



**KTH Numerical Analysis  
and Computer Science**

# **Video Based Analysis and Visualization of Human Action**

MARTIN ERIKSSON

Doctoral Thesis  
Stockholm, Sweden 2005

TRITA-NA-0438

ISSN-0348-2952

ISRN-KTH/NA/R-04/38-SE

ISBN 91-7283-926-0

CVAP 294

KTH Numerisk analys och datalogi

SE-100 44 Stockholm

SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen fredagen den 21 januari 2005 kl 14.00 i Kollegiesalen, Administrationsbyggnaden, Kungl Tekniska högskolan, Valhallavägen 79, Stockholm.

© Martin Eriksson, december 2004

Tryck: Universitetservice US AB

Omslagsbild ©Hasse Sjögren

## Abstract

Analyzing human motion is important in a number of ways. An athlete constantly needs to evaluate minute details about his or her motion pattern. In physical rehabilitation, the doctor needs to evaluate how well a patient is rehabilitating from injuries. Some systems are being developed in order to identify people only based on their gait. Automatic interpretation of sign language is another area that has received much attention. While all these applications can be considered useful in some sense, the analysis of human motion can also be used for pure entertainment. For example, by filming a sport activity from one view, it is possible to create a 3D reconstruction of this motion, that can be rendered from a view where no camera was originally placed. Such a reconstruction system can be enjoyable for the TV audience. It can also be useful for the computer-game industry. This thesis presents ideas and new methods on how such reconstructions can be obtained.

One of the main purposes of this thesis is to identify a number of *qualitative constraints* that strongly characterizes a certain class of motion. These qualitative constraints provide enough information about the class so that every motion satisfying the constraints will "look nice" and appear, according to a human observer, to belong to the class. Further, the constraints must not be too restrictive; a large variation within the class is necessary. It is shown how such qualitative constraints can be learned automatically from a small set of examples.

Another topic that will be addressed concerns analysis of motion in terms of quality assessment as well as classification. It is shown that in many cases, 2D projections of a motion carries almost as much information about the motion as the original 3D representation. It is also shown that single-view reconstruction of 2D data for the purpose of analysis is generally not useful. Using these facts, a prototype of a "virtual coach" that is able to track and analyze image data of human action is developed. Potentials and limitations of such a system are discussed in the the thesis.

The thesis consists of two main parts. The first part primarily deals with issues concerning visualization. The second part focus more on analysis of the motion.



# Acknowledgements

## Top ten people who have inspired me to pursue research

**#10. Staffan och Bengt.** The guys with the science-show on TV for kids, that taught everybody of my generation that flour on the table-hockey game makes the puck go faster...

**#9. Professor Balthazar.** The Croatian cartoon, who can invent anything by pulling the lever of his grande machine...

**#8. McGyver,** who can invent anything out of a roll of duct-tape and a couple of paper-clips...

**#7. John Badham,** the director of “War games”. Now we’re talking AI...

**#6. Niel Armstrong,** who personified the expression “The winner takes it all”...

**#5. Buzz Aldrin,** who personified the expression “Second sucks”. In my book Buzz is a winner, though...

**#4. The left-wing parties,** who claim you can eat the cake and still have it. Now, that’s science...

**#3. The right-wing parties,** who claim you can eat the cake and still have it. Unfortunately, this is no longer science...

**#2. Douglas Adams,** the author of “The hitchhiker’s guide to the galaxy”. The meaning of life cannot be 42, though. It has to be a prime...

And the **#1** person who have inspired me to pursue research is **Bamse’s grandmother** who makes the dunderhonung (dunderhoney). Ben Johnson’s medicine kit fades in comparison...

Tied for **#11.**

Everybody I have met at CVAP and CAS over the years deserve tons of credit. I am glad I had a chance to meet so many crazy people in one spot. Keep up the fantastic work.

# Contents

<b>Contents</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Human motion capture systems . . . . .	2
1.2 Automatic human motion capture . . . . .	3
1.3 State of the art . . . . .	4
1.4 Interactive marker free motion capture . . . . .	9
1.5 Motion recognition and classification . . . . .	16
1.6 Setting the stage - outline and contributions . . . . .	17
<b>2 Epipolar geometry constraints in multiple view motion tracking</b>	<b>21</b>
2.1 Software system . . . . .	22
2.2 Manual reconstruction . . . . .	22
2.3 Missing data . . . . .	22
2.4 Automatically tracked data . . . . .	23
2.5 Identification of undecidable points . . . . .	24
2.6 Estimating uncertain points . . . . .	27
2.7 Reconstruction example . . . . .	29
2.8 Chapter summary . . . . .	30
<b>3 Monocular reconstruction - prior constraints</b>	<b>33</b>
3.1 What is a prior? . . . . .	33
3.2 Bayesian inference . . . . .	34
3.3 Learning the constraint manifold - carving priors . . . . .	38
3.4 Smoothing 2D shapes . . . . .	43
3.5 Experimental examples of 2D point sets . . . . .	45
3.6 3D shapes . . . . .	49
3.7 Single view reconstruction . . . . .	55
3.8 Summary . . . . .	57

<b>4</b>	<b>Key framed single view reconstruction</b>	<b>61</b>
4.1	System overview . . . . .	62
4.2	Summary of 2D tracking . . . . .	63
4.3	3D key frames . . . . .	65
4.4	Establishing an initial estimate . . . . .	65
4.5	Fitting the smooth motion estimate to the joint data . . . . .	68
4.6	Results . . . . .	69
4.7	Concluding remarks . . . . .	71
<b>5</b>	<b>Reconstruction by qualitative selection</b>	<b>75</b>
5.1	Evaluation . . . . .	76
5.2	Reconstruction as a selection problem. . . . .	77
5.3	Pruning the search space . . . . .	78
5.4	Configuration dependencies . . . . .	79
5.5	Selecting the correct sequence . . . . .	81
5.6	Forming the final reconstruction . . . . .	86
5.7	Summary . . . . .	90
<b>6</b>	<b>2D vs. 3D data for motion based recognition and classification</b>	<b>93</b>
6.1	Comparing motion . . . . .	94
6.2	Test scenarios . . . . .	97
6.3	Finding the optimal angle . . . . .	98
6.4	Different people doing the same action . . . . .	98
6.5	Individual variation . . . . .	101
6.6	Intermediate discussion . . . . .	105
6.7	Analyzing monocular reconstructions . . . . .	108
6.8	Conclusions of monocular reconstruction . . . . .	111
<b>7</b>	<b>Action quality assessment from automatic motion capture</b>	<b>119</b>
7.1	Motion tracking revisited . . . . .	122
7.2	Temporal assessment . . . . .	124
7.3	Spatial assessment . . . . .	128
7.4	Conclusion . . . . .	130
<b>8</b>	<b>Summary and future directions</b>	<b>133</b>
8.1	Summary . . . . .	133
8.2	Future research . . . . .	134
	<b>Bibliography</b>	<b>137</b>





# Chapter 1

## Introduction

Everybody even remotely involved with any form of athletics knows about the tremendous complexity of human motion. During my years as an athlete (pole vaulter), I spent a large amount of time in front of the TV watching videotapes from my own attempts, comparing them to those of my competitors in order to understand why some of them outperformed me. My efforts in this quest, though, declined towards the end of my career. Primarily because it was too difficult to draw any fruitful conclusions. Trying to mimic the style of someone who performs better than you generally does not lead to any improvements. The optimal technique seems to vary between individual athletes. There is not generic motion pattern that everybody should strive for. For example, one of the most memorable olympic moments was Michael Johnson's race in the 200m dash in Atlanta. With tense shoulders, a backward lean and very poor knee lift, he didn't just brake the world record. He annihilated it! With his running posture, most coaches would send him back to basics and start with beginning running drills. However, Michael maintained his running style throughout his career, simply because it was his natural way to run. I doubt he would trade his gold medals for a more conventional running style. Anyone trying to mimic this style would probably be very disappointed. The history of sports is full of similar examples. The history is also full of researchers trying to find the correlation between human locomotion and the quality of the resulting performance.

When I began researching this area, the idea of developing a fully automatic and virtual "coach", consisting of two (or maybe only one) cheap video cameras, inspired me. What if I could, right after an attempt, get instant feedback from my virtual coach about technical flaws? Maybe I could even get warnings about when technical deficiencies could lead to injuries down the road. If I could have solved this to any degree of success, I would have gained an outstanding advantage over my competitors. As indicated by the fact that I never reached the olympic medal I was shooting for, I did not succeed. At least not on time. However, I did learn that a virtual coach has to distinguish between two questions:

1. What is the optimal technique.
2. How close was one trial to this optimal technique.

As an athlete, or an orthopaedic for that matter, you have to develop an idea about the optimal motion pattern. After that, it is possible to start to rate the quality of each trial, based on this idea. The first problem is an off-line task, and the second is the one to be solved on-line. The off-line task can be referred to as *motion analysis*, while the on-line task could be called *motion feedback*. One main conclusion from this thesis is that computer vision techniques are generally too blunt to provide data for advanced motion analysis, while they can provide great tools for motion feedback.

One of the most interesting area for computer vision is the field of visualization. A system that, right after a tennis point has been played, can render the action from a new angle (why not from the umpire's seat) as an animation should be very interesting for TV broadcasters. Not for the purpose of motion analysis, not for the purpose of motion feedback, but for the purpose of entertainment.

This thesis will explore some existing techniques for all these tasks, and also present some new approaches to reconstruction of human motion. The material in the thesis is primarily extensions of the results of the following publications.

- Eriksson, M. and Carlsson, S. "Maximizing validity in 2D Motion Analysis," International Conference on Pattern Recognition, 2004.
- Eriksson, M. and Carlsson, S. "Monocular Reconstruction of Human Motion by Qualitative Selection," International Conference on Face and Gesture Recognition, 2004.
- Loy, G., Eriksson, M., Sullivan, J. and Carlsson, S. "Monocular 3D Reconstruction of Human Motion in Long Action Sequences," European Conference on Computer Vision, 2004.
- Eriksson, M., Carlsson, S. "Carving Prior Manifolds Using Inequalities", IEEE Workshop on Learning in Computer Vision and Pattern Recognition, 2003.
- Eriksson, M. , Carlsson, S. "Qualitative Characterization and Use of Prior Information," Scandinavian Conference on Image Analysis, 2003.
- Sullivan, J. Eriksson, M. and Carlsson, S., "Recognition, Tracking and Reconstruction of Human Motion," Articulated Motion and Deformable Objects (AMDO) 2002
- Sullivan, J., Eriksson, M., Carlsson, S., Liebowitz, D., "Automation Multi-View Tracking and Reconstruction of Human Motion," ECCV Workshop on Vision and Modeling of Dynamic Scenes, 2002.

## 1.1 Human motion capture systems

In terms of accuracy, there are no other means that come close to the commercial motion capture systems available on the market today. Motion capture systems generally require an actor to wear special markers (reflective markers in the case of optical motion capture systems). By using several infrared cameras surrounding the actor, the 3D position of

each marker can be computed. Typically, today's systems are able to record positions of hundreds of markers in the order of 1000 Hz. While being very accurate in determining trajectories of markers in 3D, the motion capture system does not read minds. In other words, sophisticated methods are required to interpret the data. For example, if we are interested in tracking the joint center of a person's knee, we cannot, at least not without surgery (which would anyways lead to severe occlusion) place the marker at the exact center of rotation. One solution is to place one marker on "each side" of the knee, and approximate the center of rotation as the midpoint between these two markers. This is of course a rather crude model of a human joint and, as should be expected, more advanced approaches exist. For example, it is common to use a skeletal model to improve the accuracy of the motion capture data (Herda et al., 2002). Another significant problem in marker based motion capture is that the skin may be sliding with respect to the skeleton, causing errors in the reconstructed motion. Also, some markers may be occluded in some frames, requiring interpolation. Interpolating over long gaps may in term yield violations in limb length consistency and symmetry of the reconstructed skeleton. Evaluations of various approaches to solve these problems are outlined in (Halvorsen, 2002). The existence of motion capture systems greatly simplifies life for two categories of people: Researchers in biomechanics and animators.

### **Biomechanics**

In biomechanics, the possibility of acquiring exact reconstructions of movements has lead to increased knowledge about human performance in several fields. Clinical motion capture is widely used for gait analysis, where the walk pattern of for example patients rehabilitating from stroke can be monitored. By combining motion capture with other sensors, such as force plates and EMG (electromyography) measuring the muscular activity, it is possible to estimate a model about the forward kinematics of the human motion.

### **Computer Graphics**

Animating a realistic motion is difficult. Computer animators are artists, with a large supply of tricks to generate nice looking clips. However, the strongest weapon in the quest for realistic animations is the motion capture system. Some of today's computer games, based on athletics, often use motion capture clips of world class athletes, to enhance the authenticity of the game. If I play Tiger Woods, chances are high that the drive of my animated player actually looks like Tiger Wood's drive. Animating this without using motion capture data is more or less a futile task, since the individual variations between players are very minute in quantitative terms. Despite this, it is possible for the human eye to distinguish between two different players.

## **1.2 Automatic human motion capture**

What can be done if we do not want to move the Wimbledon final from the centrecourt to the motion capture lab? Actually, research in 3D reconstruction from video contains sev-

eral methods where semi automatic reconstruction of 3D motion can be done by a *degree* of human intervention. Such systems will be explored and developed in this thesis. The problem of automatically acquiring a 3D motion reconstruction or 2D motion primitives from a video sequence has intrigued researchers for many years, and a wide variety of approaches has been suggested. As most methods presented are developed for a particular application, where each application has a different set of constraints and assumptions, it is difficult to compare their level of success, since they don't play according to the same rules. Most systems developed for video based motion capture, however, share many components. Although most systems focus on different issues, there are some fundamental aspects that always have to be addressed. A canonical system designed for the task of automatic video based reconstruction should involve the following:

1. **Initialization.** Find the configuration of the human model that best complies with the video data with respect to the appearance model, in the first frame of the sequence.
2. **Tracking.** Update the configuration of the human model in subsequent frames based on video data and prior knowledge about the motion.

### 1.3 State of the art

Reconstruction of human motion is the problem of, given some sensor data, assembling a representation of the configuration of the human body at some sampled intervals. Any sensor can be used with the most common sensors being cameras of some kind. In this thesis, the main focus is on reconstruction of human motion from video sequences. The reason for this being that it is (to the best of my knowledge) the only approach possible to achieve a totally non-intrusive system. Any other method would require the person to wear some kind of special equipment. This would disqualify the system from, for instance, reconstructing motions of athletes in a competitive setting. It also makes clinical analysis much more difficult and tedious. An outline of the issues in vision based methods for motion analysis and reconstruction is shown in fig. 1.1. Depending on the objective of the system, some features in the video(s) are extracted. These features can be extracted using a model of the human body, and also a model of the motion. If the purpose is to generate a 3D reconstruction (for visualization purposes), a model is always required. If the objective is to do motion analysis in 2D, appearance based methods may suffice. Some approaches to these areas are presented next.

#### Feature extraction

The essence of computer vision is to extract useful features from a set of images. The final solution to the problem of identifying edges, ridges, blobs and image flow from video has unfortunately not been solved in this thesis. The problem of selecting what features should be extracted is very much depending on the application. Using computer vision for the purpose of motion analysis requires a correlation between some property of the motion

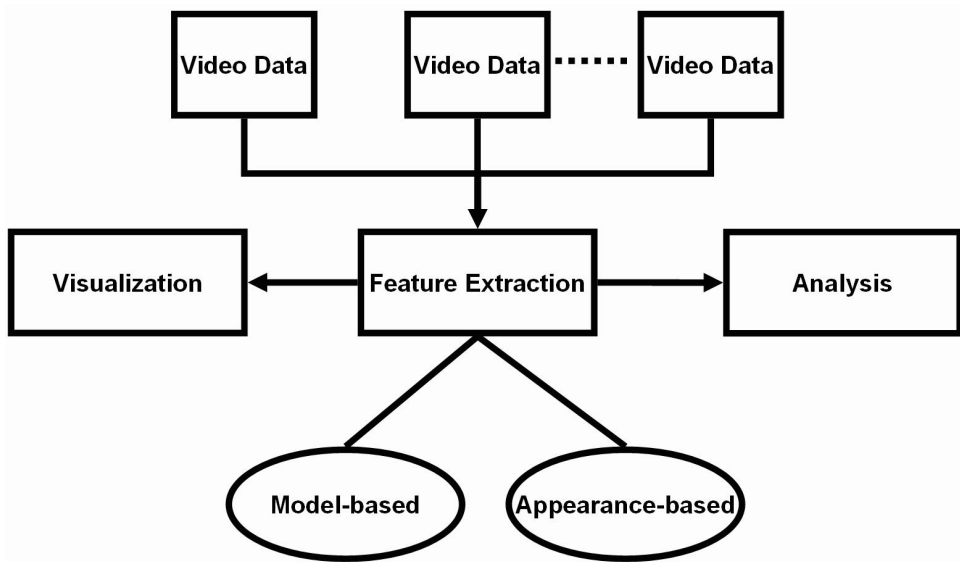


Figure 1.1: The general aspects of motion analysis and reconstruction

and a number of features in the image. For example, the objective may be to investigate the knee angle of a walking person. One solution to this problem is to locate the position of the hip, the knee and the ankle in the image. In this case, the locations of the joints in the image are the features. Designing a filter that is able to extract specific joints is difficult. However, one rather successful method is to use the contours of a person, and identify certain parts on the contour from their *shape context* (Mori and Malik, 2002). In short, the shape context is a method to identify qualitatively similar segments on two silhouettes, based on statistics on their neighborhoods. A variation of this method has also been successfully implemented by Sullivan and Carlsson (2002), where the projections of the joints of the human could be located based on the correspondence with a small set of key frame in which the joints were manually labelled.

Sometimes, the objective is not to come up with a quantitative measure (such as the knee angle), but rather to compare a number of motions. One example could be to compare a walking person to a number of walking persons in a database in order to identify who the person is. In some cases, this can be done without using a model, but rather using a set of filter responses from the raw video. For the purpose of motion analysis, it is convenient to divide the discussion into *model based* and *appearance based* methods. This may be somewhat dangerous, though, since few approaches in computer vision is totally model free.

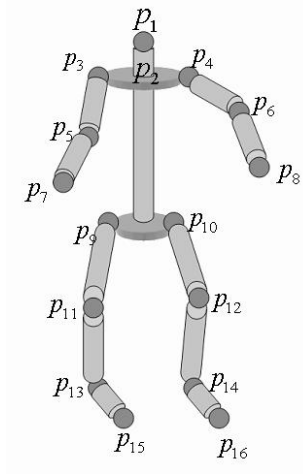


Figure 1.2: A model of the human skeleton based on 16 joints.

## Model based approaches

Most systems developed for extraction of human motion require some model of the human body. Generally, the less restrictive the models are, the more flexible will the system be in terms of capturing wide varieties of different motions. On the other hand, a very strict model is able to cope with more noise in the video, at the cost of accuracy.

The level of sophistication of the model ranges from detailed skeletons to implicit blob representations, depending on the exact objective of the system. The model can be represented as an articulated chain in 3D, where the objective is to find the configuration whose projection best complies with the image. The most natural model is a skeletal representation of the human body, such as the one shown in fig. 1.2. The limbs in such a model are usually represented as cylinders, truncated cones or ellipsoids (Hogg, 1983; Rohr, 1994; Bregler and Malik, 1998; Sminchisescu and Triggs, 2003; Sminchisescu and Triggs, 2001; Drummond and Cipolla, 2000; Eriksson and Carlsson, 2004; Sidenbladh and Black, 2002). This model is particularly useful if the motion is to be analyzed or visualized in 3D. One example of an articulated chain was presented by Liebowitz and Carlsson (2001), that uses a stick figure model to resolve metric properties of a stereo reconstruction using uncalibrated cameras. In this model, limb lengths are not required, since only the symmetry properties of the human body are exploited. Taylor (2000) used an articulated structure in order to resolve the depth in single view reconstruction, given the 2D locations of the joints. In this case, the relative limb lengths must be known. A similar approach is used by Remondino and Roditakis (2003) where a skin surface is added to the model before rendering. Herda et al. (2002) used a skeletal structure is used in order to improve the

accuracy of reconstructions obtained from optical motion capture systems.

Another approach is to use a 2D model in order to extract the primitives of the human body. If the objective is to acquire a rendered 3D reconstruction, any approach using a 2D model must find a method to map the 2D model into a 3D representation - quite the opposite to the situation where a 3D model is being used. Most 2D models are either based on silhouettes or connected patches (Wren et al., 1997; Ju et al., 1996; Ioffe and Forsyth, 2001; Sullivan and Carlsson, 2002; Mittal et al., 2003).

The skeletal model only constrain static poses of the human. Some models also exploit dynamic properties of a motion in order to track the motion over time. The dynamics can be modelled by analytically identifying possible transitions, or by using exemplar motions from a database of priors. Almost all use of dynamic models require the system to learn the dynamics from training motions. For example, the systems in (Agarwal and Triggs, 2004; Bregler, 1997) learn the dynamics of a certain motion form a set of hand labelled training data. Another approach to use implicit dynamic models is to form linear combinations of examples, in order to form a new motion that is consistent with the image data (Leventon and Freeman, 1998). An alternative method is to use a motion library in order to *select one* of the motions, and iteratively refine this motion in order to make it consistent with the image data (Park et al., 2002; Loy et al., 2004).

When a dynamic model has been learned, it is generally used in order to *track* the motion. Given a configuration of the human model in one frame, the task is to update this configuration according to the features extracted in the next video frame. In effect, tracking involves finding a configuration that complies as well as possible with the image data, while staying consequent with the priors posed on the dynamic model. It is commonplace to formulate this problem in a Bayesian framework. Given the configuration of the model  $\Theta_t$  at time  $t$ , and  $\vec{I}_t$  which is the vector of image features up to time  $t$ , the posterior distribution can be formulated as

$$P(\Theta_t | \vec{I}_t) = P(\vec{I}_t | \Theta_t) P(\Theta_t) \int_{\Theta_{t-1}} P(\Theta_t | \Theta_{t-1}) P(\Theta_{t-1} | \vec{I}_{t-1}) \quad (1.1)$$

where  $P(\vec{I}_t)$  is the likelihood function and  $P(\Theta_t)$  represents the *a priori knowledge*. The *temporal prior*  $P(\Theta_t | \Theta_{t-1})$  is usually added in order to apply the dynamics of the motion model into the inference engine. Global optimization of the posterior usually becomes very cumbersome (and therefore an interesting research issue), primarily due to the high dimensionality of the parameter space. For example, a human model of 14 limbs, each with three degrees of freedom (a rotation matrix), gives 42 parameters. In addition to this, the posterior to be maximized is very ill-behaved, with a large number of local maxima. Different configurations yield the same likelihood function, as their projections onto the image data becomes almost the same. Other sources of problems in computing the likelihood function are due to difficulties in the feature extraction, such as occlusions, motion blur, etc. Due to this rather un-collaborative search space, the posterior cannot be modelled as a uni-modal gaussian distribution, which theoretically disqualifies an MAP approach. A large number of methods how to handle such a search space have been proposed (Gavrila and Davis, 1996; Kakadiaris and Metaxas, 1996; Bregler and Malik, 1998; Wachter and

Nagel, 1999; Cham and Regh, 1999; Heap and Hogg, 1998) One common approach to this is to use variations of particle filtering and CONDENSATION (Isard and Blake, 1998), where a state space of possible configurations can be maintained and propagated over time (Sidenbladh, Black and Fleet, 2000; Sidenbladh, la Torre and Black, 2000; Deutcher et al., 2000; Cham and Regh, 1999; Heap and Hogg, 1998). Also, a hybrid Monte Carlo sampler was proposed by Choo and Fleet (2001), and was reported more efficient than point based CONDENSATION. During the past years, Sminchisescu and Triggs (2002) have reported very interesting results by new ways to find paths towards representative maxima in complicated search spaces.

