esin, Thana, Sandor, Margrit and Tin-Tin. I also appreciated the friendly and very effective UMIACS support and administrative staff. Nancy Lindley helped me stay on good terms with the Graduate School, and provided a welcome outlet for my cello-playing by organizing an annual chamber music concert at her house.

Furthermore, I would like to thank P.J. Narayanan and Pete Rander for their help in using the facilities at C.M.U.'s 3-D Studio for the experiments, and Kim's Tailor, College Park, for a body-suit job well done.

I am very indebted to my dear Ellen Koopmans; her help was instrumental to complete the Dissertation as it is in current form. She not only provided much needed support and understanding during tough times, but also was my chief editor and my skillful assistant in setting up and performing the experiments.

Finally, it is with gratitude that I acknowledge the long-term support and advice of my parents Professors Mihai Gavrila and Liana Serbescu, whose example I would like to follow.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research context

Despite the great strides computers have made over the years in terms of processing speed and data storage, there has been limited progress in making them intelligent assistants in people's everyday lives. Such an assistant would recognize you when you wake up in the morning, tell you what appointments you have today, assist you in driving your car to work, allow you to work in interactive 3-D spaces with participants at other sites, warn you if your children get in trouble at home, and finally, perhaps play your favorite music when you return home.

The main reason for this lack of capability is that computers are currently unable to perceive the environment in which they are embedded. Or, as Alex Pentland from the M.I.T. Media Lab puts it, they are "blind and deaf" [73]. They need to be spoon-fed with information by keyboard or mouse input; a very low-bandwidth and tedious form of communication. This inevitably leads to a reactive mode of operation, where typed-in commands are simply executed. An important aspect of what humans perceive as intelligence is, however, a proactive stance, based on what is currently happening in the environment and what is about to happen.

If computers are to be made aware of their environment, vision and speech are the modalities of greatest importance. It is more practical for computer vision to restrict its scope to recognizing humans and their activities, rather than aim to solve the general object recognition problem. Prior knowledge about human shape and articulation can then be used to simplify the vision problem, and a real system can be built.

1.2 Applications

There are a number of promising applications in the "Looking at People" area in computer vision, in addition to the general goal of designing a machine capable

general domain	specific area
virtual reality	- interactive virtual worlds - games - virtual studios - character animation - teleconferencing (e.g. film, advertising, home-use)
“smart” surveillance systems	- access control - parking lots - supermarkets, department stores - vending machines, ATMs - traffic
advanced user interfaces	- social interfaces - sign-language translation - gesture driven control - signaling in high-noise environments (airports, factories)
motion analysis	- content-based indexing of sports video footage - personalized training in golf, tennis, etc. - choreography of dance and ballet - clinical studies of orthopedic patients
model-based coding	- very low bit-rate video compression

Table 1.1: Applications of “Looking at People”

of interacting intelligently and effortlessly with a human-inhabited environment. These applications will be now discussed in some detail; for a summary please see Table 1.

An important application domain is smart surveillance. Here “smart” describes a system that does more than motion detection, a straightforward task prone to false alarms (there might be animals wandering around, wind blowing, etc.). A first capability would be to sense if a human is indeed present. This might be followed by face recognition for the purpose of access control. In other applications, one needs to determine what a person in the scene is doing, rather than simply signaling human presence. In a parking lot setting, one might want to signal suspicious behavior such as wandering around and repeatedly bending over to cars. In a supermarket or department store setting, valuable marketing information could be obtained by observing how consumers interact with the merchandise; this would be useful in designing a store lay-out which encourages sales. Other surveillance settings involve vending machines, ATMs and traffic.

Another application domain is virtual reality. In order to create a presence in a virtual space one needs to recover the body pose in the physical space first. Application areas lie in interactive virtual worlds, with the internet as a possible medium. The development of interactive spaces on the internet is still in its infancy; it is in the form of "chat rooms" where users navigate with icons in 2-D spaces while communicating by text. A more enriched form of interaction with other participants or objects will be possible by adding gestures, head pose and facial expressions as cues. Other applications in this domain are games, virtual studios, motion capture for character animation (synthetic actors) and teleconferencing.

In the user-interface application domain, vision is useful to complement speech recognition and natural language understanding for a natural and intelligent dialogue between human and machine. The contribution of vision to a speech-guided dialogue can be manifold. One can simply determine if a user is present to decide whether to initiate a dialogue or not. More detailed cues can be obtained by recognizing who the user is, observing facial expressions and gestures as the dialogue progresses, perhaps remembering some of the past interactions, and determining who is talking to whom in case of multiple participants. Vision can also provide speech recognition with a more accurate input in a noisy environment by focusing the attention to the spatial location of the user. This is achieved either by a post-filtering step when using a phased array of microphones or, more actively, by directing a parabolic microphone to the intended source. Finally, vision can also prove helpful for phoneme disambiguation i.e. lip reading.

One important application area in the user-interface domain is in social interfaces. They involve computer-generated characters, with "human-like" behaviors, who attempt to interact with users in a more personable way [97]. Other application areas in the user interface domain are sign-language translation, gesture driven control of graphical objects or appliances, and signaling in high-noise environments such as factories or airports.

Another application domain is motion analysis in sports and medicine. A specific application area is context-based indexing of sports video footage. In a tennis context, one may want to query a large video archive with "give me all the cases where player X came to the net and volleyed". This would eliminate the need for a human to browse through a large data set. Other applications lie in personalized training systems for various sports; these systems would observe the skills of the pupils and make suggestions for improvement. Vision-based motion analysis is also useful for in the choreography of dance and ballet, and also for clinical studies of orthopedic patients.

Finally, one can add model-based coding as a possible application domain. In a video phone setting, one could track faces in image sequences and code them in more detail than the background. More ambitiously, one might try to recover a 3-D head model initially and code only the pose and deformation parameters subsequently. It is unclear whether these applications will materialize; the first because it provides a rather modest compression gain and is specific to scenes with human faces, the second because it involves significant processing, which at least currently, is nowhere near real-time and the results are poor when compared to general-purpose compression.

In all of these applications, a non-intrusive sensory method based on vision is preferable over a (in some cases a not even feasible) method that relies on markers attached to the bodies of the human subjects or a method which is based on active sensing.

1.3 Outline

This dissertation deals with the vision-based analysis of scenes involving humans. The general approach has been to make extensive use of prior knowledge, in terms of generic 3-D human models, in order to recover 3-D shape and pose information from 2-D image sequences.

The dissertation is divided in six chapters. Chapter 2 discusses the relevant background, starting from an inter-disciplinary perspective and then focusing on the work in computer vision on the analysis of hand and whole-body movement. The pose recovery and tracking approaches have been grouped in three sections, depending on the model and the dimensionality of the tracking space which is used. The last section deals with past work on movement recognition; at this stage, the relevant features have been extracted from the images.

Chapter 3 describes a prototype vision system which uses multiple cameras, placed in the corners of a room, to observe a scene where one or more human performs some type of activity. Its aim is to recover from the multi-view images the 3-D body pose of the humans over time and subsequently, to recognize body movements. The chapter starts with a motivation of the choice to pursue a 3-D recovery approach rather than a 2-D approach.

Section 3.2 discusses the general framework for model-based tracking as used in this work. Section 3.3 covers the 3-D human modeling issues and the (semi-automatic) model-acquisition procedure which is invoked initially. Section 3.4 deals with the pose estimation component once the human model has been acquired. Included is a bootstrapping procedure to start the tracking or to re-initialize if it fails. Section 3.5 discusses the prediction and image analysis component. Section 3.6 proposes Dynamic Time Warping for movement classification. Section 3.7 presents experimental results in which successful unconstrained whole-body movement is demonstrated on two subjects. These are results derived from a large Humans-In-Action (HIA) database containing two subjects involved in a variety of activities, of various degrees of complexity.

The search for a better edge segmentation during tracking has led to the work on deformable contours reported in Chapter 4. Deformable contours are energy-minimizing models which overcome some of the problems of traditional bottom-up segmentation methods, such as edge gaps and spurious edges, by taking into account shape prediction in addition to image features. The underlying idea is to use this technique in the before-mentioned tracking system of humans, by initializing the deformable contours at the predicted location of the human(s) in the new frame. Section 4.1 discusses related work. In Section 4.2 the Hermite representation is proposed for deformable contour finding, together with an optimization procedure based on dynamic programming. The experiments are described in Section 4.6.

The last part of the dissertation, Chapter 5 deals with initial results on recursive 3-D head shape estimation from monocular images. Given reasonably noisy feature-tracks, 3-D head motion is estimated recursively using a Kalman filter. Using this motion estimate and a generic 3-D head model, fitted to a frontal view, initial results are reported on recovering 3-D head shape from contours.

Finally, Chapter 6 contains the conclusions of this dissertation together with suggestions for future work.

1.4 Problem formulation

The main purpose of this thesis is to present a set of techniques which allow vision-based 3-D pose recovery and tracking of unconstrained whole-body human movement without the use of markers. Related to this goal is an investigation of image segmentation techniques that can take advantage of a model-based tracking approach. Towards the end of the thesis, preliminary attention has been given to the problem of 3-D head model acquisition from monocular head-shoulder images.

The current prototype system for 3-D whole-body pose recovery and tracking operates under the following conditions

- The system knows only about humans. It does not model any other objects in the scene (e.g. tables, chairs) and can therefore be thrown off by large occlusions of human body-parts by these objects.

- The system uses multiple cameras. Although many of the techniques described here apply to the monocular case as well, it is acknowledged that multiple cameras greatly aid in successful 3-D tracking by allowing better object localization and motion disambiguation. Multiple cameras are especially helpful for 3-D model-acquisition.

- The cameras are calibrated. For the followed 3-D recovery approach by synthesis it is necessary to know the relative positioning of the cameras and their effective focal length in order to know how 3-D structure (with respect to one camera or to the world) is mapped onto pixel coordinates. Camera calibration is done initially.

- The system uses a simplified model for the human body. The body is modeled as an articulated rigid body and does not account for loose fitting clothes, loose hair and muscle deformations. In practice, these simplifications are acceptable; the purpose of 3-D modeling is not to allow highly realistic renderings of the human, but to capture shape sufficiently accurate to support image segmentation. The system does not model dynamics such as the notion of support. It is aware, though, of joint angle limits.

- Model-acquisition is done semi-automatically in an initialization procedure. This involves frontal and sideway views of each body part derived from an externally supplied contour segmentation.

- The initial 3-D pose of the human at the start of tracking is approximately known. For the case of a single human in the scene standing upright, this assumption can be relaxed.

- There is no pose ambiguity. Each body part is visible in the images, or, in the case of occlusion, its location can be determined by model constraints. The system maintains only one estimate of the current pose and it cannot handle conditions where, due to occlusion, a variety of poses are acceptable. Also, the system does not know when to stop or start tracking an occluded body part.

No image segmentation is assumed given to the system other than for the model-acquisition stage. In particular, no feature point-correspondences (e.g. at joints) between model and image are assumed given. Moreover, the system does not even attempt to recover point-features in images for correspondence with the model or across multiple views, because of lack of such identifiable features over the whole human body. The system uses perspective projection.

The main problem of this thesis, as formulated above, is challenging because it involves

- segmentation of rapidly changing 2-D scenes

- recovering 3-D structure and motion

- dealing with articulated (non-rigid) motion

- handling (self) occlusions

Chapter 2

Background

There has been keen interest in human movement from a wide variety of disciplines. In psychology, there have been the classic studies on human perception by Johansson [45]. His experiments with moving light displays (MLD) showed that human observers can almost instantly recognize biological motion patterns even when presented with only a few of these moving data points. This suggested that recognition of moving parts could be achieved directly from motion, without structure recovery. In the hand gesture area, there have been many studies on how humans use and interpret gestures, see for example work by McNeill [63]. Quek [76] has put this in the context of vision-based human-computer interfaces.

In kinesiology (i.e. biomechanics) the goal has been to develop models of the human body that explain how it functions mechanically. The increase of movement efficiency has also been an issue. The first step involves motion capture by placing active or passive markers on the human subject. Typically, the data undergoes kinematic analysis followed by the computation of forces and torques, see [17].

In choreography, there has been long-term interest in devising high-level descriptions of human movement for the notation of dance, ballet and theater. Some of the more popular notations have been the Labanotation, the Ekshol-Wachmann and the Effort-Shape notation. Across the variety of notation systems there has been little consensus of what these general-purpose descriptions should be. Badler and Smoliar [7] provide a good discussion.

Computer graphics has dealt with the synthesis of human movement. This has involved devising realistic models for human bodies for applications in crash simulations, workplace assessment and entertainment. Some of the issues have been how to specify spatial interactions and high-level tasks for the human models. See [7] [6] [61].

The reported work in vision has increased significantly over the past three years, following the “Looking at People” workshop in Chambery (1994) and the two “Automatic Face and Gesture Recognition” workshops in Zürich (1995) and Killington (1996). Some of it has now also reached the popular scientific press [73]. This chapter discusses previous work dealing with the vision-based analysis of hand and whole-body movement. The hand and whole-body tracking problems are discussed together because of their similarities (i.e. both involve articulated structures).

Previous work has dealt with human body segmentation, pose recovery, tracking and action recognition. It is useful to consider the following dimensions when classifying previous vision work:

- the type of models used (stick figures, volumetric models, none),
- the dimensionality of the tracking space (2-D or 3-D),
- sensor modality (visible light, infra-red, range),
- sensor multiplicity (monocular, stereo),
- sensor placement (centralized vs. distributed) and
- sensor mobility (stationary vs. moving).

For convenience, the discussion is organized in three parts, based the first two dimensions: the 2-D model-free approach, the 2-D model-based approach and the 3-D model-based approach. This is followed by a discussion of the remaining work. Earlier reviews were given by Aggarwal *et al.* [1] and Cedras and Shah [19].

2.1 The 2-D model-free approach

One possible approach to action recognition has been to bypass a structure recovery step altogether and to use simple "low-level", model-free 2-D features from a region of interest. This approach has been especially popular for applications of hand pose estimation in sign language recognition and gesture-based dialogue management.

For hand pose estimation, the region of interest is typically obtained by background image subtraction or skin color detection. This is followed by morphological operations to remove noise. The extracted 2-D features are based on hand shape, movement and/or location of the interest region. For shape, Freeman *et al.* [31] use x-y image moments and orientation histograms, Hunter *et al.* [43] use rotationally invariant Zernike moments and Kjeldsen and Kender [53] use the cropped region directly. Others [94] [21] [25] [91] consider the motion trajectories of the hand centroids. Pose classification is based on hard-coded decision trees [94] [21] [25], nearest neighbor criteria [43], neural networks [53] or Hidden Markov Models [91]. Some additional constraints on pose can be imposed using a dialogue structure where the current state limits the possible poses that can be expected next.

Similar techniques have been applied for model-free whole-body action recognition. A $K \times K$ spatial grid is typically superimposed on the interest region, after a possible normalization of its extent. In each of the $K \times K$ tiles a simple feature is computed, and these are combined to form a $K \times K$ feature vector to describe the state of movement at time t. Polana and Nelson [75] use the sum of the normal flow within a tile as feature, Yamamoto et al. [103] use the number of black pixels in the thresholded tile and Takahashi et al. [93] define an average edge vector for each tile. Darell and Pentland [24] use the image pixels directly for their correlation-matching approach. Action recognition is subsequently considered as a time-varying pattern matching problem and a number of techniques apply which will be discussed later, in Section 2.4.

General-purpose motion-based segmentation and tracking techniques have also been used for applications such as people counting. Shio and Sklansky [88] aim to recover the average 2-D image velocity of pedestrians in a traffic setting. They obtain a motion field based on correlation techniques over successive frames. The motion field is smoothed both spatially and temporally to reduce the effects of non-rigid motion and measurement errors. A quantization of the field is then followed by an iterative merging step which results in regions with similar motion direction. Segen and Pingali [86] group partially-overlapping feature tracks over time in a real-time implementation.

2.2 The 2-D model-based approach

This section discusses work which uses prior knowledge of how the human body (or hand) appears in 2-D, taking essentially a model- and view-based approach to segment, track and label body parts. Since self-occlusion makes the problem quite hard for arbitrary movements, many systems assume a-priori knowledge of the type of movement and/or the viewpoint under which it is observed.

A number of researchers have analyzed scenes involving human gait parallel to the image plane. Geurtz [33] performs hierarchical and articulated curve fitting with 2-D ellipsoids. Niyogi and Adelson [68] [69]advocate segmentation over time because of robustness; their procedure involves finding human silhouettes with deformable contours in X-T space [68] or deformable surfaces in X-Y-T space [69]. Guo, Xu and Tsuji [36] propose obtaining a 2-D stick figure by obtaining the skeleton of the silhouette of the walking human and matching it to a model stick figure. They use a combination of link orientations and joint positions of the obtained stick figure as features for a subsequent action recognition step. Ju, Black and Yacoob [48] use a parametrized motion model to analyze gait constrained to a plane. The legs are modeled a set of connected planar patches.

An early attempt to segment and track body parts under more general conditions is made by Akita [3]. The assumption here is that the movement of the human is known a-priori in the form of a set of representative stick figure poses or "key frames". These would be of help when the the tracking of body parts fails. Unfortunately, it is not clear from the paper how well the proposed segmentation and tracking algorithms perform. Without a-priori knowledge of the type of movement being performed, Long and Yang [60] track the limbs of the human silhouette by tracking anti-parallel lines (apars). They develop methods to deal with occlusions, resulting in the appearance, disappearance, merging and splitting of the apars. Leung and Yang [58] report progress on the general problem of segmenting, tracking and labeling of body parts from a silhouette of the human. Their approach features moving edge detection, ribbon tracking and a number of structural constraints, including the concept of support. Wren *et al.* [101] take a region-based approach. Their real-time system models and tracks the human body as a connected set of "blobs"; these are image regions defined by similar color statistics. Heuristics based on spatial relationships are used for the labeling of body parts. Finally, Kahn and Swain [49] describe a system which detects humans pointing laterally.

2.3 The 3-D model-based approach

In this section we discuss work that aims to recover 3-D articulated pose over time, i.e. joint angles with respect to an object-centered [62] coordinate system. 3-D motion recovery from 2-D images is often an ill-posed problem. In the case of 3-D human tracking, however, one can take advantage of the large available a-priori knowledge about the kinematic and shape properties of the human body to make the problem tractable. Tracking also is well supported by the use of a 3-D model which can predict events such as (self) occlusion and (self) collision. Once 3-D tracking is successfully completed, one has the benefit of being able to use the 3-D joint angles as features for movement matching, which are viewpoint independent and directly linked to the body pose. Compared to 3-D joint coordinates, joint angles are less sensitive to variations in the size of humans.

A quick and accurate, yet obtrusive, method to obtain 3-D joint data with multiple cameras involves placing easily-identifiable markers on the joints and obtaining 3-D data by triangulation. In the remainder, the discussion deals with approaches to articulated motion recovery which are not based on triangulation. They use a single camera (unless stated otherwise) and the model (i.e. stick figure or volumetric) is assumed given.

One approach to 3-D articulated pose recovery from a sequence of single-view images is to use a divide-and-conquer technique. This involves decomposing the object into a number of simple (rigid or articulated) parts, solving for motion and depth of the subparts and verifying whether the parts satisfy the necessary constraints. Shakunaga [87] identifies such a set of primitive articulated structures for which he solves the pose recovery problem using angles between line features in the image.

To avoid unfavorable combinatorics at the verification step, it is beneficial to propagate constraints from part to part. The primitives of O'Rourke and Badler [71] are box-shaped regions which represent possible joint locations in 3-D. These regions are initially refined by the measurement of the joints in the images (assumed given) and the orthography assumption. A constraint propagation procedure is then applied based on the known distances between connected joints. The verification procedure involves an iterative search procedure on the refined 3-D uncertainty regions, in which angular and collision constraints are verified using the 3-D model.

Other work has used projective geometry. The constraint propagation scheme of Chen and Lee [22] starts at the human head and continues via the torso to the limbs. An interpretation tree is built to account for the spatial ambiguity which arises from the fact that there are two possible poses of a link (of known length) in 3-D which result in the same 2-D projection. This interpretation tree is pruned later for physically implausible poses. Chen and Lee's assumption of six known feature points on the head to start the procedure and the overhead of the interpretation tree makes their approach somewhat less appealing. Zhao [104] has a similar problem formulation but does not maintain the interpretation tree, considering instead only one pose at the time. He monitors when spatial ambiguities are encountered and disambiguates them by temporal coherence. Holt *et al.* [42] provide a constraint propagation scheme for human gait, where one joint remains at a fixed location. Motion constraints are also incorporated at the earliest stages. The core of their system involves solving a polynomial system of equations. Other approaches have imposed general constraints on the articulated motion, such as the "fixed-axis" [99] or in-plane [40] assumptions of rotations.

Hel-Or and Werman [38] describe a technique for articulated pose recovery based on the fusion of constraints and measurements using a Kalman filter framework. Kakadiaris and Metaxas [51] use a physics-based approach where various forces act on the different parts to align them with the image data. Constraint forces enforce point-to-point connectivity between the parts.

Other approaches to 3-D articulated motion use parametrized models where the articulation constraints are encoded in the representation itself. One such approach [28] [104] [80] [81] [35] uses feature correspondence between model and image to update pose by inverse kinematics, a common technique in robot control theory [89]. The state space maps onto image space by a non-linear measurement equation which takes into account the coordinate transformations at various articulation sites and the 3-D to 2-D projection. Inverse kinematics involves inverting this mapping to obtain changes in state parameters which minimize the residual between projected model and image features. The procedure involves a linearization of the measurement equation, as defined by the Jacobian matrix, and a gradient-based optimization scheme. The inverse kinematics approach can also be taken with multiple cameras when no feature correspondence between cameras is assumed. One simply concatenates the residual from the available camera views, see for example [81].

Another approach using parametrized models does not attempt to invert a non-linear measurement equation. Instead it uses the measurement equation directly to synthesize the model and uses a fitting measure between synthesized and observed features for feedback, see [41] [29] [82] [74] [70] [56]. Pose-recovery can then be formulated as a search problem which entails finding the pose parameters of a graphical human model whose synthesized appearance is most similar to the actual appearance of the real human. These systems do not need to assume point correspondences between model and image, they match based on occluding contours. Ohya and Kishino [70] use a global search strategy based on genetic algorithms and Kuch and Huang [56] generate random moves locally. Straightforward extensions to the multi-camera case have also been employed in simulations [70].

Finally, a different approach to articulated pose recovery altogether has been proposed by Heap and Hogg [37]. Their example-based approach allows shape deformations as well and is based on a principal component analysis of 3-D positional data.

Deriving feature correspondences between model and image remains, of course, an open issue. Approaches [71] [22] [104] [42] which assume that joint correspondences are given make strong assumptions. Moreover, relying exclusively on a few joint correspondences makes the resulting approach [28] [80] quite sensitive to occlusion. To alleviate this, some work has used evaluation measures for the model-to-image fit based on image regions in the neighborhood of the projected model contours. These include measures based on correlation (on a raw or smoothed LOG-filtered image) [35] [81], normal distance from projected model contours to image edges [37] and straight-line distance metrics [82]. The approach by Yamamoto and Koshikawa [102] uses optical flow. Others assume feature correspondence is given to their pose recovery algorithm [87] [22] [104] [42].

The last part of this section reviews the previous work in terms of experimental results on real data. Dorner [28] tracks articulated 3-D hand motion (palm motion and finger bending/unbending) with a single camera in several examples. However, her system requires colored markers on the joints and cannot handle occlusions. Rehg and Kanade [80] do not require markers. They track an 8-DOF partial hand model (movement of palm in a 2-D plane and three fingers) with one camera and a full 27-DOF hand model with two cameras in near real-time from the hand silhouette (hand is against a black background). Occlusion cannot be handled, although a later version [81] does tolerate some occlusion; a successful tracking example is shown where one finger moves over the other finger, with the rest of the hand fixed. Heap and Hogg [37] show initial tracking results on hand model and hand pose recovery; several questions about occlusion and implausible model shapes still remain open.