### **Model free approaches**

Even though very few methods are completely model free, some of them use rather weak models (Bradski and Davis, 2002; Little and Boyd, 1998). In motion analysis, weak models are generally used for the purpose of classification. It is possible to compare two filter responses in order to classify two motions as "similar" or "dissimilar". Methods using this approach will probably not be able to answer questions along the line of "what was the difference". For classification, though, this may not be required. In (Bobick and Davis, 2001) a *Motion Energy Image* is created by superimposing several binary frames of a motion on top of each other, yielding a good signature for the motion. One important observation in this work is that recognition can be performed even from very low resolution video, which is also the case in (Efros et al., 2001). Here, the image flow of the motion is used to generate a rather discriminating signature. In (Little and Boyd, 1998) the system starts with a very weak model. However, as the shape that is being tracked (a human arm in the example), a model is created based on the physical properties of the deformed object.

### **Analysis**

There are several aspects of analysis. At one end of the spectrum are biomechanical studies, where joint angles of athletes or patients, are measured with an extreme accuracy, in order to identify minute details of a certain part of the motion. At the other end of the spectrum, we have the task of coarsely classifying certain actions, such as whether a person is walking or dancing. Of course, each of these tasks has its own set of tools and restrictions. A nice review of visual analysis systems developed up to 1999 is given in (Gavrila, 1999). In biomechanics, researchers can generally not obtain results with enough accuracy without using intrusive equipment. For seemingly simpler tasks, such as coarse classification, the object is to achieve quick results, with as little information as possible. For example, one common research problem is to identify certain activities in a regular video sequence, such as extracting all dancing scenes of a Fred Astair movie or extracting all forehand strokes from a tennis match. Another popular application based on non-intrusive motion analysis involves surveillance systems, where suspicious actions are to be identified from surveillance cameras (Mittal et al., 2003). Also, people identification, where a person can be identified based on gait is a popular research issue (BenAbdelkader and Cutler, 2002; BenAbdelkader, 2002; Little and Boyd, 1998; Lee and Grimson, 2002; Carlsson, 2000). An-



other typical application requiring analysis of the video data is sign language recognition (Holden and Owens, 2003). Recognizing sign language is a particularly difficult problem, since it involves motion of the hands, as well as fingers, which are difficult to extract from regular video.

Analysis can be carried out using appearances of the 2D sequences, or by first reconstructing the motion and perform the analysis in 3D. The later approach generally requires multiple views to be useful. However, a hot research topic is to perform 3D reconstruction using only one camera.

## **Visualization**

The purpose of analysis is to extract motion details from sensor data (generally video). In the field of visualization, we have the opposite objective - how do we render an action in a nice looking fashion. This is generally a task for animators whose artistic skills are required in order to achieve realistic motions. In order to do this, animators must be very familiar with several topics of motion analysis and kinematics in order to understand the locomotion of humans and animals. Specifically, one common method in computer graphics is to use motion capture data in order to obtain exact examples of a certain motion. These specific examples must then be modified in order to fit animated characters of different anatomy than the test subject (Gleicher, 1998; Hodgins and Pollard, 1997). Also, individual variations in the dynamics are required in order to avoid that all characters move in the same way, and to adjust a characters locomotion based on mood (Rose et al., 1998). In computer animation, the motion of a character must be modelled in a high level fashion, where the details are learned from prior knowledge about the motion (Guo and Robergé, 1996). In other words, the parametrization of the motion must be fairly low in dimensionality. Preferably, the positions of end-effectors (feet and hands) is enough. This means that the inverse kinematics must be solved in a plausible fashion, in order to have the character move naturally in a virtual environment, and also to transit smoothly between different motions (Rose et al., 2001; Lee et al., 2002; Lee and Shin, 1999).

## **1.4 Interactive marker free motion capture**

Even though using a commercial motion capture system is superior in most regards, there are many situations where a manually operated motion capture system is desired. One such situation is when recording motions of live athletic events, since athletes cannot be expected to wear special equipment during competition. Generally, in an interactive motion capture system, the sensors that automatically register the 2D positions of the markers are replaced by a human, clicking points with a mouse. In its most basic form, motion capture can be performed using only one camera. By using multiple cameras, more accurate reconstructions can be obtained. In both cases, a model of the human being reconstructed is required. However, in the multi view case, the model can be very simple.

## Single view

The most basic approach in order to achieve a 3D motion, given a single video sequence of an action, was presented by Taylor (2000). In principle, the reconstruction is achieved by modelling the human body as an articulated chain. The links of the chain correspond to limbs of the body, and the connections between the links are the human joints (elbows, knees, etc.). The task is to find the configuration of the chain that reprojects the joints back to the clicked joint positions from the image data. In this method, orthographic projection is assumed. Any camera model is of course possible; however, the camera parameters must be known beforehand, or figured out by using some auto calibration method. The algorithm itself does not provide enough information to solve any camera parameters. Assume that we are using 16 joints to model the human body. Further, we must assume to know the limb lengths of the person being reconstructed as well. In fact, by knowing the ratios of the limb lengths, a reconstruction can be obtained that is correct up to a scale factor. The human model will look like the skeleton in 1.2. In Taylor's methodology, no temporal aspects are taken into consideration. Each frame of the video is reconstructed separately. Generally, the resulting reconstruction may be a bit jerky, due to unprecise clicking by the user. This requires some postprocessing in terms of temporal smoothing, in order to make the sequence look good (if the purpose of reconstruction is visualization). The first step of the reconstruction algorithm requires the user to click (probably using the mouse) on each of the 16 joints of the person. Generally, this is a quite straight forward task, unless some points are occluded. In that case the user has to play a guessing game, and approximate the location of the occluded joint. Extensions to this algorithm could of course incorporate automatic tracking of the joints (Sullivan and Carlsson, 2002) (in this case, the problem of occlusion becomes yet more severe of course, since the guessing game is left to the computer). After the clicking, we have all joint positions in 2D,  $\{p_1, p_2, \dots, p_{16}\}$  as well as the desired distances between two joints in 3D,  $\{l_1, l_2, \dots, l_{15}\}$  as given by the limb length in the skeletal model. The depth between two neighboring joints in the articulated chain is computed by:

$$\Delta Z_{a,b} = \sqrt{l_{a,b}^2 - \|p_a - p_b\|^2} \quad (1.2)$$

Now, the reconstruction is correct up to a *binary ambiguity*. Given only the 2D data, it is generally impossible to know if a limb points towards or away from the camera. This means that, given an articulated chain of  $L$  links, and the projections of the joints, there are  $2^L$  possible solutions. An example of this phenomenon of a 3-link chain is illustrated in fig. 1.3. While some of the work in this thesis involves automatic disambiguation by using priors, we conclude for the moment that in most cases, it is a relatively easy task for the user to do this manually, by looking at the video. However, sometimes when the limbs are near parallel to the image plane, it may be a bit problematic. In fig. 1.4 one frame from a tennis sequence is shown, together with a number of possible reconstructions. All reconstructions are possible; however some of them look a bit "suspicious".

As a summary of manual single view reconstruction, we conclude that the following issues make the task relatively difficult:

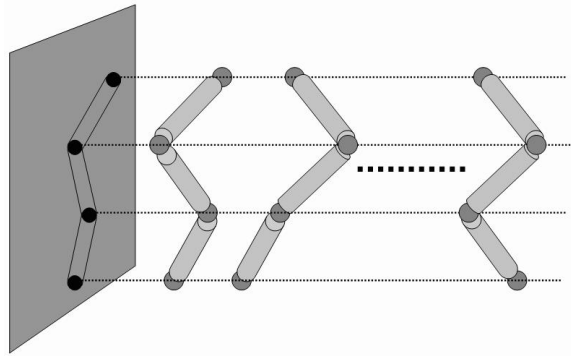


Figure 1.3: Given a projection of an articulated chain, each limb can point either towards or away from the image plane.

- **Unknown limb lengths.** In the reconstruction system developed for the experiments in this thesis, the user is allowed to test different limb lengths, as well as scales, in order to obtain a realistic and nice looking reconstruction.
- **Binary ambiguity.** Even though the user is familiar with possible human poses, this can be problematic nevertheless. It becomes particularly obvious when reconstructing complex athletic activities, such as gymnastic or pole vaulting.
- **Identify center of rotation.** Modelling the human as an articulated chain is a crude way. However, few other methods are feasible. The center of the joint is hidden inside the limbs, which means that the points selected by the user must be regarded as coarse estimates. For example, anatomically the shoulder joint is far from being a simple spherical joint. Another problematic joint is the hip joint, since the center of rotation in this case is hidden far inside the body. Of course, more advanced models can be used on the cost of simplicity.
- **Occlusion.** Occluded points must be estimated. Since small errors in joint positions can have a tremendous impact on the visual appearance of the reconstruction, this leads to significant problems in some cases.
- **Temporal smoothness.** If a sequence is to be reconstructed, it is important to maintain limb length consistency throughout the motion. This may be surprising, but in some frames one set of limb lengths appears to yield a correct solution, while in other frames, a different set is to prefer.

One important lesson to learn from this is that single view reconstruction of human motion is very difficult. Obviously, it is a difficult task for a human which means that we should give heaps of credit to systems actually achieving relatively good reconstructions automatically.

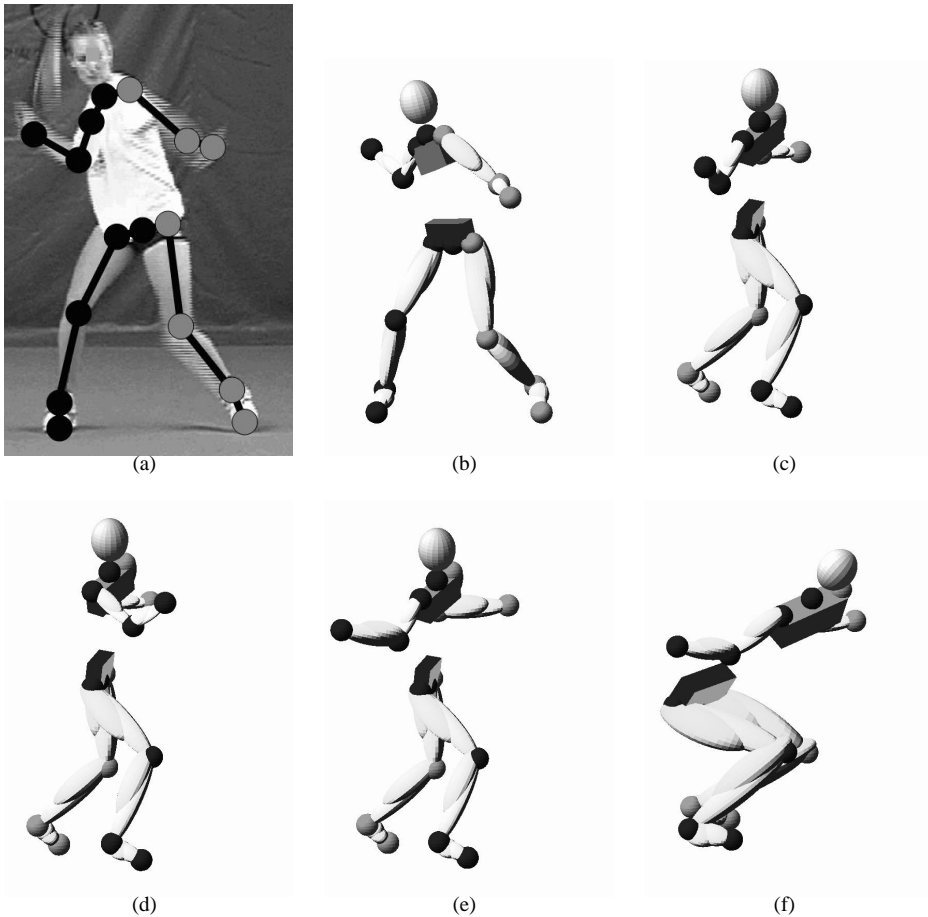


Figure 1.4: Possible reconstructions, given the projected joint centers (a). (b) shows the reconstruction from the viewpoint of the camera. (c)-(f) show possible reconstructions from a side view. In (f), the limb lengths of the model have changed.

### Multi view

By using multiple cameras, we are much better prepared to reconstruct the third dimension of a human motion sequence. The method implemented in the motion capture system developed during this thesis is the one by Liebowitz and Carlsson (2001). Their method uses two cameras, and assumes orthographic projection. As in the case of manual single view reconstruction, the user must click on the joint center locations of the person in each frame of both sequences. By using the Tomasi-Kanade factorization (Tomasi and Kanade,

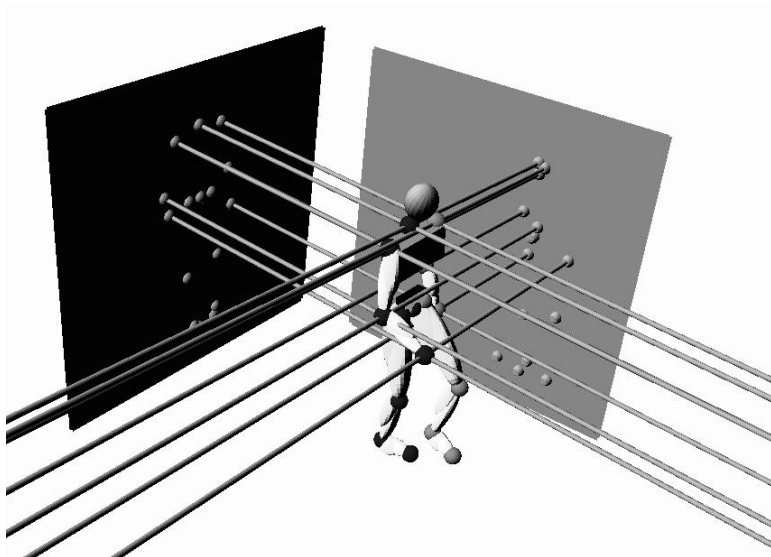


Figure 1.5: The projection of a skeleton using two orthographic cameras.

1992), a 3D reconstruction can be obtained that is correct up to an *affine transformation*. By using a simple model of the human skeletal structure, the affinely correct reconstruction can be upgraded to metric correctness. The model is very simple, and only poses two constraints:

- Symmetry. The right arm has the same length as the left.
- Constant length. Each limb has the same length throughout the sequence.

By utilizing this, no knowledge about the orientation of the cameras is required, as long as each camera can be approximated to yield a scaled orthographic projection.

### Tomasi-Kanade factorization for calibrated stereo pairs

Fig. 1.5 illustrates the situation at hand. Given the input (the projected points in the two cameras), compute the 3D structure (the walking person). In the figure, only one frame of the sequence is shown; the method, though, works by considering all joints in all frames at the same time. Each frame of each of the views provides a set of image features (joint locations)

$$X = p_1, p_2, \dots, p_n$$

where the joint locations are represented as in-homogeneous point coordinates. All frames in view  $v$  can then be concatenated into one matrix  $\mathcal{V}_v$  of size  $2 \times nF$  where  $F$

is the number of frames in the sequence. The *measurement matrix*,  $W$ , is constructed by normalizing  $\mathcal{V}_v$ , and stacking these normalized point sets of each view into one matrix. If we consider the case of two cameras, the measurement matrix will have the form

$$W = \begin{pmatrix} \mathcal{V}_1 \\ - \\ \mathcal{V}_2 \end{pmatrix} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^{nF} \\ y_1^1 & y_1^2 & \dots & y_1^{nF} \\ x_2^1 & x_2^2 & \dots & x_2^{nF} \\ y_2^1 & y_2^2 & \dots & y_2^{nF} \end{pmatrix} \quad (1.3)$$

where  $x_j^i$  indicates the x-coordinate of the  $i$ :th point in view  $j$ .

The points in the measurement matrix should, according to the model, result from orthographically projected 3D points. In other words,

$$\begin{aligned} \mathcal{V}_1 &= \mathcal{C}_1 S \\ \mathcal{V}_2 &= \mathcal{C}_2 S \end{aligned}$$

where

$$S = [ P_1 \quad P_2 \quad \dots \quad P_{nF} ]$$

is the  $3 \times nF$  *structure matrix* of 3D coordinates and  $\mathcal{C}_1$  and  $\mathcal{C}_2$  are the  $2 \times 3$  orthographic camera matrices. The camera matrices can be stacked into one *motion matrix*

$$M = \begin{pmatrix} \mathcal{C}_1 \\ - \\ \mathcal{C}_2 \end{pmatrix} \quad (1.4)$$

The measurement matrix is a product of the motion matrix and the structure matrix

$$W = MS$$

Since the product of  $M$  ( $4 \times 3$ ) and  $S$  ( $3 \times nF$ ) should have rank 3 or less,  $W$  must be rank-deficient. By applying singular value decomposition on  $W$

$$W = U\Sigma V^T \quad (1.5)$$

where  $\Sigma$  is a diagonal matrix of singular values  $\sigma_1, \sigma_2, \sigma_3, \sigma_4$  where  $\sigma_4$  should be close to zero (exactly zero with perfect data), the motion matrix is obtained by taking the first three columns of  $U$ , and the structure matrix  $S$  is obtained by taking the first three rows of  $V^T$ . Unfortunately, this process only generates a reconstruction that is correct up to an *affine transformation*. An example of such a reconstruction is shown in fig. 1.6. However, we should not panic, since the next section will show how the reconstruction can be upgraded to metric correctness.

### Metric rectification

As shown before, the measurement matrix can be written as a product of the motion matrix and the structure matrix. However, any full-rank  $3 \times 3$  matrix  $A$  can be inserted as

$$W = MA^{-1}AS \quad (1.6)$$

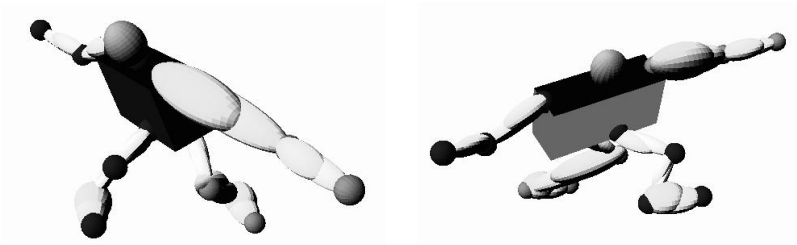


Figure 1.6: A stereo reconstruction that is correct modulus an affine transformation.

In other words, if the reconstructed structure is modified (skewed, rotated and scaled), the reconstructed cameras can compensate for that in order to yield the same measurement matrix. Intuitively, we want to identify the correction matrix, that enforces symmetry and constant limb length constraints. Additional constraints can be obtained by the fact that the camera matrices should correspond to orthographic cameras. However, this is not necessary, since enough constraints are obtained by symmetry and constant limb length assumptions. Camera constraints can be useful, though, in the case where non-stationary cameras are used. Considering only the structure matrix,  $S$ , there exists a matrix,  $A$  s.t.

$$S^m = AS \quad (1.7)$$

where  $S^m$  is the metrically correct structure matrix. Further, we are only interested in the reconstruction that is correct up to a similarity transform; i.e. we do not care about rotation. Thus, by RQ decomposition, we can extract the similarity components (an orthonormal rotation matrix) out of  $A$ :

$$A = RU \quad (1.8)$$

which leaves us with the upper triangular matrix  $U$ . The distance between two points  $S_i$  and  $S_j$  in the affinely correct structure and  $S_i^m$  and  $S_j^m$  in the metrically correct structure  $S^m$  are related as

$$(S_i - S_j)^T(S_i - S_j) = (S_i^m - S_j^m)^T U^T U (S_i^m - S_j^m) \quad (1.9)$$

Thus, every pair of distances between points known to be the same, according to symmetry and constant limb lengths, puts a constraint on the matrix  $U$ , by

$$(S_i^m - S_j^m)^T U^T U (S_i^m - S_j^m) = (S_k^m - S_l^m)^T U^T U (S_k^m - S_l^m) \quad (1.10)$$

For example,  $S_i^m$  and  $S_j^m$  can be the locations of the left elbow and wrist in one frame, while  $S_k^m$  and  $S_l^m$  correspond to the right elbow and wrist in the same frame. Since  $U$  has 5 unknowns, at least 5 constraints must be obtained. By using a larger number of constraints, the system will be over-constrained and can be solved by least-squares techniques. Fig. 1.7

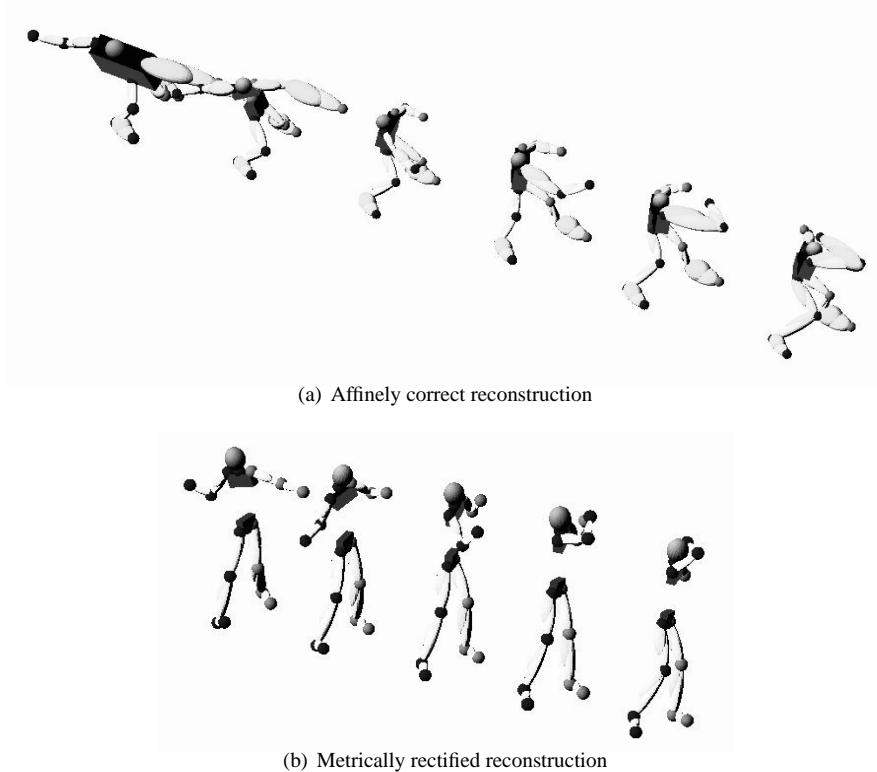


Figure 1.7: (a) shows the reconstruction of a motion sequence using Tomasi-Kanade factorization. (b) shows the same reconstruction after the metric rectification process.

shows an affinely correct reconstruction of a 5-frame sequence. Some limbs tend to change lengths significantly during the motion. In the same figure, the reconstruction after metric rectification is shown.

## 1.5 Motion recognition and classification

The last chapter of this thesis will try to put together an embryo of the automatic coaching system mentioned before. Before doing this, the problem will be theoretically analyzed, in order to understand the limitations of automatic classification of human motion. There are two important tasks in automatic motion analysis:

1. Motion classification
2. Quality assessment



The first problem deals with comparing motions in order to decide what category a certain motion belongs to. It also involves tasks such as gait recognition or sign language recognition. Generally, its purpose is to categorize a motion with respect to already known motions. As mentioned before, gait recognition has received a large amount of attention. One reason for this is probably that it is relatively easy to gather data from walking people. Also, a good solution to the gait recognition problem could be rewarding. Particularly in physical rehabilitation.