In terms of experimental results on whole-body movement using a single camera, Hogg [41] and Rohr [82] deal with the restricted movement of gait (parallel to image plane). The movement is essentially in 2-D with no significant movement in depth. Given that gait is modeled a-priori, the resulting search space is one-dimensional. Downton and Drouet [29] attempt to track unconstrained upper-body motion but must conclude that tracking gets lost due to propagation of errors. Both Goncalves *et al.* [35] and Kakadiaris and Metaxas [51] track one arm while keeping the shoulder fixed. Goncalves *et al.* [35] furthermore assume that the 3-D shoulder position is known. Finally, Perales and Torres [74] describe a multi-view system which requires input from a human operator.

In almost all previous approaches on real data it has been difficult to ascertain how good the 3-D pose recovery results are; no ground truth is given and no orthogonal camera view is available to, at least visually, verify the recovered pose along the depth dimension.

2.4 Action recognition

There are different ways to view human action recognition. A narrow interpretation considers it simply as a classification problem involving time-varying feature data. This consists of matching an unknown test sequence with a library of labeled sequences which represent the prototypical actions. A complementary problem is how to learn the reference sequences from training examples. Both learning and matching methods have to be able to deal with small data and time scale variations within similar classes of movement patterns.

Rangarajan *et al.* [79] match motion trajectories of selected points by a parametrization based on the locations where significant changes in direction or speed occur. Matching between reference and test trajectories allows a fixed amount of time-offset, using a Gaussian-convoluted reference parametrization. Goddard [34] represents activities by scenarios; a sequence of events with enabling conditions, and time-constraints between successive events. Each possible scenario is matched and given a measure of appropriateness, depending on the cumulative confidence in the scenario, the likelihood that its “next” event has occured, and the time-constraints. No learning takes place in the previous two methods. Campbell and Bobick [18] use a phase-space representation in which the velocity dimensions are projected out, discarding the time component of the data altogether. This makes the learning and matching of patterns simpler and faster, at the potential cost of an increase in false positives. Other general techniques for time-varying data analysis have been used as well: Dynamic Time Warping (DTW) [24] [93] [100], Hidden Markov Models (HMM) [77] [103] [91] and Neural Networks (NN) [36] [83].

Another aspect of human action recognition are static postures; sometimes it is not the actual movement that is of interest but the final pose (for example, pointing). Herman [39] describes a rule-based system to interpret body posture given a 2-D stick figure. Although the actual system is applied on a toy problem (in baseball), it does make the point to use qualitative pose measures together with other attributes such as facing direction and contact. It also emphasizes the importance of contextual information in action recognition.

Chapter 3

3-D body tracking and movement recognition

3.1 Motivation for the followed approach

This work takes a 3-D recovery approach to body tracking and movement recognition using a joint-parametrized graphical model for synthesis. Rather than employing a gradient-based optimization scheme, it uses local search in the pose parameter space of the model. Matching between synthesized model and image is based on occluding contours. Multiple cameras are used; they use perspective geometry. This approach is motivated as follows.

Recognition systems using 2-D features have been able to claim early successes in the analysis of human movement. For applications with typically a single human, constrained movement and a single viewpoint (i.e. recognizing gait parallel to the image plane, lateral pointing gestures, small vocabulary of distinct hand gestures) the 2-D approach often represents the easiest and best solution.

The aim of this work, in contrast, is to deal with unconstrained and complex (multi) human movement (e.g. humans wandering around, making different gestures while walking and turning, social interactions such as shaking hands and dancing). It is deemed unlikely that this can be achieved by a purely 2-D approach. A 3-D approach leads to a more accurate, compact representation of physical space which allows a better prediction and handling of occlusion and collision. It leads to meaningful features for action recognition, which are directly linked to body pose. Furthermore, the 3-D recovery approach is of independent interest for its use in virtual reality applications.

The advantage of using joint-parametrized human models and synthesis for 3-D pose recovery is that the resulting approach takes advantage as much as possible of prior 3-D knowledge and relies as little as possible on error-prone 2-D image segmentation. Unlike other work [71] [22] [28] [104] [42] [80] no point feature correspondences are needed between model and image (e.g. at the joints). Matching is based on whole (occluding) contours and regions, rather than based on a few points.

It is recognized that an inverse kinematics approach with the associated gradient-based optimization scheme has the advantage that it exploits gradient cues in the vicinity of a minimum and therefore can be very efficient in some cases, see for example [80]. For robustness, a local non-greedy search method was chosen here instead, taking the higher computational cost in stride. The main reason for choosing for a non-greedy search procedure is that in this application a gradient-based method is very likely to get stuck in a local minimum, i.e. to converge to a sub-optimal or undesired solution. There are two main reasons why this is to be expected.

First, the measurement equation is highly non-linear. It contains a composition of various non-linear rotation matrices at the articulation sites and the full 6-DOF rigid transformation matrix of the root. The measurement equation will also have to include the non-linearity of the perspective projection. At the same time, the sampling ratio at which measurements are obtained is limited to frame rate. This is a problem for fast movements of locomotion and gesticulation; large parameter deviations will be poorly captured by a linearization of the measurement equation around the current state. Second, the measurements can be noisy or incorrect. No known points-correspondences are assumed between 3-D model and 2-D images. The correspondence of points on the occluding contours is prone to errors because of the aperture problem. Incorrect correspondences can be made altogether because of the existence of nearby noise edges or edges of different occluding contours ([80] [35] [51] do not have noisy data, their tracked object is white and the background is black).

A non-greedy search method also promises to be more robust over time; if it fails to find a good solution at time t, there is still a chance that it may recover at time $t + 1$, if the search area is sufficiently wide.

In terms of comparison with the previous systems which use a similar approach to 3-D model-based pose recovery based on synthesis, one can note that the current work deals with automatic tracking (unlike [74]), unconstrained full-body motion (unlike any previous work) and real data (unlike [71] [70]). Working with real data involves errors in body modeling, camera calibration and image segmentation which are difficult to simulate (for example, [71] [70] do not account for these errors). To build a robust system for the above conditions, several improvements are proposed in terms body models, search procedure and search evaluation measure (see next sections). Similar to [70], a multi-view approach is used to mitigate the effects of occlusion and allow better 3-D object localization.

3.2 Model-based tracking

The general framework of the proposed tracking system is inspired by the early work of O'Rourke and Badler [71]. It is illustrated in Figure 3.1. Four main components are involved: prediction, synthesis, image analysis and state estimation. The prediction component takes into account previous states up to time t to make a prediction for time $t+1$. It is deemed more stable to do the prediction at a high level (in state space) than at a low level (in image space), allowing an easier way to incorporate semantic knowledge into the tracking process. The synthesis component translates the prediction from the state level to the measurement (image) level, which allows the image analysis component to selectively focus on a subset of regions and look for a subset of features. Finally, the state-estimation component computes the new state using the segmented image.

The above framework is general and can also be applied to other model-based tracking problems. The next sections will discuss how the components are implemented in this system for the case of tracking humans.

3.3 3-D body modeling and model acquisition

3-D graphical models for the human body generally consist of two components: a representation for the skeletal structure (the “stick figure”) and a representation for the flesh surrounding it. The stick figure is simply a collection of segments and joint angles with various degree of freedom at the articulation sites. The representation for the flesh can either be surface-based (using polygons, for example) or volumetric (using cylinders, for example). There is a trade-off between the accuracy of representation and the number of parameters used in the model. Many highly accurate surface models have been used in the field of graphics [6] to model the human body, often containing thousands of polygons obtained from actual body scans. In vision, where the inverse problem of recovering the 3-D model from the images is much harder and less accurate, the use of volumetric primitives has been preferred to “flesh out” the segments because of the lower number of model parameters involved.

For the purposes of tracking 3-D whole-body motion, a 22-DOF model (3 DOF for the positioning of the root of the articulated structure, 3 DOF for the torso and 4 DOF for each arm and each leg) is used, without modeling the palm of the hand or the foot, and using a rigid head-torso approximation. See [6] for more sophisticated modeling. Here, the root of the articulated structure is kept fixed at the center of the torso. Transformations between different coordinate systems occur at sites of articulation (the joints) and are described by homogeneous coordinates $\mathbf{x} = (x, y, z, 1)^T$ using

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \tag{3.1}$$

where

$$\mathbf{H} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \tag{3.2}$$

and $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{T}$ is a 3×1 translation vector. The transformations are applied in fixed order, starting at the root and following the tree-structure which represents the connectivity of the segments in the articulated figure. In the case of human stick figure modeling, $\mathbf{T}$ is constant (no prismatic joints) and $\mathbf{R}$ is variable at articulation sites. $\mathbf{R}$ is compactly described by the three Euler angles (ϕ, θ, ψ) [89], i.e. the joint angles.

$$\mathbf{R} = \begin{pmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{pmatrix} \tag{3.3}$$

Human pose $\mathbf{P}(t)$ is thus represented by the time-varying 22-dimensional parameter vector

$$\begin{aligned} \mathbf{P}(t) = \quad & (\mathbf{P}_{\mathbf{ROOT}}(x,y,z),\ \mathbf{P}_{\mathbf{TORSO}}(\phi,\theta,\psi), \\ & \mathbf{P}_{\mathbf{L\_SHOULDER}}(\phi,\theta,\psi),\ \mathbf{P}_{\mathbf{L\_ELBOW}}(\theta), \\ & \mathbf{P}_{\mathbf{R\_SHOULDER}}(\phi,\theta,\psi),\ \mathbf{P}_{\mathbf{R\_ELBOW}}(\theta), \\ & \mathbf{P}_{\mathbf{L\_HIP}}(\phi,\theta,\psi),\ \mathbf{P}_{\mathbf{L\_KNEE}}(\theta), \\ & \mathbf{P}_{\mathbf{R\_HIP}}(\phi,\theta,\psi),\ \mathbf{P}_{\mathbf{L\_KNEE}}(\theta)) \end{aligned} \tag{3.4}$$

where (ϕ, θ, ψ) are the Euler angles. Throughout this work, the Euler angles (ϕ, θ, ψ) at the shoulders and hips will be called the elevation -, abduction - and twist angles and the Euler angles θ at the elbows and knees will be called the flexion angles.

Regarding the shape, it was felt that simple cylindrical primitives (possibly with elliptic XY-cross-sections), as in [29] [41] [82] [35], would not represent body parts such as the head and torso accurately enough. Therefore, the class of *tapered super-quadrics* [65] is employed; these include such diverse shapes as cylinders, spheres, ellipsoids and hyper-rectangles.

Their parametric equation $\mathbf{e} = (e_1 e_2 e_3)$ is given by [65]

$$\mathbf{e} = a \begin{pmatrix} a_1 C_u^{\epsilon_1} C_v^{\epsilon_2} \\ a_2 C_u^{\epsilon_1} S_v^{\epsilon_2} \\ a_3 S_u^{\epsilon_1} \end{pmatrix} \tag{3.5}$$

where $-\pi/2 \leq u \leq \pi/2, -\pi \leq v \leq \pi$ and where $S_\theta^\epsilon = sign(sin\theta)|sin\theta|^\epsilon$, and $C_\theta^\epsilon = sign(cos\theta)|cos\theta|^\epsilon$. Furthermore, $a \geq 0$ is a scale parameter, $a_1, a_2, a_3 \geq 0$ are aspect ratio parameters and ϵ_1, ϵ_2 are "squareness" parameters. Adding linear tapering along the z-axis to the super-quadric leads to the parametric equation $\mathbf{s} = (s_1 s_2 s_3)$ [65].

$$\mathbf{s} = \begin{pmatrix} (\frac{t_1 e_3}{a a_3} + 1) e_1 \\ (\frac{t_2 e_3}{a a_3} + 1) e_2 \\ e_3 \end{pmatrix} \tag{3.6}$$

where $-1 \leq t_1, t_2 \leq 1$ are the tapering parameters along the x and y axes. So far, satisfactory modeling results have been obtained with these primitives alone (see experiments); a more general approach also allows the deformation of the shape primitives [9] [72] [65].

Thus, human body shape $\mathbf{S}$ is considered time-invariant; it is represented by the parameter vector

$$\begin{aligned} \mathbf{S} = \quad & (\mathbf{S_{HEAD}}, \mathbf{S_{NECK}}, \mathbf{S_{TORSO}}, \\ & \mathbf{S_{UPPER\_ARM}}, \mathbf{S_{LOWER\_ARM}}, \\ & \mathbf{S_{UPPER\_LEG}}, \mathbf{S_{LOWER\_LEG}}) \end{aligned} \tag{3.7}$$

with $\mathbf{S_k} = (a^k, a_1^k, a_2^k, a_3^k, \epsilon_1^k, \epsilon_2^k, t_1^k, t_2^k)$ the super-quadrics parameters for body part k.

Ideally, both pose $\mathbf{P}(t)$ and shape $\mathbf{S}$ are recovered simultaneously from the images. In this thesis, a model-acquisition stage is required initially to obtain shape S. Thereafter, pose tracking only involves determining $\mathbf{P}(t)$.

The shape parameters $\mathbf{S_k} = (a^k, a_1^k, a_2^k, a_3^k, \epsilon_1^k, \epsilon_2^k, t_1^k, t_2^k)$ are derived in the model-acquisition stage from the projections of occluding contours in two orthogonal views, parallel to the zx- and zy-planes. This involves the human subject facing the camera frontally and sideways. The assumption made here is that 2-D segmentation of each body part is given in the two orthogonal views (a way to obtain such a segmentation is proposed by Kakadiaris and Metaxas [50]). The shape estimation procedure is as follows. First, the two longitudinal 2-D axes of a projected body part are used to recover the 3-D major axis the body part by triangulation. The contours of the body parts are back projected for each view onto the plane through the 3-D major axis parallel to the image plane. This gives 3-D occluding contour data. A coarse-to-fine search procedure is used over a reasonable range of parameter space $\mathbf{S_k}$ to determine the best-fitting quadric. Fitting uses chamfer matching (see the next section) as a similarity measure between the fitted and back-projected contours. Figure 3.2 shows frontal and side views of the recovered torso and head for two persons: DARIU and ELLEN. Figure 3.3 shows their complete recovered models in a graphics rendering. These models are used in the tracking experiments of Section 3.7.

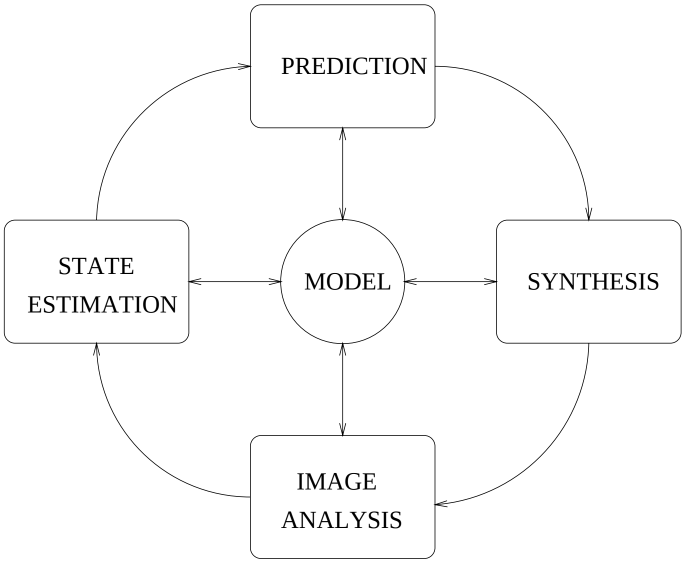


Figure 3.1: The tracking cycle

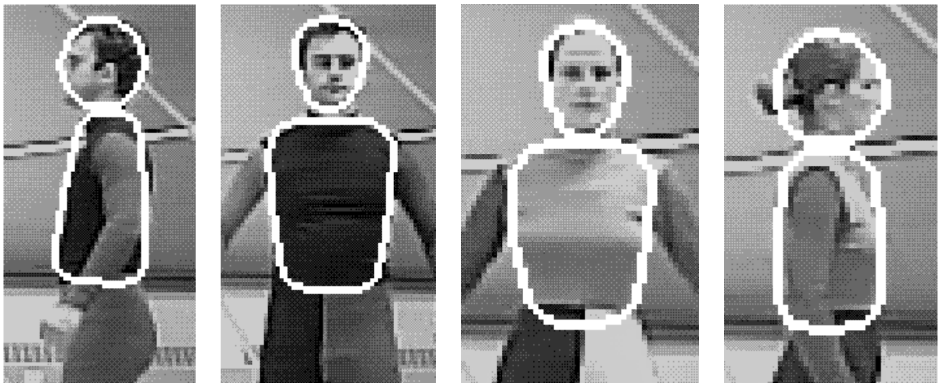


Figure 3.2: Frontal and side views of the recovered torso and head for the DARIU and ELLEN model

3.4 3-D pose estimation

The features of the generate-and-test approach to pose recovery and tracking are: similarity measure between synthesized model and image, search procedure, initialization and multi-view integration. These are discussed in turn. See Figure 3.4 for a schematic overview.

- Similarity measure

In this approach the similarity measure between model view and actual scene is based on arbitrary edge contours rather than on straight line approximations (as in [82], for example); we use a robust variant of *chamfer matching* [10]. The *directed* chamfer distance $DD(T, R)$ between a test point set T and a reference point set R is obtained by summing the distance contributions of the individual points t of set T, $dd(t, R)$; the latter is defined as the distance from t to the nearest point in set R

$$DD(T, R) \equiv \sum_{t \in T} dd(t, R) \equiv \sum_{t \in T} min_{r \in R} \parallel t - r \parallel \tag{3.8}$$

and its normalized version is

$$\overline{DD}(T, R) \equiv DD(T, R)/|T| \tag{3.9}$$

$DD(T, R)$ can be efficiently obtained in a two-pass process by pre-computing the chamfer distance to the reference set on a grid with the desired resolution. The distance map $D[i][j]$ with $i = 0, ..., N + 1$ and $j = 0, ..., M + 1$ contains initially two values: 0 at feature point locations (here, edge points) and “infinity” elsewhere. If $D[0][0]$ is considered the upper left corner and $D[N + 1][M + 1]$ the lower right corner of the grid, the forward pass shifts a 3×3 window row by row from left to right, from the upper left corner to the lower right corner of the grid, locally minimizing the distance at the window center with respect to the upper-diagonal entries in the current window.

```
for (i:=1; i<=N; i++)
  for (j:=1; j<=M; j++)
      D[i][j] := min(D[i][j], D[i][j-1]+2, D[i-1][j]+2,
                     D[i-1][j-1]+3, D[i-1][j+1]+3);
```

Conversely, the backward pass shifts a 3×3 window row by row from right to left, from lower right corner to the upper left corner of the grid, locally minimizing the distance at the window center with respect to the lower-diagonal entries in the current window.

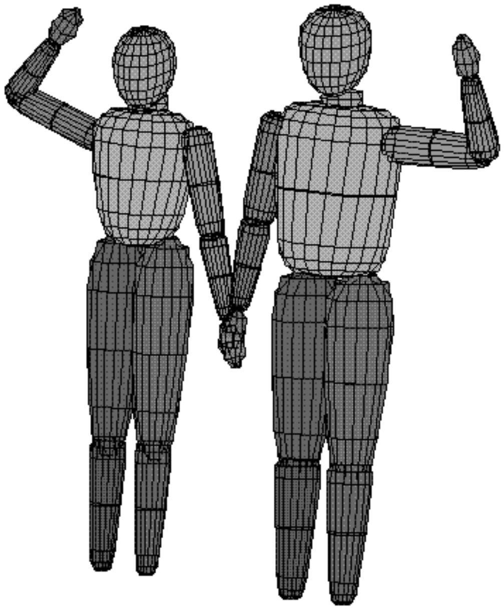


Figure 3.3: The recovered 3-D models ELLEN and DARIU say “**hi!**”

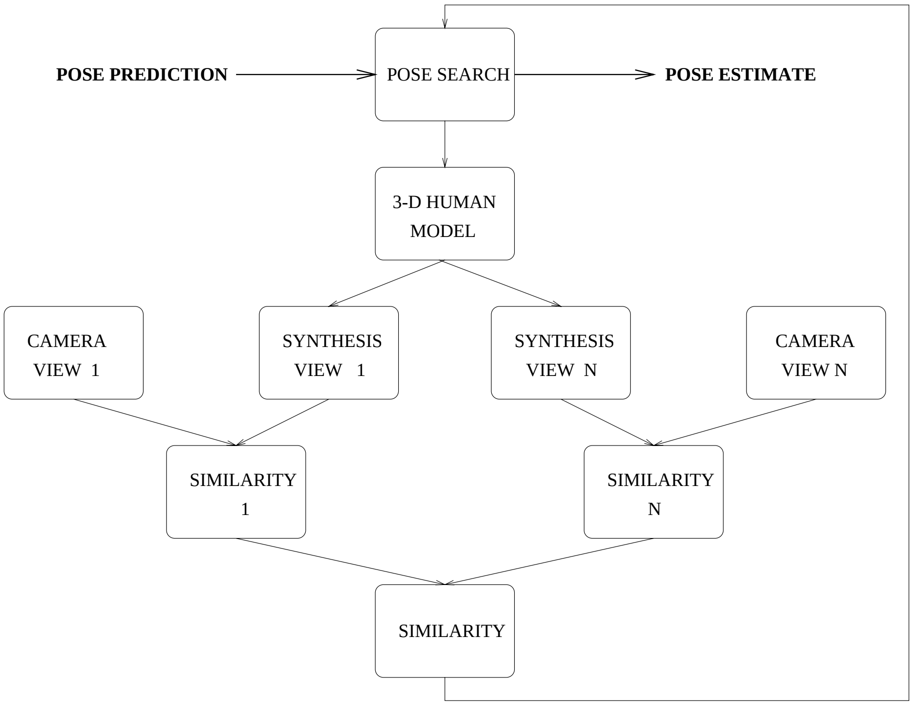


Figure 3.4: The pose-search cycle

```
for (i:=N; i>=1; i--)
  for (j:=M; j>=1; j--)
      D[i][j] := min(D[i][j], D[i+1][j]+2, D[i][j+1]+2,
                     D[i+1][j-1]+3, D[i+1][j+1]+3);
```

The resulting distance map is the so-called "chamfer image". See Figure 3.5 for an example edge image and the corresponding chamfer image. As seen by the above code fragment, the chamfer image can be computed very fast. Once it is obtained, computing the chamfer distance involves simply indexing onto this grid with the points of the reference set; the resulting grid values are subsequently added.

It would be even more efficient if we could use only $DD(M, S)$ during pose search (as done in [10]), where M and S are the projected model edges and scene edges, respectively. In that case, the scene chamfer image would have to be computed only once, followed by fast access for different model projections. However, using this measure alone has the disadvantage (which becomes apparent in experiments) that it does not contain information about how close the reference set is to the test set. For example, a single point can be really close to a large straight line, but we may not want to consider the two entities very similar. Therefore, the *undirected* normalized chamfer distance $\overline{D}(T, R)$ is used

$$\overline{D}(T, R) \equiv (\overline{DD}(T, R) + \overline{DD}(R, T))/2 \tag{3.10}$$

A further modification is to perform outlier rejection on the distribution $dd(t, R)$. Points t for which $dd(t, R)$ is above a user-supplied threshold θ are rejected outright; the mean μ and standard deviation σ of the resulting distribution is used to reject points t for which $dd(t, R) > \mu + 2\sigma$. With these changes the chamfer matching becomes similar to the modified Hausdorff distance as used in [44].