The other item is quite different, and rather ill defined. The primary targets for a system doing quality assessment are athletes and coaches. The question whether it is possible to create a system that actually performs better than a human coach in some way is interesting. There are a number of systems available today that assist coaches in analyzing motion; however, very few systems are able to give instant feedback, without any human intervention. Those systems that do generate feedback generally require the athlete to wear intrusive equipment.

Generally, there are two approaches to quality assessment. The first approach is to compare a trial to a reference (for instance of a world class performer) and define a similarity measure such that similarity to the "good" motion indicates that the trial was successful. The other method is to manually explain to the system what a "good" trial should look like, and have the system automatically rate how close the athlete was to this "modelled reference". The two approaches are quite similar, since they both must convert the input video into a representation that is useful in order to rate the requested part of the performance. When comparing two motions, it is not always necessary to compare everything. For instance in a golf stroke, the most discriminating part of the motion is done by the upper body. As we will see in the last chapter, the system must be flexible enough so the user can quickly and easily explain to it how the comparison or analysis should be carried out.

Chapter 7 will discuss how two motions can differ temporally as well as spatially. Two motions may consist of exactly similar poses, but one motion might be carried out faster than the other. This is probably the most interesting thing for an athlete to know. For example, a weight lift that is carried out fast has much greater impact on the body than a slow lift, even though the weights are the same. Generally, the *power* that an athlete is able to generate is more important than the weight on the bar.

## 1.6 Setting the stage - outline and contributions

The purpose of this thesis is to introduce a number of new concepts in the field of motion analysis and visualization. The goal has been to produce a thesis that is understandable to a large audience, while still being scientific enough to initiate a discussion among the leaders in the field. The main contributions of each chapter is described next.

## Visualization

- **Chapter 2** will go over a number of useful techniques in reconstruction of human motion using two cameras. The focus of this chapter is how to handle the problem of erroneous data in stereo reconstruction. There are methods to obtain 3D reconstruction given certain feature points. The chapter will illustrate that epipolar constraints have to be incorporated into a reconstruction system.
- **Chapter 3** will introduce the concept of qualitative constraints in motion reconstruction. The main ideas will be explained using 2D shapes, before demonstrating how the method extends to 3D shapes, and eventually to single view reconstruction. Simplified, the task dealt with in chapter 3 is to compute a shape (human motion for example) given noisy data. The general method to solve this is to use a large set of training shapes that are used as examples, in order to compute what the perturbed shape should look like. This generally means that the result will be biased towards the training data. The contribution of chapter 3 is to show how to use a qualitative measure in order to obtain less biased reconstructions. Further, the proposed method works even when the number of training shapes is small.
- **Chapter 4** explains how to complement existing techniques in tracking and single view reconstruction with the use of *3D key frames* in order to obtain good looking reconstructions. Systems that perform automatic reconstruction of single view data generally fail after a few seconds of tracking. Chapter 4 will demonstrate that by adding a small amount of manual work to the tracking and reconstruction process, significantly longer sequences can be tracked and reconstructed. The added amount of work consists of manually constructing a small number of 3D poses. A reconstruction of a 36 seconds long tennis sequence is presented.

## Analysis

- **Chapter 5** can be regarded as a continuation of the theory introduced in chapter 3. It will be shown how a qualitative measure can be more effective than a euclidian measure in order to rate similarity between human motions. The discussion is rather theoretical, but the chapter will demonstrate a full single view reconstruction algorithm. The main difference from previous approaches is the concept of comparing similarity of two motions using the qualitative measure.
- **Chapter 6** will use motion capture data in order to explore how much information is lost when going from three down to two dimensions. While much research has been done in order to compare different motions in 2D, little is known how well such a comparison is valid in the original 3D motion. In other words, if the projections of two point sets are similar, how similar are the original 3D point sets? The chapter also addresses a very important aspect of monocular reconstruction. Given 2D video data, is it possible to gain any information about the original 3D motions by first generating a monocular 3D reconstruction, or is it more accurate to compare the

motions in 2D? By investigating this, an understanding of under what conditions monocular reconstruction can actually help classification is presented.

- **Chapter 7** will wrap up the thesis, by going back to the discussion about the "virtual coach". How useful will computer vision techniques be in order to automatically assess the quality of a motion? The chapter deals with philosophical issues as well as presenting useful results, based on methods from previous chapters. The chapter should be regarded as the most important chapter in terms of future directions of research. There is very little research in computer vision that addresses the problem of quality assessment from a practical point of view of a coach or athlete. The purpose of this chapter is to change this, and bring up the end user to the agenda.



## Chapter 2

# Epipolar geometry constraints in multiple view motion tracking

Recovering the 3D structure of human motion is an intensively explored problem. Generally, the problem involves identifying the 3D coordinates of a number of representative joints of the body at each time step throughout the motion. By modelling the body as an articulated chain where the joints are connected by limbs of fixed lengths, a 3D skeleton of the motion can be computed. Fortunately, there exists many well behaved methods to determine the 3D coordinates of a point in space, given its 2D projections in a number of sequences taken from different viewpoints. How many viewpoints are required to obtain a good reconstruction? This of course depends on what is meant by "good". In theory, two calibrated cameras, forming a stereo pair, yield enough information in order to recover the 3D motion. This requires each joint to be visible in each frame of the sequence in both cameras, which is generally not the case. As mentioned from the introduction, the most accurate method to obtain a 3D representation of human motion is to use motion capture systems. By using only video sequences of athletes not wearing any special equipment, the problem is much more challenging. However, relatively good solutions to this problem are potentially very rewarding, since they can recover 3D motions of persons under more realistic settings than in a laboratory. One particularly attractive application is to reconstruct athletic motion in a competition setting. Even though a pure camera based 3D reconstruction (with a limited number of cameras) cannot be expected to yield solutions accurate enough for any finer level of biomechanical research, they may be interesting for a coach or an athlete. These issues will be covered in chapter 7.

This chapter will extend the discussion in the introduction about Tomasi-Kanade factorization followed by metric rectification. The purpose is to give some intuition about how to handle non-perfect data, and how to estimate data that is lost, due to the inherent problems of computer vision. Obviously, given perfect input data, the algorithm will always deliver an exact reconstruction. In the case of automatic tracking, perfect input data is of course nothing but an unrealistic desire, which means that reconstruction errors will inevitably occur. This chapter illustrates how fundamental epipolar constraints can be used in or-

der to identify potential tracking errors. Similar constraints, together with the rigid link properties of an articulated structure, are shown to be useful in order to "fill in" missing data. By doing this in an intelligent fashion, a visually plausible reconstruction is obtained even when the input data can not be fully trusted. Such a reconstruction is of course not likely to be useful for accurate biomechanical analysis, but may lead to a visually plausible animation.

## 2.1 Software system

During the course of this thesis, a software package has been developed in order to illustrate the research issues of human motion capture. Down the road, it may also turn out useful for animators, researchers in biomechanics, game developers and sports broadcasters. One module of the system is designed for stereo reconstruction of un-calibrated cameras. Further, this module is designed for two different purposes:

- Reconstruction of manually marked data.
- Reconstruction of automatically tracked data.

In principle, the two versions don't differ much. The main difference is that the automatically tracked data is much more contaminated with noise. Reconstruction therefore requires more attention in terms of handling suspicious input.

## 2.2 Manual reconstruction

In its most trivial form, the software package developed for 3D reconstruction consist of a GUI in which the user manually marks the positions of the joints of the human skeleton. When the entire sequence has been marked in both views, the Tomasi-Kanade factorization algorithm creates a reconstruction that is correct up to an affine transformation. The metric rectification of (Liebowitz and Carlsson, 2001) is then applied in order to obtain a metrically correct reconstruction. A snapshot of the software is shown in fig. 2.1. The entire motion sequence is regarded as one point cloud, where the joint locations of each individual frame are stacked on top of each other. In order for this to be approximated by an affine camera, the size of the point cloud has to be small compared to the distance to the cameras. If the camera is moving (panning, tilting or zooming) a special case of the algorithm has to be applied. For the functionality of the software system presented in this thesis, stationary non-zooming cameras are sufficient.

## 2.3 Missing data

In the perfect world, each joint is clearly visible and identifiable in every frame of the sequence. Unfortunately, perfect worlds are quite rare. For the 3D reconstruction system to be practically useful, a method of dealing with contaminated data is implemented. As in most vision systems, the primary sources of errors are:

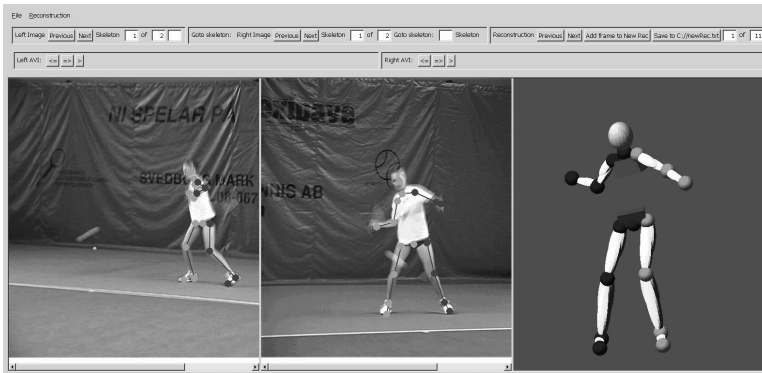


Figure 2.1: Snapshot of the reconstruction module. The user manually marks the feature points in each frame of the sequence from two views. A 3D reconstruction is displayed in the rightmost window.

- **Self occlusion.** Some joints are occluded by the rest of the body.
- **Image blur.** Some motions are very rapid (for instance a golf stroke, or a hard tennis stroke), and a typical 50 Hz camera will cause severe blur.
- **Loose clothing.** It is important to keep in mind that when reconstructing an articulated chain (in this case the human body), it is crucial to identify the center of rotation of each joint. Such a point is of course hidden inside the human body, and always has to be approximated, even if tight clothes are worn. If the subject is not wearing tight fitting clothes, this deviation from the center of rotation gets yet more problematic.

Fig. 2.2 illustrates these sources of errors. A robust system has to deal with the fact that some of the locations have to be estimated. When the system is executed in manual mode, the best method to estimate joint centers of occluded points, or points in frames suffering from severe motion blur, is to simply leave the guessing to the user. In order to assist the user, the system can provide cubic splines of the point trajectories of each joint, in order to facilitate for interpolation of invisible points. There are of course other methods to estimate joint locations based on the structural constraints of the motion. This will require a set of training motions from where the constraints can be learned. Issues related to this are discussed in the next chapter. For this module of the reconstruction system, the generic assumption of smooth trajectories is provided as help to the user.

## 2.4 Automatically tracked data

Manual reconstruction of 3D structure is very useful in order to obtain correct 3D data from live events. One appealing feature would be to obtain the reconstruction very soon after the

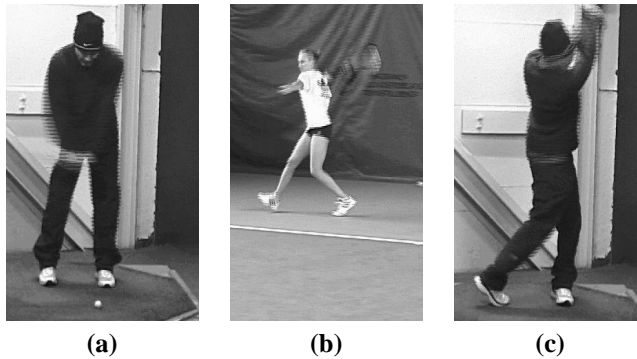


Figure 2.2: Examples of problematic frames. In (a) the image blur makes it very difficult to accurately localize the hands of the player. In (b), the right side of the woman is mostly occluded, making it difficult to identify the right hand and elbow. In (c), the loose clothes make it difficult to identify for instance the hip positions, even though all the joints of the golfer are visible.

action has taken place. This could be used to show a virtual replay of a point in tennis, right after the point has been finished. In such an application, manual selection of feature points is disqualified, since it simply takes too long to finish. Due to this, some automatic tracker of the joints is required. For the software system presented here, the tracking system is based on the work by (Sullivan and Carlsson, 2002). The tracker does sometimes deliver inaccurate joint positions, due to the regular problems of computer vision. Specifically, the sources of errors are the same as the ones listed in section 2.2. This means that the system performing reconstruction of automatically tracked data must be able to:

- Identify undecidable points.
- Identify tracking errors.
- Correct erroneously tracked data.
- Insert undecidable points.

## 2.5 Identification of undecidable points

In some cases, the tracking algorithm can itself identify when a position of a point is impossible to compute. This is a built-in feature in the tracking algorithm. During preprocessing, the tracking algorithm has learned the joint locations in a number of key frames of the sequence. For each individual frame, a distance measure is computed to every key frame in order to classify the frame. When the frame is classified, each individual feature point is transferred from the key frame to the actual frame, according to a local similarity



measure. If this local similarity measure is above a threshold, that point is considered to be *undecidable* in that frame. Generally, this occurs when a point is occluded, as in the case the middle picture of fig. 2.2, where the right hand and elbow are occluded.

### Identification of tracking errors

The joint locations delivered always contain a certain amount of noise. Generally, the amount is small. Sometimes, though, severe tracking errors occurs. These errors must be identified, which is typically achieved by some strategy of outlier detection. This system uses two main indicators of classifying a tracked joint location as an outlier:

1. Residual in the affine reconstruction.
2. Residual in the metric rectification.

The measurement matrix consist of the joint locations of the motion sequence from two different views. The Tomasi-Kanade factorization then decomposes the measurement matrix into one possible structure matrix and one possible motion matrix.

$$W = MS \quad (2.1)$$

where

$$S = \begin{bmatrix} X_1 & X_2 & \dots & X_n \\ Y_1 & Y_2 & \dots & Y_n \\ Z_1 & Z_2 & \dots & Z_n \end{bmatrix} \quad (2.2)$$

are the 3D coordinates of the affine reconstruction and

$$M = \begin{bmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \end{bmatrix} \quad (2.3)$$

is the motion matrix encoding the two  $2 \times 3$  affine cameras  $\mathcal{C}_1$  and  $\mathcal{C}_2$ . Thus, the measurement matrix,  $W$  should be a rank 3 matrix. Given two affine cameras, the fundamental matrix (F-matrix) can now be computed. The F-matrix under orthographic projection has the form

$$F = \begin{bmatrix} 0 & 0 & c \\ 0 & 0 & d \\ a & b & 0 \end{bmatrix} \quad (2.4)$$

and relates two orthographically projected points as

$$x^T F x' = 0 \quad (2.5)$$

where  $x$  and  $x'$  are the corresponding 2D points in the two sequences in homogeneous representation. The validity of each corresponding point pair, as reported by the tracking algorithm, can now be estimated by the residual of equation 2.5. Since we expect a relatively small number of outliers, a point pair  $x, x'$  is classified as erroneous iff  $x^T F x' > t$

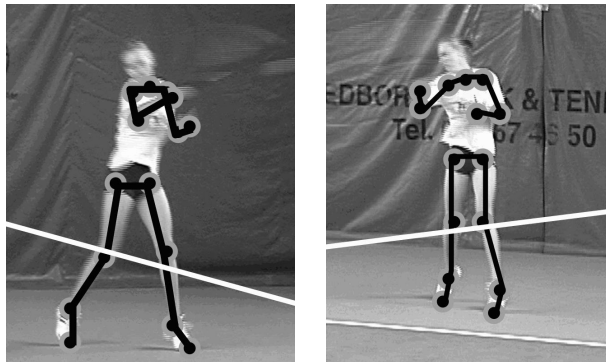


Figure 2.3: The epipolar constraints dictate the lines on which each joint should reside, with respect to the corresponding point in the other image. In this example, the right knee is fairly well tracked, since the point resides near the epipolar line in both images.

where  $t$  is a manually set threshold. If we expect a large number of outliers, a RANSAC classification could be used. Geometrically, this residual is represented as the sum of distances of the two points from the epipolar line in the two images, as illustrated in fig. 2.3. Note that it is very unlikely that both points in a correspondence are erroneously tracked. Nevertheless, from this classification alone, it is impossible to decide in which of the two sequences the tracking has gone wrong. It is of course possible to use a number of priors in order to decide this; a discussion that will be postponed to the next chapter, where the use of prior information will be deeply discussed. This classification alone suffers from one major flaw: It only detects errors as long as the direction of the error is perpendicular to the epipolar line. A severe tracking error can thus still yield a perfectly consistent epipolar geometry under orthographic projection. Such a situation is shown in fig. 2.4, where the right knee in the left image has drifted along the epipolar line, and is thus classified as correctly tracked.

Even if the direction of the error is along the epipolar line, it is still possible to catch the error. This is accomplished in the next step of the reconstruction algorithm, where the affinely correct reconstruction is upgraded to metric correctness. As explained in the introduction, such a rectification involves identifying the  $3 \times 3$  rectifying matrix  $H$ , that will enforce the symmetry constraints of a human, and also enforce the constant limb length constraints (limbs don't grow along the motion). Every pair of segments that are supposed to have the same length will pose a constraint on this rectifying matrix. When  $H$  has been identified, and the affine structure is upgraded to metric, we can look at the resulting limb lengths, and conclude that limbs of significantly deviating lengths has one or two endpoints that are erroneously tracked in one of the sequences.

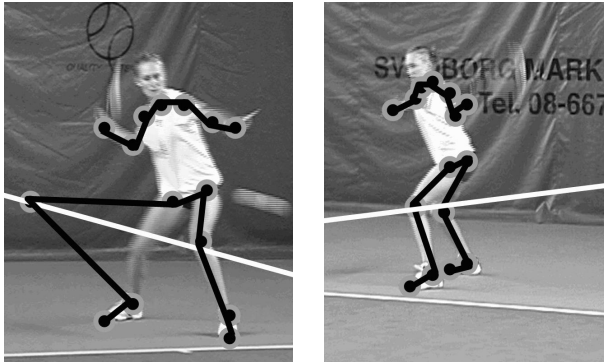


Figure 2.4: In this case, the right knee is wrong in the left image. Nevertheless, this measurement matrix will yield an affinely consistent solution since the erroneous point has drifted along the epipolar line.

## 2.6 Estimating uncertain points

This far the discussion has focused on some commonly used methods to identify (or at least make educated guesses) when something has gone wrong in the tracking. The question then arises: How should the errors be corrected? The answer depends on the purpose of the reconstruction, and how severe the errors are. If the tracking is performed at a relatively high frame rate, and the errors are rare, the best approach to fill in gaps in the joint trajectories is to use a cubic spline. On the other hand, if errors occur frequently, and the frame rate is rather low, there are a number of other tricks to consider.

As the human body is represented as an articulated chain, there are many constraints that can be used in order to insert a missing point. What constraints can be used depends on the confidence of the neighboring joints in the chain. Another factor is whether it is possible to identify in which view the tracking has failed. In this system, this is generally possible if the tracking module has reported an error. However, if the error was caught by in-consequences in the epipolar geometry, this is generally difficult. The general assumption when estimating missing locations is that the limb lengths have been computed with a plausible accuracy. Depending on the location of the joint in the skeletal hierarchy, a number of observations can be made. The most important observations are illustrated below.

### Last joint in a chain

If the last joint in a chain is missing (for example the wrist) and the location of its neighbor joint (elbow in this case) is found plausibly, the length of the limb connecting these joints (forearm) will constrain the location of the missing joint to the surface of a sphere centered

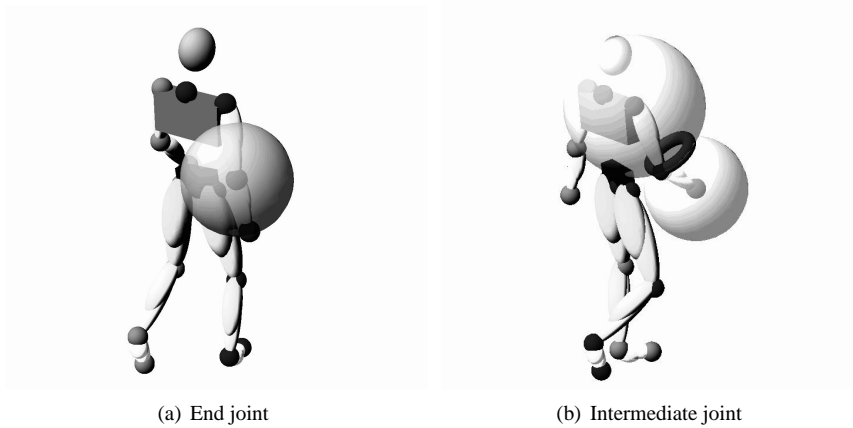


Figure 2.5: Structural constraints posed by the articulated chain structure. (a) illustrates the constraining sphere in the case the right hand would be missing. The sphere is centered at the elbow, and the radius is set to the length of the forearm. (b) illustrates the same thing for an intermediate joint - in this case the elbow. One constraining sphere is centered at the right shoulder, and one is centered at the right hand. The dark circle shows the intersection of these two spherical surfaces; the circle on which the elbow should reside.

at the neighboring joint, with a radius equal to the length of the limb. This is illustrated in fig. 2.5(a).

### Intermediate joint

If the missing joint is not an end joint, the problem can be constrained further. In this case, the missing joint has two neighbors, each constraining the location to the surface of a sphere, indicating that the missing joint must reside on the circle corresponding to the intersection of the two spheres, as illustrated in fig. 2.5(b). This of course assumes that both neighbors have been located plausibly.

### Temporal constraint

If the missing joint has been located correctly in the *temporal proximity*, it is possible to constrain the joint location further, by selecting the location on the constraining sphere (or circle) that best fits into the temporal constraints. In principle, this can be achieved by performing a cubic spline interpolation of the missing data, followed by enforcing the limb length constraints. This situation is illustrated in fig. 2.6

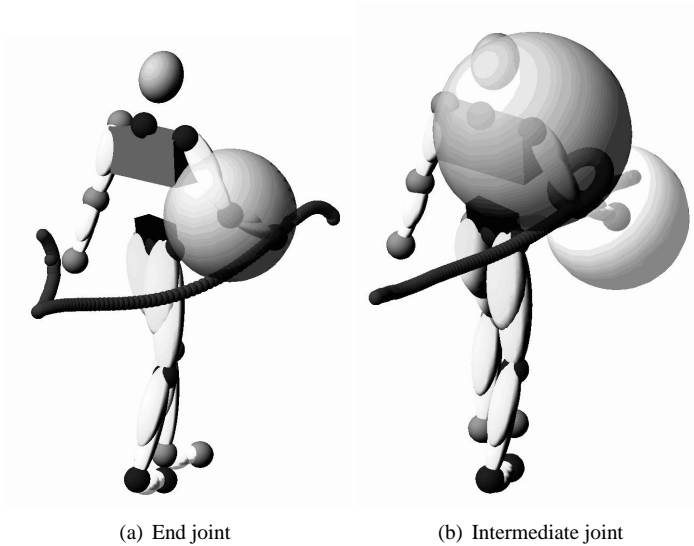


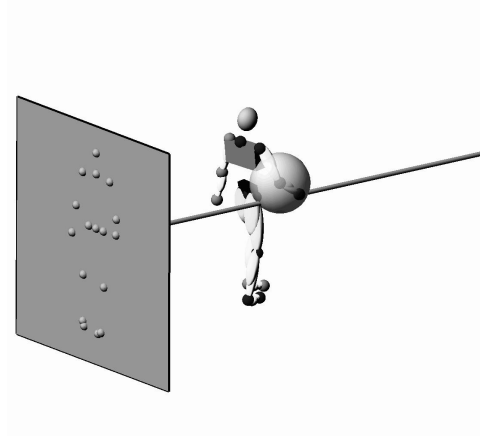
Figure 2.6: Temporal constraints. These figures illustrate how the temporal history of a joint can be used in order to further constrain the search area if the joint is missing. In (a) the constraining trajectories for the right hand is shown. (b) shows the same situation for the elbow - an intermediate joint further constrained by two spherical surfaces.

### Single view constraints

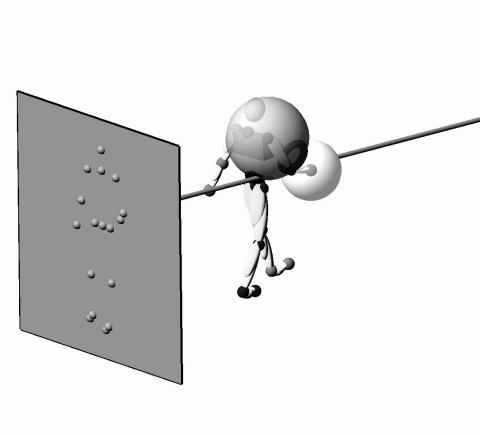
Under some circumstances, it is possible to identify in which of the views the tracking has failed. This means that the location of the missing joint is reduced to a projection line in space, given by the joint location in the correctly tracked view. As this line will generally intersect the constraining sphere(s) in two points given from the earlier constraints, there are only two locations to choose from. This situation is illustrated in fig. 2.7.

## 2.7 Reconstruction example

Fig. 2.8 shows an example of a reconstruction. The points used in this case were obtained from the automatic tracking system. During the course of the tracking, some of the joints were missing, due to occlusion. These errors were reported from the tracking module, indicating that the single view constraints could be used. Further, some errors were trapped by violations to the epipolar geometry; for example in frame 5 (the 5:th pose from the right), the left leg and right arm display suspicious behavior. By using some of the methods described previously, the final reconstruction turned out quite similar to the reconstruction obtained from hand clicked data.



(a) End joint



(b) Intermediate joint

Figure 2.7: Single view constraints. These figures illustrate the situation when a joint has been localized in one view, but not in the other. This yields a constraining line in space (assuming orthographic projection). This line intersects the constraining spheres in at most two points, giving very strong constraints.

## 2.8 Chapter summary

A number of methods in order to handle the problem of missing data has been demonstrated in this chapter. The methods should be regarded as a toolkit, and a complete algorithm of how to use these methods has intentionally been left out. Instead, a heuristic for how to go about the problem has been presented. The main goal of a reconstruction system is for

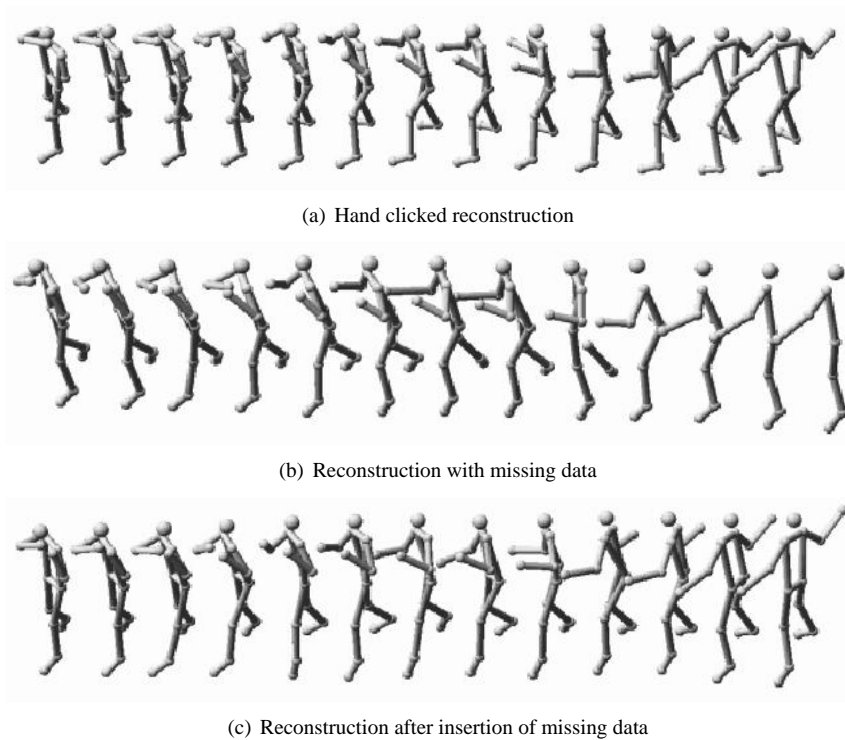


Figure 2.8: Example of how missing points can be filled in, using the methods described in this chapter. **(a)** shows the "correct" reconstruction obtained from hand clicked data. **(b)** shows the resulting reconstruction based on the automatically tracked data. In **(c)** the reconstruction is cleaned up, by filling in missing points. Also, the right forearm looks a bit peculiar in reconstruction **(b)**; something that has been corrected in the final reconstruction.

it to be fully automatic. In practice, however, a system performing multi view reconstruction needs to be manually adaptable during operation. Too high confidence in automatic recovery methods will generally lead to frequent errors. However, by allowing the system to occasionally interrupt execution and wait for human guidance, the result will be significantly better. Also, a successful system should be able to learn from the human guidance, and learn typical constraints during execution. The problem of multi view reconstruction is mathematically solved. However, the software issues for how a user friendly system implementing these methods still remains to be solved. The contents of this chapter should be regarded as an embryo of such a system, where the intention is to optimize the trade-off between manual intervention and fast results.





## Chapter 3

# Monocular reconstruction - prior constraints

Reconstructing 3D human motion from a single view is an ill-posed problem. It has earlier been shown that given a video sequence, there is an infinite number of reconstructions that would yield a zero-residual with respect to the video. The problem is yet more difficult if no markers are attached to the person performing the motion, since this inevitably implies that tracking errors will occur. Given this rather pessimistic overview, how could it be that researchers, like Sminchisescu and Triggs (2003), have succeeded in obtaining very nice reconstructions which give a very strong impression of being correct? The answer lies in the use of *prior information*.

### 3.1 What is a prior?

Without prior knowledge, there is little we could accomplish in life. An artist cannot make a painting of a black knight riding into the sunset, unless he or she has some sort of concept about what a horse looks like. Or a knight. Not to mention the sunset. In fact, most of the things we see, we have *almost* seen before. Why are we still impressed by the painting? We already know what a knight, a horse and a sunset look like. Probably because the painter was able to capture subtle details of the knight that differ slightly from the template we have stored in our minds. In other words, it appears to be much more efficient to capture differences from known standards, than to capture an entire scene from scratch. In order to learn something, you almost have to know it already! One answer to the question can thus be that a prior is something that we already know. We don't need to read the manual every time we drive a new car. All we have to do is identifying the differences from the car we usually drive, and pray that our prior assumptions hold. For example, the car will probably turn left if we turn the steering wheel left, etc. Using priors carelessly, can of course be devastating, for example if we use our car driving priors when taking over a Boeing 767. Generally, humans are very good at judging when to use priors, and how to weight their relevance to the situation. Computers on the other hand

are a bit less flexible. This chapter deals with a number of approaches to have a computer incorporate prior information. The target of the discussion will be human motion, and their 3D reconstructions. Before getting there, other shapes will be investigated as well, in order to present some new and very promising concepts. Most of the methods in this chapter can be put in a Bayesian formulation, which is commonplace when it comes to combining priors with new data. Therefore, we start our journey through the world of priors by reviewing the foundation of Bayesian inference.

### 3.2 Bayesian inference

One of the most characteristic properties of a computer vision algorithm is its tendency to occasionally deliver contaminated results. The completely noise-free vision system has yet to be developed. This is one reason why Bayesian inference is a frequently used methodology in computer vision problems today (Bowden, 2000; Brand, 1999; Bregler, 1997; Bregler and Malik, 1998; Leventon and Freeman, 1998; Pavlovic and Rehg, 2000). Providing a theoretically well founded approach for recognition, restoration and reconstruction, it is a technically sound tool for vision algorithms. The use of priors allows the computer to always come up with a result, no matter how bad the input data is. For extremely poor data, reliance on priors becomes essential. One example is the system developed in this thesis, where a 3D reconstruction is obtained from monocular video. The basic situation of this is depicted in fig. 3.1. The key issue is of course how much the algorithm should listen to the data, and how much it should rely on the priors. In the case of monocular reconstruction, the entire concept becomes yet more intriguing, since we need to use the priors even though the input 2D data is perfect, since the depth ambiguity can never be resolved without any knowledge about the human structure, or about the dynamics of the motion being reconstructed. In its most abstract form, Bayesian inference involves finding the most likely event, given a certain set of observations. For example, we may want to find the most likely outcome,  $X$ , given the, potentially noisy, observation  $X'$ . This is done by maximizing the left hand side (the *posterior*) of the equation:

$$P(X|X') = P(X'|X)P(X) \quad (3.1)$$

$P(X'|X)$  is the *likelihood* function, and  $P(X)$  is the *prior* - or the probability distribution of an outcome given no observation. One of the main arguments against Bayesian approaches is the subjectiveness of the prior. Rather than disputing this, it is more fruitful to admit that designing a plausible model for the a-priori distribution involves some creative work. The remainder of this chapter describes a new approach of dealing with training data in order to design a prior distribution.

#### Two faces of a prior

The most common way to design a prior model is to use a set of training data, exemplifying the process to be modelled. Given such a training set, some decisions have to be made about the properties of the model. Decisions such as if a polynomial of a certain degree

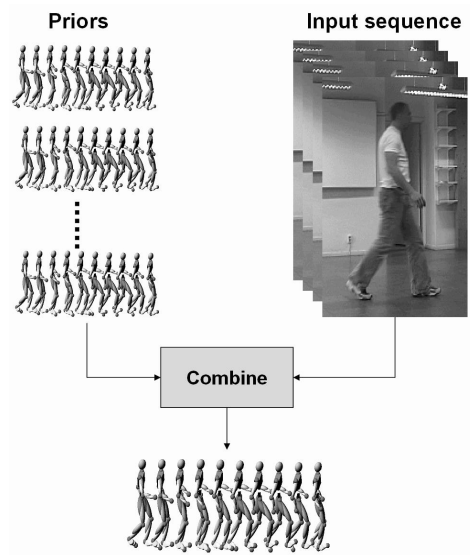


Figure 3.1: The principle of how a Bayesian inference engine combines the input data with prior information.

should be used for regression. This can be regarded as *meta modelling*, since it involves deciding from a category of models. After such a choice has been made, we have implicitly made a *qualitative assumption about the process*. The next step is to decide the *parameters* of the model (e.g. the polynomial of selected degree). This can be regarded as designing the *quantitative* properties of the prior. In practice, it turns out that the probability density functions (quantitative properties) for the model are learnt from the training set, while the qualitative choices about the model are based purely on heuristic grounds. This means that if the training data is insufficient, the solution after regression with the learnt parameters may be very biased toward the training data. In some situations, a less biased result would be obtained if the *qualitative* as well as the *quantitative* part could be learned from the training data.

### Inference with non-linear priors

The model of the prior information about a process to be estimated from contaminated or insufficient data, can be thought of as a manifold in a parameter space. All realizations of the process are represented by a point on this manifold. As an example, suppose we want to estimate the shape of a planar curve from a set of noisy points. Ideally we should have the prior probability distribution of the curve together with the distribution of the noise in order to formulate the problem as that of Bayesian estimation. A qualitative characterization of priors on the other hand could just mean that we are considering the class of convex curves.

If we let the curve be represented by the set of points  $p_1 \dots p_n$  for which we have noisy measurements,  $p'_1 \dots p'_n$ , the Bayesian inference problem can be formulated as that of finding the convex set  $p_1 \dots p_n$  that maximizes the posterior

$$P(p'_1 \dots p'_n | p_1 \dots p_n) \quad (3.2)$$

In fact, this is the likelihood function, but since all convex shapes are considered equally likely with respect to the prior, the likelihood function and the posterior are proportional. The constraint manifold in this case constitutes points in  $\mathbb{R}^{2n}$ , representing the convex sets  $p_1 \dots p_n$ . Fig. 3.2 shows an example of fitting a convex set of points to a noisy set using the sum of squared errors as the posterior, i.e. least squares fitting. Convexity of a point

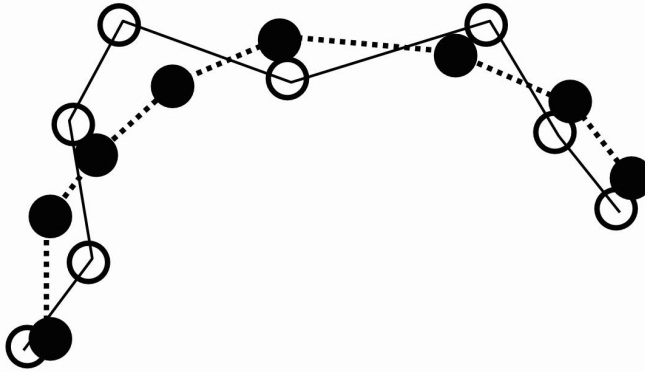


Figure 3.2: Least squares regression of a point set with convexity constraint. The solid points correspond to the curve after regression.

set can be expressed algebraically by the determinant conditions:

$$\begin{vmatrix} p_i & p_j & p_k \\ 1 & 1 & 1 \end{vmatrix} > 0 \quad (3.3)$$

for all triplets of ordered points  $i < j < k$ . The sign of the determinant measures the orientation of the point triplet. For the solution of problems like these we are therefore faced with the problem of optimizing non-linear functions with non-linear inequality constraints. Problems like this, defined over sets of points  $p_i$  in space and time, will be considered. The constraint manifold  $\mathcal{M}$  will generally be a semi algebraic set consisting of combinations of equality and inequality constraints.

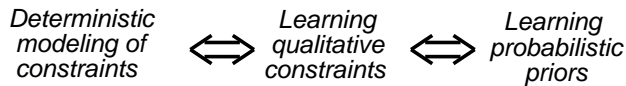
$$\mathcal{M} = \mathcal{M}_1 \mathcal{M}_2 \quad (3.4)$$

The structure of the prior probability density on the manifold  $\mathcal{M}$  is generally not considered. From a Bayesian point of view this is assumed to be equally distributed and the

constraint manifold defines the set where the prior pdf is non-zero. Given that we have the measurements  $p'_1 \dots p'_n$ , the constrained optimization problem can be solved by maximizing the posterior:

$$\max_{p_1 \dots p_n \in \mathcal{M}} P(p'_1 \dots p'_n | p_1 \dots p_n) \quad (3.5)$$

The basic idea can be seen as that of identifying completely impossible events and prevent these from showing up in the results. This is a very weak characterization of the prior information compared to a complete estimation of the pdf and it will consequently allow the measured data more weight in the final estimate. No matter how poor our data are however, we will never get unrealistic results in the final estimation provided we have accurately estimated the constraint manifold. The approach can be seen as an intermediate between the extremes of classical deterministic modelling and the complete unconstrained probabilistic modelling. The constraints are modelled by choosing appropriate classes of



constraint functions but for very complex classes a learning procedure is used in order to detect the appropriate subset of the constraints that actually characterize the manifold.

## Related work

Bayesian inference has been a common tool in reconstruction of human motion. For example, Leventon and Freeman (1998) used a set of motion capture data to form a set of basis function for the Bayesian prior. Bowden (2000) showed how to use non-linear models of deformation in order to allow for non-linear point distribution models. Pioneering work in defining dynamic manifolds for human motion was done by Brand (1999) who reconstructed human body poses from 2D shadows, assuming the correct reconstruction to reside on a manifold in configuration space. Gomes and Mojsilovic (2002) presented a variational approach to define a manifold, given a set of sample points. For the priors discussed in this chapter, the dynamic manifold is defined in terms of qualitative properties of point sets. Aichholzer et al. (2002) presented the concept of qualitatively equivalent point sets, which is a corner stone for the ideas in this chapter. When dealing with 3D human motion, the entire motion sequence can be regarded as a shape in space-time. Physics has shown to provide nice constraints on such shapes (Witkin and Kass, 1988; Ngo and Marks, 1993). In extension to this, when the dynamics of a certain motion has been learned, methods have been developed to enforce these constraints onto new characters. This is generally referred to as re-targeting of motion (Gleicher, 1998), and is an important problem for animators.

### 3.3 Learning the constraint manifold - carving priors

In the previous example, the manifold was relatively easy to formulate; defining the set of convex curves does not even require any training data - the term *convex* is well-defined. As more complex and non-intuitive constraints are added, the procedure of defining the characteristics of the training data becomes much more intriguing. The goal is to have the system automatically learn all constraints by itself from a set of examples. This procedure - defining a manifold given a few examples - resembles the act of a sculptor carving out a statue in a rock. The final product should capture the essence of the training data and nothing more. If the manifold is too broad, regression may end up with a solution too far away from the training set. If the manifold is too narrow, the solution will be severely biased towards the training data. Carving in a rock is a tough problem. Learning constraints from samples is only slightly easier. Therefore we need to provide the system with well-defined rules, before the artistic work may begin.

#### Shape representation

All shapes and motion in this thesis are modelled as point sets. The coming discussions is no exception. When dealing with human motion, the body will again be represented by the normal skeletal model, as shown in fig. 1.2. The locations of 16 joints define a pose, in each frame  $f$  and is represented in the matrix

$$X_f = [p_1 p_2 \dots p_{16}]^T$$

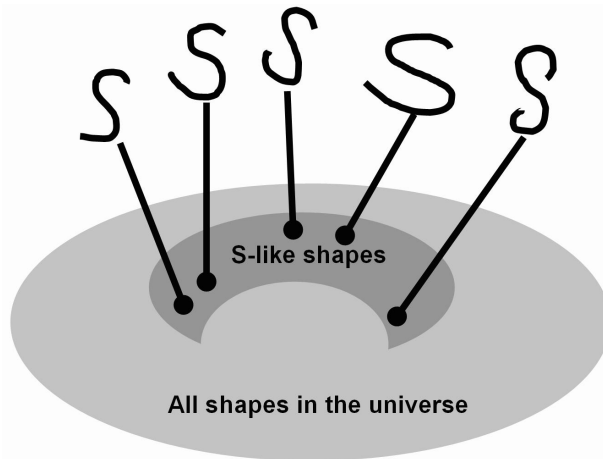
where  $p_n \in \mathbb{R}^2$  for 2D points and  $p_n \in \mathbb{R}^3$  for 3D points. A motion sequence will be represented by stacking the points of each pose on top of each other into one tall matrix. Thus, the sequence  $X_{1:N} = (X_1, X_2, \dots, X_N)$  is represented as

$$X_{1:N} = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_N \end{bmatrix}$$

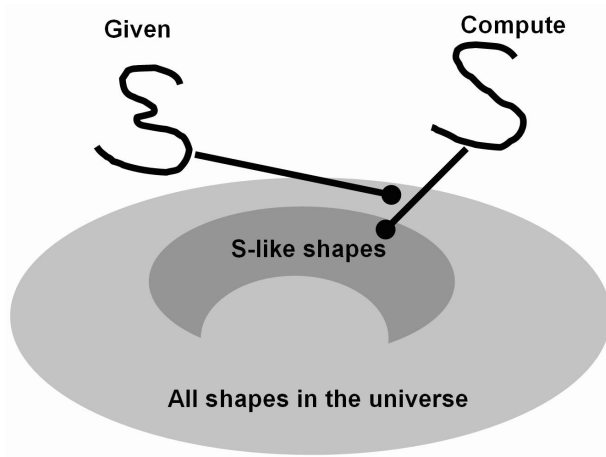
During the discussion about qualitative constraints, two dimensional curves will be used in addition to human motion. These curves are generally hand drawn shapes (letters and symbols), and are also represented as point sets. The point set is obtained by sampling uniformly along the curve. The number of sample points may vary from case to case.

#### Shape categories

From section 3.2 we understand that the objective is to, given a contaminated shape, find the nearest shape residing on the manifold of the shape category. In fig. 3.3(a) a theoretical manifold of the category of s-like-shapes is shown. This manifold has been generated from 5 examples (how this is done will be revealed soon). In (b), an example of how a correct s-like-shape is found, given a perturbed shape. More generally, the point set



(a)



(b)

Figure 3.3: In (a) the manifold generated from 5 samples of the letter "S" is shown. In (b), an example of how a shape from the manifold is found, given a noisy version of the letter "S".

$X$  will belong to a specific class which has a certain variation within it. Examples such as sampled handwritten characters and spatio-temporal body locations in 2D and 3D are considered. However, the general methodology extends far beyond these examples. The shape categories used here in order to represent the ideas of qualitative constraints are based on point set order types. Recall the example of the convex curve, that was defined by the signs of a number of determinants of matrices created by neighboring points. Let's generalize this concept, in order to classify categories not as obvious as convex shapes. Suppose we write the letter "S" repeatedly and sample the  $x, y$ - coordinates to get points  $p_1 \dots p_n$  along the contour (Fig 3.4). We now want to capture the constraint manifold  $\mathcal{M}$

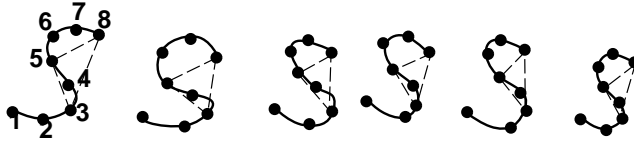


Figure 3.4: Examples of handwritten characters. Points 3, 5, and 8 is an example of a triplet that has a consistent sign of orientation throughout the set of characters.

for the set of points  $p_1 \dots p_n$ . This discussion will concentrate on the set  $\mathcal{M}_1$  of inequality constraints. It is possible to learn a set of inequality constraints  $C(p_1 \dots p_n) > 0$  from the training samples  $p_1^t \dots p_n^t$   $t = 1, 2, \dots$ . We hope this set of constraints will be able to characterize the constraint manifold  $\mathcal{M}$  in a reasonable way. The selection of this specific set will be made based on prior knowledge of the structure of the class of events considered. If we take the example of the letter "S" we see that it can be described qualitatively in terms of convexity properties of subsets of it's coordinate points. We could therefore test the specific set of constraints given by the determinants:

$$C(p_i, p_j, p_k) = \begin{vmatrix} p_i & p_j & p_k \\ 1 & 1 & 1 \end{vmatrix} \quad (3.6)$$

and select all those combinations of points  $i, j, k$  for which we consistently have the same sign of the determinant:

$$C(p_i^t, p_j^t, p_k^t) > 0 \quad (3.7)$$

for all examples  $t = 1, 2, \dots$  in the training set. This subset of constraints will then be used to characterize the constraint manifold  $\mathcal{M}_1$  for the letter "S".

The physics and geometry of the problem under consideration should ideally dictate the initial choice of tentative constraint functions  $C_i$ . The constraint functions, though, do not have to be related to the actual problem per se. The important thing is that they can be used to capture or approximate the support of the priors. The specific choice of determinants as above defines a general class of constraints that can model specific important qualitative shape properties such as convexity. They are therefore interesting candidates for a large class of qualitative modelling problems.