One notes that other measures could (and) have been used to evaluate a hypothesized model pose, which work directly on the scene image: correlation (see [35] and [81]) and average contrast value along the model edges (a measure commonly used in the snake literature). The reason that was opted for preprocessing the scene image (i.e. applying an edge detector) and chamfer matching is that it provides a gradual measure of similarity between two contours while having a long-range effect in image space. It is gradual since it is based on distance contributions of many points along both model and scene contours; as two identically contours are moved apart in image space the average closest distance between points increases gradually. This effect is noticeable over a range up to threshold θ, in the absence of noise. The two factors, graduality and long-range, make (chamfer) distance mapping a suitable evaluation measure to guide a search process. Correlation and average contrast along a contour, on the other hand, typically provide strong peak responses but rapidly declining off-peak responses.

- Search

Search techniques are used to prune the N dimensional pose parameter space (see also [70]). The search space $\mathbf{\Sigma}$ at time t is defined by a discretization of the pose parameter space by units of η_i along the various dimensions i. $\mathbf{\Sigma}$ is restricted to the neighborhood of the predicted pose $\hat{\mathbf{P}} = (\hat{p}_1, .., \hat{p}_N)$ as specified by positive and negative deviations Δ_i^+ and Δ_i^-, resp.

$$\begin{aligned} \Sigma &= \{\{p_1\} \times .. \times \{p_N\}\}, \\ \{p_i\} &= \{\hat{p}_i - \Delta_i^-, \ .., \ \hat{p}_i + \Delta_i^+\}, \ \ step\ \eta_i \end{aligned} \tag{3.11}$$

Best-first search [67] is used to search the state space $\mathbf{\Sigma}$. search. This local search procedure involves maintaining the set of states already evaluated and picking at each iteration the state with the best evaluation measure to expand next. The expansion of a state involves evaluating the neighboring states and adding them to the set of states considered; a "neighboring" state is defined here to be the states whose parameters differ by one unit increment η_i with those of the original state. Thus the expansion of a state involves a maximum of $2 * N$ new states. A state corresponding to a physically impossible pose is dismissed a-priori without the need for evaluation by synthesis.

A local search technique is used for pose-recovery because a reasonable initial state can be provided by a prediction component during tracking or by a bootstrapping method at start-up. The use of a well-behaved similarity measure derived from multiple views, as discussed before, is likely to lead to a search landscape with fairly wide and pronounced maxima around the correct parameter values; this can be well detected by a local search technique such as best-first. Nevertheless, the fact remains that the search-space is very large and high-dimensional (22 dimensions per human, in our case); this makes "straight-on" search daunting. The proposed solution to this is *search space decomposition*. The decomposed search space Σ^* is defined as

$$\Sigma^* = (\Sigma_1, \Sigma_2) \tag{3.12}$$

$$\Sigma_1 = \{\{p_{i_1}\} \times .. \times \{p_{i_M}\} \times \hat{p}_{i_{M+1}} \times .. \times \hat{p}_{i_N}\} \tag{3.13}$$

$$\Sigma_2 = \{\tilde{p}_{i_1} \times .. \times \tilde{p}_{i_M} \times \{p_{i_{M+1}}\} \times ... \times \{p_{i_N}\}\} \tag{3.14}$$

where $(\tilde{p}_{i_1}, .., \tilde{p}_{i_M})$ denotes the best solution to searching Σ_1. Thus initially a subset of parameters is searched while keeping the others at their predicted values. Subsequently, the remaining parameters are searched while keeping the first parameters at their best value. This search space decomposition can be applied recursively and can be represented by a tree in which non-leaf nodes represent search spaces to be further decomposed and leaf nodes are search spaces to be actually processed. The proposed recursive pose recovery scheme of K humans is illustrated in Figure 3.6. In order to search for the pose of the i-th human in the scene humans 1, ..., $i-1$ are synthesized with the best pose parameters found earlier, and humans $i+1$, ..., K are synthesized with their predicted pose parameters. The best torso/head configuration of the i-th human is searched for while keeping the limbs at their predicted values, etc.

In practice, it has been observed that it is more stable to include the torso-twist parameter in the arms (or legs) search space, instead of in the torso/head search space. This is because the observed contours of the torso alone are not very sensitive to twist. Given that the root of the articulated figure is kept fixed at the torso center, the dimensionalities of the search spaces to be considered are 5, 9, and 8, respectively.

One could try to increase the parallelism of the search decomposition by searching different parameter subspaces independently, for example, searching for each human separately while keeping the other humans at their predicted pose. Evidently, there are potential coupling effects between the pose parameters of multiple humans due to occlusions. These coupling effects are stronger if one extends the parallelism to the various body limbs of a single human.

- Initialization

The bootstrapping procedure for tracking currently handles the case where moving objects (i.e. humans) do not overlap and are positioned against a stationary background. The procedure starts with background subtraction, followed by a thresholding operation to determine the region of interest; see Figure 3.7. This operation can be quite noisy, as shown in the figure. The aim is to determine from this binary image the major axis of the region of interest; in practice this is the axis of the prevalent torso-head configuration. Together with the major axis of another view, this allows the determination of the major 3-D axis of the torso. Additional constraints regarding the position of the head along the axis (currently, implemented as a simple histogram technique) allow a fairly precise estimation of all torso parameters, with the exception of the torso twist and the limbs parameters, still to be searched for.

The determination of the major axis can be achieved robustly by iteratively applying a principal component analysis (PCA) [46] on data points sampled from the region of interest. Let $(\mathbf{x}_i, i = 1, ..., N)$ describe the data points sampled from the foreground region, denote their mean by μ. The best-fitting axis (minimizing the sum of squared perpendicular distances to the axis) goes through μ and its direction is given by the eigenvector $\mathbf{v}_{\lambda_{max}}$ associated with the largest eigenvalue of the data covariance matrix $\mathbf{C}$

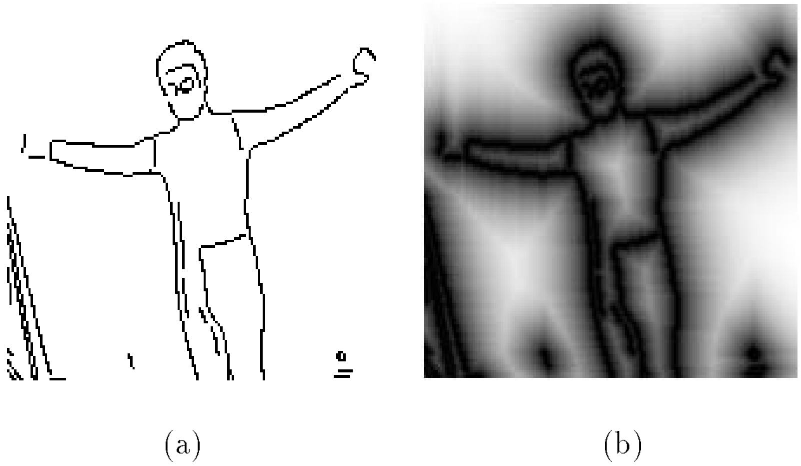


Figure 3.5: (a) Edge and (b) corresponding chamfer image

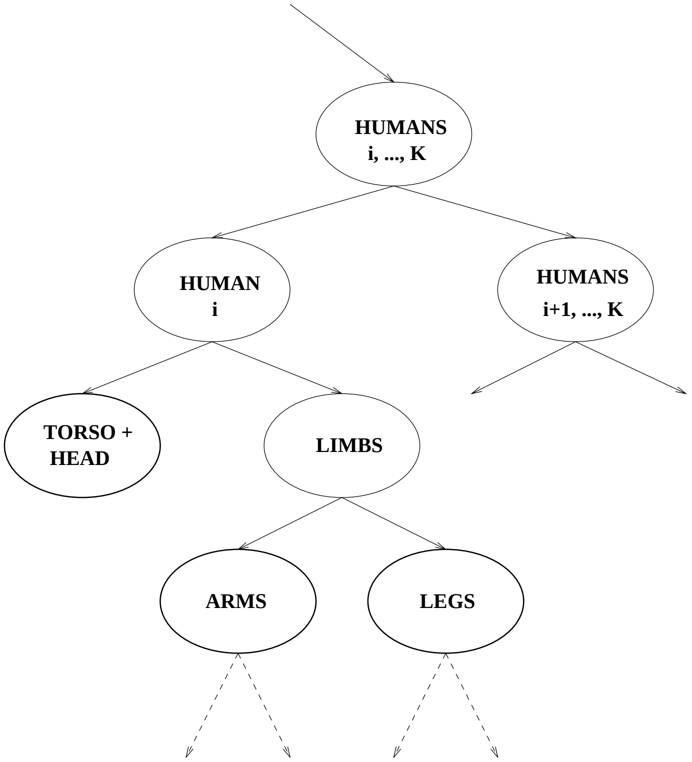


Figure 3.6: A decomposition of the pose-search space

$$\mathbf{C} = \sum_{i=1}^{N} (\mathbf{x}_i - \mu) (\mathbf{x}_i - \mu)^T \tag{3.15}$$

At each iteration, the distribution of the distances from data points to the best-fitting axis is computed. Data points whose distances to the current major axis are more than the mean plus twice the standard deviation are considered outliers and removed from the data set. This process results in the removal of the data points corresponding to the hands if they are located lateral to the torso, and also of other types of noise. The iterations are halted if the parameters of the major axis vary by less than a user defined fraction from one iteration to another. In Figure 3.7 the successive approximations to the major axis are shown by straight lines in increasingly light colors.

- Multi-view approach

By using a multi-view approach we achieve tighter 3-D pose recovery and tracking of the human body than from using one view only; body poses and movements that are ambiguous from one view can be disambiguated from another view. The appearance of the human model is synthesized for all the available views, and the appropriateness of a 3-D pose is evaluated based on the similarity measures for the individual views (see Figure 3.4).

3.5 The other components

The first implementation of the prediction component was in batch mode and consisted of a constant acceleration model for the pose parameters. In other words, a second degree polynomial was fitted at times $t, ..., t - T + 1$, and its extrapolated value at time $t + 1$ was used for prediction. It was found that this prediction scheme did not necessarily perform better than the trivial strategy of taking the current state estimate for prediction. This is because the sampling rate of measurements at frame rate is too low to allow simple kinematic models to be valid for prolonged periods when dealing with fast movement such as walking. In some cases, the constant acceleration model performes worse than the trivial constant parameter model, for example when trying to predict parameters related to the motion of an swinging arm; the predicted parameters would have the arm shoot through. The lack of strong kinematic models also made the application of the Kalman filter for state estimation less appealing. The situation might be different if more sophisticated kinematic models were to be formulated, which would model movements such as walking.

The synthesis component uses a standard graphics renderer to give the model projections for the various camera views.

The image analysis component applies an edge detector to the images, performs linking, and groups the edges into constant curvature segments. Some conservative steps are taken to filter out irrelevant edge segments (i.e. which are not part of the occluding contours of body parts). As an initial step, background subtraction is applied to obtain a foreground region which is then used as a mask for the edge segments. Further filtering involves the contours of the synthesized model; edge segments are accepted if their directed chamfer distance to the model edges (all combined) is less than a user-specified threshold. This process facilitates the partial removal of unwanted contours which could disturb the scene chamfer image.

The next step assigns scene edge pixels to different body units to allow matching per body unit during the decomposed search. The body units are defined as the torso and head (combined) and the four limbs. The edge assignment is done in a greedy manner; edge pixels are labeled based on what body unit of the predicted model is closest; this is determined by accessing the corresponding model chamfer images. In case multiple viable assignments (the distance ratio between the closest and second closest unit is higher than a user supplied threshold (≤ 1), the edge pixels are assigned to both body units. The end result is that both model and scene edges are labeled and the corresponding chamfer images have been computed per body unit. This allows computation of the undirected chamfer distances, as was discussed in previous section. Observe that the remaining noise edge pixels, if relative few compared to the data pixels, are filtered out by the proposed outlier-rejection technique.

3.6 Movement recognition

Movement recognition is considered here in the restricted sense of time-varying pattern matching. A variant of Dynamic Time Warping (DTW) [66] is discussed that can deal with unsegmented pattern sequences. Compared to Hidden Markov Models (HMM) and Neural Networks (NN), DTW is conceptually simple and quite effective, allowing flexibility in time-alignment between test and reference pattern to allow correct classification.

For patterns containing time-varying data, Dynamic time warping (DTW) (see [66] for an overview) is a well-known technique to match a test pattern with a reference pattern if the time scales are not perfectly aligned. Denote the test pattern as $\mathbf{T}(n), n = 1, 2, ..N$, where $\mathbf{T}(n)$ is an application-dependent k-dimensional feature vector. Let $\mathbf{R}(m), m = 1, 2, ..M$ be a reference pattern. Dynamic time-warping usually assumes that the endpoints of the two patterns have been accurately located (i.e. segmented) and formulates the problem as finding the optimal path from point $(1, 1)$ to (n, m) on a finite grid. The optimal path can be found efficiently by dynamic programming. Point (i, j) on a path represents matching the i-th feature vector of the reference pattern with the j-th feature vector of the test pattern.

Since human movements are continuous, one cannot assume that test patterns have been accurately segmented. Therefore, the DTW method is used at each time instant t of a test sequence, choosing a sufficiently large fixed time-interval N to search for the reference pattern "backwards" in time. The test pattern thus consists of the features derived from time $t - N + 1$ to t. This is similar to the Continuous Dynamic Programming (CDP) method proposed recently in [93] with some difference in the cumulative distance computations.

Without any loss of generality, assume the feature values have been normalized i n the $[0, 1]$ range. Define the distance $d(i, j)$ between $\mathbf{R}(i)$ and $\mathbf{T}(j)$ according to the (unweighted) L_1 norm:

$$d(i,j) \;=\; ||\mathbf{R}(i) - \mathbf{T}(j)||_{L_1} = \sum_k |R_k(i) - T_k(j)| \tag{3.16}$$

The DTW method involves two stages: a forward stage in which cumulative distances are computed between test and reference pattern, and a backward stage in which the optimal path is traced back. The cumulative distance $S(i, j)$ between reference pattern $\mathbf{R}(i)$ and test pattern $\mathbf{T}(j)$ is given as follows:

Boundary expressions

$$\begin{aligned} S(i,0) &= \infty \quad (1 <= i <= M) && (3.17)\\ S(i,1) &= \infty \quad (2 <= i <= M) && (3.18)\\ S(1,j) &= j\ d(1,j)\ (1 <= j <= N) && (3.19) \end{aligned}$$

for $2 <= j <= N$:

$$S(2,j) = min \begin{cases} S(1,j-2) + \frac{3}{2}(d(2,j-1) + d(2,j)) \\ S(1,j-1) + 2d(2,j) \\ S(1,j) + d(2,j) \end{cases} \tag{3.20}$$

Non-boundary expressions for $3 <= i <= M, 2 <= j <= N$

$$S(i,j) = min \begin{cases} S(i-1,j-2) + \frac{3}{2}(d(i,j-1) + d(i,j)) \\ S(i-1,j-1) + 2d(i,j) \\ S(i-2,j-1) + \frac{3}{2}(d(i-1,j) + d(i,j)) \end{cases} \tag{3.21}$$

After the above expressions have been evaluated, $S(N, M)$ denotes the sum of he distances between matching feature vectors along the optimal path. By keeping track of the predecessor on the optimal path at each grid point during the forward process, one can trace back on the optimal path, starting from (N, M). Note that the slope of the optimal path is constrained between 1/2 and 2, thus

N needs to be no larger than 2 M. Let $(1, s)$ be the starting point on the optimal path to (N, M). The output of the DTW matching algorithm is

$$D(t) = \frac{S(N, M)}{k(t - s + M)} \tag{3.22}$$

which is the average distance between corresponding feature values of test and reference pattern on the optimal path, with the averaging over the length of the optimal path $(t - s + N)$ and over the dimensions k.

In practice one may have a set of P (segmented) training patterns of the same class. A standard clustering approach can be used to divide this set in $K < P$ groups, based on distance measure D, from which K reference patterns are chosen. Weighting of the k dimensions in the distance function $d(i, j)$ can be based on the variance of feature values of the different patterns of a group, when warped to the reference pattern (or the "longest" one as in [24]).

The complexity of the DTW matching method is $O(N \times M)$. Speed-up can be achieved by applying the matching method every other t^* time step, or only if the N feature vectors describing the test pattern are sufficiently "close" to the M feature-vectors describing the reference pattern in k-dimensional space, discarding the time-component of the data for the moment (see also [18]).

3.7 Experiments

A large data base was compiled containing multi-view images of human subjects involved in a variety of activities. These activities are of various degrees of complexity, ranging from single-person hand waving to the challenging two-person close interaction of the Argentine Tango.

Experimental set-up

The data was taken from four (near-) orthogonal views (FRONT, RIGHT, BACK and LEFT) with the cameras placed wide apart in the corners of a room for maximum coverage; see Figure 3.9. The background is fairly complex; many regions contain bar-like structures and some regions are highly textured (observe the two VCR racks in lower-right image of Figure 3.9). The subjects wear tight-fitting clothes. Their sleeves are of contrasting colors, simplifying edge detection somewhat in cases where one body part occludes another.

Because of disk space and speed limitations, the more than one hour's worth of image data was first stored on (SVHS) video tape. A subset of this data was digitized (properly aligned by its time code (TC)) and makes up the HIA database, which currently contains more than 2500 frames in each of the four views.

Calibration

The cameras were calibrated in a two-step process. First, the intrinsic camera parameters were recovered. This was done for each camera separately, placing a movable planar grid pattern close to each camera in order to calibrate for an as large field of view (FOV) as posible, see Figure 3.8a. The relative large effective focal lengths of the lenses used (they turned out to correspond to FOVs of 29.6, 29.2, 29.2 and 32.8 degrees, respectively) allowed a pin-hole approximation for the viewing geometry throughout the entire experiments with no need to account for lens distortion. The only intrinsic parameters to be recovered were thus the effective focal lengths.

The second step involved recovering the extrinsic camera parameters, recovering the orientations and positions of the cameras with respect to each other. This was done for pairs of cameras (the calibration pattern was not visible from all cameras simultaneously), see Figure 3.8b. Both intrinsic and extrinsic calibrations were performed using an iterative, non-linear least squares method developed by Szeliski and Kang [92].

Calibration does not have to be perfect. Sufficient is an accuracy for which the image localization errors resulting from calibration are small compared to those resulting from 3-D body modeling or image processing. To obtain an indication of the quality of the four-camera calibration in terms of 3-D space, the positioning of camera BACK was computed with respect to (the opposite) camera FRONT, along two paths, via camera LEFT and camera RIGHT. The corresponding transformation matrices are

$$\mathbf{H}_{B-L-F} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} -0.9812 & -0.0517 & 0.1860 & -700.8 \\ -0.0574 & 0.9979 & -0.0257 & 415.6 \\ -0.1842 & -0.0360 & -0.9822 & 8170.4 \\ 0.0000 & 0.0000 & 0.0000 & 1.0 \end{pmatrix} \tag{3.23}$$

$$\mathbf{H}_{B-R-F} = \begin{pmatrix} \mathbf{R}^{'} & \mathbf{T}^{'} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} -0.9887 & -0.0528 & 0.1398 & -462.8 \\ -0.0576 & 0.9980 & -0.0309 & 442.3 \\ -0.1379 & -0.0386 & -0.9896 & 8153.9 \\ 0.0000 & 0.0000 & 0.0000 & 1.0 \end{pmatrix} \tag{3.24}$$

Define the normalized position deviation $\overline{\mathbf{\Delta}}_T$ between the two transformations as

$$\overline{\mathbf{\Delta}}_T = \frac{||\mathbf{T} - \mathbf{T}^{'}||_2}{||\mathbf{T}||_2} \tag{3.25}$$

and define the x-axis deviation $\mathbf{\Delta}_{R_x}$ to be the angle between $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{R}^{'}_{\mathbf{x}}$, i.e.

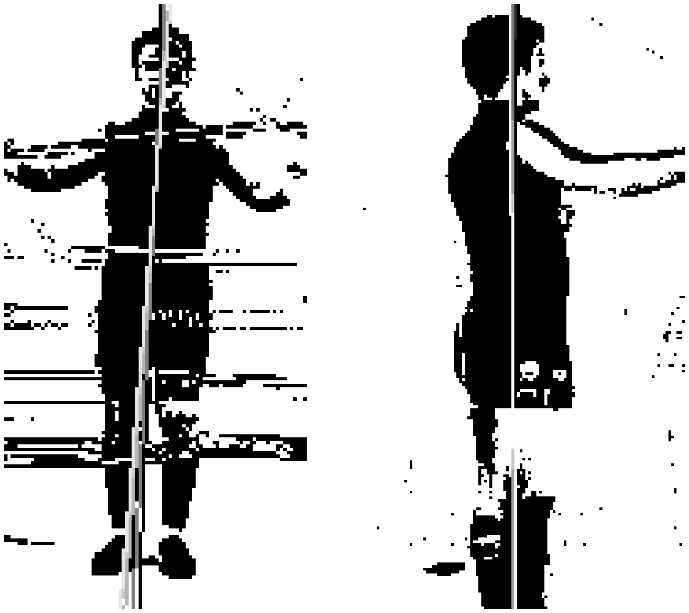
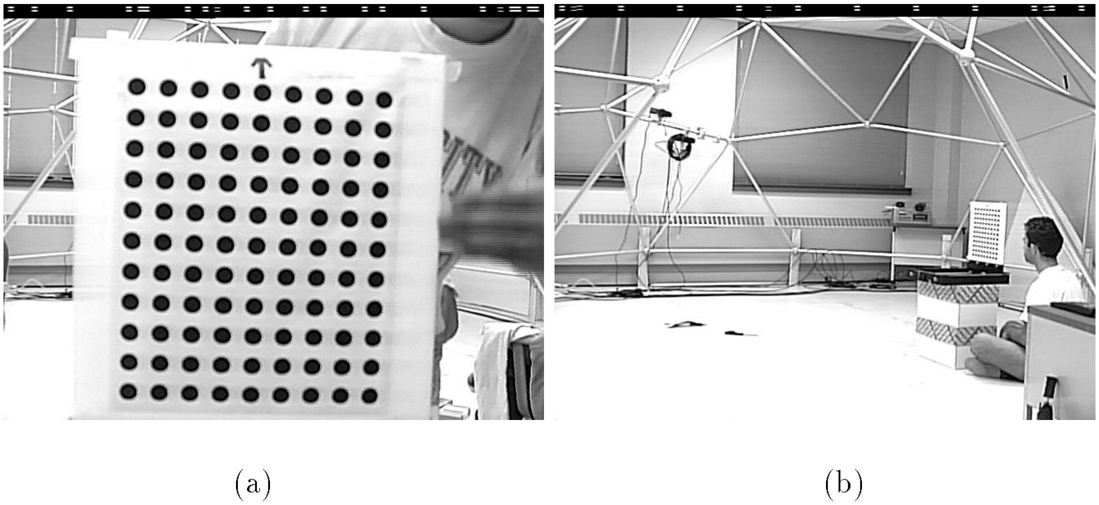


Figure 3.7: Robust major axis estimation using an iterative principal component fit (cameras FRONT and RIGHT). Successive approximations to the major axis are shown in lighter colors.



(a)

(b)

Figure 3.8: Calibration for (a) intrinsic parameters and (b) extrinsic parameters

the first column of the rotation matrices $\mathbf{R}$ and $\mathbf{R}^{`}$. Then,

$$\mathbf{\Delta}_{R_x} = cos^{-1}(\frac{\mathbf{R_x} \cdot \mathbf{R}^{`}_{\mathbf{x}}}{||\mathbf{R_x}||_2 \; ||\mathbf{R}^{`}_{\mathbf{x}}||_2}) \tag{3.26}$$

Define similarly $\mathbf{\Delta}_{R_y}$ and $\mathbf{\Delta}_{R_z}$. Then

$$\overline{\mathbf{\Delta}}_T = 0.029, \;\; \mathbf{\Delta}_{R_x} = 2.7 \text{ deg}, \;\; \mathbf{\Delta}_{R_y} = 1.0 \text{ deg } and \, \mathbf{\Delta}_{R_z} = 2.6 \text{ deg} \tag{3.27}$$

are indeed close to the desired value of zero.

Another measure of calibration quality considers errors in the image plane and uses epipolar lines. Given two cameras, C and C_R, an image point p_{C_R} of camera C_R can be the projection of any 3-D point P lying on the half-line L from focal point O_{C_R} through the image plane at location p_{C_R}. The projection of this half-line onto the image plane of camera C denotes the possible image locations of the point corresponding to p_{C_R}; this latter half-line is called an epipolar line. Correct calibration will result in the intersection of the epipolar line with the projection of P onto the image plane of C, p_C.

The epipolar lines drawn in Figure 3.9 in the RIGHT, BACK and LEFT camera views correspond to selected points in the FRONT view. One can see that corresponding points lie very close to or on top of the epipolar lines. Observe how all the epipolar lines emanate from one single point in the BACK view: the FRONT camera center lies within its view.

Implementation

The current system is implemented under A.V.S. (Advanced Visualization System). Following its data flow network model, it consists of independently running modules, receiving and passing data through their interconnections.

The A.V.S. network implementation of the current system is shown in Figure 3.10; it bears a close resemblance to the pose search cycle shown earlier in Figure 3.4. The parameter space was bounded in each angular dimension by ± 15 degrees, and in each spatial dimension by ± 10 cm around the predicted parameter values. The discretization was 5 degrees and 5 cm, respectively. These values were kept constant during tracking.

The individual limb joint angles were also constrained to lie in the ranges specified by Table 3.7. Additional constraints could be placed on combinations of joint angles, based on physical considerations such as collision or limitations of the twist angle in certain poses.

The number of pose search iterations was typically 1100 for each time instant: 200 iterations for the torso, 500 iterations for the torso-twist and arms, and the remaining 400 iterations for the legs.

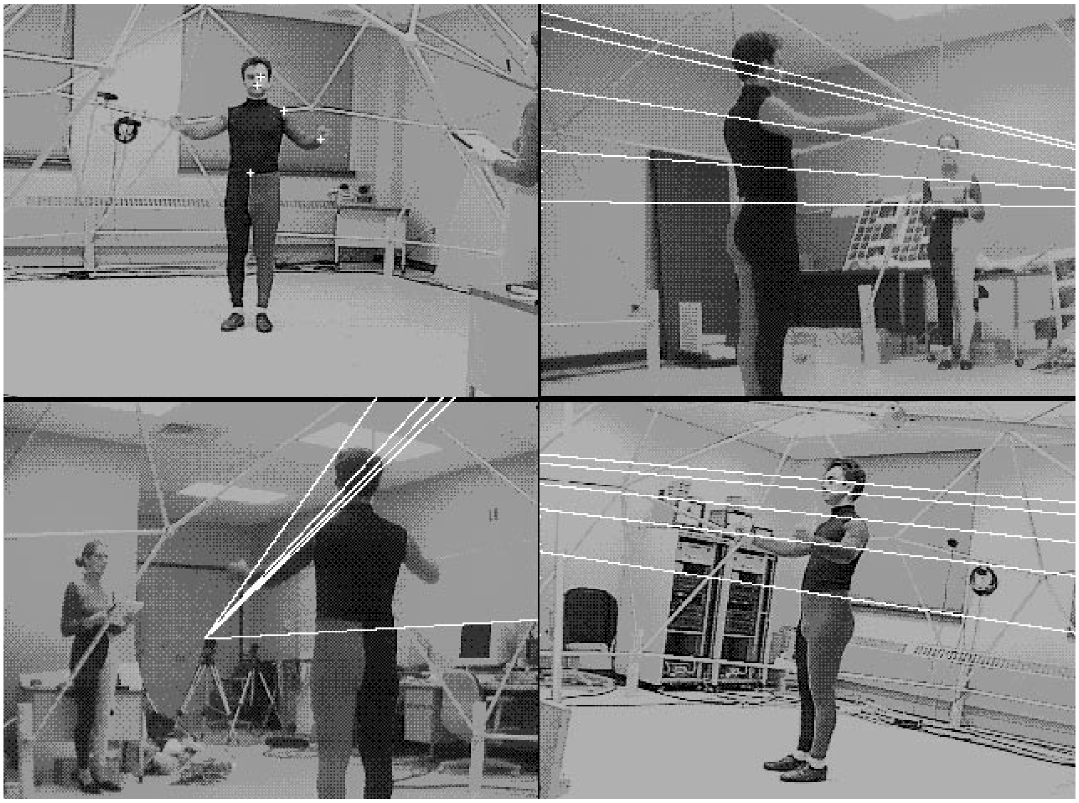


Figure 3.9: Epipolar geometry of cameras FRONT (upper-left), RIGHT (upper-right), BACK (lower-left) and LEFT (lower-right): epipolar lines are shown corresponding to the selected points from the view of camera FRONT

	min (degrees)	max (degrees)
arm elevation	0	180
abduction	-45	180
twist	90	-90
flexion	0	145
leg elevation	0	145
abduction	none	none
twist	none	none
flexion	0	145

Table 3.1: Joint angle ranges

Results

The first thing to be shown is that the followed calibration and human modeling procedures allow accurate 3-D localization of the human in the scene. This in turn allows the model prediction to be useful for image segmentation during tracking. To show this, the pose- parameters of the acquired model were manually optimized, and the resulting model projections (shown in white) were superimposed on the four orthogonal views, with the mapping given by the obtained calibration. Figure 3.11 shows the result for two instances; a quite close fit can be observed between the model projections and the human contours in the scene.

Next, results are shown regarding the image processing component of the tracking system. Figure 3.12 shows a typical multi-view image of the Humans-In-Action database, involving a single participant in motion. The result of applying edge detection and linking, using a method developed by Sarkar and Boyer [85] is shown in Figure 3.13. Figure 3.14 shows the effects of masking the edges with the foreground region, the latter which is obtained by background subtraction. This is the essentially the input to the pose recovery algorithm, note that the quality of edges is quite bad. Torso-axis estimation is shown in Figure 3.15. Figure 3.16 illustrates the edge assignment to various body units; this is done before starting the decomposed pose search.

Figures 3.17 and 3.18 illustrate tracking for persons DARIU and ELLEN, respectively. The movement performed can be described as raising the arms sideways to a 90 degree elevation with a 90 degree flexion, followed by rotating both elbows forward. Moderate opposite torso movement takes place for balancing as arms are moved forward and backwards. The current recovered 3-D pose is illustrated by the projection of the model in the four views, shown in white, as before. The displayed model projections include for visual purposes the edges at the intersections of body parts; these were not included in the chamfer matching process. It can be seen that tracking is quite successful, with a good fit for the recovered 3-D pose of the model for the four views. Figure 3.19 shows some of the recovered pose parameters for the DARIU sequence.

Figure 3.20 shows the result of movement recognition using *Dynamic Time Warping (DTW)*; for the time-interval in which the elbows rotate forward, we use the left hand pose parameters derived from the ELLEN sequence as a template (see Figure 3.20a) and match them with the corresponding parameters of the DARIU sequence. Matching with DTW allows (limited) time-scale variations between patterns. The result is given in Figure 3.20b, where the DTW dissimilarity measure drops to a minimum when the corresponding pose pattern is detected in the DARIU sequence.

Figure 3.21 illustrates an instance of whole-body tracking of person DARIU for a movement that can be described as walking and turning. The figures show that tracking is successful initially. The current system does lose track, eventually. Figure 3.21d shows a characteristic failure situation due to incorrect image segmentation. Here the predicted torso pose (not shown) was somewhat inaccurate, it partially overlaped the right arm. Due to the greedy edge labeling scheme employed, a significant amount of arm edges are labeled as the torso pixels, with the inevitable result that the similarity optimization leads to an incorrect torso placement.

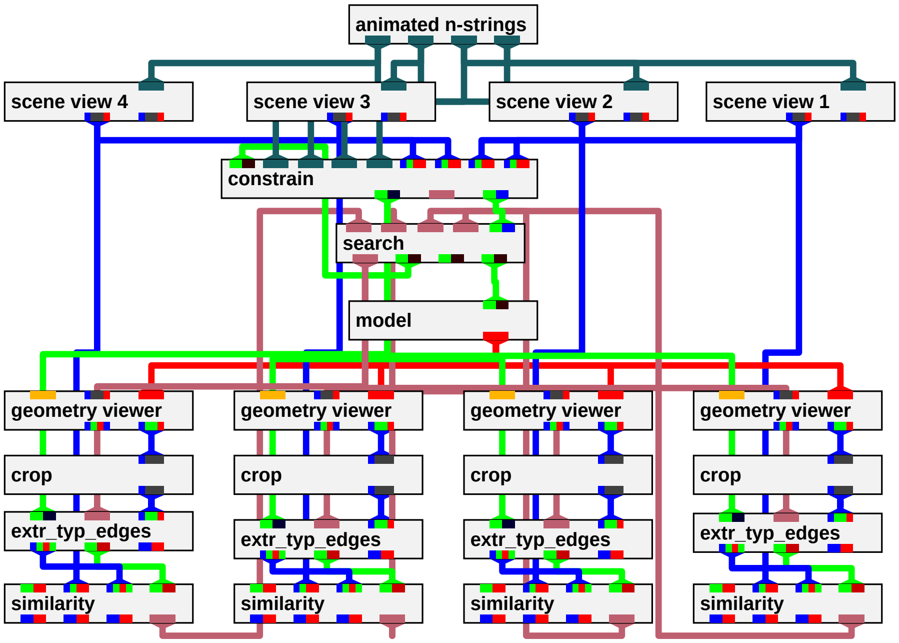
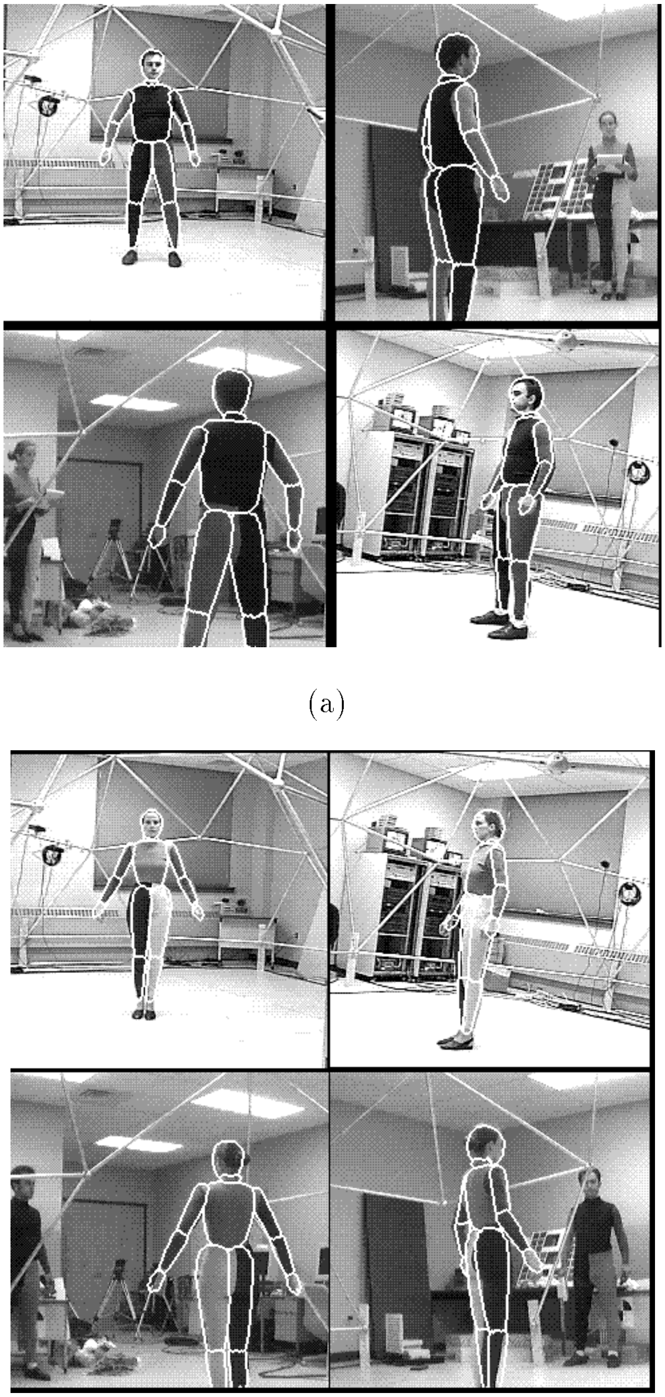


Figure 3.10: The A.V.S. network implementation of the system.



(a)

(b)

Figure 3.11: Manual 3-D model positioning: (a) DARIU (b) ELLEN

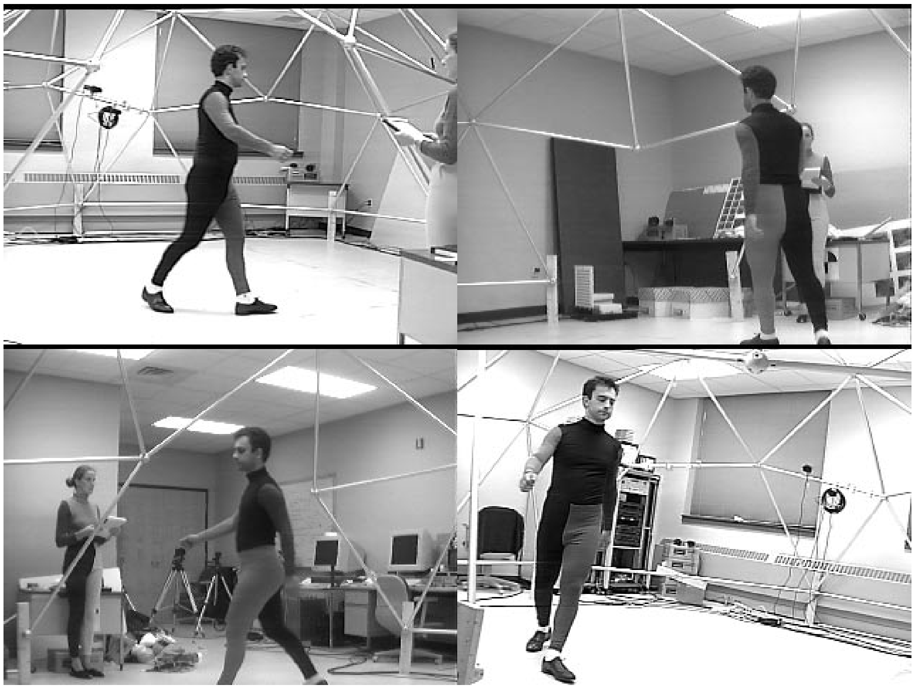


Figure 3.12: Original images

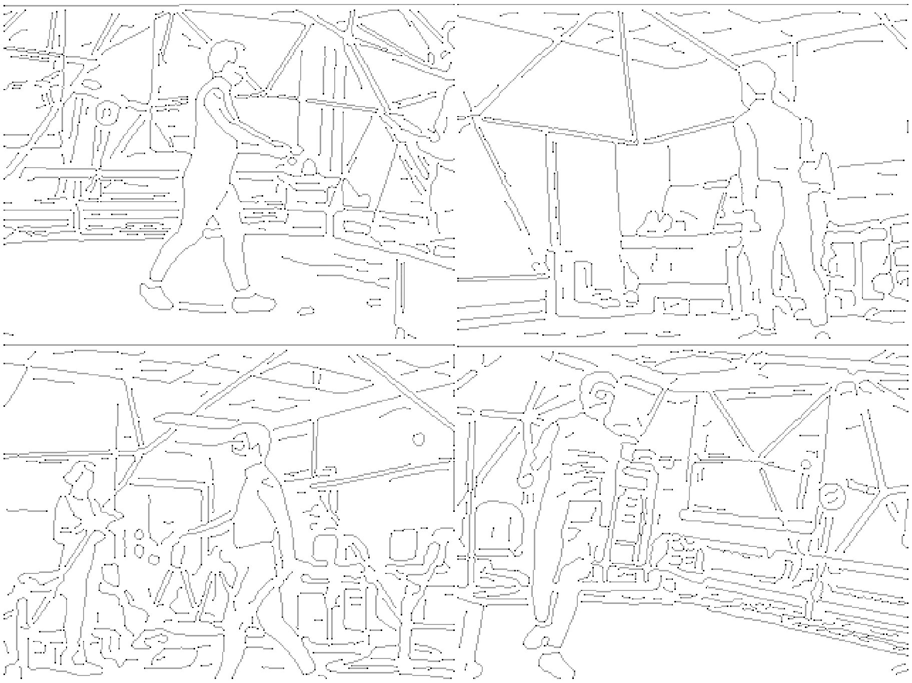


Figure 3.13: Edge images

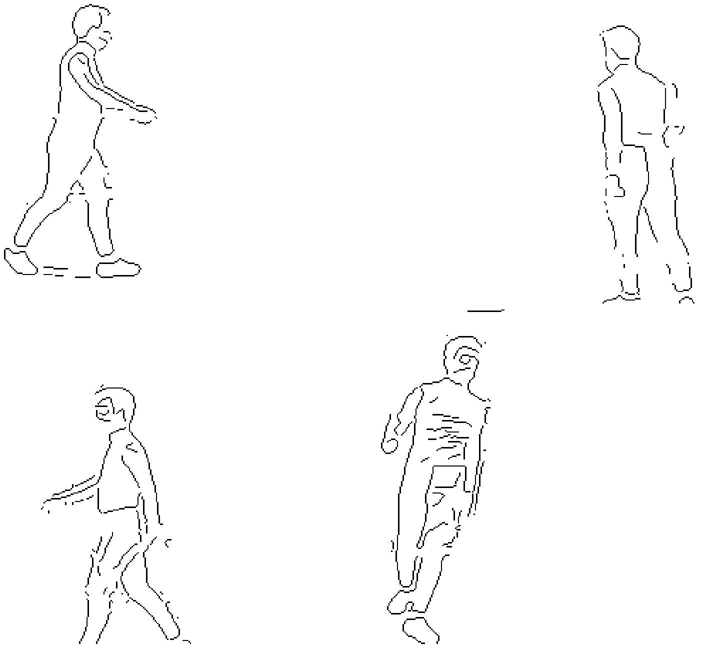


Figure 3.14: Foreground edge images

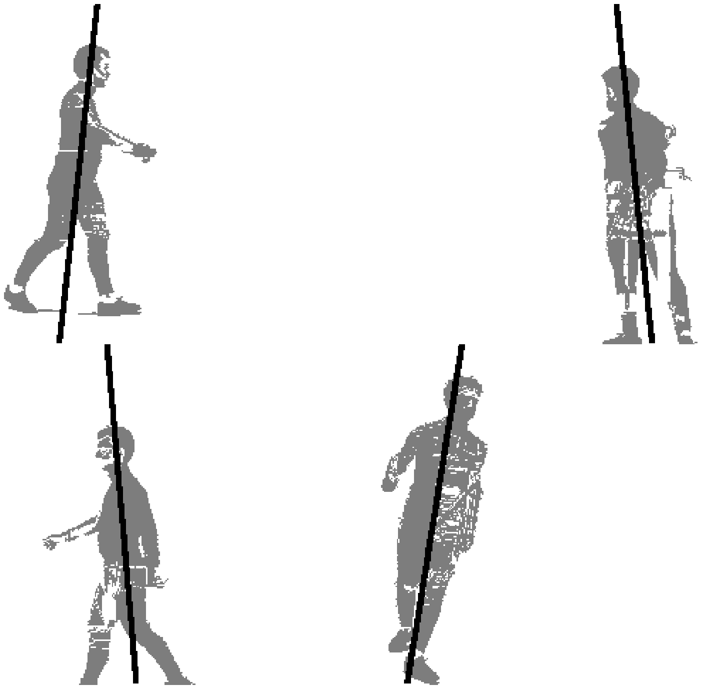
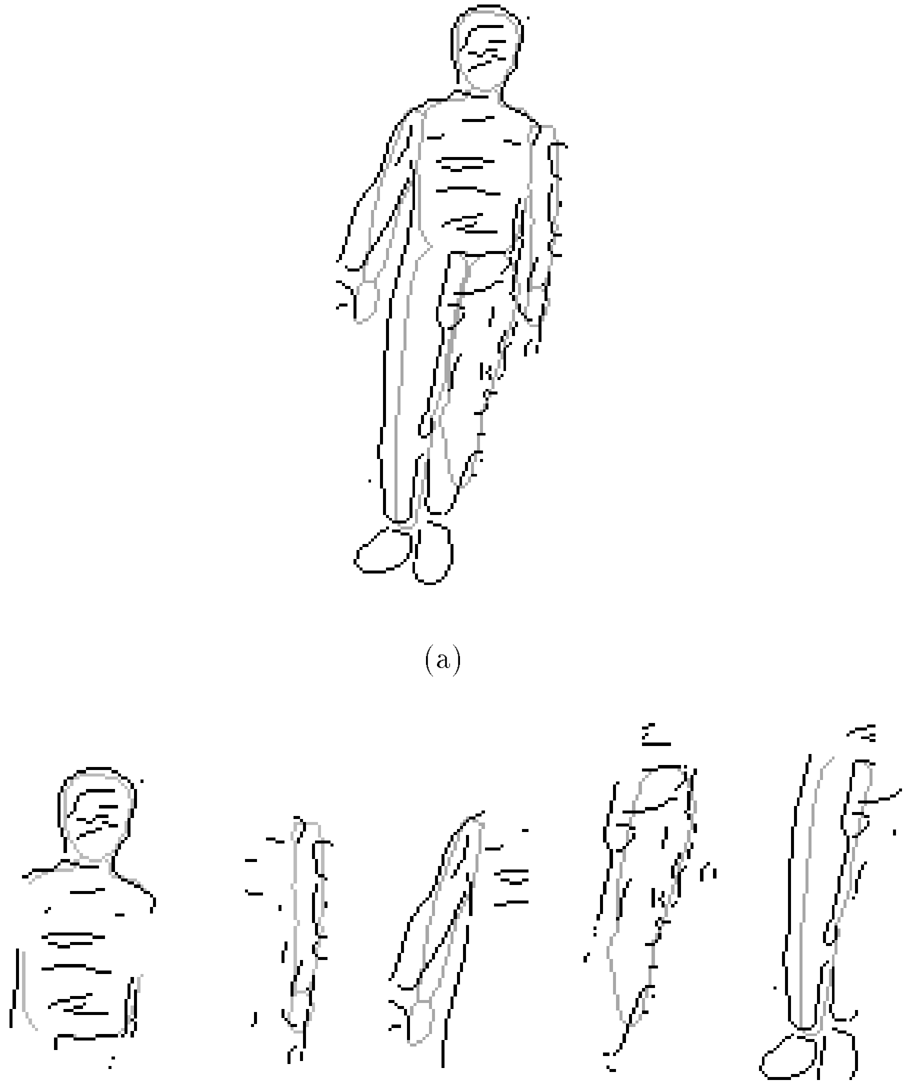