**Representation of the constraint manifold**

The inequality constraints ( $\mathcal{M}_1$ ) of the constraint manifold ( $\mathcal{M}$ ) from equation 3.4 can be represented in a *constraint table*. In fig. 3.5 all 14 qualitatively distinct configurations of a set of four points are shown. Each of these configurations can be coded in a table shown in fig. 3.5, where each ordered triplet is listed, together with the sign of the determinant. The table could also be represented in one column by only listing the triplets with positive

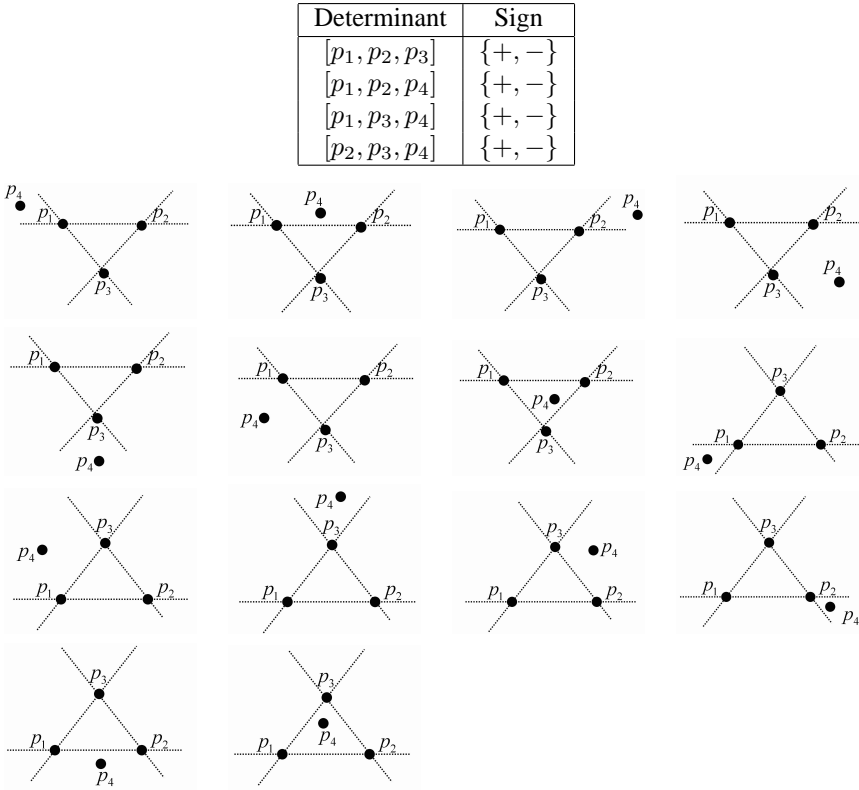


Figure 3.5: There are 14 qualitatively distinct configurations of a point set of cardinality 4. The table shows how a certain configuration can be represented.

determinants; if a constraining triplet has a negative determinant, it can be made positive by changing the order of the points in the triplet. Since we want to be able to enumerate the triplets in the format  $[p_i, p_j, p_k]$   $i < j < k$  a two-column table is used.

A constraint table, representing a certain class of point sets, is a subset of this table. This subset consists of the triplets representing the true qualitative constraints of the class. In order to identify which triplets should go into the constraint table, a number of training shapes are required. Every point triplet whose determinant has the same sign in *every*

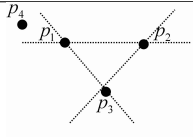
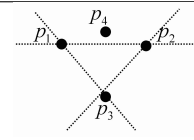
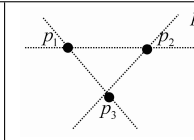
Determinant	Sequence 1	Sequence 2	Sequence3	Sequence 4	Constraint
$[p_1p_2p_3]$	+	+	-	+	?
$[p_1p_2p_4]$	+	+	+	+	+
$[p_1p_2p_5]$	-	-	-	-	-
$[p_1p_2p_6]$	-	+	+	-	?
...	...	...	...	...	...
$[p_{n-2}p_{n-1}p_n]$	-	-	-	-	-

Figure 3.6: The constraint table computed using four training sequences. In the last column, the determinant sign of each constraining triplet is shown. If the triplet is not constraining a question mark is shown.

*training shape* is added to the constraint table, together with the sign of that particular determinant. If only one training shape is used, a point set must have exactly the same constraint table as this training shape in order to belong to the class. As new training shapes are added, constraints are removed from the constraint table, and the class gets less constrained. In fig. 3.6 an imaginary constraint table with respect to four training sets is shown. Rather than removing non-constraining triplets, a question mark in the last column is used to indicate that the triplet is not a constraining triplet (i.e its determinant has different signs in different training sets).

**Example**

Consider the first three point sets in fig. 3.5, together with their constraint tables. This is shown in compact form as:

			
$[p_1p_2p_3]$	+	+	+
$[p_1p_2p_4]$	-	-	-
$[p_1p_3p_4]$	+	-	-
$[p_2p_3p_4]$	+	+	-

In this table, the first two determinants have the same sign for all three point sets. Thus, the resulting constraint table for the small training set is

$[p_1p_2p_3]$	+
$[p_1p_2p_4]$	-

Geometrically, this can be interpreted as that the points  $p_3$  and  $p_4$  must reside on different sides of the line spanned by  $p_1$  and  $p_2$ . As the point sets gets larger, it will be more difficult to put words on the constraints.

### 3.4 Smoothing 2D shapes

Given a potentially perturbed point set  $X' = [p'_1, p'_2, \dots, p'_n]$  we want to find the point set  $X = [p_1, p_2, \dots, p_n]$  maximizing equation 3.5. The algorithm iterates over the steps outlined in fig. 3.7. Step 2 improves the evolving model with respect to the measurement

Step 1:	Select an initial $X$ s.t. $X \in \mathcal{M}$
Step 2:	Gradient descent: $X_{i+1} = X_i - \nabla_X P(X' X)$
Step 3:	Enforce qualitative constraints: $X_{i+1} \in \mathcal{M}$
Step 4:	Goto Step 2

Figure 3.7: The iteration steps of finding the optimal 2D point set  $X$  given a perturbed point set  $X'$ .

data, while step 3 reprojects the model to  $\mathcal{M}_1$  (the manifold of allowable configurations).

#### Initial selection

In order to start the iteration, the algorithm needs an initial estimate. The only requirement of this estimate is that it resides on the manifold,  $\mathcal{M}$ . The most straight forward way to do this, is to select one shape randomly from the training data. Other methods are possible. For instance, a linear combination of training sets may yield an initial estimate closer to the solution (Howe et al., 1999; Bowden, 2000; Bregler and Omohundro, 1994). However, since  $\mathcal{M}$  is generally not a convex set, a linear combination of training sets may not reside on  $\mathcal{M}$ .

#### Gradient descent

In order to refine an estimate, the algorithm adapts the point set according to the gradient of the posterior  $\nabla_X P(X|X')$ , which is simply the likelihood function. We use the sum of squared differences, as shown in equation 3.8 as an estimate of this function.

$$\begin{aligned} \nabla_X P(X|X') &= \lambda \|X - X'\| \\ 0 < \lambda &\leq 1 \end{aligned} \tag{3.8}$$

A large  $\lambda$  will result in a faster convergence, while the refined point set may be pushed far away from  $\mathcal{M}$ , making it harder to reproject. The experiments in this chapter, used  $\lambda = 0.2$ . Proving the optimal choice is however beyond the scope of this study.

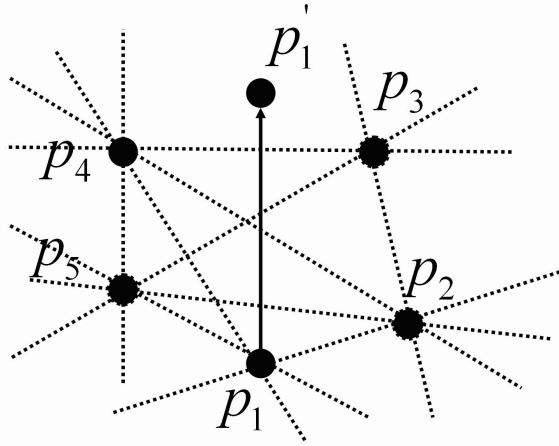


Figure 3.8: Enforcing the qualitative constraints in a point set. If point  $p_1$  has to move to the other side of line  $(p_3, p_4)$ , it also crosses lines  $(p_2, p_5)$ ,  $(p_3, p_5)$  and  $(p_2, p_4)$ , which may lead to new violations.

### Enforcing the qualitative constraints

Given a set of points  $X = [p_1, \dots, p_n]$ , and a constraint table  $T$  encoding the manifold  $\mathcal{M}_1$ , how do we update  $X$  so that it complies with  $T$ ? Generally, this is a computationally expensive problem. However, if we assume that the point set  $X$  is "relatively close" to  $\mathcal{M}_1$ ,  $X$  can be projected onto  $\mathcal{M}_1$  in reasonable time. In order to visualize the complexity of the problem, note that a point set changes its qualitative configuration as soon as one point crosses any line defined by two other points in the point set. Assume that  $[p_1 p_3 p_4]$  in fig. 3.8 is a violating triplet with respect to  $T$ . Further, we want to fix this violation by moving point  $p_1$  to the other side of line  $(p_3, p_4)$ . After this alteration,  $p_1$  has also crossed lines  $(p_2, p_5)$ ,  $(p_3, p_5)$  and  $(p_2, p_4)$ , which may cause new violations, depending on the constraint table. Clearly, the problem is non-trivial, and the presented algorithm should not be considered optimal. In fact, under some circumstances, the algorithm fails to resolve all violations. Finding and proving an optimal algorithm is a very intriguing problem. At this stage, we settle for the following heuristic that generally resolves *most* violations. In a point set  $X$ , a violating triplet is a triplet whose determinant is inconsistent with constraint table  $T$ . The algorithm iterates over the following steps until all violations are resolved, or until no more violations can be resolved.

1. Select the point  $p_i \in X$  that is a member of the largest number of violating triplets with respect to the constraint table.
2. For each violating triplet containing  $p_i$ , create a new point set by moving  $p_i$  to the other side of the line defined by the other two points in the triplet.

3. From all the new point sets, keep the point set with the least number of violating triplets.

Step 2 refers to "the other side of the line". In this work the point is simply moved slightly across the line, in order to minimize the alteration. Finding the optimal choice of how far each point should be reprojected could be a very interesting problem from an optimization point of view.

### 3.5 Experimental examples of 2D point sets

The algorithm was tested on a selection of point sets representing hand drawn characters and shapes, as well as point sets representing 2D projections of human motion. The results of these cases are demonstrated below.

#### Hand drawn shapes

For this experiment, a number of curves were manually drawn and spatially sampled, in order to generate point sets. It is assumed that the point correspondences between the point sets is known. In fig. 3.9 the effect of adding training shapes is shown. The graph displays the number of constraining triplets after each of the ampersand shapes is added to the training set. After adding the first two shapes, the number remains rather constant. As the last three shapes are added, the number of constraints drops quickly, due to the strong perturbations of those shapes. In particular, as the last training shape is added (the mirrored shape) only 10 constraints were left.

In fig. 3.10 the result of smoothing a perturbed shape based on training data is shown. The training set of a class of point sets is shown in the first column. The second column shows a noisy point set from the same class. The last column shows the result of the algorithm. As can be seen, a rather small number of training data is required by the algorithm in order to identify and correct apparent perturbations.

#### Projected human motion

As described in section 3.3, there is no conceptual difference between human motion and the shapes described in the previous section. Human motions are represented as point sets as well. This section shows that the same principle applies to projected human motions. The algorithm is applied on data from a number of sequences from athletics. Again, the point set of a motion is generated by stacking the point set of each time frame into one tall matrix. This means that both temporal and structural constraints will be considered by the constraint table, rather than only constraining each frame individually. By doing this, the algorithm attempts to identify qualitative constraints of the *trajectories* of each point, relative to each other. In the first case, some examples of pole vaulting were considered. The most critical phase of the vault is chosen - the part of the run-up when the vaulter plants the pole into the box. Each sequence is made up of three frames. In order to define the manifold, seven training sequences were used. Five of these training sequences are shown

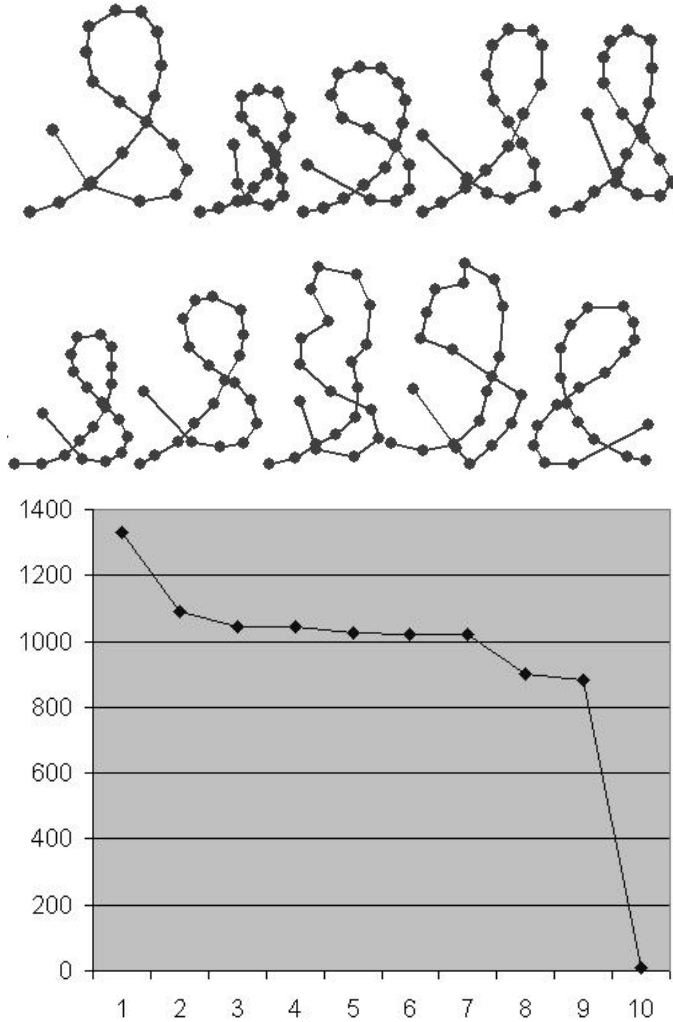


Figure 3.9: The training shapes for the ampersand experiment. The graph shows the number of constraining triplets in the constraint table after each ampersand is added to the training set. After the last shape, a backwards ampersand, has been added to the training set, no constraining triplets exist anymore.

in fig. 3.11<sup>1</sup>. Fig. 3.12 shows how the number of constraints decreases, as new examples

<sup>1</sup>These sequences are taken from [www.stabhochsprung.com](http://www.stabhochsprung.com). Courtesy of Mr. Herbert Czington

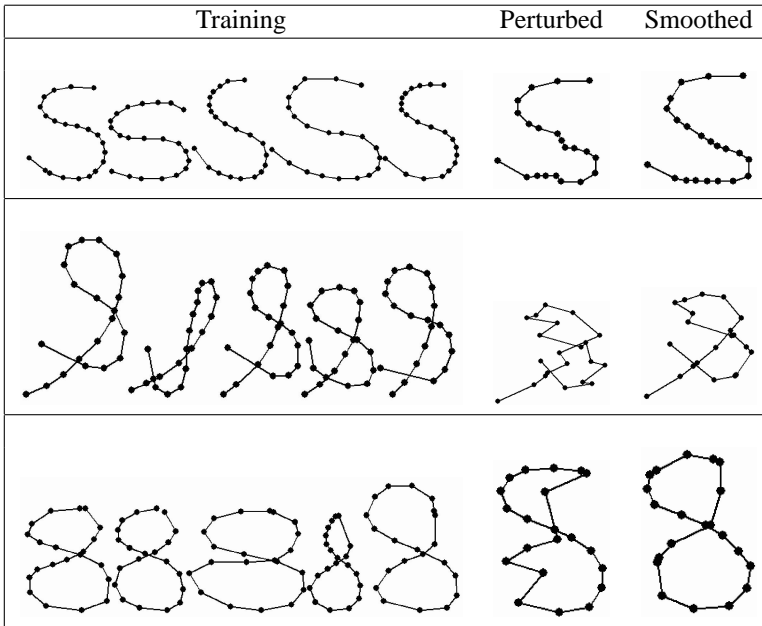


Figure 3.10: The first column shows the training shapes used to learn the qualitative constraints. The second column shows a perturbed instance of a shape coming from this category. The last column shows the corrected perturbed shape.

are added. Fig. 3.13 shows the result of the smoothing where a few significant errors were introduced manually. The top row shows the original sequence, the second row shows the perturbed sequence, with the errors pointed out by bold arrows. This shape contained 405 violating triplets. The constraint table generated from the training sequences contained 11,019 rows (=constraining triplets). The last row shows the result of the smoothing. Obviously, the result is fairly close to the original shape.

In fig. 3.14 the results from smoothing a weightlifting (snatch) sequence, and a tennis sequences contaminated with significant gaussian noise are presented. Also, one tennis sequences with a couple of apparent manual errors is smoothed. The constraint table for the weightlifting sequence was based on 5 training sets, and contained 15,984 rows. The perturbed point set violated 2,961 of these before being smoothed. The constraint table for the severely distorted tennis sequence used 8 training sequences, and consisted of 12,296 rows. In the distorted tennis sequence 918 constraining triplets were violated.

### Missing data

One interesting application for the qualitative descriptors is the case when some points are missing. This is rather common when dealing with point sets generated from an automatic

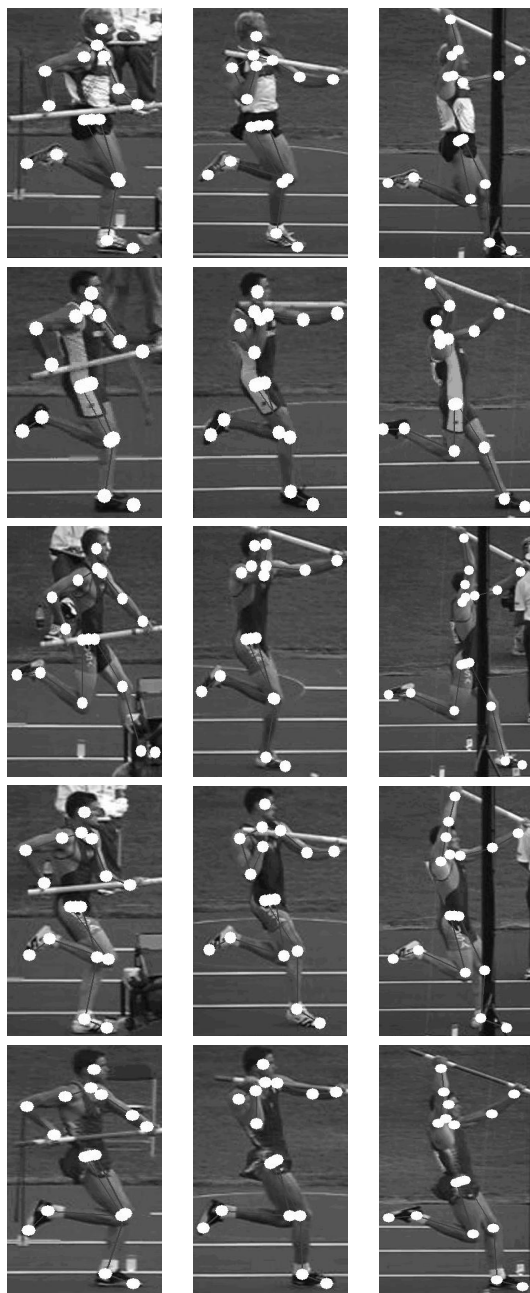


Figure 3.11: Five of the seven training sequences for the pole vault experiment.



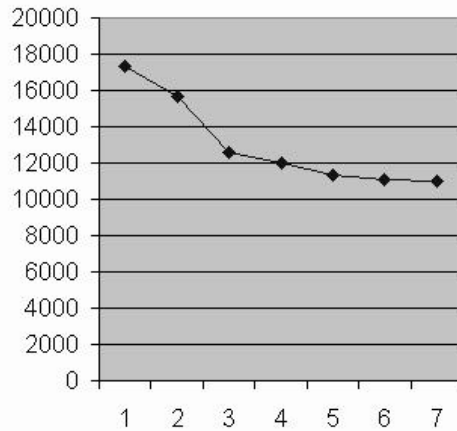


Figure 3.12: The number of constraining triplets versus new training data for the pole vault experiment.

tracking module. In this case, the positions of the missing points can be estimated based on the manifold of possible events. In this experiment, each missing point is inserted into the point set so that its position is consistent with the points already in the set. In order to achieve this, the new point is randomly placed into a number of locations, in order to find the location resulting in the fewest number of violating triplets. After this location is found, the reprojection procedure was executed, as described previously. The results from this experiment are shown in fig. 3.15, where two sequences are completed, based on qualitative constraints. The first sequence is again a pole vault sequence where the points corresponding to the vaulter's left arm is missing in all frames. In the other sequence, the vaulter's upper body is missing in the middle frame.

### 3.6 3D shapes

All algorithms shown thus far extend perfectly well to 3D data. The problem is computationally more demanding of course, since we are dealing with point sets in 3D space. Instead of using determinants of point triplets we now use determinants of point quadruples. Also, the determinant of the quadruple changes sign when one point crosses the plane spanned by the other three points, as illustrated in fig. 3.16. The procedure for reprojecting a shape onto the manifold of possible events is the same as in the 2D-case, with a couple of modifications:

1. Select the point  $p_i \in X$  that is a member of the largest number of violating quadruples with respect to the constraint table.

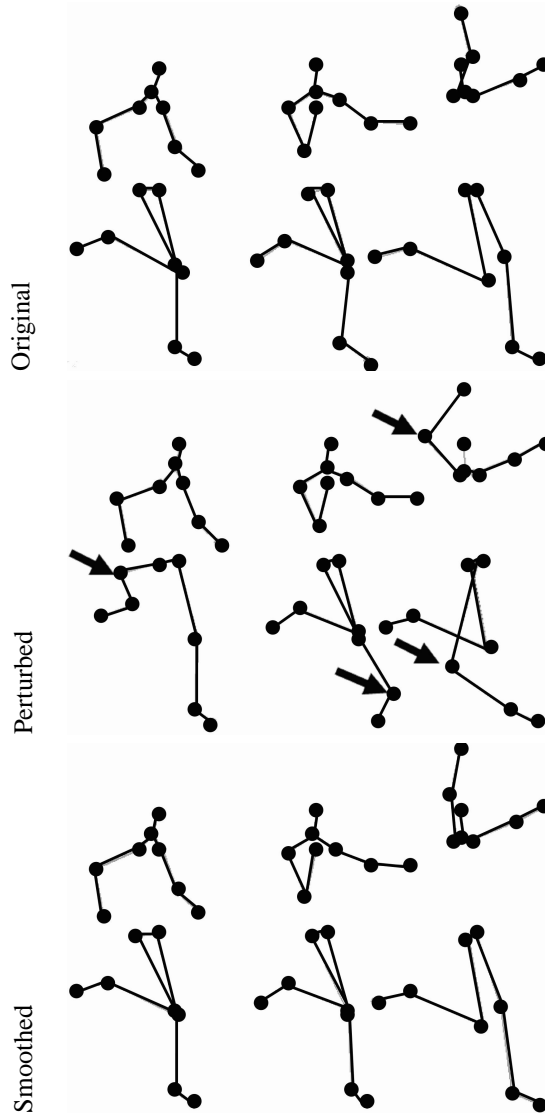


Figure 3.13: The result on smoothing a pole vault sequence with distinct errors (pointed out by the arrows in the middle row).