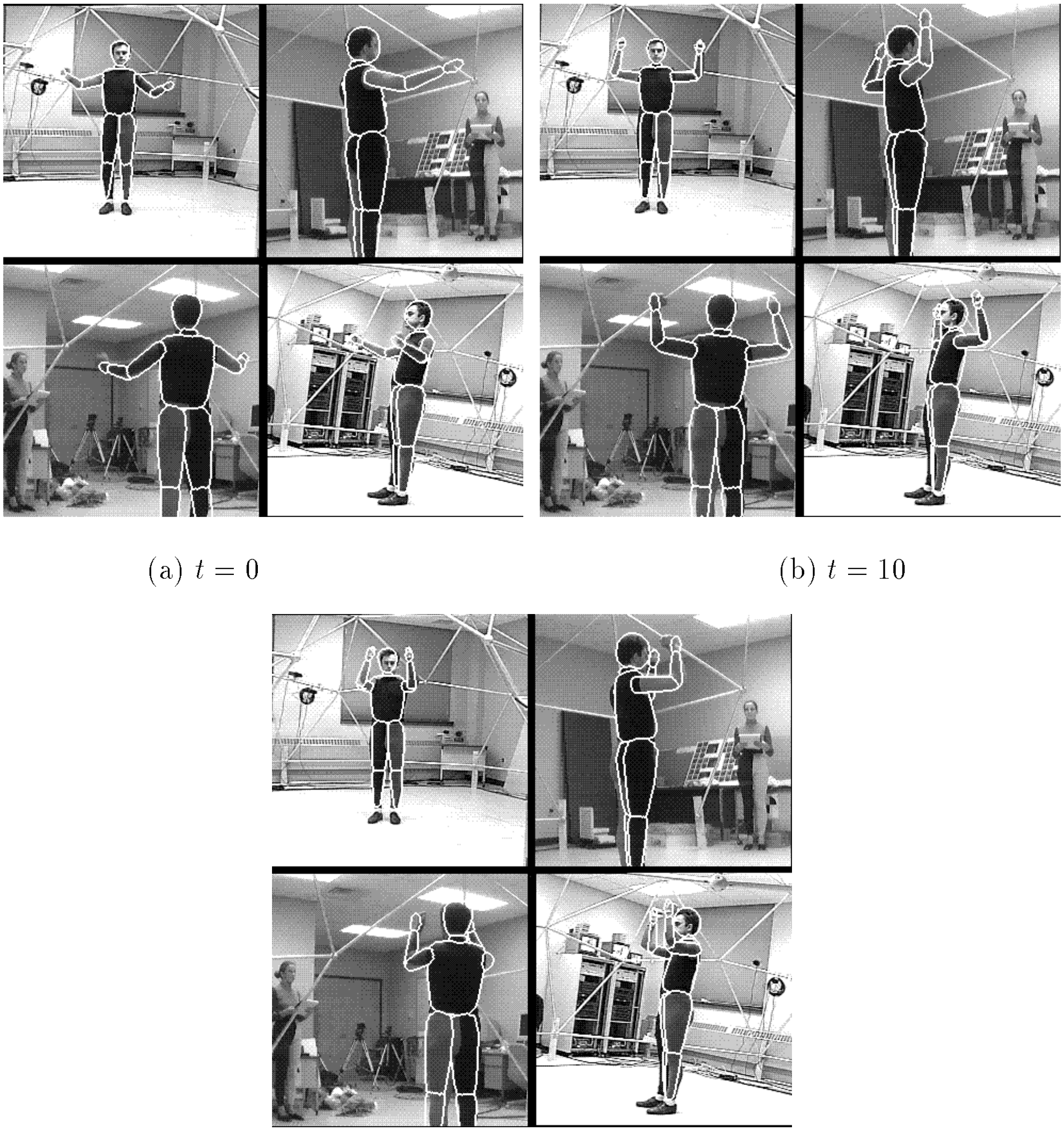
Figure 3.15: Robust axis fit images



(a)

(b) (c) (d) (e) (f)

Figure 3.16: Edge pixel assignment to various body units: (a) all edge pixels, (b) torso-head, (c) left arm, (d) right arm, (e) left leg, (f) right leg

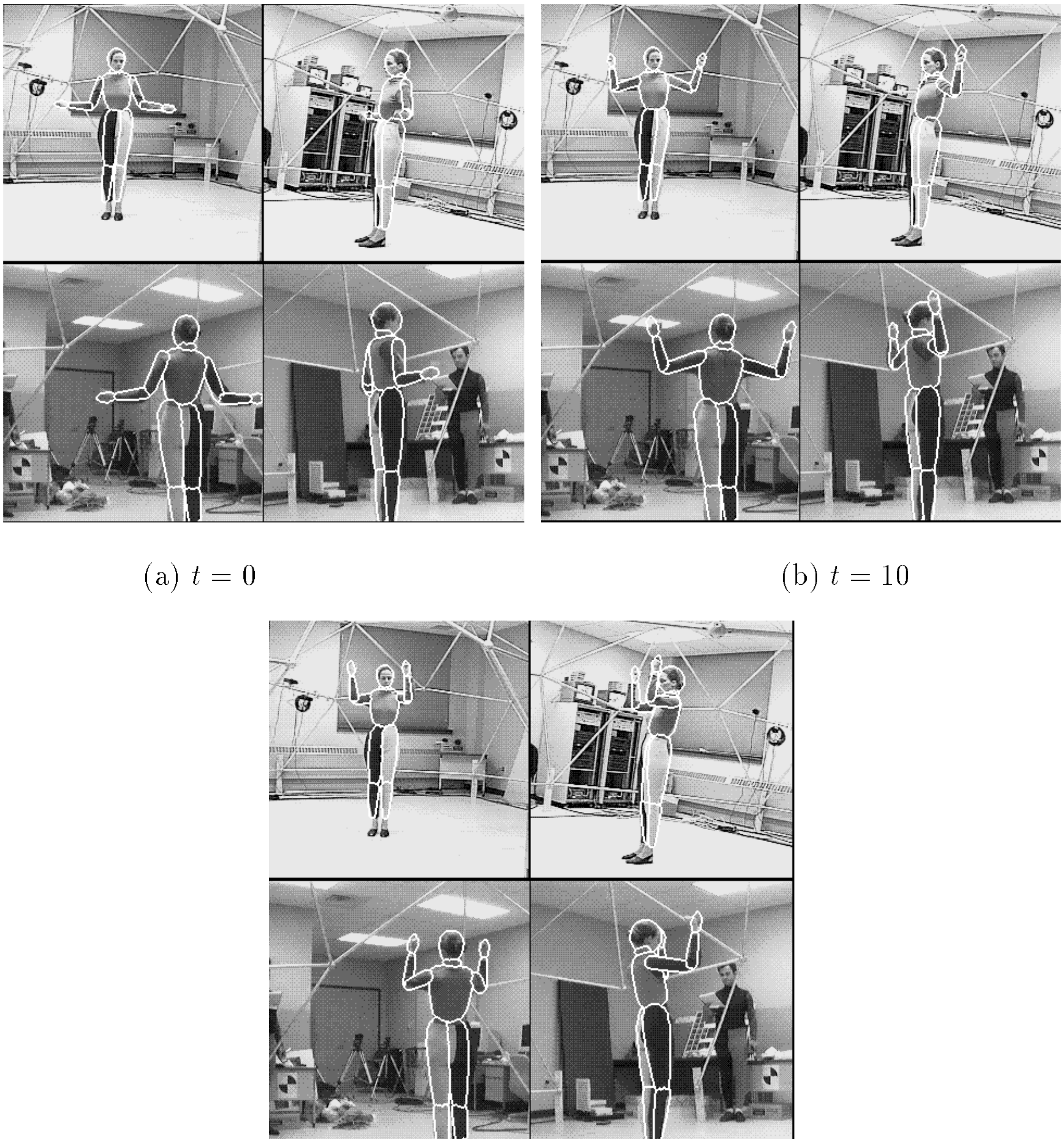


(a) $t = 0$

(b) $t = 10$

(c) $t = 25$

Figure 3.17: Tracking sequence D-TwoElbowRot ($t = 0, 10, 25$), cameras FRONT, RIGHT, BACK and LEFT.

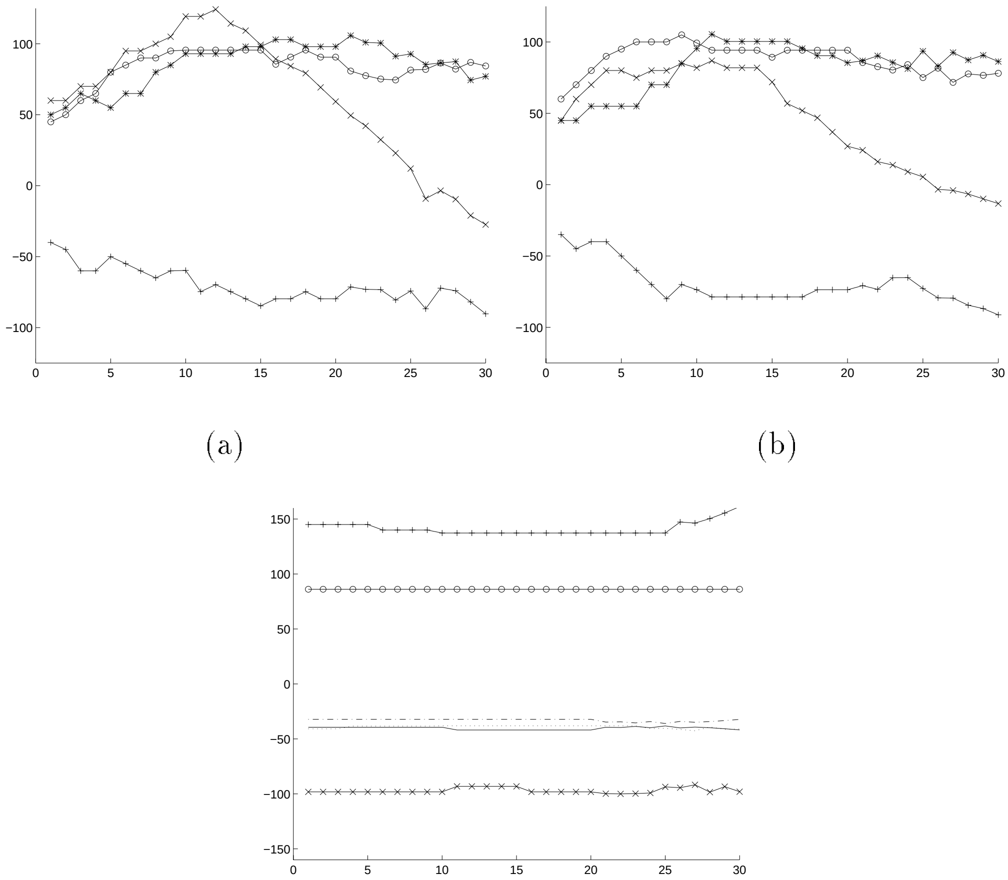


(a) $t = 0$

(b) $t = 10$

(c) $t = 25$

Figure 3.18: Tracking sequence E-TwoElbowRot ($t = 0, 10, 25$), cameras FRONT, RIGHT, BACK and LEFT.

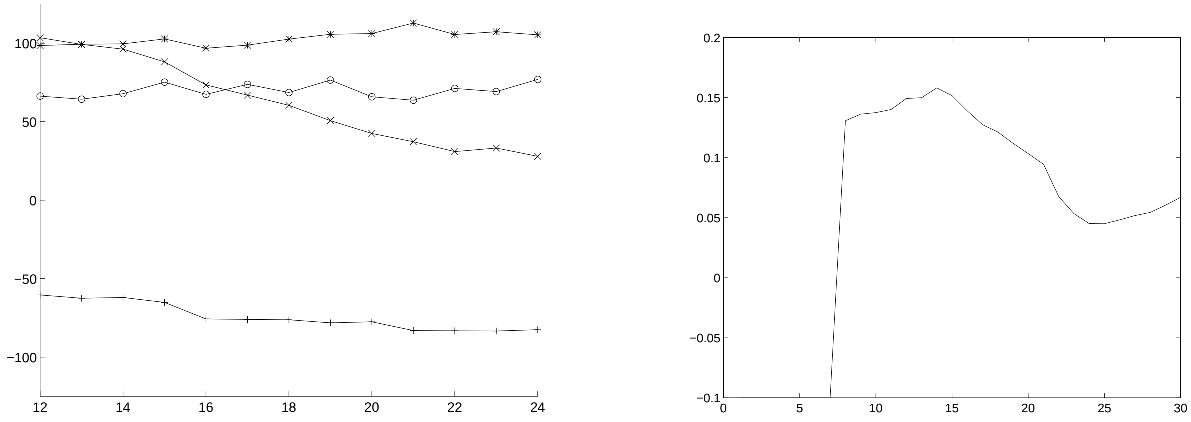


(a)

(b)

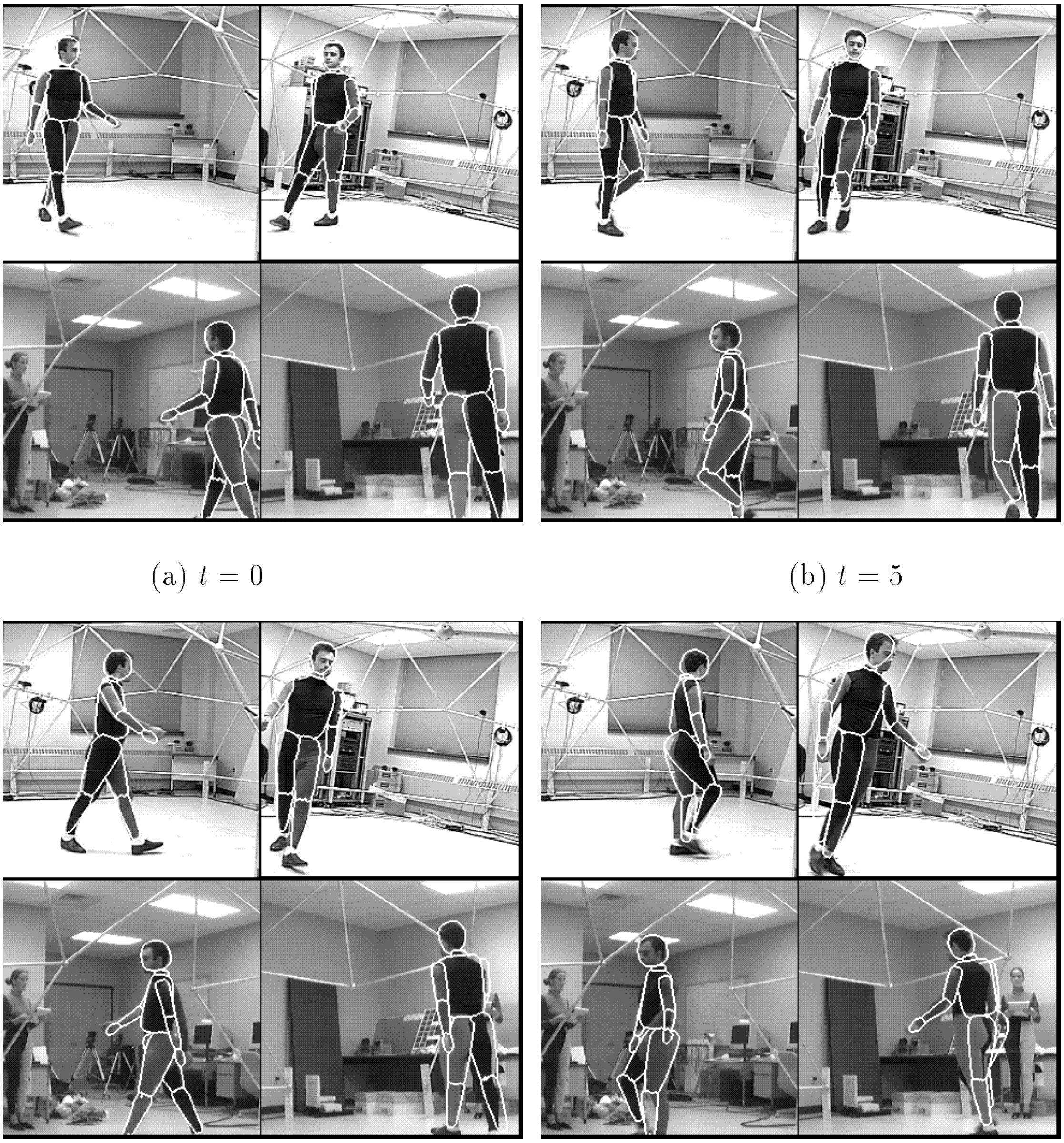
(c)

Figure 3.19: Recovered 3-D pose parameters vs. frame number, D-TwoElbRot; (a) and (b): LEFT and RIGHT ARM, abduction- (x), elevation- (o), twist- (+) and flexion-angle (*) (c): TORSO, abduction- (x), elevation- (o), twist-angle (+) and x- (dot), y- (dashdot) and z-coordinate (solid). Along the vertical axes, angles are in degrees, positions are in cm.



(a) (b)

Figure 3.20: (a) A template T for the left arm movement, extracted from E-TwoElbRot; (b) DTW dissimilarity measure of matching template T with the LEFT ARM pose parameters of D-TwoElbRot.



(a) $t = 0$

(b) $t = 5$

(c) $t = 10$

(d) $t = 15$

Figure 3.21: Tracking sequence D-TurnWalk ($t = 0, 5, 10, 15$), cameras FRONT, RIGHT, BACK and LEFT.

Chapter 4

Hermite deformable contours

Image segmentation by boundary finding is one of the central problems in computer vision. This is because amongst features that can be used to distinguish objects from their backgrounds, such as color and texture, shape is usually the most powerful. For detecting instances of objects with fixed and known shape, the Hough-transform or a template matching technique is well suited (see [84] [8]). For cases where there exists some flexibility in the object shape (either with respect to a previous frame in a tracking application, or with respect to a user supplied shape in an interactive object delineation setting) deformable contour models have found widespread use.

Deformable contours (also called active contour models, or "snakes") are energy-minimizing models for which the minima represent solutions to contour segmentation problems. They can overcome problems of traditional bottom-up segmentation methods, such as edge gaps and spurious edges, by the use of an energy function that contains shape information in addition to terms determined by image features. The additional shape information can be seen as a regularization term in the fitting process. Once placed in image space, the contour deforms to find the most salient contour in its neighborhood, under the influence of the generated potential field.

An extensive amount of work has been reported on deformable contours since their emergence in the late eighties; among others [4], [13]-[27], [32]-[78], [90]-[98]. A useful way to characterize the different approaches is along the following dimensions:

- contour representation
- energy formulation (internal and external)
- contour propagation mechanism (spatial and temporal)

The various contour representations that have been used previously are reviewed in Section 4.1. A new local representation is proposed for the deformable contour framework, based on Hermite interpolating cubics, see Section 4.2. Its use has several advantages, as will become apparent. The main plus is that it handles both smooth and polygonal curves naturally.

The solution to the contour finding problem is formulated by a *maximum a posteriori* (MAP) criterion. This leads to an internal energy formulation which contains squared terms of deviations from the expected Hermite parameter values. The external energy terms describe the typical image gradient correlations. See Section 4.3. The resulting energy minimization is performed by dynamic programming which gives the optimal solution to contour finding for a certain search region, see Section 4.4.

One of the well-known limitations of deformable contours is that their initial placement has to be close to the desired object boundary in order to converge. In tracking applications, this assumption might be violated. To keep the problem computationally tractable, the effects of transformation and deformation are decoupled, see Section 4.5.

Experiments on a variety of images are presented in Section 4.6.

4.1 Related work

Contour representations can be roughly divided into two classes, depending on whether they are *global* or *local*. Global representations are those where changes in one shape parameter affect the entire contour, and conversely, local change of the contour shape affects all parameters. Global representations are typically compact, describing shape in terms of only a few parameters. This is an advantage in a recognition context, i.e. when trying to recover these parameters from images, because of lower complexity. A useful class of shapes easily modeled by a few global parameters are the super-quadrics [95], which are generalizations of ellipses that include a degree of "squareness". To these shapes, one can add global deformations, such as tapering, twisting and bending [9]. A more general global representation is the Fourier representation [90]. It expresses a parametrized contour in terms of a number of orthonormal (sinusoidal) basis functions. Arbitrary contours can be represented in any detail desired, given a sufficient number of basis functions.

Local representations control shape locally by various parameters. This flexibility makes local representations well suited in a shape reconstruction context, as is the case when deforming a contour to fit image data. The simplest contour representation is an ordered list of data points. More compact representations describe contours in terms of piecewise polynomials. Each segment of the parametrized contour $(x_i(t), y_i(t))$ is described by a polynomial in t. The lowest-degree interpolating polynomial is of degree one, leading to a contour representation by polylines and polygons. More flexibility is possible by the use of higher order polynomials, generally cubic polynomials; they are the lowest degree polynomials for which derivatives at the endpoints can be specified. Higher order polynomials tend to bounce back and forth in less controlable fashion and therefore are used less frequently for interpolation purposes.

Natural cubic splines are piecewise third degree polynomials which interpolate control points with C^0, C^1 and C^2 continuity. The natural cubic spline parameters depend on all control points, which makes it a global representation. B-splines on the other hand, have a local representation, where contour segments depend only on a few neighboring control points. This comes at a price of not interpolating the control points. The same C^0, C^1 and C^2 continuity as natural splines is now achieved at the join points of connecting segments. By replicating control points, one can force the B-spline to interpolate the control points. A last interesting property is that the B-spline can be specified such that it performs a least-squares fit on the available data points.

In previous work, three local representations have been used for deformable contour finding: point representations, polygonal chains and uniform B-splines. These representations have the following disadvantages when used for the contour finding task.

Manipulating contours on the fine scale offered by pixel-by-pixel representations leads typically to high computational cost (for example, note the high complexity incurred in [32]). The incorporation of a-priori shape information in this featureless representation is difficult. If, on the other hand, a contour is represented by a few (feature) points, and contour finding only considers image data in the local neighborhood of these points, no use is made of data at intermediate locations which makes the approach prone to image noise.

The polygonal chain representation [27] overcomes some of these problems. However, it is not well suited to represent curved objects well, requiring many control points to be adequate. In an interactive object delineation setting, this is tedious. For tracking applications, the placement of control points close to each other, typical also of point representations, leads to stability problems. This is because for most contour finding approaches using local representations, a-priori shape information is encoded for each control point with respect to its neighboring control points (i.e. curvature [52][78] [98], affine coordinates [57]). If control points are close together, small deviations due to image noise or contour propagation will result in large changes of local shape properties.

B-splines present an efficient and natural way to represent smoothly curved objects. For objects with sharp corners they are less suited; the C^2 continuity smooths out any regions of high curvature of a contour. The fact that B-splines do not interpolate the control points can be considered a drawback in an interactive object delineation setting (think of a physician pointing to specific locations in medical images). The before mentioned use of control point duplication can take care of this, but then straight line segments appear around the newly C^0 continuous control point. Without user intervention, when to duplicate control points becomes a difficult decision; for example, Menet [64] duplicates control points in regions where after M steps of contour deformation, the curvature is higher than a user-supplied threshold θ.

4.2 The Hermite representation

The previous considerations lead us to propose the Hermite representation for deformable contour finding. Hermite contours are piecewise cubic polynomials, which interpolate the control points $\mathbf{p_0}, ..., \mathbf{p_N}$. In each interval, the Hermite cubic $Q(\mathbf{s}, t) = [x(\mathbf{s}, t)\ y(\mathbf{s}, t)]$ is specified by the positions $\mathbf{p_{i-1}}$, $\mathbf{p_i}$ and tangent vectors $\tau^+_{\mathbf{i-1}}$, $\tau^-_{\mathbf{i}}$ at the endpoints.

Let $\mathbf{Q}$ be an arbitrary cubic polynomial

$$\mathbf{Q} = \mathbf{T} \cdot \mathbf{C} \tag{4.1}$$

where

$$\mathbf{T} = [t^3\ t^2\ t^1\ 1] \qquad \mathbf{C} = \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix}$$

with tangent vector $\mathbf{Q'(t)}$

$$\mathbf{Q'} = \mathbf{T'} \cdot \mathbf{C} = [3t^2\ 2t\ 1\ 0]\ \cdot\ \mathbf{C} \tag{4.2}$$

Given Hermite parameter matrix

$$\mathbf{H_i} = [\mathbf{h_{i_x}} \mathbf{h_{i_y}}] = [\mathbf{p_{i-1}}\ \mathbf{p_i}\ \tau^+_{\mathbf{i-1}}\ \tau^-_{\mathbf{i}}]^T \tag{4.3}$$

the corresponding Hermite coefficient matrix $\mathbf{C_{H_i}}$ can be derived as [30]

$$\mathbf{C_{H_i}} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{H_i}$$

The Hermite parameters are collected in state vector $\mathbf{H}$ for later use

$$\mathbf{H} = \begin{bmatrix} \tau^-_{\mathbf{0}} \\ \mathbf{p_0} \\ \tau^+_{\mathbf{0}} \\ \ldots \\ \tau^-_{\mathbf{N}} \\ \mathbf{p_N} \\ \tau^+_{\mathbf{N}} \end{bmatrix} \tag{4.4}$$

When considering the same criteria of usefulness for the contour finding problem as discussed in previous section for the point-, polygon- and spline-based representations, one note that the Hermite representation

- can efficiently represent both smooth and sharp contours. This is because smooth contours are well represented by the Hermite interpolating cubics, while at the same time, arbitrary sharp corners can be easily generated at the control points by the adjustment of the left and right tangent vector parameters
- interpolates the control points
- is explicit in those features that can be measured from image data: position and direction of gradient at control points. This allows to prune the search space during contour finding, as we will see in next section.

4.3 MAP formulation

A *maximum a posteriori* (MAP) criterion is formulated for the solution of the contour finding problem. The aim is to find from all possible contours the contour which matches the image data best, in a probabilistic sense. Let $\mathbf{d}$ be the image to be matched and $\mathbf{t_H}$ be the image template corresponding to the Hermite parameters $\mathbf{H}$. Desired is $\mathbf{H_{MAP}}$ which maximizes the probability that $\mathbf{t_H}$ occurs given $\mathbf{d}$, e.g. $P(\mathbf{t_{H_{MAP}}}|\mathbf{d})$. $\mathbf{t_{H_{MAP}}}$ is then the *maximum a posteriori* solution to the problem. Bayes rule gives

$$\begin{aligned} P(\mathbf{t_{H_{MAP}}}|\mathbf{d}) &= \max_{\mathbf{H}} P(\mathbf{t_H}|\mathbf{d}) \\ &= \max_{\mathbf{H}} \frac{P(\mathbf{d}|\mathbf{t_H})\ P(\mathbf{t_H})}{P(\mathbf{d})} \end{aligned} \tag{4.5}$$

where $P(\mathbf{d}|\mathbf{t_H})$ is the conditional probability of the image given the template, and $P(\mathbf{t_H})$ and $P(\mathbf{d})$ are the prior probabilities for template and image, respectively. Taking the natural logarithm on both sides of eq.(4.5) and discounting $P(\mathbf{d})$, which does not depend on $\mathbf{H}$, leads to an equivalent problem of maximizing objective function U

$$\begin{aligned} U(\mathbf{t_{H_{MAP}}}, \mathbf{d}) &= \max_{\mathbf{H}} U(\mathbf{t_H}, \mathbf{d}) \\ &= \max_{\mathbf{H}} (lnP(\mathbf{t_H}) + lnP(\mathbf{d}|\mathbf{t_H})) \end{aligned} \tag{4.6}$$

The above equation describes the trade-off between a-priori and image-derived information.

If the image to be matched is considered as a noise corrupted template with additive and independent noise that is zero-mean Gaussian, we have $P(\mathbf{d}|\mathbf{t_H}) = P(\mathbf{d}|\mathbf{t_H} + \mathbf{n}) = P(\mathbf{n}|\mathbf{d} - \mathbf{t_H})$, thus

$$P(\mathbf{d}|\mathbf{t_H}) = \prod_{\mathbf{t_H}(x,y)} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(d(x,y)-t_H(x,y))^2}{2\sigma_n^2}} \tag{4.7}$$

and

$$\mathit{ln}\ P(\mathbf{d}|\mathbf{t_H}) \ = \ \mathit{constant} + \sum_{\mathbf{t_H}(x,y)} \frac{(d(x,y) - t_H(x,y))^2}{2\sigma_i^2} \tag{4.8}$$

This last term can be replaced by correlation term $\mathbf{d} \cdot \mathbf{t_H}$, approximating $||\mathbf{d}||$ and $||\mathbf{t_H}||$ by constants. For $||\mathbf{d}|| = 1$ and $||\mathbf{t_H}|| = 1$ we obtain

$$\max_{\mathbf{H}} \ \mathit{ln}\ P(\mathbf{d}|\mathbf{t_H}) \ \approx \ \min_{\mathbf{H}} (1 - \mathbf{d} \cdot \mathbf{t_H}) \ = \ \min_{\mathbf{H}} \ E_{ext} \tag{4.9}$$

A similar derivation was described by Rosenfeld and Kak [84] and Staib and Duncan [90].

In the above derivation, the data values of $\mathbf{d}$ and $\mathbf{t_H}$ at image location (x, y), $d(x, y)$ and $t_H(x, y)$, were assumed scalar; $d(x, y)$ typically represents an edge magnitude (as derived by a Sobel operator) and $t_H(x, y)$ contains a normalized value of 1 if image location (x, y) lies on the Hermite contour or 0 if it lies outside. For the case of non-scalar data types (e.g. $d(x, y)$ and $t_H(x, y)$ represent the image intensity gradient and the contour normal, respectively) one needs to adapt the noise model of Equation 4.7.

The prior probability for a Hermite contour $\mathbf{H}$ is modeled as

$$P(\mathbf{H}) = P(\mathbf{H}|\overline{\mathbf{H}}) \ = \ \mathit{constant} \cdot \prod_i e^{-\frac{(\mathbf{H}[i] - \overline{\mathbf{H}}[i])^2}{2\sigma_i^2}} \tag{4.10}$$

where $\overline{\mathbf{H}}$ represents an expected contour. $\overline{\mathbf{H}}$ is typically obtained as the sample mean of contours generated in a training phase, or as the contour obtained by prediction during tracking. σ_i acts as a weighing measure for the various dimensions. In case of an open contour, we set the σ's of τ_0^- and τ_N^+ to a non-value.

In tracking applications the contour typically undergoes a transformation T (for example, translation, rotation and scale) for which one does not want to penalize. The above modeling assumes that any transformation on the contour which one does not want to penalize has already been performed, before eq.(4.10) is applied. Any further contour change is considered as deformation from an expected contour and thus penalized.

Taking the natural logarithm gives

$$\max_{\mathbf{H}} \ \mathit{ln}P(\mathbf{t_H}) \ = \ \min_{\mathbf{H}} \sum_i \frac{(\mathbf{H}[i] - \overline{\mathbf{H}}[i])^2}{2\sigma_i^2} \ = \ \min_{\mathbf{H}} \ E_{int} \tag{4.11}$$

4.4 Optimization

There are many ways to solve the resulting minimization problem

$$\min_{\mathbf{H}} E \;=\; \min_{\mathbf{H}} \, (E_{int} + E_{ext}). \tag{4.12}$$

Variational calculus methods have been used extensively for continuous parameter spaces where derivative information is available [20] [52] [64] [78] [90] [95]. For discrete search spaces one possibility is to use A.I. search techniques. A discrete enumeration technique based on dynamic programming (DP) is used here which was popularized by Amini *et al.* [4], and used since by [32] [57]. The advantages of dynamic programming with respect to variational calculus methods are in terms of stability, optimality and the possibility to enforce hard constraints [4]. For dynamic programming to be efficient compared to the exhaustive enumeration of the possible solutions, the decision process should be *Markovian*. This is typically the case if the a priori-shape component E_{int} contains a summation of terms which only depend on parameters which can be derived locally along the contour.

For the case of open contours, our objective function can be written as

$$\begin{aligned} E \;=\; & E_1(\mathbf{p_0}, \tau_{\mathbf{0}}^+, \tau_{\mathbf{1}}^-, \mathbf{p_1}) \;+\; \ldots \;+ \\ & E_N(\mathbf{p_{N-1}}, \tau_{\mathbf{N-1}}^+, \tau_{\mathbf{N}}^-, \mathbf{p_N}) \end{aligned} \tag{4.13}$$

Applying the dynamic programming technique to our formulation involves generating a sequence of functions of two variables, s_i with $i \;=\; 0..N-1$, where for each s_i a minimization is performed is over two dimensions. s_i are the optimal value functions defined by

$$\begin{aligned} s_0(\tau_{\mathbf{1}}^-, \mathbf{p_1}) \;&=\; \min_{\mathbf{p_0}, \tau_{\mathbf{0}}^+} E_1(\mathbf{p_0}, \tau_{\mathbf{0}}^+, \tau_{\mathbf{1}}^-, \mathbf{p_1}) \\ s_i(\tau_{\mathbf{i+1}}^-, \mathbf{p_{i+1}}) \;&=\; \min_{\mathbf{p_i}, \tau_{\mathbf{i}}^+} (s_{i-1}(\mathbf{p_i}, \tau_{\mathbf{i}}^+) \;+ \\ &\quad\; E_i(\mathbf{p_i}, \tau_{\mathbf{i}}^+, \tau_{\mathbf{i+1}}^-, \mathbf{p_{i+1}}) \;) \\ &\quad\; i \;=\; 1 \;..\; N-1 \end{aligned} \tag{4.14}$$

If $\mathbf{p_i}$ and $\tau_{\mathbf{i}}^-$ $(\tau_{\mathbf{i}}^+)$ range over N_P and N_T values at each index i, the complexity of the proposed algorithm is $O(NN_P^2N_T^2)$.

The above formulation is for open contours. For closed contours, where the first and last control point are defined equal, we apply the same algorithm as for the open contour case, yet repeat it for all N_P possible locations of the first (last) control point, while keeping track of the best solution. The complexity increases to $O(NN_P^3N_T^2)$.

Speed-up can be achieved by a multi-scale approach. Here contour finding is first done on a lower resolution image to find an approximated contour. This can be done with a coarse discretization of the parameter space (i.e. requiring smaller N_P and N_T for the same parameter range). At the finer level, the originally desired discretization can be achieved by decreasing the parameter range to lie around the solution found at the coarse level.

At the same scale, the algorithm can be sped up by discounting unprobable control point locations before starting the DP search. A measure of "unprobability" can be specified in terms of weak image gradient strength or dot product between measured and expected gradient directions (the latter are explicit in the Hermite representation). If all the candidate control point locations are rated similarly (e.g. standard deviation of ratings below a threshold), it is more robust to consider all.

In addition, for closed contours, one can use only a single pass DP for closed contour and to optimize for the remaining $E_0(\mathbf{p_N}, \tau^+_{\mathbf{N}}, \tau^-_{\mathbf{0}}, \mathbf{p_0})$ while assigning to $\mathbf{p_0}$ and $\mathbf{p_N}$ the optimal values found for the open contour case. Of course, all these speed-up procedures loose the optimality property of DP. Nevertheless, the last two methods which were implemented performed satisfactory in practice.

4.5 Tracking

The high computational cost of dynamic programming, and of other search methods which do not get stuck in the closest local minimum, makes search only feasible in a limited neighborhood. For interactive contour delineation this is fine, since the user is likely to place well-positioned control points, very close to the desired contour. In tracking applications this requirement is often unrealistic. On the other hand, it is our observation that the effects of deformation are often small from frame to frame once rigid motion is accounted for.

The effects of motion and deformation on the contour are therefore decoupled, first, transformation parameters $T = [\mathbf{t}, \phi, s]$ are searched for, with $\mathbf{t}$, ϕ and s denoting translation, rotation and scaling. T is found with respect to the undeformed contour, after which search continues for the deformation parameters. The first stage is robustly performed by template matching (or Hough Transform [57]) on a Gaussian-blurred gradient image.

The second stage is the DP approach described earlier. Both stages use motion prediction methods; template matching at time $t + 1$ searches in a parameter range centered around predicted transformation $\overline{T}(t+1)$ using predicted template $\overline{\mathbf{H}}(\mathbf{t+1})$. $\overline{\mathbf{H}}(\mathbf{t+1})$ is also the initial contour of DP search.

For simplicity, currently $\overline{T}(t+1) = T(t)$ and $\overline{\mathbf{H}}(t+1) = \mathbf{H}(\mathbf{t})$ is used. More general, $\overline{T}(t+1) = p(t+1)$ where p is a best fitting n-th order polynomial at (t-M, T(t-M)) ..., (t-1, t). Similar consideration holds for $\overline{\mathbf{H}}(\mathbf{t+1})$. If the time-span M in which a n-th order model holds is large it is efficient to use a recursive predictor such as the Kalman filter.

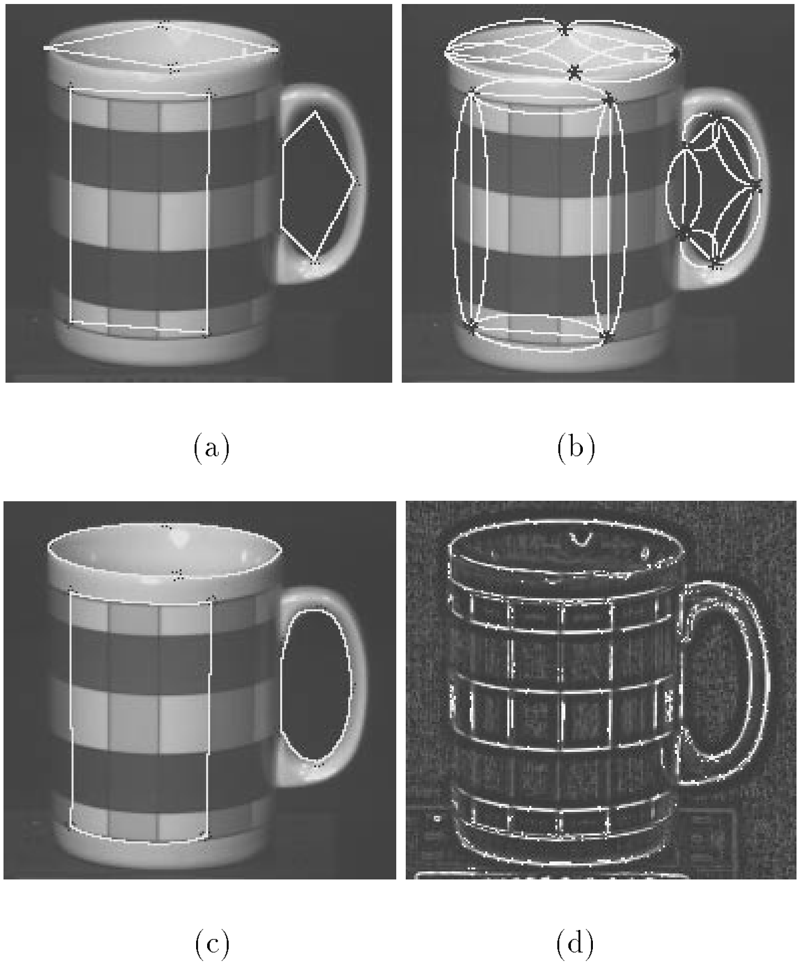
4.6 Experiments

Several experiments have been performed with the proposed combination of Hermite representation and template-plus-DP search in both interactive as tracking settings. The associated template matching parameters were range and discretization of the transformation parameters (translation, rotation and scale). DP-related parameters included the initial values of the Hermite parameters, their range and discretization, as well as the weighing parameters. The locations considered around control point $\mathbf{p_i}$ lied on a rectangular grid with x-axis perpendicular to $\mathbf{p_{i+1}} - \mathbf{p_{i-1}}$. The Hermite gradients $\tau_\mathbf{i}$ were described in terms of length l_i and direction ϕ_i. Typically, $N = 5$, $N_P = 9$ ($N_P = 4$ after pruning), $N_l = 3$, $N_\phi = 9$.

Figure 4.1 demonstrates versatility of the Hermite representation. Different initial contours are placed by the user as shown in Figure 4.1a. Figure 4.1b shows the search region covered by DP for the initial control point placement; for each contour segment the Hermite cubics are shown corresponding to ($\phi_{i_{max}}$, $\phi_{i+1_{max}}$) and ($\phi_{i_{min}}$, $\phi_{i+1_{min}}$) for fixed (initial) control point locations and $l = l_{max}$. Many different Hermite contours which lie within this search region are not displayed. Figure 4.1c shows the result of contour finding by DP. One observes a wide variety of shapes that have been accurately described by the Hermite representations, from the smoothly varying contour of the mug rim to the sharp corners of the square pattern, with a curved horizontal segment joining at the corner. It compares favorably with a possible representation by polygonal chains, splines or Fourier descriptors.

For completeness, we also show in Figure 4.1d the conditioned Sobel gradient image, which is used by the DP algorithm. A conditioned image is used instead of the original Sobel image in order to amplify weak but probable edges. This is done based on local considerations, taking into account mean μ and standard deviation σ in a $n \times n$ neighborhood. A linear remapping is applied on the image data at (x, y) if σ is greater than a user specified threshold.

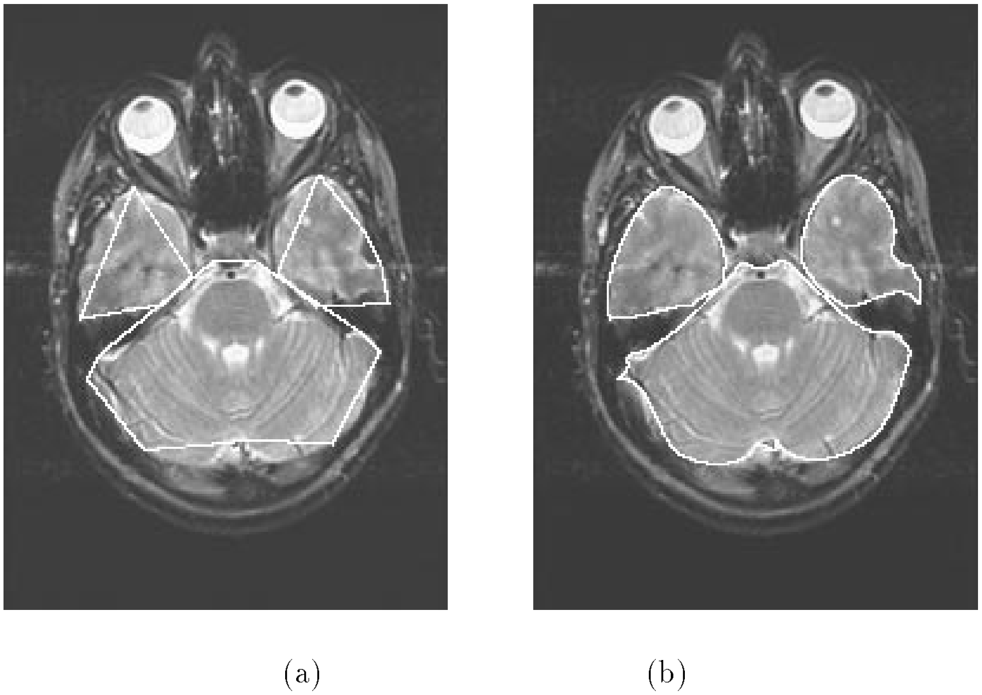
Figure 4.2 shows different instances of initial placement and contour detections on a MR image of the human brain. Figure 4.3 shows a tracking sequence of a head using the proposed combination of coarse-scale template matching and DP. Finally, Figure 4.4 shows a sequence where different face features are tracked. Here the measure of fit between Hermite contour and image was based on greyscale statistics (mean and variance) along the inside and outside boundaries of contour segments.



(a) (b)

(c) (d)

Figure 4.1: Mug image (a) contour initialization (b) search region (c) contour detection (d) conditioned Sobel image



(a) (b)

Figure 4.2: Brain MR image (a) contour initialization (b) contour detection

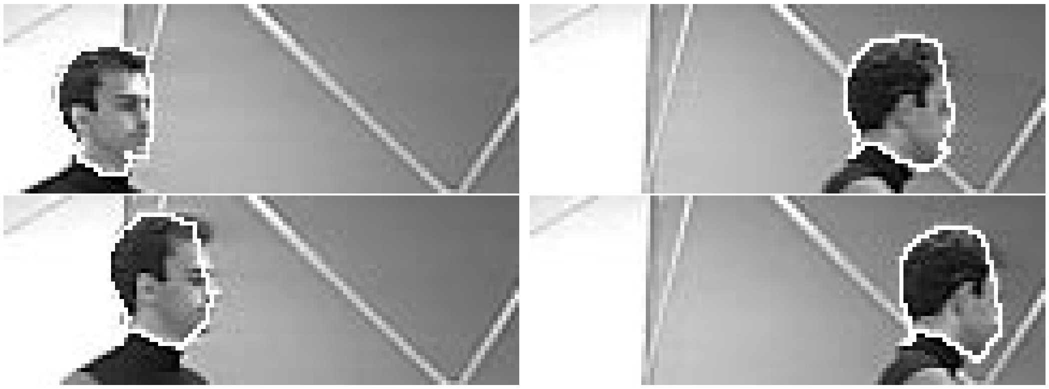


Figure 4.3: A head tracking sequence (t = 0, 8, 24, 30)

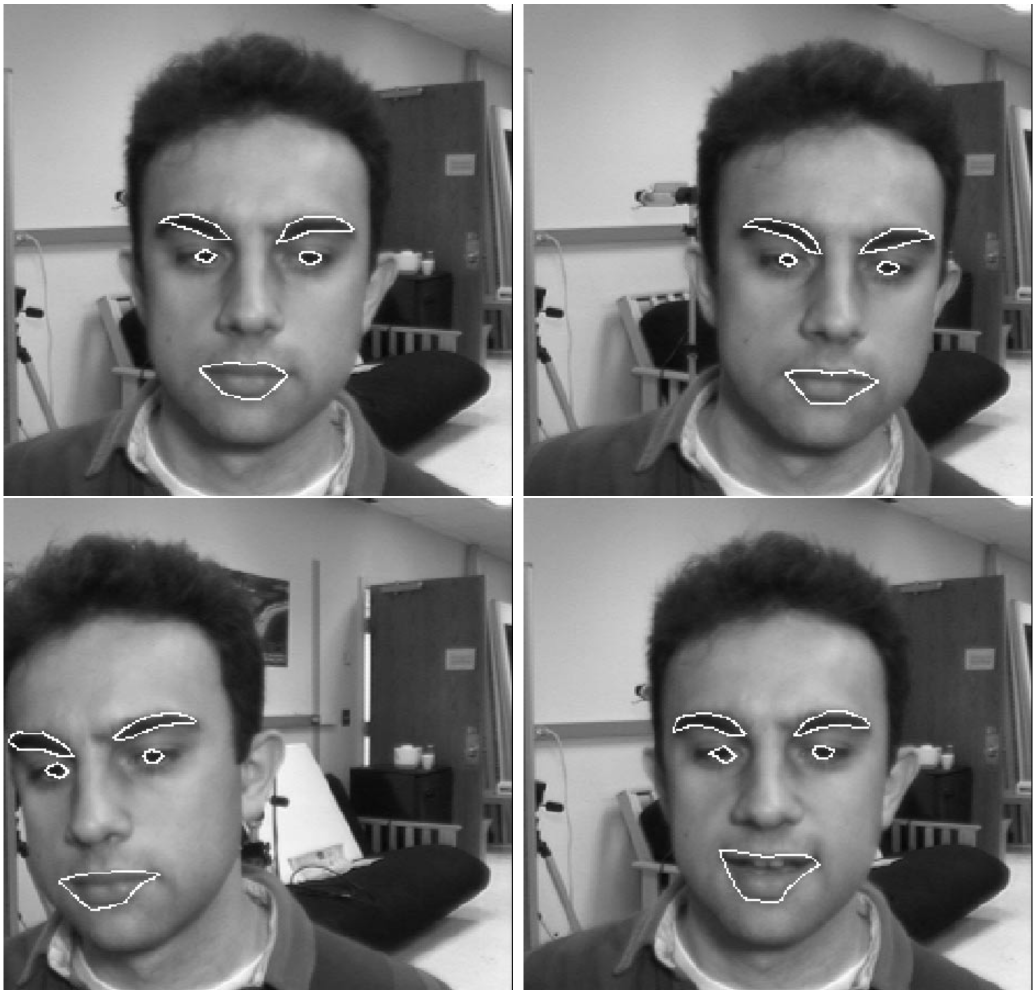


Figure 4.4: A face tracking sequence (t = 0, 20, 60, 120)

Chapter 5

Towards 3-D head model acquisition

The acquisition of accurate 3-D textured head-models from images is important for many applications such as computer animation, virtual reality, video resolution enhancement and low bit-rate image coding [2]. For example, many successful movies have come to rely on high precision scanned 3-D (head) models of actors for their special-effects; twisting, bending and morphing an actor's appearance to the degree desired. Some TV commercials have followed suit, most notably leading to toddlers "breakdancing" and basketball teams consisting of replicated star-players.