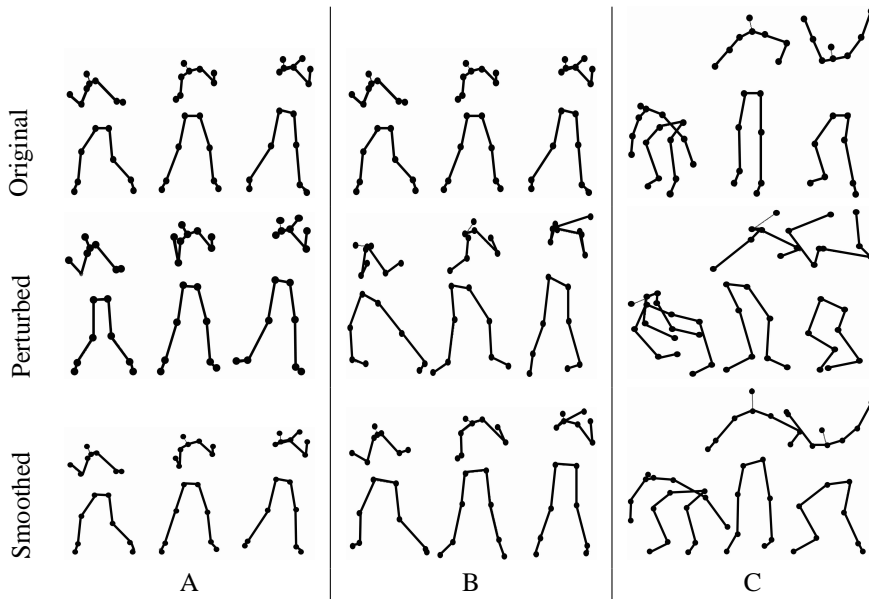


Figure 3.14: The smoothing algorithm applied to different 2D shapes and motions. The original shape in the top row is followed by the same shape perturbed with gaussian noise. The last row shows the qualitatively smoothed shape.

2. For each violating quadruple containing  $p_i$ , create a new point set by moving  $p_i$  to the other side of the plane spanned by the other three points in the quadruple.
3. From all the new point sets, keep the point set with the least number of violating quadruples.

The shapes considered here are exclusively shapes representing human motion. Similar to the 2D case, a representation of motion is constructed by stacking the point set of each pose into one tall matrix. A sequence of human motion can be visualized as a space-time shape, as shown in fig. 3.17. Since the entire motion is regarded as one shape, spatial as well as temporal constraints are implicitly considered. The 2D algorithm (fig. 3.7), for performing smoothing of a perturbed shape is slightly modified in order to handle the 3D case, as shown in fig. 3.18. The main difference is the addition of *step5 - the equality constraints* and is discussed next.

### Equality constraints

Working in 3D space, rather than in a 2D projection, makes for larger computational demands. Fortunately, we can partly compensate for this by incorporating the *equality constraints*, introduced as  $\mathcal{M}_2$  in equation 3.4. This is referred to as *rigid-link constraints*. It

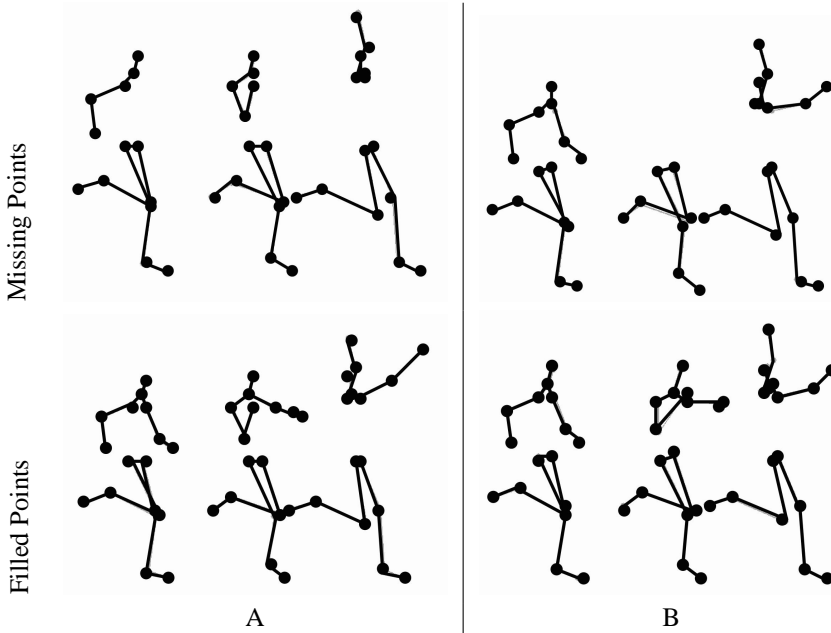


Figure 3.15: Filling in missing points. The top row shows two sequences with a number of points missing. The bottom row shows how the missing points can be filled in by maintaining the qualitative characteristics from the training set. In the left sequence, the vaulter's left arm is missing. In the right sequence, the upper body in the middle frame is missing.

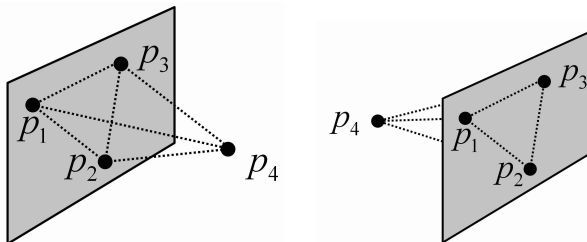


Figure 3.16: 3 points define a plane. The determinant of 4 points indicates what side of this plane the 4th point resides on. In this figure the two point quadruples are of different order types.

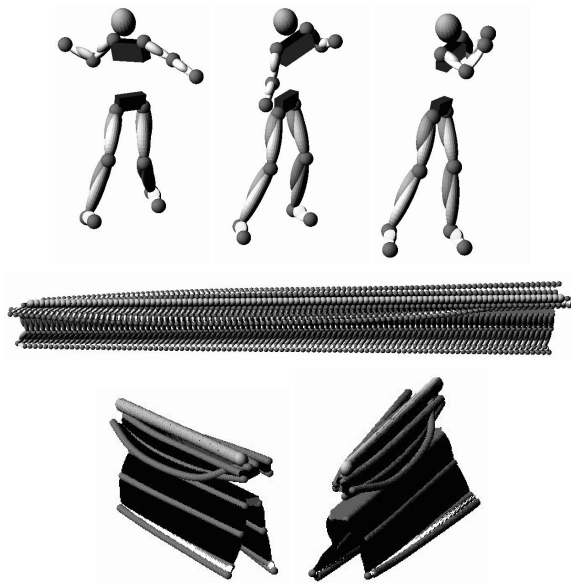


Figure 3.17: Motion shape. The top row shows three key frames from a forehand stroke. The second and third row show a shape representation of the motion from three different viewpoints.

Step 1:	Select an initial $X$ s.t. $X \in \mathcal{M}$
Step 2:	Translate, rotate and scale $X$ to fit the target 3D shape.
Step 3:	Gradient descent: $X_{i+1} = X_i - \nabla_X P(X' X)$
Step 4:	Enforce qualitative constraints: $X_{i+1} \in \mathcal{M}_1$
Step 5:	Enforce equality constraints: $X_{i+1} \in \mathcal{M}_2$
Step 6:	Goto Step 3

Figure 3.18: The iterative algorithm for finding the optimal 3D point set  $X$ .

turns out that when working in the 3D world, the equality constraints are often sufficient to approximate the manifold  $\mathcal{M}$ . Rigid-link constraints ensure that the distance between the points in a number of pairs in a point set are equal. Specifically for human motion these point pairs are identified from the following two observations:

- **Symmetry constraints.** The left limbs of the person must be of the same length as the right.
- **Temporal consistency.** The length of a limb remains constant throughout the motion.

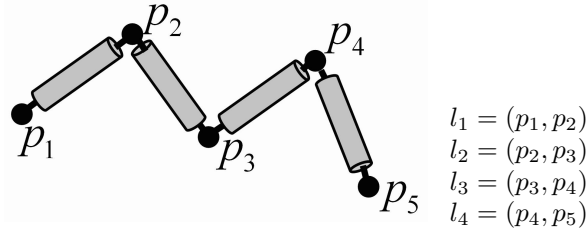


Figure 3.19: An articulated chain of four rigid links.

Given a kinematic chain  $K = (l_1, l_2, l_3, l_4)$  as in fig. 3.19 and the perturbed point set  $X = [p_1, p_2, p_3, p_4, p_5]$  and the required length of the constrained links,  $U = [u_1, u_2, u_3, u_4]$  we can enforce the rigid-links constraints (equality constraints) by updating  $X$  according to equation 3.9.

$$end[l_j] = start[l_1] + \sum_{i=1}^j u_i d_i \quad (3.9)$$

where  $d_i$  is the direction of link  $i$ , and is computed by

$$d_i = \frac{end[l_i] - start[l_i]}{\|end[l_i] - start[l_i]\|} \quad (3.10)$$

and  $start[l_i]$  and  $end[l_i]$  are the start and end points of segment  $l_i$ . The limb lengths  $U$  are computed as the average length amongst all corresponding limbs in the sequence. In order to satisfy the equality constraints as well as the inequality constraints, the algorithm iteratively enforces both these constraint categories until convergence is reached.

### Results of 3D smoothing

In these experiments, the motions are represented by three key frames (as was the case in 2D). The intermediate frames are interpolated, in order to improve the visual appearance. Two different motions were tested. Tennis (a forehand stroke) and walking (a single stride).

#### Tennis

In the tennis experiment, the system learned the constraint manifold from four forehand strokes. The algorithm was then given the sequence  $X'_T$  shown from two viewpoints in fig. 3.20. As can be seen,  $X'_T$  was prepared with some apparent errors in the motion of the right elbow. Obviously, the right arm bends the wrong way towards the end of the sequence, which is a violation of the learned qualitative constraints. The algorithm outlined in fig. 3.18 was then executed, in order to estimate the point set  $X_T \in \mathcal{M}$ , residing as close as possible to  $X'_T$ . The results from the qualitative smoothing are shown below the perturbed sequence in fig. 3.20.

### Walking

The other experiment involves a walking person. The training set in this case consists of three training sequences. Again, the input sequence was prepared with an apparent error; the right knee of the person bends the wrong way. As shown in fig. 3.21, this is corrected after executing the smoothing algorithm. The errors in the input sequences are designed in order to simulate errors that frequently occur in single view reconstruction, where errors frequently result in the joints pointing the "wrong way" with respect to the image plane.

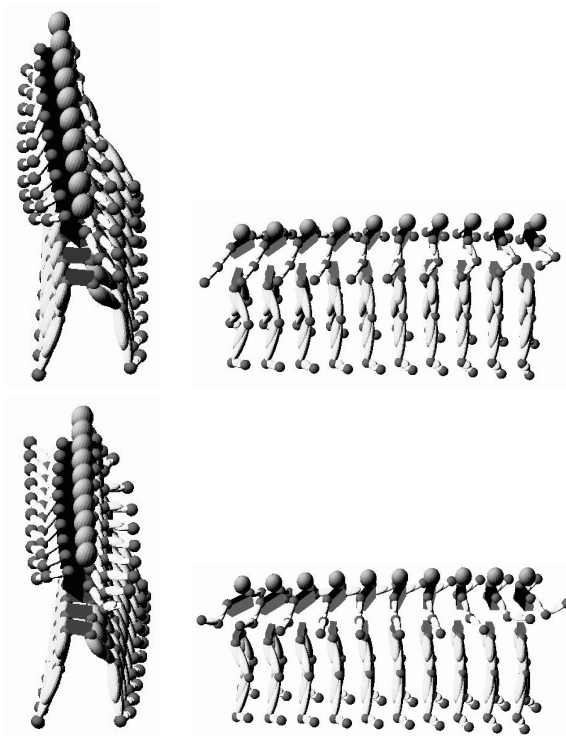


Figure 3.20: The results of qualitative smoothing of a perturbed tennis sequence. In the top sequence (as seen from two view points) the trajectory of the right hand is erroneous. In the sequence below, the trajectory is fixed, according to qualitative constraints.

## 3.7 Single view reconstruction

We have now reached the stage to tackle the goal of the chapter - 3D reconstruction of a motion given its 2D projection. More specifically, we are interested in doing this when the 2D data is perturbed with noise, since this is generally the case when dealing with

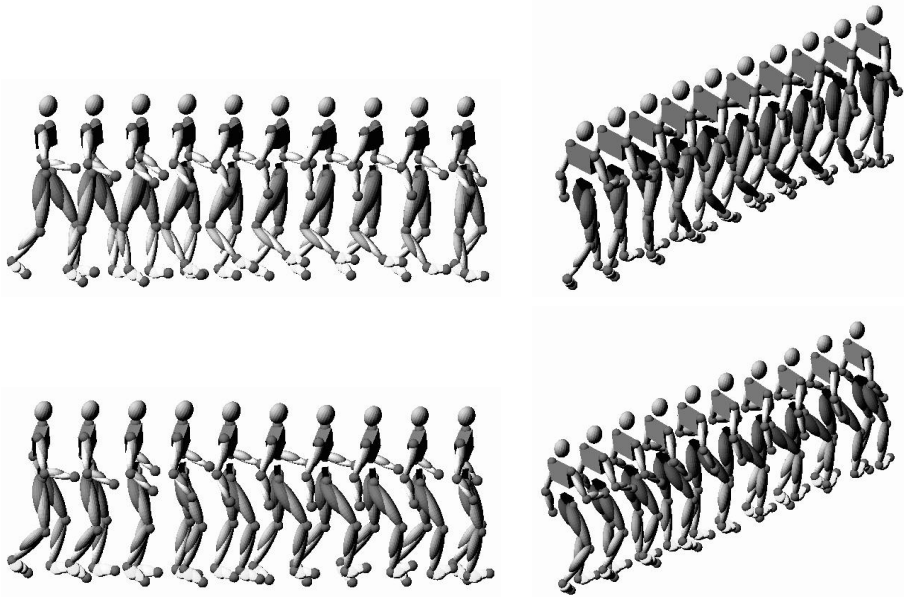


Figure 3.21: The results of qualitative smoothing of a perturbed walking sequence. In the perturbed sequence (on top) the right knee bends the wrong way. This apparent violation of the qualitative constraints is fixed in the bottom sequence.

automatically tracked data. In general, when the 2D data contains very little, or no noise, it turns out that  $\mathcal{M}_1$  is enough to estimate  $\mathcal{M}$  (the manifold of allowable motions). As an increasing amount of gaussian noise is introduced, this is no longer the case. In fact, only enforcing the rigid link constraints does not disqualify any configurations. In this case, the system will always come up with a solution that eliminates the projection error, no matter how poor the data is. This can of course be avoided by enforcing constraints dictating maximal limb lengths and certain allowed values for the ratios of the limb lengths. However, the qualitative constraints partly take care of this dilemma. The first thing at hand, though, is to upgrade the iteration process outlined in fig. 3.18 to handle the case of single view reconstruction. The new algorithm is outlined in fig. 3.22. The only difference here compared to the previous case is in step 2, where the initial shape is transformed by a similarity transformation, in order to minimize the projection error. There are many solutions of such pose estimation problems. The system developed here uses a Leuvenberg-Marquart optimization, in order to find the optimal rotation matrix and scale factor.

In order to show the potential power of the qualitative constraints, one walking sequence was projected down to 2D, using a synthetic orthographic camera, positioned perpendicular to the direction of the walk. This 2D data was contaminated with an increasing amount of noise. Each point  $p$  in the projected point set  $X$  was perturbed by an amount



Step 1:	Select an initial $X$ s.t. $X \in \mathcal{M}$
Step 2:	Translate, rotate and scale $X$ in order that its reprojection fits 2D data.
Step 3:	Gradient descent: $X_{i+1} = X_i - \nabla_X P(X' X)$
Step 4:	Enforce qualitative constraints: $X_{i+1} \in \mathcal{M}_1$
Step 5:	Enforce equality constraints: $X_{i+1} \in \mathcal{M}_2$
Step 6:	Goto Step 3

Figure 3.22: The iteration steps of finding the optimal point set  $X$

of  $mN(0, 1)$ , a sample from a zero-mean normal distribution times the magnitude  $m$ . The noisy data was then used as input to the reconstruction algorithm. The algorithm was executed in the following two modes:

1. Using only the rigid link constraints. At each step of the iteration, the evolving solution was only reprojected back to  $\mathcal{M}_1$ .
2. Using rigid link constraints, as well as the inequality constraints. In this mode, the evolving solution was reprojected to  $\mathcal{M} = \mathcal{M}_1\mathcal{M}_2$ .

The gradient descent in step 3 is slightly modified when the qualitative constraints are enforced. Computationally, it is very complex to reproject a solution back to manifold of allowable configurations if the solution is pushed too far away from  $\mathcal{M}$ . Therefore the gradient descent was iteratively applied until the number of violating quadruples exceeded a threshold value, rather than using a fixed stepsize. The result is shown in fig. 3.23. As can be seen, the rigid-link constraints are not very good at handling severe noise. This is implicitly due to the fact that the input data is given too much weight. The rigid-link constraints are too weak to control the iteration. On the other hand, when the inequality constraints are enforced, the iteration is much more restrictive in terms of listening to the image data. The learned inequality constraints force the solution to resemble a more natural walk.

### 3.8 Summary

Qualitative constraints are useful in order to capture fundamental properties of shapes and, in particular, human motion. The purpose of qualitative constraints is to, from a small set of examples, identify the most fundamental properties of the class of shapes, or motion. This is useful in smoothing perturbed shapes in order to avoid extreme bias towards the training data, while still enforcing the resulting shape to belong to the broad class. The constraints presented here have been based on identifying certain spacial properties of subsets of point in the point set representing the shape. No claims have been made that these properties are optimal in defining qualitative classes. The key point to be made is that qualitative constraints are useful. How to define them optimally is to some extent problem dependent.

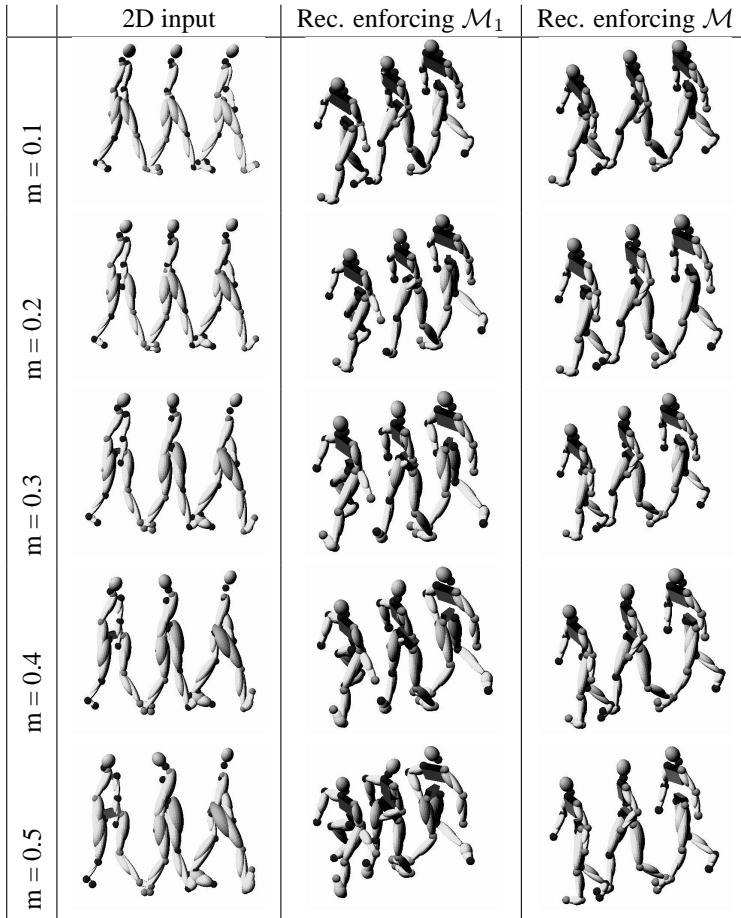


Figure 3.23: The results of a single view reconstruction. The left column shows the value of  $m$  (magnitude of noise). The second column shows the input data. The third column shows the result of a single view reconstruction enforcing only rigid-link constraints. The rightmost column shows the result when the inequality constraints are enforced as well.

One issue with the constraints defined in this chapter is the computational complexity. Long sequences of 3D motion generate very large point sets. Thus, future work involves designing methods to reproject large perturbed shapes to the manifold of allowable configurations in a computationally efficient manner.

**Beyond single view reconstruction - SYNTHESIS**

One of the final goals for developing qualitative constraints for 3D motion is to allow the random generation of new motions of a certain class. For example, for the purpose of developing a computer game for tennis, we might want to implement a totally random player. That is a player with an individual, artificial technique, that is not acquired from a real player. By "carving out" a manifold representing all possible variations within a class of motions, this could be accomplished by simply draw one random sample from the manifold. The manifolds described in this chapter capture the essence of certain motions. However, they have turned out to be too broad in order to generate random motions. Motions residing on the generated manifold sometimes don't agree with our subjective idea about the motion. This can be solved by identifying new constraints that have to hold, in order for the motion to belong to the class. This is of course a very subjective task, and the analogy to the sculptor becomes yet clearer - we have to further refine the statue and shave off some of its unnecessary parts. This thesis has set the stage - and hopefully inspired some artists out there to start carving manifolds.



## Chapter 4

# Key framed single view reconstruction

In the previous chapter, we learned the importance of using prior information in order to obtain a truthful 3D reconstruction of human motion. A method to automatically learn the priors from a set of training data was presented. The methods are applicable when reconstructing short distinctive sequences, where enough training data is available. Some problems arise, though, under the following conditions:

- There is no training data.
- There is too much variation in the motion for automatic learning of priors (the manifold of allowable configurations is too broad).
- The sequence to be reconstructed is very long.

The last item can be solved by splitting up the sequence into shorter "snippets", for which training data is available. This of course adds complexity to the problem. Especially so, if each snippet also suffers from the second bullet in the list above; too large individual variations. This chapter will deal with much tougher sequences than was the case in the previous chapter. Unfortunately, this means that the automatic learning of the constraint manifold is temporarily aborted. The reason for this is to demonstrate that, with little manual intervention, it is possible to obtain a reconstruction that is visually pleasing, and to a very large extent reflects the true action taking place. Again, automatically tracked 2D data will be used as input, which means that priors are again required for two obvious reasons:

1. Handle tracking errors.
2. Resolve the depth ambiguity.

In this chapter the priors are manually added to the system for every specific sequence to be reconstructed. This may seem to be impractical for most useful applications. However, the methodology is designed so that, by some manual intervention and creativity, *very long sequences* can be reconstructed *almost* automatically, as soon as the initial intelligence is put into the system. The ideas are demonstrated on a long sequence of a tennis game.

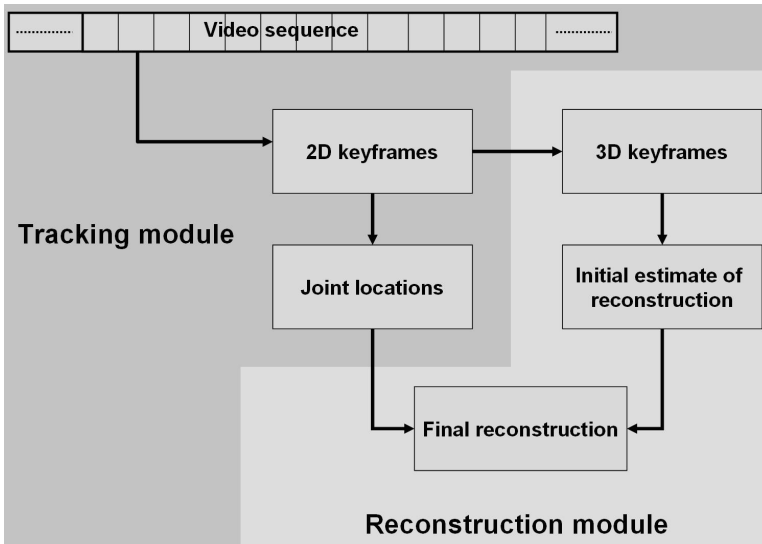


Figure 4.1: Schematic diagram of the activities in the reconstruction system.