Apart from these and similar applications which require very accurate 3-D head models and which can afford acquisition in controlled environments (by active sensing) and costly equipment, the arrival of cheap single-camera systems on top of PCs and workstations is likely to facilitate a 3-D head model acquisition ability from monocular images for the general public. Although less accurate than systems based on structured light or passive stereo, such monocular systems could provide sufficient realistic models to allow face animation, for example as part of a "personalized" human-computer interface, including speech. In this setting, one can reasonably assume a degree of user cooperation in obtaining his or her 3-D head model. This can range from the desirable none to some light form of scripting, where the user has to perform a pre-determined head movement (for example, starting in a frontal view and turning side-ways), and even to requiring some assistance in the image processing part (for example in resetting features when they get lost during tracking).

In this chapter, the head-model estimation problem is examined in the above single-camera-on-desk setting, with the scope initially restricted to the case of a rigid (but unknown) head shape. The main features of the followed approach are the integration of motion and structure estimates over time in the recursive framework of the Kalman-filter [15] and the use of occluding contours to incrementally obtain a more accurate head shape. The use of a recursive framework facilitates a real-time implementation, although this was not pursued here.

The outline of this chapter is as follows. Section 5.1 provides an overview of related work. Section 5.2 discusses a Kalman filter implementation for motion and structure recovery following Azarbayejani and Pentland [5]. Section 5.3 contains the initial approach to the problem followed by the obtained results, in Section 5.4.

5.1 Related Work

Active sensing techniques allow very precise 3-D shape recovery. Structured (visible or infrared) light is used to illuminate parts of the object distinctively with a line or dot pattern. Using properly calibrated sensors from two or more viewpoints, one can easily find point correspondences between the views and recover 3-D shape by triangulation; this involves placing the object or sensors on a motion platform with known motion for full surface coverage. There are several commercial products based on this method, for example by Cyberware.

A number of researchers have considered passive stereo techniques, where no special lighting is required. Koch [55] describes a method where camera motion is estimated by regularized optical flow; the recovered motion is used to fuse successive depth maps together. Akimoto *et al* use only frontal and profile views of heads to warp a generic 3-D model. Some simplifying assumptions are made, among others that two cameras are placed in precise orthogonal configuration with horizontal and aligned epipolar geometry.

Another related line of research has dealt with the general problem of recovering shape from occluding contours assuming either known motion [105], controlled motion [23] (object placed on turntable), or the use of multiple cameras [47]. Various ways are proposed to deal with the "aperture problem".

For the monocular case, most work on 3-D head models has concentrated on tracking with a known and fixed model, initialized with various degrees of user assistance in a frontal view [11], [54] [96]. Some have extended the model with non-rigid deformations as controlled by known Action Units to account for facial expressions [16], [59]. Motion estimation has been achieved by either optical flow or by a generate- and-test strategy. Because of the reliance on a fixed model fitted to the frontal view, these methods encounter difficulties when dealing with significant head rotations.

There has been little work done on the simultaneous estimation of both shape and head motion. Some ideas have been proposed in [14], [12] and [54], which entail updating the mesh structure after the overall rigid head motion has been estimated by optical flow. Many questions remain in terms of the computational cost and robustness of the resulting algorithms. Recently, some interesting work by DeCarlo and Metaxas [26] has dealt with the incorporation of optical flow in a deformable model framework.

5.2 Motion and structure estimation

This work relies on the extended Kalman Filter (EKF) to estimate motion and structure recursively. The use of temporal information, in the form of feature tracks combined with a motion model, makes the approach less prone to error accumulation than some of the successive two-frame motion estimation approaches mentioned earlier. The EKF formulation developed by Azarbayejani and Pentland [5] is used, which is summarized below.

The camera model is the pinhole model

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} \frac{1}{1 + Z_C \beta} \tag{5.1}$$

where (X_C, Y_C, Z_C) is the location of a 3-D point in the camera reference frame, (u, v) is the image location and $\beta = 1/f$ is the inverse focal length, the latter which can be estimated. 3-D feature location is expressed in terms of image location (u, v) and unknown depth α

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} u\beta \\ v\beta \\ 1 \end{pmatrix} \tag{5.2}$$

Pointwise structure is described with one parameter per point. Under the assumption of zero-mean noise on the image locations, it was shown [5] that features effectively only have one degree of freedom in 3-D space. Even when measurement biases exists, most uncertainty remains along the depth dimension, justifying the structure parametrization $(\alpha_1, ..., \alpha_N)$ for N features. Restricting the otherwise $3N$ dimensional parameter space to N dimensions increases stability of the filter.

Translation is estimated by $(t_X, t_Y, t_Z\beta)$. Image locations are related to camera, structure and motion parameters by eq.5.1 and

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C\beta \end{pmatrix} = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \beta \end{pmatrix} \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{5.3}$$

The 3-D rotation is defined as the rotation of the object reference frame relative to the camera reference frame. Interframe rotation is expressed in Euler angles $(\omega_X, \omega_Y, \omega_Z)$ centered around zero, which is combined with an external quaternion representation (q_0, q_1, q_2, q_3) to maintain an estimate of global rotation.

The state vector used in the Kalman filter consists of $N + 7$ parameters, 6 for motion, 1 for camera geometry and N for structure

$$\mathbf{x} = (t_X, t_Y, t_Z\beta, \omega_X, \omega_Y, \omega_Z, \beta, \alpha_1, ..., \alpha_N) \tag{5.4}$$

The state model in the EKF has been chosen trivially as identity plus noise, assuming no a-priori information about system dynamics. The measurement equation is given by combining Equations 5.1, 5.3 and 5.2. The RHS (u, v) in eq.5.2 defines the image location of the feature in the initial frame, and the LHS (u, v) in eq.5.1 represents the measurement. The issue of scale is resolved by fixing the depth α of a point throughout the sequence. This is achieved by setting its initial variance to zero.

5.3 Head model acquisition

The initialization procedure consists of warping a generic 3-D head model to a frontal view of a head. Fitting requires locating facial features in the image and computing their XY coordinates in 3-D space by back-projection onto a plane of constant depth $Z = Z_0$, the distance from the camera to the center of the head; this determines the scale of the model. The corresponding XY vertex coordinates of the generic head model are mapped onto these. To interpolate non-feature model vertices, an identical triangulation is defined on the XY coordinates of both the model and image features; non-feature vertices in a particular "model" triangle are mapped onto the corresponding "image" triangle using bi-linear interpolation. In absence of more detailed information about depth, a constant scale factor is applied to the Z axis to warp the model. This scale factor is set equal to the XY distance ratio between two features in the original and warped model (i.e. the eyes). The head is centered at depth Z_0.

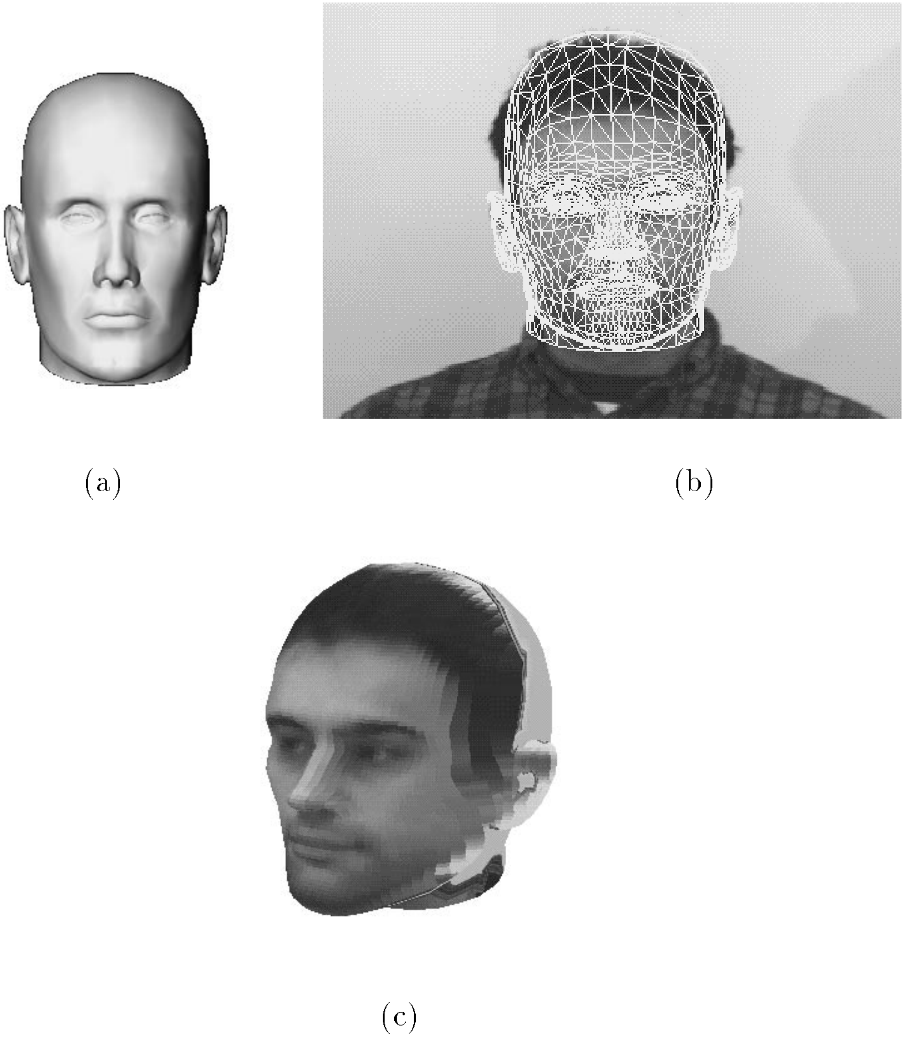
As a first attempt to obtain 3-D head data from image sequences, we run the Kalman filter to obtain motion estimates which are used to animate the warped generic model. At each iteration, image points on the outline of the head are back-projected and it is determined where the corresponding rays come closest to the vertices on the occluding contour of the animated 3-D model; these points are called "attachment" points. These attachment points are transformed and collected in an object-centered coordinate system for further processing.

5.4 Experiments

Figure 5.1 shows the generic and the fitted 3-D head model. The feature points for the model-warping were the eyes, nostrils, mouth, chin, top and sides of the head. Their location was determined interactively. Figure 5.1c shows the result of texture mapping the model of Figure 5.1b. The model-warping gives fairly realistic results for the eye-nose-mouth area for large rotations, however, cheek and hair regions are poorly represented.

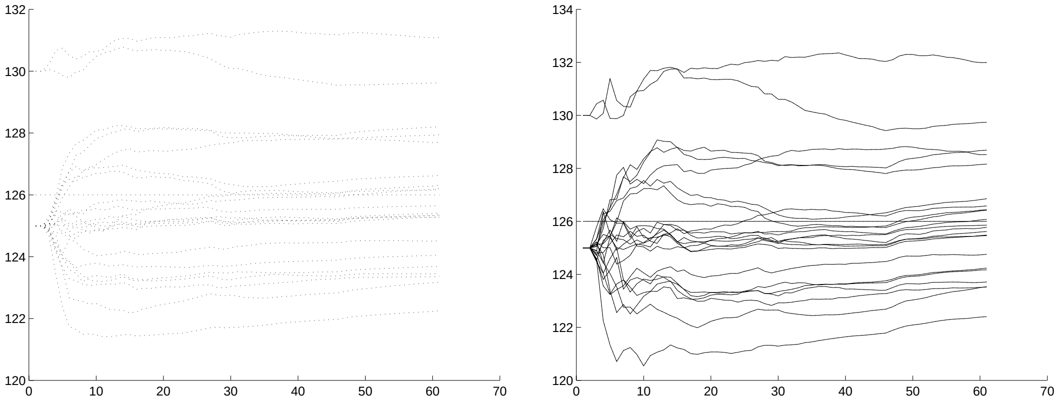
An image sequence is considered of a face turning from a left to a right profile view. No assumption is made about the motion performed as the Kalman filter is applied. The features correspond to left/right corners of eyes and mouth, nostrils, and skin marks; they were tracked manually for the moment. Figure 5.6ab shows the used features in the frontal view and the remaining visible ones in the right profile view. A noisy sequence was also generated by adding uniform noise of ±3 pixels to the feature tracks (the image size is 760 x 484). Parameter β was computed from the camera and lens specifications and not estimated. The structure and motion estimates of the Kalman filter are given in Figures 5.2, 5.3, 5.4 and 5.5, for unperturbed ($\Delta n = 0$) and perturbed input ($\Delta n = 3$). The depth of one of the eye corners was fixed at $Z = 126$ cm, which was the depth of the corresponding vertex in the warped model after placing it at $Z = Z_0 = 130$ cm. The two features close to the ears were initialized at $Z = 130$ cm, all others were initialized at $Z = 125$ cm. Figure 5.2 shows structure converges within 15 frames for the unperturbed data, somewhat longer for the perturbed case. The addition of noise does not have a major impact on the motion estimates as can been seen in Figures 5.3 and 5.4. Figure 5.5 shows a good performance of the Kalman filter for the unperturbed case; the error between the measured and estimated image locations is overall less than 0.4 % of the image extent, i.e. less than 3 pixels. Evidently, this error increases with perturbed data, partly because the measurements are noisy themselves.

Figure 5.7 shows the attachment points connected in a mesh, as it "wraps around" the head, for the perturbed case. The attachment points for the unperturbed case are shown in Figure 5.8. A fairly close fit can be observed.



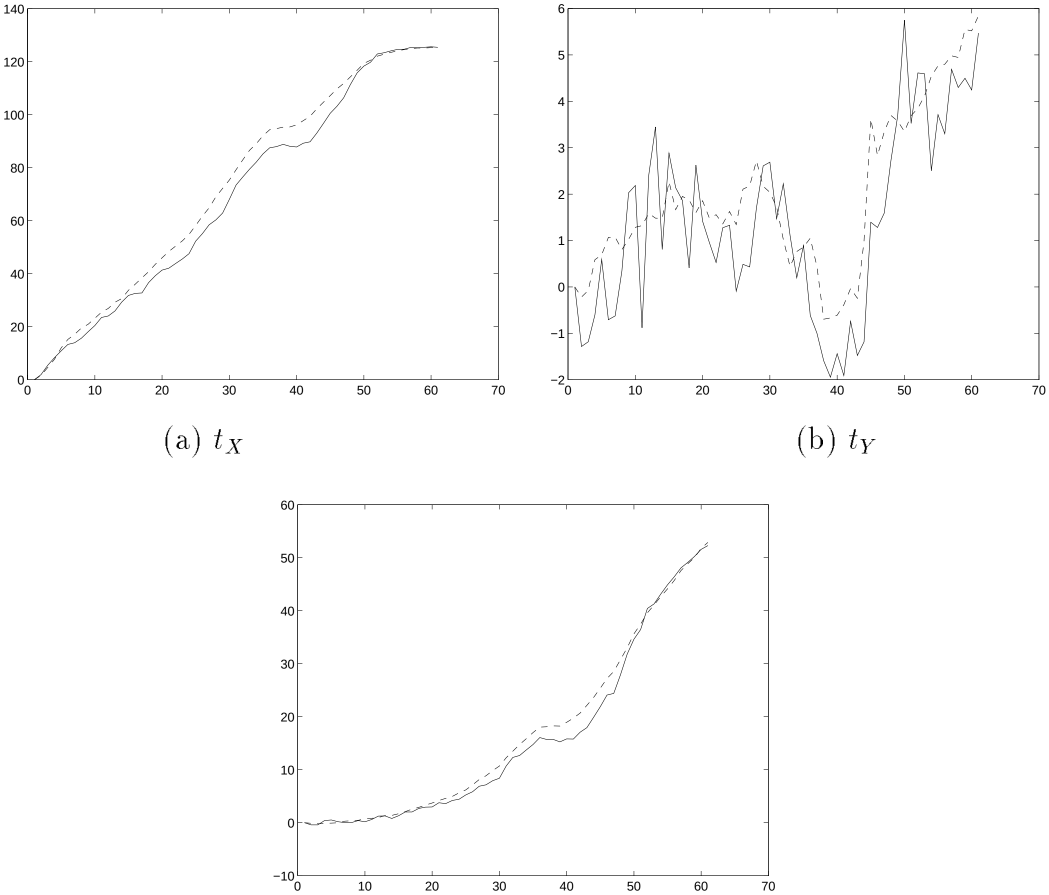
(a) (b) (c)

Figure 5.1: (a) Generic 3-D head model, (b) fitted model, (c) texture-mapped model



(a) (b)

Figure 5.2: Kalman filter structure estimates (α_i in cm) vs. frame number: (a) noise $\Delta n = 0$ pixels, and (b) noise $\Delta n = 3$ pixels.

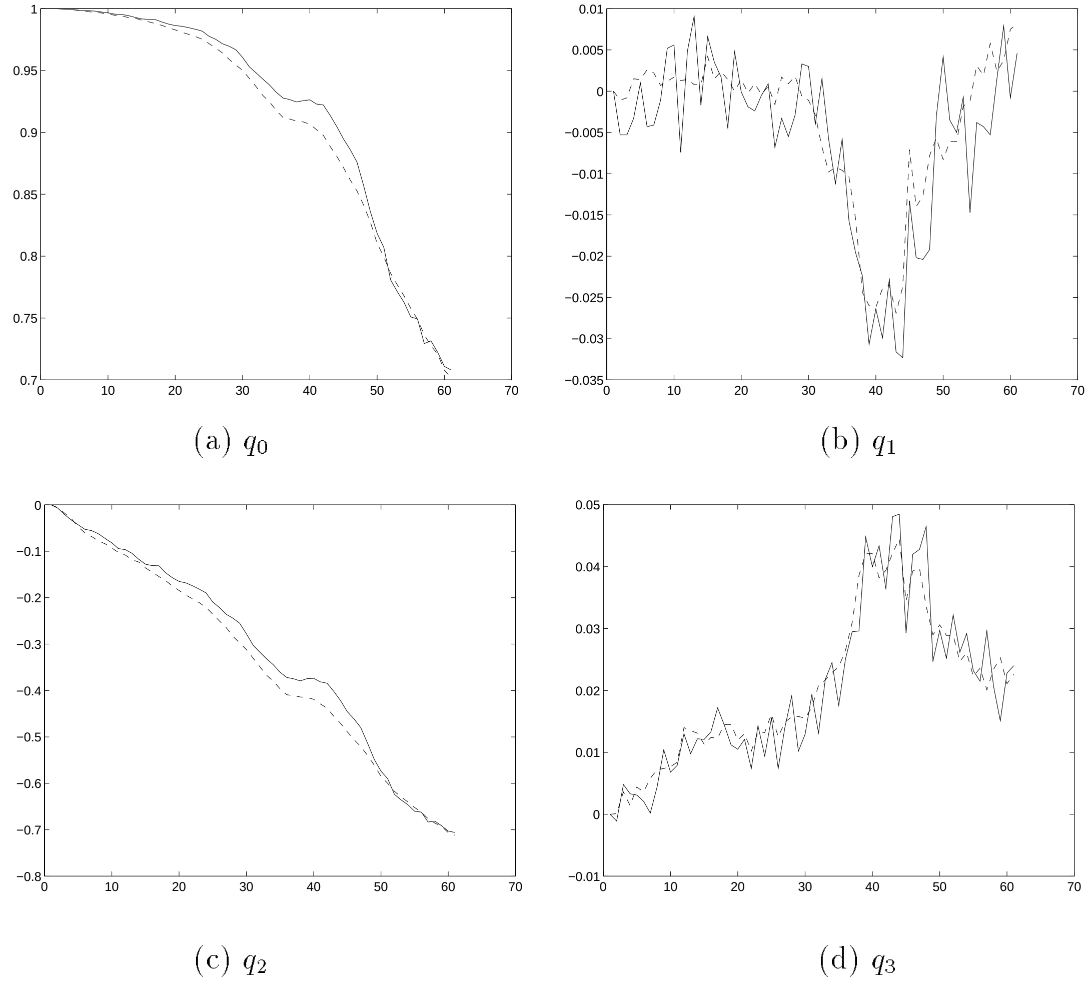


(a) t_X

(b) t_Y

(c) t_Z

Figure 5.3: Kalman filter translation estimates (t_X, t_Y and t_Z in cm) vs. frame number: no noise added (dotted) vs. uniform noise (± 3 pixels) added (solid).



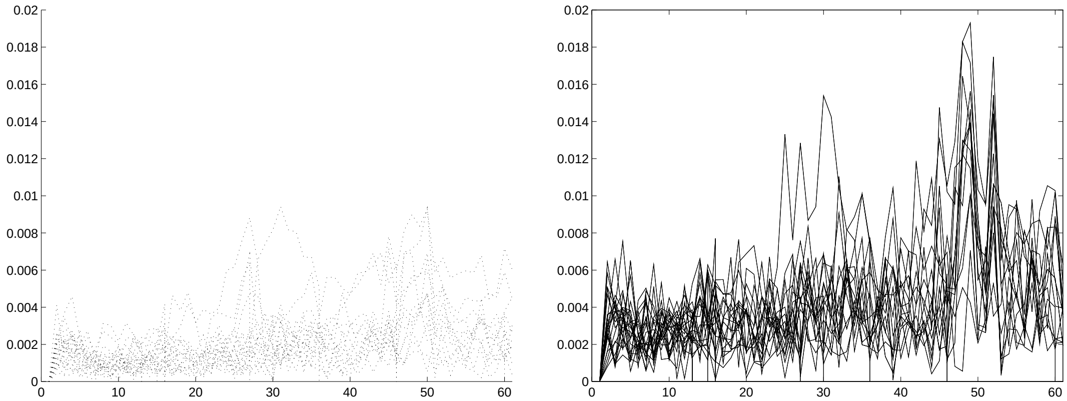
(a) q_0

(b) q_1

(c) q_2

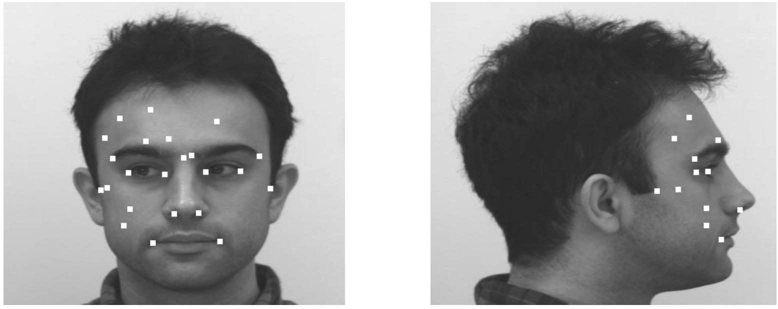
(d) q_3

Figure 5.4: Kalman filter rotation estimates (unit quaternions) vs. frame number: no noise added (dotted) vs. uniform noise (±3 pixels) added (solid).



(a) (b)

Figure 5.5: Residual errors of feature location (in fractions of image plane size): (a) noise $\Delta n = 0$ pixels, and (b) noise $\Delta n = 3$ pixels.



(a) (b)

Figure 5.6: The features in the frontal view (a) and in the profile view (b)

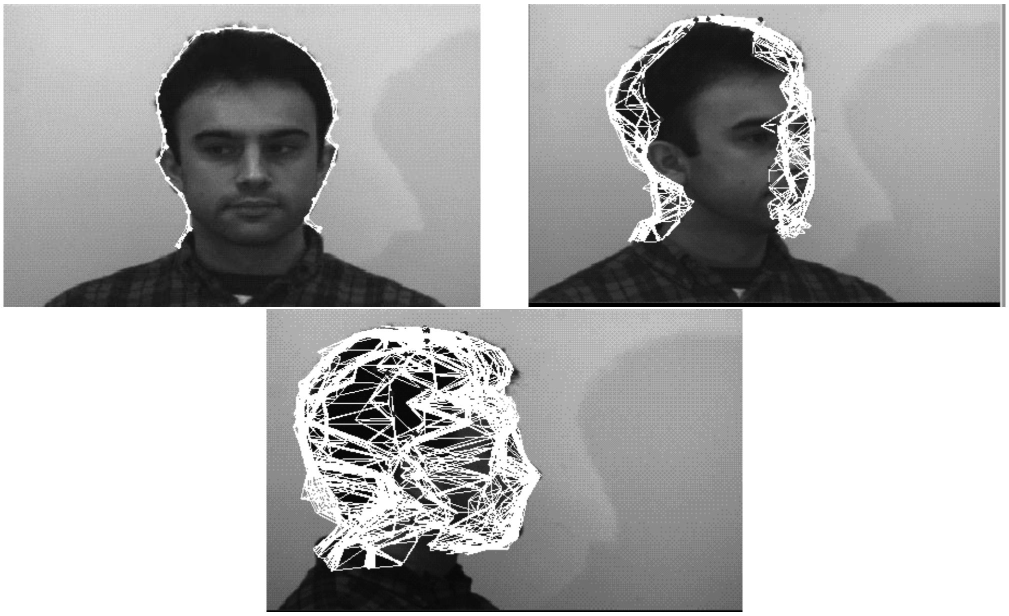
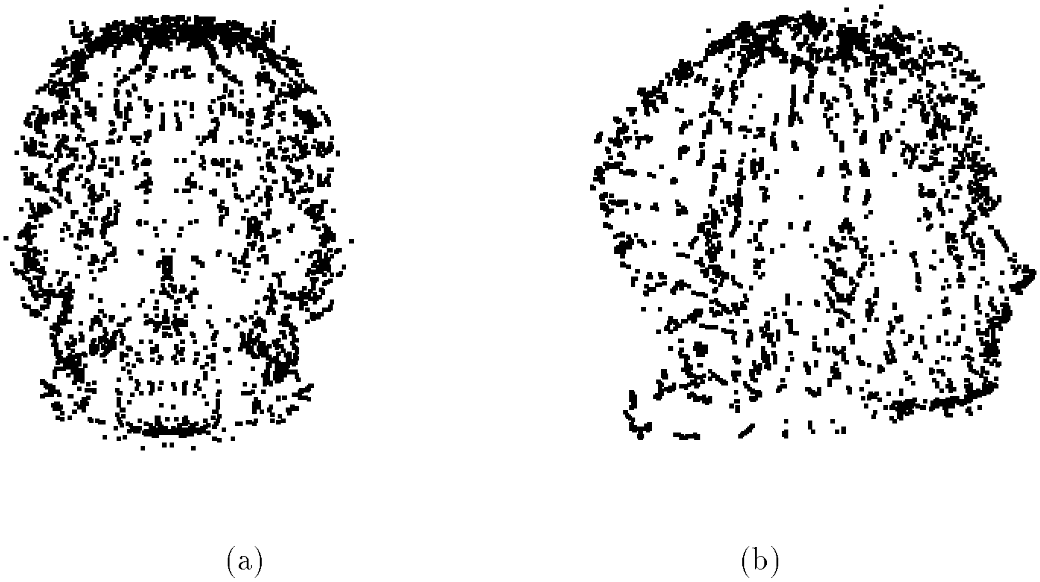


Figure 5.7: 3-D head mesh acquisition from contours (perturbed input)



(a) (b)

Figure 5.8: 3-D head point data from contours (unperturbed input)

Chapter 6

Conclusions and future work

6.1 Summary

This thesis has presented a system for vision-based 3-D tracking of unconstrained whole-body movement without the use of markers. The 3-D recovery approach was motivated by two considerations, to obtain a more meaningful feature set for action recognition and to obtain 3-D data for applications such as virtual reality. Pose-recovery was formulated as a search problem and entailed finding the pose parameters of a graphical human model whose synthesized appearance is most similar to the actual appearance of the real human in the multi-view images. The models used for this purpose were acquired from the images semi-automatically. A decomposition approach and a best-first technique was used to search through the high dimensional pose parameter space. A robust variant of chamfer matching was used as a fast and well-behaved similarity measure between synthesized and real edge images. A variant of Dynamic Time Warping was introduced to allow the matching of unsegmented movement patterns.

A large Humans-In-Action database has been compiled which contains multi-view images of human subjects involved in a variety of activities. The results obtained in this thesis demonstrate, for the first time, successful 3-D tracking of unconstrained whole-body movement on real images. In particular, the following two conclusions can be drawn from the experimental results. First, the calibration and human modeling procedures support a (perhaps surprisingly) good 3-D localization of the model such that its projection matches the all-around camera views. This is good news for the feasibility of *any* multi-view 3-D model-based tracking method, not just for the proposed one. Second, the proposed pose recovery and tracking method based on, among others, the chamfer distance as similarity measure, is indeed able to maintain a good fit over time.

This thesis has also introduced Hermite deformable contours to improve on 2-D edge segmentation by using shape prediction. Their representation was shown to have advantages over point-, polygonal- and spline-based representations in terms of versatility, stability and controlability. A decoupled approach to contour tracking was proposed based on template matching on a coarse scale to account for motion effects, and dynamic programming on a finer scale to account for the deformation effects. These ideas were demonstrated on images from a variety of domains.

Finally, first steps have been made towards the recursive 3-D head model acquisition from monocular head-shoulder images. The motion estimate obtained by a Kalman filter and a generic 3-D head model fitted to a frontal view were used to obtain rough 3-D head shape from a sequence of occluding head contours.

6.2 Future work

There are many ways to extend the current work on 3-D body tracking. An important area for improvement is image segmentation. The approach taken in this thesis has been to restrict work in 2-D to a minimum and place all the burden on a combinatoric 3-D search in order to demonstrate the feasibility of 3-D model-based tracking. More emphasis on image segmentation is likely to result in higher algorithm efficiency by a restriction of the relevant search space, similar to the ideas of O'Rourke and Badler [71]. For example, pose search restrictions could be derived from the detection of the medial axes of limbs or the detection of skin-colored regions like the hand and the face. Further reductions of the search space could be achieved by triangulation on these features. The resulting approach would place the here proposed pose recovery methods more in the context of pose verification. At the same time, further work on 2-D labeling of body parts is needed to allow the tracking system to bootstrap (or to recover) from a wider variety of poses. The robustness of pose recovery can be increased if contour features are first labeled before brought into correspondence. The incorporation of Hermite deformable contours in the system would allow better edge segmentation and also provide temporal cues. Furthermore, region-based features (e.g. color) are likely to be useful to complement the current edge-based approach.

Another area of improvement is the use of more sophisticated human models (e.g. flexible torso) which include some notions about dynamics (e.g. support). Their automatic acquisition from images containing human movement also needs to be addressed. Further improvement deals with reasoning about occlusion and knowing when to initiate or stop tracking of body parts.

Finally, it would be interesting to develop a symbolic component on top of the tracking component which would allow reasoning on a semantic level about what human movement is observed in the scene. This would involve defining and recognizing generic movement primitives (e.g. "extending hand") or poses (e.g. "holding object"), placing the events into a database together with previous knowledge, and allowing inferences by user-supplied rules. The symbolic component might also formulate the vision tasks to be executed, for example, changing the mode of tracking from a fine-scale (with each body part tracked) to a coarse scale (with human body considered as a whole) and vice versa, depending on context.

It is expected that with these improvements, 3-D based vision systems will have greatly improved capabilities to deal with complex human movement successfully. This might include analyzing the Argentine Tango, see Figure 6.1.

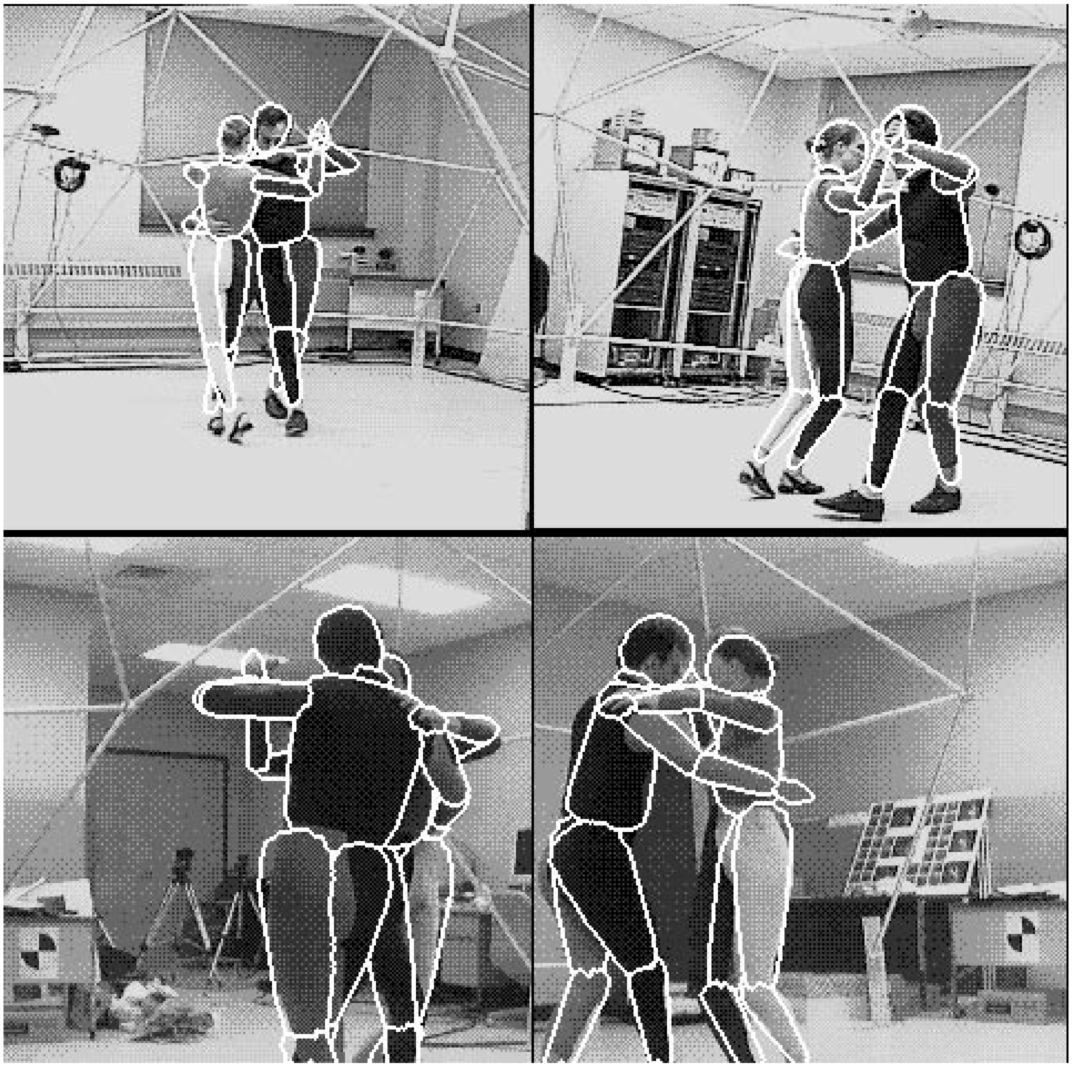


Figure 6.1: "Manual" 3-D pose recovery for a pair dancing the Argentine Tango (cameras FRONT, RIGHT, BACK and LEFT)

Bibliography

[1] J. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Articulated and elastic non-rigid motion: A review. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 2–14, Austin, 1994.

[2] K. Aizawa and T. Huang. Model-based image coding: Advanced video coding techniques for very low bit-rate applications. *Proceedings of IEEE*, 83(2):259–271, 1995.

[3] K. Akita. Image sequence analysis of real world human motion. *Pattern Recognition*, 17(1):73–83, 1984.

[4] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[5] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6), 1995.

[6] N. Badler, C. Phillips, and B. Webber. *Simulating Humans.* Oxford University Press, Oxford, 1993.

[7] N. Badler and S. Smoliar. Digital representations of human movement. *ACM Computing Surveys*, 11(1):19–38, 1979.

[8] D. Ballard and C. Brown. *Computer Vision.* Prentice-Hall, Eaglewood Cliffs, 1982.

[9] A. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18:21–30, 1984.

[10] H. Barrow et al. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.

[11] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *International Conference on Pattern Recognition*, Vienna, 1996.

[12] M. Bichsel. Human face recognition. In *International Workshop on Automatic Face and Gesture Recognition*, Zurich, 1995.

[13] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.

[14] G. Bozdagi, A.Tekalp, and L. Onural. Simultaneous 3-d motion estimation and wire-frame model adaptation for knowledge-based video coding. In *ICASSP*, pages 413–416, 1996.

[15] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):90–99, 1986.

[16] D. Cai, X. Wang, and H. Liang. Several key problems in model-based image sequence compression by using interframe A.U.s correlation. In *International Conference on Image Processing*, pages 409–413, 1994.

[17] T. Calvert and A. Chapman. Analysis and synthesis of human movement. In *Handbook of Pattern Recognition and Image Processing: Computer Vision*, pages 432–472. Academic Press, 1994.

[18] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *International Conference on Computer Vision*, pages 624–630, Cambridge, 1995.

[19] C. Cedras and M. Shah. Motion-based recognition, a survey. *Image and Vision Computing*, 13(2):129–154, 1995.

[20] A. Chakraborty, M. Worring, and J.S. Duncan. On multi-feature integration for deformable boundary finding. In *International Conference on Computer Vision*, pages 846–851, 1995.

[21] C. Charayaphan and A. Marble. Image processing system for interpreting motion in American Sign Language. *Journal of Biomedical Engineering*, 14(15):419–425, 1992.

[22] Z. Chen and H. Lee. Knowledge-guided visual perception of 3-D human gait from a single image sequence. *IEEE Transactions on on Systems, Man and Cybernetics*, 22(2):336–342, 1992.

[23] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, 1992.

[24] T. Darrell and A. Pentland. Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, New York, 1993.

[25] J. Davis and M. Shah. Gesture recognition. Technical Report CS-TR-93-11, University of Central Florida, 1993.

[26] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with application to human face shape and motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–238, San Francisco, 1996.

[27] P. Delagnes, J. Benois, and D. Barba. Active contours approach to object tracking in images sequences with complex background. *Pattern Recognition Letters*, 16:171–178, 1995.

[28] B. Dorner. Hand shape identification and tracking for sign language interpretation. In *Looking at people, International Joint Conference on Artificial Intelligence*, Chambery, 1993.

[29] A. Downton and H. Drouet. Model-based image analysis for unconstrained human upper-body motion. In *IEE International Conference on Image Processing and its Applications*, pages 274–277, 1992.

[30] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics.* Addison-Wesley, 1996.

[31] W. Freeman et al. Computer vision for computer games. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 100–105, Killington, 1996.

[32] D. Geiger et al. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3), 1995.

[33] A. Geurtz. *Model-based Shape Estimation.* PhD thesis, Department of Electrical Engineering, Polytechnic Institute of Lausanne, 1993.

[34] N. Goddard. Incremental model-based discrimination of articulated movement direct from motion features. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 89–94, Austin, 1994.

[35] L. Goncalves et al. Monocular tracking of the human arm in 3-D. In *International Conference on Computer Vision*, pages 764–770, Cambridge, 1995.

[36] Y. Guo, G. Xu, and S. Tsuji. Understanding human motion patterns. In *International Conference on Pattern Recognition*, pages 325–329, 1994.

[37] T. Heap and D. Hogg. Towards 3-D hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 140–145, Killington, 1996.

[38] Y. Hel-Or and M. Werman. Recognition and localization of articulated objects. *International Journal of Computer Vision*, 19(1), 1996.

[39] M. Herman. *Understanding body postures of human stick figure.* PhD thesis, Department of Computer Science, University of Maryland, 1979.

[40] D. Hoffman and B. Flinchbaugh. The interpretation of biological motion. *Biological Cybernetics*, 42:195–204, 1982.

[41] D. Hogg. Model based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[42] R. Holt, A. Netravali, T. Huang, and R. Qian. Determining articulated motion from perspective views: A decomposition approach. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 126–137, Austin, 1994.

[43] E. Hunter, J. Schlenzig, and R. Jain. Posture estimation in reduced-model gesture input systems. In *International Workshop on Automatic Face and Gesture Recognition*, pages 290–295, Zurich, 1995.

[44] D. Huttenlocher, G. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[45] G. Johansson. Visual perception of biological motion and a model for its analysis. Perception and Psychophysics, 14(2), 1973.

[46] I. Jolliffe. *Principal Component Analysis.* Springer Verlag, New York, 1996.

[47] T. Joshi, N. Ahuja, and J. Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. In *International Conference on Computer Vision*, pages 290–295, Cambridge, 1995.

[48] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parametrized model of articulated image motion. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, 1996.

[49] R. Kahn, M. Swain, P. Prokopowicz, and J. Firby. Gesture recognition using the perseus architecture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 734–741, San Francisco, 1996.

[50] I. Kakadiaris and D. Metaxas. 3-D human body model acquisition from multiple views. In *International Conference on Computer Vision*, pages 618–623, Cambridge, 1995.

[51] I. Kakadiaris and D. Metaxas. Model-based estimation of 3-D human motion with occlusion based on active multi-viewpoint selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 81–87, San Francisco, 1996.

[52] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[53] R. Kjeldsen and J. Kender. Toward the use of gesture in traditional user interfaces. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 151–156, Killington, 1996.

[54] R. Koch. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), 1993.

[55] R. Koch. 3-d modeling of human heads from stereoscopic image sequences. *Submitted to Symposium der Deutsche Arbeitsgemeinschaft für Mustererkennung*, 1996.

[56] J. Kuch and T. Huang. Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *International Conference on Computer Vision*, pages 666–671, Cambridge, 1995.

[57] K. Lai and R. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:1084–1090, 1995.

[58] M. Leung and Y. Yang. First sight: A human body outline labeling system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):359–377, 1995.

[59] H. Li, P. Roivainen, and R. Forcheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), 1993.

[60] W. Long and Y. Yang. Log-tracker, an attribute-based approach to tracking human body motion. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(3):439–458, 1991.

[61] N. Magnenat-Thalman and D. Thalmann. Human modeling and animation. In *Computer Animation*, pages 129–149. Springer-Verlag, 1990.

[62] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three dimensional shapes. *Proceedings Royal Society London B*, 200:269–294, 1978.

[63] D. McNeill. *Hand and Mind - What Gestures Reveal about Thought.* The University of Chicago Press, Chicago and London, 1992.

[64] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes: implementation and application to stereo. In *ARPA Image Understanding Workshop*, pages 720–726, 1990.

[65] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

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

[67] N. Nilsson. *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers, 1980.

[68] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in x-y-t. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[69] S. Niyogi and E. Adelson. Analyzing gait with spatiotemporal surfaces. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 64–69, Austin, 1994.

[70] J. Ohya and F. Kishino. Human posture estimation from multiple images using genetic algorithm. In *International Conference on Pattern Recognition*, pages 750–753, 1994.

[71] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.

[72] A. Pentland. Automatic extraction of deformable models. *International Journal of Computer Vision*, 4:107–126, 1990.

[73] A. Pentland. Smart rooms. *Scientific American*, 274(4):54–62, 1996.

[74] F. Perales and J. Torres. A system for human motion matching between synthetic and real images based on a biomechanic graphical model. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 83–88, Austin, 1994.

[75] R. Polana and R. Nelson. Low level recognition of human motion. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, Austin, 1994.

[76] F. Quek. Eyes in the interface. *Image and Vision Computing*, 13(6), 1995.

[77] L. Rabinier. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings IEEE*, 77(2):257–285, 1989.

[78] P. Radeva, J. Serrat, and E. Marti. A snake for model-based segmentation. In *International Conference on Computer Vision*, pages 816–821, Cambridge, 1995.

[79] K. Rangarajan, W. Allen, and M. Shah. Matching motion trajectories using scale space. *Pattern Recognition*, 26(4):595–610, 1993.

[80] J. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European Conference on Computer Vision*, pages 35–46, 1994.

[81] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *International Conference on Computer Vision*, pages 612–617, Cambridge, 1995.

[82] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 59(1):94–115, 1994.

[83] M. Rosenblum, Y. Yacoob, and L.S. Davis. Human emotion recognition from motion using a radial basis function network architecture. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 43–49, Austin, 1994.

[84] A. Rosenfeld and A. Kak. *Digital Picture Processing.* Academic Press, 1982.

[85] S. Sarkar and K. Boyer. Optimal infinite impulse response zero crossing based edge detectors. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54:224–243, 1991.

[86] J. Segen and S. Pingali. A camera-based system for tracking people in real-time. In *International Conference on Pattern Recognition*, pages 63–67, Vienna, 1996.

[87] T. Shakunaga. Pose estimation of jointed structures. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 566–572, 1991.

[88] A. Shio and J. Sklansky. Segmentation of people in motion. *IEEE Workshop on Visual Motion*, pages 325–332, 1991.

[89] M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley and Sons, Inc., 1989.

[90] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992.

[91] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using hidden markov models. In *International Symposium on Computer Vision*, pages 265–270, Coral Gables, 1995.

[92] R. Szeliski and S.-B. Kang. Recovering 3-D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.

[93] K. Takahashi et al. Recognition of dexterous manipulations from time-varying images. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 23–28, Austin, 1994.

[94] S. Tamura and S. Kawasaki. Recognition of sign language motion images. *Pattern Recognition*, 21(4):343–353, 1988.

[95] D. Terzopoulos and D. Metaxas. Dynamic 3-D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[96] A. Tsukamoto, C. Lee, and S. Tsuji. Detection and pose estimation of human face with synthesized image models. In *International Conference on Pattern Recognition*, pages 754–757, 1994.

[97] M. Turk. Visual interaction with lifelike characters. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 368–373, Killington, 1996.

[98] N. Ueda and K. Mase. Tracking moving contours using energy-minimizing elastic contour models. In *European Conference on Computer Vision*, 1992.

[99] J. Webb and J. Aggarwal. Structure from motion of rigid and jointed objects. In *Seventh International Joint Conference on Artificial Intelligence*, pages 686–691, Vancouver, 1981.

[100] A. Wilson and A. Bobick. A state-based technique for the summarization and recognition of gesture. In *International Conference on Computer Vision*, pages 382–388, Cambridge, 1995.

[101] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Killington, 1996.

[102] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–665, 1991.

[103] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[104] J. Zhao. *Moving Posture Reconstruction from Perspective Projections of Jointed Figure Motion.* PhD thesis, University of Pennsylvania, 1993.

[105] J.Y. Zheng. Acquiring 3-d models from sequences of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):163–178, 1994.