## 4.1 System overview

The system consists of two main modules; the tracking module and the reconstruction module. The tracking module extracts the feature points defining the joints of the human in each frame. The reconstruction module handles the generation of 3D data based on the 2D information delivered from the tracking. Further, the reconstruction is built as an add-in to an existing, state-of-the-art tracking implementation. The best feature of the reconstruction module is that it barely adds any complexity to the system in terms of additional manual labor. In principle, the reconstruction works in a similar fashion as in the previous chapter. It starts with an initial estimate of the final reconstruction and iteratively refines this estimate. In the previous chapter, this estimate was chosen from a set of training data. In this case no training data exists, so the initial estimate must be selected differently. This is actually done manually, but with much help of some prior work done by the tracking algorithm. This initial estimate is then refined based on the tracked points delivered by the tracking module. Again, during this refinement, care is taken, so that the reconstruction stays on the manifold of allowable motions. In this case, the manifold is strictly heuristical and rather simple, and not automatically learned as in the previous chapter. The main objective is to present the reconstruction module. Since some of its components strongly rely on the behavior of the tracking, an overview of the main features of the tracker is included in the presentation. An outline of the system is illustrated in fig. 4.1.

## 4.2 Summary of 2D tracking

The tracking algorithm is the one developed in (Loy et al., 2004). Briefly, the tracking module performs the following activities: Given an input video sequence of  $N$  frames,

$$V_{1:N} = (V_1, V_2, \dots, V_N)$$

the silhouette of the person being reconstructed is extracted in each frame. Thus, the video sequence is transformed into a sequence of silhouettes:

$$S_{1:N} = (S_1, S_2, \dots, S_N)$$

According to a similarity measure,  $d(S_i, S_j)$  defined on the silhouettes, a *small* set of frames,  $K \subseteq \{1, \dots, N\}$  is extracted in order to serve as *key frames*.  $S_K$  is the set of silhouettes associated with  $K$ . Also associated with  $K$  is another set  $W_K \subseteq \{1, 2, \dots, N\}$ , which is defined as the set of silhouettes *relatively similar* to at least one silhouette in  $S_K$ . More specifically

$$i \in W_K \Leftrightarrow \exists_{j \in K} \{d(S_i, S_j) < \gamma\} \quad (4.1)$$

where  $\gamma$  is a similarity threshold. The details of the threshold selection is left out of this discussion. Simultaneously with the key frame selection, an association map,

$$A : \{1, \dots, N\} \mapsto K$$

is defined, associating each frame with its most similar key frame. For all silhouettes  $S_i \notin S_{W_K}$ , we define  $A(i) = -1$ . Lastly, the strength of the association of silhouette  $S_i$  with  $A(i)$  is denoted as  $a(i)$ . Intuitively, this strength should be  $d(S_i, S_{A(i)})$ ; however, the function is discretized as follows:

$$a(i) = \begin{cases} 0 & i \in W_K \\ 1 & i \notin W_K \wedge \exists_{j \in W_K} |i - j| < 4 \\ 2 & \text{otherwise} \end{cases}$$

In other words, an association gets the value 0 if the frame is well represented, value 1 if the frame is relatively close (spatially in the sequence) to a well represented frame. All other frames get the value 2. While one purpose of  $K$  is to serve as a "summary" of the entire sequence, another equally important purpose is to assist in localizing the exact joint locations in frames of similar appearance. In order to facilitate for the later, the user has to manually click on these joint locations in all key frames. This is the only manual intervention required by the system. The good news is that the key frames generated for one small part of the sequence (say 30 seconds) will be enough for most parts of the entire game, modulus a couple of unexpected incidents, for example when one of the players throws a racket in agony. This relies on the hypothesis that most actions during a game are fairly repeated. In most athletic activities, this is a valid assumption. In fact, tennis is a very complex exercise compared to highly repeatable actions such as long jump or

golf. On the other hand, very non-repeated activities such as dancing or ice hockey would probably require more key frames. In the tennis sequence used in this chapter, key frames are identified separately for upper and lower body. The 25 key frames for the upper body are shown in fig. 4.2. Manually marking the joints in 25 frames is not an overwhelming task. By using the the silhouettes and the set of key frames, the tracking algorithm is able



Figure 4.2: Key frames identified in the 36 seconds long tennis sequence.

to deliver the sequence

$$P_{1:N} = (P_1, P_2, \dots, P_N)$$

which is the sequence representing the joint locations in each frame. Since 15 joint centers were used, each  $P_i$  is a  $15 \times 2$  matrix.

The important deliverables from the tracking to the 3D reconstruction module is summarized in the box below:

***Delivered from 2D tracking:***

$K \subseteq \{1, \dots, N\}$	The set of key frame indexes
$A(i)$	The key frame assignment map.
$a(i)$	The strength of the key frame assignment.
$P_{1:N} = (P_1, \dots, P_N)$	The positions of the joint centers in the sequence.
$P_K$	$\{P_i \in P : i \in K\}$



### 4.3 3D key frames

Recall from the introduction, that the problem of reconstructing the 3D pose of a human, given its 2D projection is ill-posed. Given the projection of the skeletal joint locations  $P_t$  at time  $t$ , the number of qualitatively different reconstructions,  $X_t$ , is bounded by  $2^L$ , where  $L$  is the number of links, (Taylor, 2000) as each link can point either towards or away from the image plane, as illustrated in fig. 1.3. For an  $N$  frame sequence, the number of possible reconstructions explodes to  $(2^L)^N$ . This enormous search space can be pruned by imposing the physiological limitations of the human body (Sminchisescu and Triggs, 2003) and bounding the motion between adjacent frames. However, automatic approaches for finding the correct configuration only works on relatively short sequences, using today's state-of-the-art systems. It must also be remembered that the limb lengths of the person being reconstructed are un-known, and the reconstruction will only be correct up to the limb length estimate.

After this review of the problems of 3D reconstruction, let's move on to the good parts. The 2D tracking module has already done the job in identifying the set of key frames,  $P_K$  (remember that  $P_{1:N}$  is the sequence of extracted joint centers).  $P_K$  can now easily be extended into the set

$$X_K = \{X_i : i \in K\}$$

where  $X_i$  is the *pose* or 3D representation of  $P_i$ . Acquiring  $X_K$  is easily done manually, since the 2D tracking scheme has *already budgeted* for manual labelling of feature points in each key frame. Therefore, extending the process of marking points in  $P_K$ , to also include the depth disambiguation is a very cheap upgrade. These 3D reconstructed key poses  $X_K$  now provides an approximate basis of the 3D poses exhibited in the *entire sequence*. Fig. 4.3 shows some of the key frames together with their manually constructed 3D key poses. Thus with a limited amount of manual effort we have obtained some very strong priors. The next section describes how these 3D key poses are used to create a smooth initial estimate,  $X_{1:N}^0$ , of the 3D configuration of the player throughout the sequence.

### 4.4 Establishing an initial estimate

One of the prioritized features of this system is for it to always come up with a solution, even when the tracking module fails (maybe due to severe occlusion, motion blur or just pure bad luck). The purpose is not to produce a reconstruction accurate enough for detailed biomechanical analysis. The key to always be able to deliver "something" is to identify some frames of the sequence where the tracking seems to have delivered very accurate data. The key poses associated with these frames are then used as interpolation points in order to create a smooth and relatively correct estimate. However, since such an initial estimate is only based on a small (in this example 25) number of key poses, this reconstruction would not provide much variation throughout the sequence. In this tennis example, all forehands would therefore look the same, except for some temporal variations (each stroke may be of different duration). In order to capture the details of the motion,

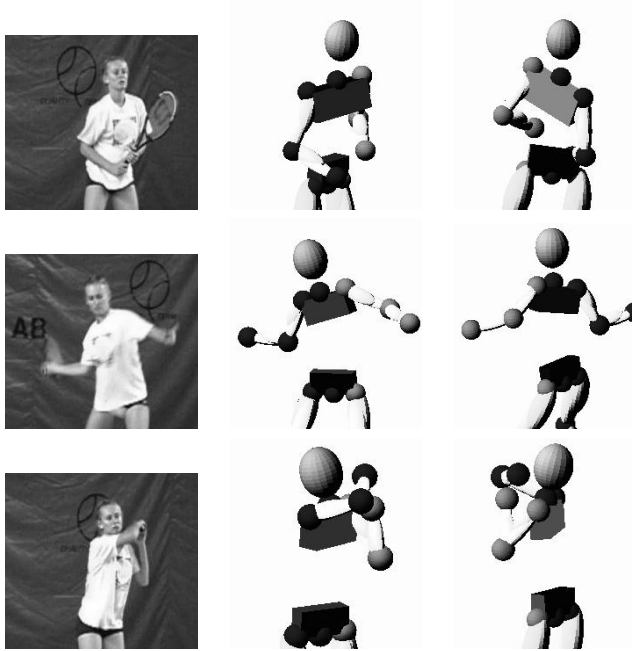


Figure 4.3: Three of the upper body key frames are shown to the left. The two rightmost columns show the 3D reconstructions of the key frames from two different viewpoints.

this initial estimate must be refined in order to capture finer details of the sequence. From the 2D tracking module, we do have the key frame assignments,  $A(i)$  and the weights of these assignments,  $a(i)$ . From the manual 3D key pose construction, we also have  $X_K$ . From this information alone, we can actually create a *keyed* representation of the entire sequence, simply by constructing the sequence:

$$X_{1:N}^0 = (X_{A(1)}, X_{A(2)}, \dots, X_{A(N)})$$

In case  $X_{A(i)} = -1$ ,  $X_{A(i-1)}$  can be used instead. This idea is illustrated in fig. 4.4. Since some of the associations are relatively weak, the initial estimate  $X_{1:N}^0$  may contain several bad poses. Therefore, a subsequence of the best (most likely) poses should be chosen, and used in order to interpolate a better initial estimate. In addition to  $a(i)$ , another indicator of how well a certain key pose fits an associated frame will be used. This is done by testing how well the key pose orthographically projects back to the associated frame. Let's thus define the projection correctness,  $r(i)$  as:

$$r(i) = \min_c \| \mathcal{C} X_{A(i)}^T - P_i^T \|$$

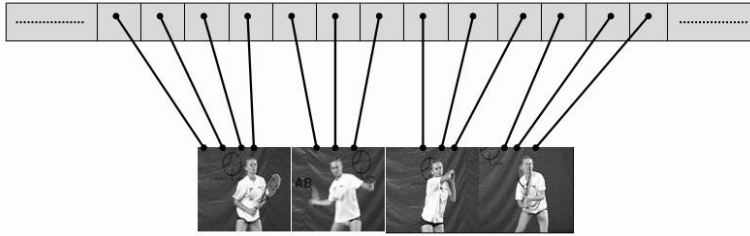


Figure 4.4: Assignment of key frames to each frame in the sequence.

where  $\mathcal{C}$  is an orthographic camera. A compound estimate,  $b(i)$  can now be defined as a weighted sum

$$b(i) = \alpha a(i) + \beta r(i)$$

where  $\alpha$  and  $\beta$  are parameters. In this implementation, only associations with  $a(i) = 0$  are considered. This is facilitated by setting  $\alpha = \text{inf}$  and  $\beta = 1$ . It is also important that the distance between two interpolation points do not become too large. In order to avoid this,  $X_{1:N}^0$  is split up into 20 frame runs. In each of these 20 frame runs, the strongest association is selected. This ensures a relatively robust basis for interpolation. Given this basis, the initial estimate is refined by spherical linear interpolation (SLERP)(Shoemake, 1985) between the selected key poses. An illustration of this process is shown in fig. 4.5.

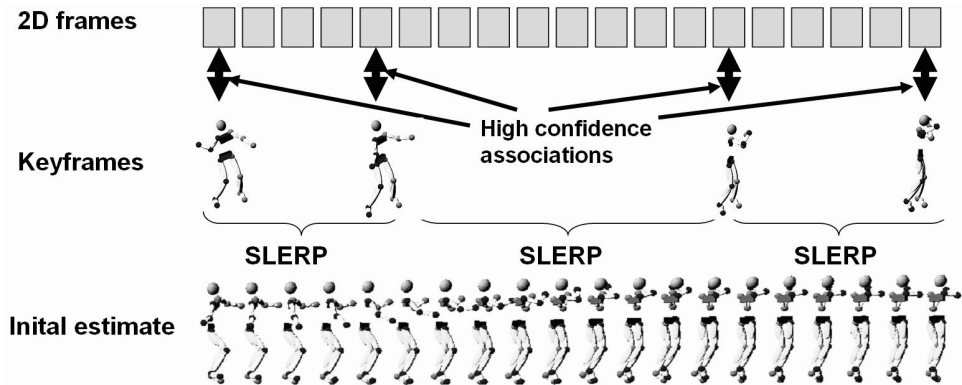


Figure 4.5: Visualization of the generation of a smooth and plausible trajectory of the 3D skeleton that approximates the content of the video.

### Spherical Linear Interpolation (SLERP)

SLERP is a common method for interpolation between two rotation matrices, and is thus very well suited for interpolation between two skeletal poses. As mentioned in the introduction, the skeleton can be represented as a set of joint locations in cartesian space, or as a set of rotation matrices indicating the orientation of each limb with respect to the previous limb in the articulated chain. In principle, when interpolating between two skeletal poses, intermediate rotation matrices for each limb are inserted. Specifically, each limb in the chain is represented by the transformation matrix

$$T = \begin{bmatrix} R_z(\alpha)R_y(\theta)R_x(\phi) & t \\ \mathbf{0} & 1 \end{bmatrix}$$

where  $R_z, R_y$  and  $R_x$  are rotation matrices, and  $t$  is the translation (limb length). The most straightforward way of interpolating between two matrices,  $T_1$  and  $T_2$  would be to interpolate each individual rotation matrix, by inserting rotation matrices corresponding to intermediate angles. This method, however, suffers from the problem of sometimes going through singularities, resulting in bizarre solutions. The best method to avoid this problem is by representing the transformations by quaternions. A rotation matrix is represented in quaternion form as

$$q = [\cos(\theta/2), w \cdot \sin(\theta/2)]$$

where  $w$  is a unit axis around which the rotation takes place. Given two quaternions,  $q_0$  and  $q_1$ , at times 0 and 1 respectively, intermediate rotations can be computed as

$$q_t = (q_0 \cdot q_1^{-1})^t \quad (4.2)$$

By performing this operation on every limb in the articulated chain, we can obtain smooth and natural transitions between two poses. Animators may argue that this sometimes is not good enough, since the entire motion will be jerky due to rapid changes in angular velocities at the interpolation points. In order to avoid this, spherical cubic interpolation should be used. In this case, however, our interpolated sequence is only intended to serve as an initial estimate of the sequence, and will be strongly refined later, by listening to the tracked data. Therefore, SLERP is definitely good enough for this purpose.

## 4.5 Fitting the smooth motion estimate to the joint data

The last task is to refine the 3D reconstruction by allowing the localized joint locations  $P_{1:N}$  to influence  $X_{1:N}^0$ . The  $P_{1:N}$  may contain outliers, be corrupted by noise and suffer from missing estimates due to self occlusion. To ensure robustness to these factors, the final estimate of  $X_{1:N}$  is forced to be a valid trajectory of a human skeleton. Let's define  $\mathcal{M}_N$  as the manifold describing all valid trajectories of the sequence of length  $N$ . Then:

$$\hat{X}_{1:N} = \arg \min_{X_{1:N}} E(X_{1:N}, P_{1:N}) \quad \text{subject to} \quad \hat{X}_{1:N} \in \mathcal{M}_N. \quad (4.3)$$

where  $E$  is a cost function based on the sum of squared differences between  $P_{1:N}$  and the orthographic projection of  $X_{1:N}$  (denoted by  $X'_{1:N}$ ):

$$E(X_{1:N}) = \| X'_{1:N} - P_{1:N} \|^2 \quad (4.4)$$

There is no easy characterization of  $\mathcal{M}_N$ , so enforcing  $\hat{X}_{1:N}$  to belong to  $\mathcal{M}_N$  is difficult. Here  $\mathcal{M}_N$  has been defined by the following criteria:

1. Each pose must exhibit symmetry properties - the left limbs must be of equal length as the right limbs.
2. The length of each limb must remain constant throughout the sequence.
3. The trajectory of each joint throughout the sequence must be smooth.

By additional work, it is possible to refine  $\mathcal{M}_N$  to more accurately reflect the motion being reconstructed. Again, the sculptor from the previous chapter should maybe be called in to perform some carving. By construction  $\hat{X}_{1:N}^0 \in \mathcal{M}_N$ . Therefore, it is used as

Step 1:	Translate, rotate and scale $\hat{X}_{1:N}^0$ to fit the 2D data
Step 2:	Set $i = 1$ .
Step 3:	Gradient descent: $\hat{X}_{1:N}^i = \hat{X}_{1:N}^{i-1} - \lambda \nabla_{X_{1:N}} E _{\hat{X}_{1:N}^{i-1}}, \quad 0 < \lambda \leq 1.$
Step 4:	Enforce constraints: $\hat{X}_{1:N}^i \in \mathcal{M}_N$ .
Step 5:	Increment $i$ by one and goto Step 3. (until convergence)

Figure 4.6: The iteration steps involved in finding  $\hat{X}_{1:N}$ .

the initial guess for the solution of the minimization problem posed in equation (4.3). Figure 4.6 gives an outline of how the minimization proceeds. At the end of each iteration, enforcing  $\hat{X}_{1:N}^i \in \mathcal{M}_N$  is approximated, by resetting the limb lengths to their correct value, and applying a low-pass filter to the trajectories of each joint. A large  $\lambda$  yields faster convergence, but makes it more difficult to reproject the solution back onto  $\mathcal{M}_N$ .  $\lambda = 0.2$  was again used for these experiments. Figure 4.7 shows how a 3D key frame is refined in order to match the image data. Here the same 3D key frame is modified to form two different reconstructions to match different forehand frames, capturing the subtle differences between the two forehand strokes.

## 4.6 Results

In order to examine the correctness of the reconstructions, the original tennis sequence was recorded from two viewpoints. The footage of one of the cameras was used for reconstruction, while the other was used in order to compare the result from another viewpoint. This setup is illustrated in fig. 4.8.

**Robustness:** As mentioned before, by continuously reprojecting the evolving solution

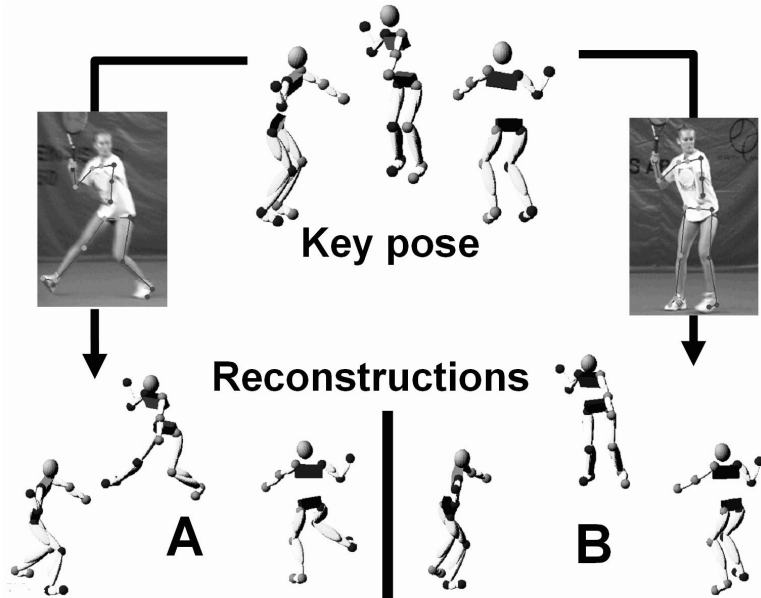


Figure 4.7: Result of refining a key pose based on image data. The key pose is refined according to the different images, resulting in two different 3D poses, A and B. Each pose is shown from three different viewpoints.

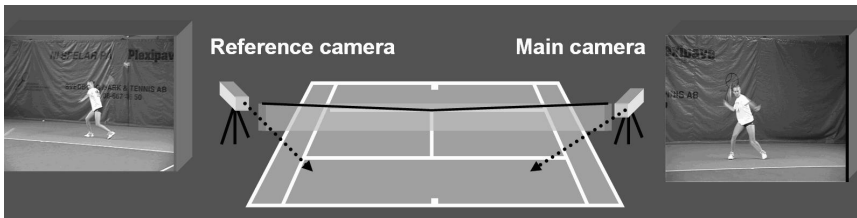


Figure 4.8: The camera positions used in the experiment

back onto the manifold, the algorithm will be able to handle frames where the tracking has gone wrong. In fig. 4.9 some examples of frames where the tracking has obviously failed in delivering correct data are shown. Nevertheless, the reconstructed poses appear to be correct.

**Long sequence:** In fig. 4.10 the footage and reconstruction of a 36 seconds long tennis sequence is shown. The sequence is sub-sampled, so that every 50:th frame is shown.

**Selected stroke:** In fig. 4.11 The resulting reconstruction of one selected forehand stroke is shown.

## 4.7 Concluding remarks

The idea of key framed reconstruction is a good complement to the ideas presented in the previous chapter. It seems like a system intended for 3D reconstructions must be able to handle different amount of manual intervention, depending on the situation. In cases where strong priors are available, little manual intervention should be required. However, when reconstructing a completely unknown motion sequence, some human knowledge must be added to the system. In this chapter, this is done by selecting a few very characteristic poses of the sequence, and reconstruct these manually. Also, the definition of the manifold of allowable configurations is not a trivial task. A system utilizing this approach should provide the user with a set of tools for this job. Such tools could be various strategies of learning priors from examples. In this chapter, the manifold was primarily defined by using symmetry and constant limb length constraints. Also, smoothness conditions play an important role. For the future system being developed, one big task is to design a plausible GUI providing valuable tools for the creative sculptor to carve manifolds.

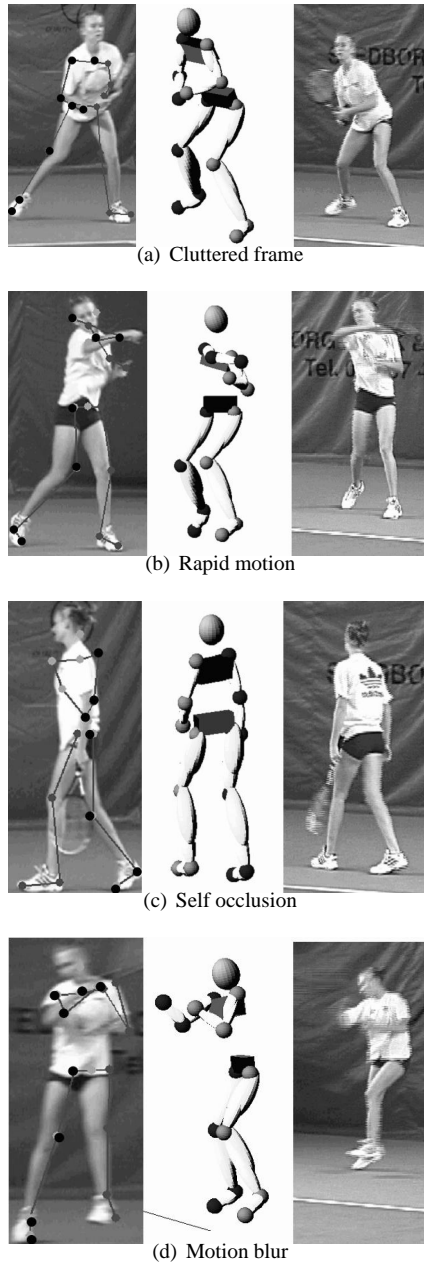


Figure 4.9: Examples of how problematic frames can be reconstructed despite erroneous tracking.





Figure 4.10: Result of the reconstruction of the entire sequence. Every 50th frame of the 36s long sequence is shown together with the image from the reference camera.

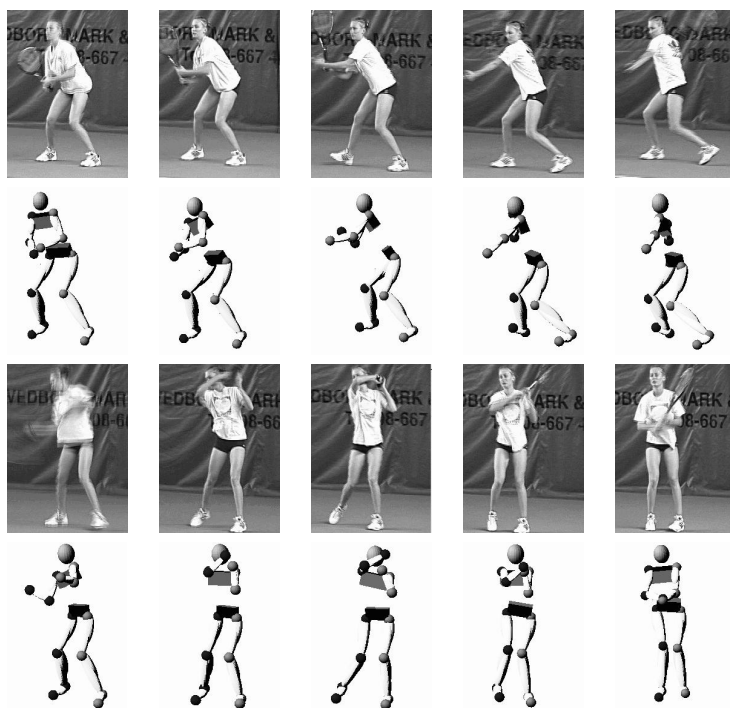


Figure 4.11: Result of the reconstruction of one of the forehand sequences, together with the footage from the reference camera.

## Chapter 5

# Reconstruction by qualitative selection

In this chapter, the discussion about qualitative constraints, as were introduced in chapter 3, continues. The problem of single view reconstruction will remain in focus, but the problem is now formulated as a discrete search problem, rather than as an optimization problem. As pointed out before, monocular reconstruction is an ill-posed problem, since depth information is irretrievably lost due to the imaging process. Despite this, many systems (as mentioned before) still perform quite well in this quest. Generally their success can be attributed to a combination of a good model of the motion and good prior data. The best systems today are generally able to maintain a 3D reconstruction for a few seconds, before slipping into erroneous configurations. The motions being reconstructed are generally quite smooth, like walking or jogging. One of the main difficulties is the initialization of the model. In the previous two chapters, we have seen how information about exemplar motions can be used in order to initialize the reconstruction. This is strongly related to the initialization problem in model based tracking. Given a correct pose in one frame, smoothness priors can perform quite well in updating the model according to the image data. At least for a while, until the model drifts off to an erroneous minima. Note that the erroneous minima can actually be a global minima, but still be incorrect. The reason for this being that the cost function does not perfectly reflect the human motion constraints. In most cases, such erroneous minima can be identified by inspection. Quite often, though, there are multiple solutions to the reconstruction problem that actually look equally good. In such cases, it becomes necessary to play a guessing game, and the luck becomes an essential factor for success. The good news in such a case is that since the motion was only recorded from one view, nobody can prove that the reconstruction actually picked the wrong configuration (unless the researcher admitted that he or she also used a reference camera!). In this chapter, the discussion about how to use prior information continues. The problem is discretized into a search problem, where the task is to search for the most likely configuration of a model, given some image data. The most common pitfalls are discussed and exemplified, and a new approach to select the most likely configuration based on a qualitative measure is presented.

## Comments on the results

It is worth mentioning at this point, that the purpose of this chapter is not to present a full system for monocular reconstruction. The purpose is to yet further emphasize the power of automatically learned qualitative constraints. The methods presented are, however, very useful in rating the likelihood of a candidate solution to a reconstruction problem. The methods also appear to have potential in other areas, which require shape to be represented in a broader, and more qualitative way.

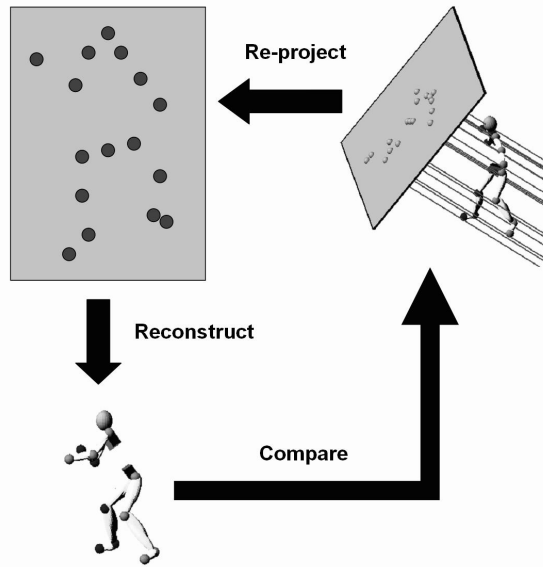


Figure 5.1: The generic setup for the experiments in this chapter. One motion capture sequence is projected using a synthetic orthographic camera. The reconstruction algorithm is executed on the projected data. The result is then compared to the original sequence.

## 5.1 Evaluation

The most common evaluation criteria for a reconstruction is its visual appeal. If the reconstruction looks good, it is generally considered to be correct. Unless the purpose of the reconstruction is to perform motion analysis in 3D, this highly subjective measure is probably the most effective. Nevertheless, it is sometimes interesting to compare a reconstruction to the ground truth, in order to verify the correctness in a more objective fashion. One method to facilitate this is to record the motion with a motion capture system while filming it, in order to use the video to perform the reconstruction, and the motion capture

data to compare the results with. The discussion here will simplify things even further, by projecting the motion capture data to a synthetic 2D image, on which the monocular reconstruction is performed. This eliminates many sources of errors and makes it easier to rate how well a discrete configuration selection method performs in picking the correct motion. The process is illustrated in fig. 5.1.

## 5.2 Reconstruction as a selection problem.

As most approaches to reconstruction have to assume a large amount of uncertainty in terms of image data (tracking has a tendency to occasionally fail), the optimization problem is generally formulated in order to generate a well-behaved cost function combining image data and prior likelihoods about the model. If we allow ourselves to deal with perfect image data, we can use the fact that the number of possible binary configurations of the reconstruction is bounded by  $2^L$  where  $L$  is the number of links of the articulated chain used to model the human skeleton. Depending on the model, this number generally becomes quite large. For a sequence of  $F$  frames, the number of possible reconstructions explodes to  $(2^L)^F$ . This enormous search space can be pruned by imposing the physiological limitations of the human body (Sminchisescu and Triggs, 2003) and bounding the motion between adjacent frames. In (Park et al., 2002) this problem was solved by fitting a reference motion to the image data and using the binary configuration from the reference motion. While sometimes giving good results, this approach can be a bit problematic in some situations since the algorithm immediately commits to one of the  $(2^L)^F$  configurations (the configuration of the reference motion). Another problem is that the orientation of the reference motion with respect to the image plane has to be perfectly computed in order for the configuration to be correct. Small errors in estimating the orientation of the pose generally lead to suspicious configurations in the final reconstruction. In chapter 3, the problem of these suspicious configurations were fixed by enforcing qualitative constraints, which involved reprojecting the motion back to a manifold of allowable configurations. The purpose of the study in this chapter is to present the problem of reconstruction as a discrete selection problem, rather than a classical optimization problem. Further, this formulation leads to a new measure in order to compare the similarities of shapes. The qualitative constraints introduced in the previous chapters will be shown to possess some appealing properties in order to compare human motions. The formulation of the reconstruction as a selection problem is divided into the following two tasks:

1. Prune the search space, in order to rule out the vast majority of the possible configurations at an early stage.
2. Design a measure that picks the best of the possible configurations, based on priors.

Both of these issues involve the use of priors which again will be a number of example motions. As before, some of the constraints will be automatically identified from the training data. Whether the first step (pruning the search space) is required or not, depends on the complexity of the skeletal model of the human, but also on the complexity of the selection criteria in step 2.

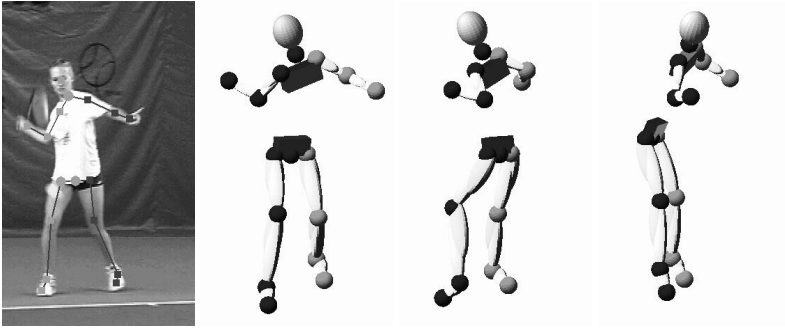


Figure 5.2: Different binary configurations. The left image shows one frame of a tennis sequence together with the extracted feature points. The second image is a reconstruction that appears to be correct. The third image show a reconstruction that is correct with respect to the image data; however, it possesses some apparent errors, violating physical constraints of the human body. The last image shows a reconstruction that is also correct with respect to the image data, and does not violate any physical constraints. However, it violates some specific constraints of the pose it is supposed to reconstruct.

## Limitations

In the previous chapters we have seen the qualitative properties pose constraints primarily on poses only. Even though some temporal constraints have been identified, some work remains in order to make the computation of the temporal constraints efficient enough for most practical applications. In this chapter, we have the similar situation; the qualitative constraints will be used to select likely poses at each time frame. In order to compile these likely poses into one coherent motion, smoothness constraints will be used. However, the main contribution of this chapter is not to generate perfect motions, but rather to demonstrate how a qualitative similarity measure can be used to select likely poses from a set of hypotheses.

## 5.3 Pruning the search space

The reconstruction obtained from most binary configurations are obviously wrong. In fig. 5.2 three reconstructions of the 2D points representing a pose from a tennis sequence are shown. All reconstructions in the figure are consistent with respect to the image data. However, two of them upon inspection are wrong. One of them obviously violates some physical constraints of the human body, while one of them violates constraints for the specific pose (it does not look like a tennis pose). Both of these erroneous configurations should definitely be removed from the search space, and preferably at an early stage. Evaluating all the  $2^L$  possible solutions to the reconstruction would be tough to handle for a human skeleton of 15 links. In this section, it is shown how to make an early rejection of im-

possible poses, given their 2D projection. To avoid considering all possible configurations and then pruning the majority of them out, the approach must be constructive and involves generating the possible configurations.

The method used in order to achieve this task uses the fact that the orientation of one limb with respect to the image plane is highly dependent of the configuration of other limbs, within a class of motions. These dependencies can be identified by orienting a synthetic image plane in a large number of ways with respect to the example motions.

## 5.4 Configuration dependencies

This discussion begins by reviewing Taylor's method for manual 3D reconstruction, explained in the introduction. This method is used in this chapter, with the goal of automating the disambiguation of the binary configuration of the reconstruction. The model of the human body is represented in the same way as in the proceeding chapters; an articulated chain with 15 joints connected by rigid links. A motion sequence is a point set,

$$X_{1:N} = \{X_1, X_2, \dots, X_N\}$$

where each

$$X_f \in \mathbb{R}^{3n} = \{p_1, p_2, \dots, p_n\}$$

is the pose in frame  $f$ , represented by a selection of human joints, and  $p_i \in \mathbb{R}^3$  is the 3D point of joint  $i$ . Further, the human skeleton is modelled as an articulated chain with  $L = n - 1$  links, connected at the  $n$  rotational joints.  $X_f$  can be projected onto the image plane  $\Pi$ , using orthographic projection, in order to obtain  $X_f^P$ . Given  $X_f^P$ , the number of qualitatively different reconstructions,  $X'_f$ , is bounded by  $2^L$  (assuming orthographic projection and known limb lengths), as each link can point either towards or away from the image plane. Earlier, the term binary configuration has been introduced, which is the state of an articulated chain with respect to the image plane  $\Pi$ , as illustrated in fig. 1.3. For a point set,  $X_f$ , representing the pose of the motion in frame  $f$ , its binary configuration with respect to the image plane,  $\Pi$ , can be represented as a string

$$b_f \in B_f = \{0, 1\}^L$$

where 0 means that the limb points outwards (away from the image plane) and 1 means that the limb points inwards (towards the image plane). Given  $X_f^P$  of the projected joint locations of the person in frame  $f$ , the reconstruction,  $X'_f$ , can be computed using Taylor's method:

$$X'_f = \mathcal{R}_T(c, b_f, X_f^P) \quad (5.1)$$

where  $c$  contains the parameters of the human model (order of the limbs and the limb lengths) and  $b_f$  is the binary configuration with respect to  $\Pi(X_f^P)$ , the image plane of  $X_f^P$ .  $B_f$  is the set of all possible binary configurations. Without any restrictions,

$$|B_f| = 2^L \quad (5.2)$$

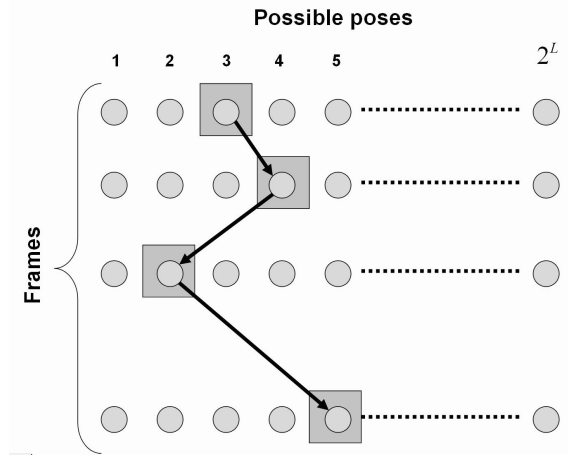


Figure 5.3: Each of the circles represent a possible pose (configuration of the model) in a certain frame. The objective is to find a path from the first to the last frame, that is most likely to generate a correct motion. Obviously, an exhaustive search through this space would be computationally expensive.

resulting in the cumbersome search problem as previously discussed (see fig. 5.3). We now use the fact that the binary configurations of the links are strongly dependent on each other. These dependencies can be identified by investigating a number of examples. We have introduced the reconstruction operator  $\mathcal{R}_T(c, b_f, X_f^P)$ . Let's further define the inverse operator as

$$b_f = \mathcal{R}_T^{-1}(c, \Pi, X_f) \quad (5.3)$$

which is the binary configuration of  $X_f$  with respect to the image plane  $\Pi$ . Given a set,  $\mathcal{T}_f$ , of example poses for the pose in frame  $f$ , the list of all binary configurations that can be obtained from the examples will be a subset  $B'_f \subseteq B_f$  given by

$$B'_f = \bigcup_{X \in \mathcal{T}_f} \bigcup_{\Pi} \mathcal{R}_T^{-1}(c, X_f, \Pi) \quad (5.4)$$

To compute  $B'_f$  in a systematic fashion, we use the fact that given an articulated structure of  $L$  limbs that is randomly rotated in front of a plane  $\Pi$ , its binary configuration with respect to  $\Pi$  only changes when two limbs are simultaneously parallel to  $\Pi$ ; something that occurs  $L(L - 1)$  times. Every time this happens, 4 new configurations are added to  $B_f$ , due to the 4 configurations after infinitesimal rotations around the two limbs that are parallel to  $\Pi$ . This is illustrated in fig.5.4. As the number of examples are added, the number of possible configurations increase. However, the number of new configurations declines with every new example, due to their structural similarity. This is illustrated in the graph in fig. 5.5, where possible binary configurations of one pose of a number of walking



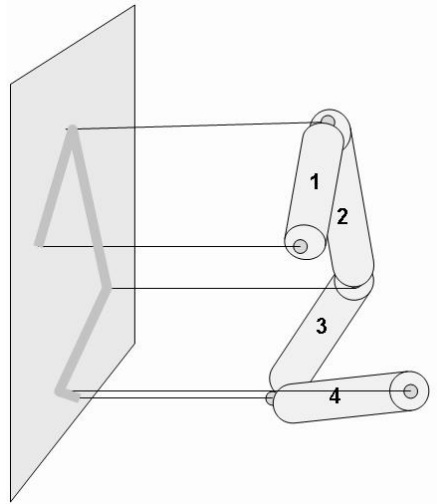


Figure 5.4: A changeover in configuration. In this case, limb 1 and limb 2 are both parallel to the image plane. If limb 1 is the root segment, limb 3 points towards the image plane, while limb 4 points away from the image plane. Any infinitesimal rotation (except for rotations around limb 1 and limb 2), of this structure will put it into one of the following four binary configurations:  $[0, 0, 1, 0]$ ,  $[0, 1, 1, 0]$ ,  $[1, 0, 1, 0]$ ,  $[1, 1, 1, 0]$

sequences are plotted versus the number of example poses<sup>1</sup>. As new examples are added, the increments in the number of new binary configurations are reduced. At the end, when all 9 training shapes are taken into account, 742 possible binary configurations were found. This is a large improvement from the original exponential complexity.

## 5.5 Selecting the correct sequence

From the computation explained in the previous section, a set of possible reconstructions is obtained for each pose of the sequence. We want the reconstruction to somewhat resemble the training data, but also be faithful to the image data. Taylor's method ensures that the reconstruction, no matter which configuration we chose, will be perfectly consistent with the image data. Given this, the only remaining task is to select the reconstruction that is most likely to be correct. This is now a feasible task, since we have strongly reduced the number of possibilities. The remaining issue now is the *definition of correctness*. We want the reconstruction to be qualitatively similar to the training data. Therefore we need to capture the essence of each pose (in this case the poses of a walking sequence) in a

<sup>1</sup>The walking sequences are from the CMU motion database.

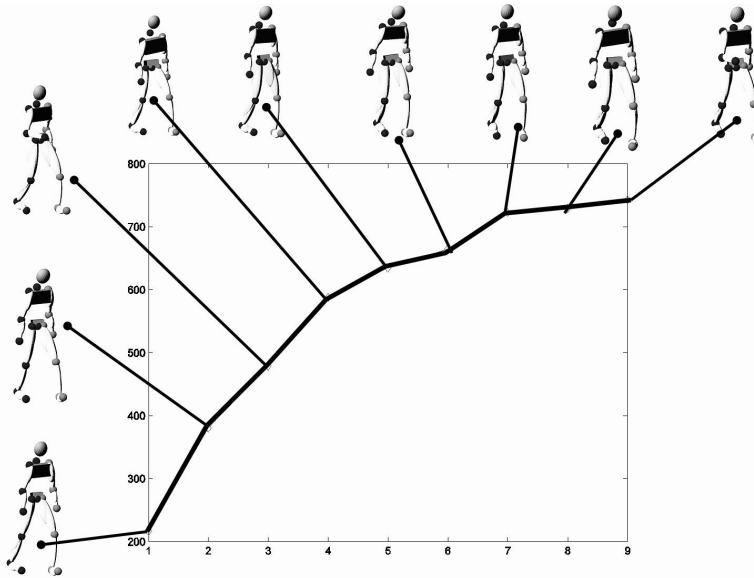


Figure 5.5: The number of possible binary configurations plotted versus the size of the training set. As new examples are added, new possible binary configurations are added.

more generic way than just comparing its euclidian distance to one reference motion. As a consequence, the normal constraints of the human body should also be enforced (elbows don't bend backwards, etc.).

### Qualitative measure

Earlier, the concept of qualitative constraints were introduced and exploited for the purposes of monocular reconstruction. In this chapter, the qualitative constraints are used to select the most appropriate point set from the set of hypotheses contained in the pruned search space. In principle, the same definition of the manifold of allowable configurations introduced before is again used here. However, the formulation is slightly changed. The discussion begins by reviewing the definition of allowable configurations, and then define a similarity measure that represents the qualitative distance from this manifold. Again, each pose throughout the motion is regarded as a separate shape, and the constraints computed on each individual pose. Previously, a shape (represented as a point set) was considered to belong to a certain class if each constraining  $n$ -tuple (quadruples in the case of 3D shapes) of points were of the same order type as the analogous  $n$ -tuples in the example shapes. Again, if we consider a quadruple of points in the point set, we define the order type of that quadruple as the sign of the determinant of the matrix with these points as its columns (in homogeneous coordinates). The geometric interpretation of this is shown in 3.16. The

configuration of a point set,  $X$ , can be represented as a string

$$C_X \in \{\{-1, 1\}^{\binom{n}{4}}\} \quad (5.5)$$

where  $n$  is the size of the point set and the symbols are assigned as

$$C_X = \left[ \text{sign} \left( \begin{vmatrix} p_1 & p_2 & p_3 & p_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} \right), \dots, \text{sign} \left( \begin{vmatrix} p_{n-3} & p_{n-2} & p_{n-1} & p_n \\ 1 & 1 & 1 & 1 \end{vmatrix} \right) \right] \quad (5.6)$$

We can define a distance measure in this configuration space, in order to measure the qualitative similarity of two point sets  $X$  and  $Y$ . In this work the hamming-distance of the strings encoding two configurations is used as distance measure.

$$d(X, Y) = h(C_X, C_Y) = \sum_{i \leq \binom{n}{4}} |C_X[i] - C_Y[i]| \quad (5.7)$$

In other words, two point sets are identical if every possible quadruple of points have the same order type. One possible cost function to use in order to select the best reconstruction is to take the sum of the distances between the reconstruction and all shapes in the training set, individually. However, this method will not reflect the property of the shape class optimally, as the majority of the point quadruples are not very characteristic for the class. Because of this, we use the set  $\mathcal{T}$  of training shapes in order to identify the most characteristic quadruples. These are defined to be the quadruples belonging to the same order type in *all training shapes*. If we denote the set of these indices as  $J$ , the new cost of a shape is defined as

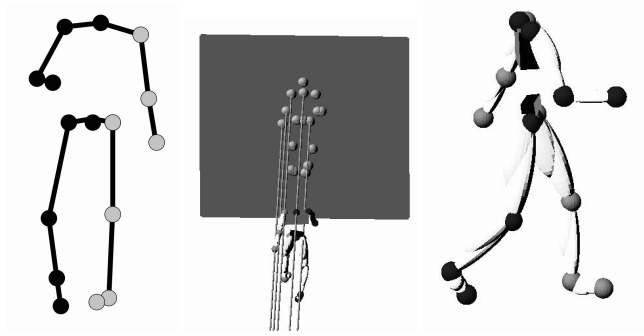
$$c(X) = \sum_{j \in J} |C_X[j] - C_Y[j]| \quad (5.8)$$

for any  $Y \in \mathcal{T}$ .

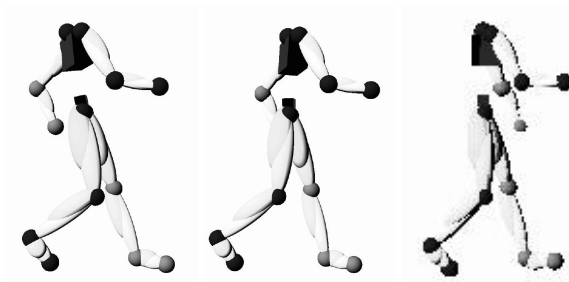
The main reason for proposing a qualitative selection criteria rather than, for example, a euclidian approach is to avoid very awkward configurations. This becomes particularly useful when the motion to be reconstructed is significantly different from the example motions, as discussed next.

### Why a qualitative measure?

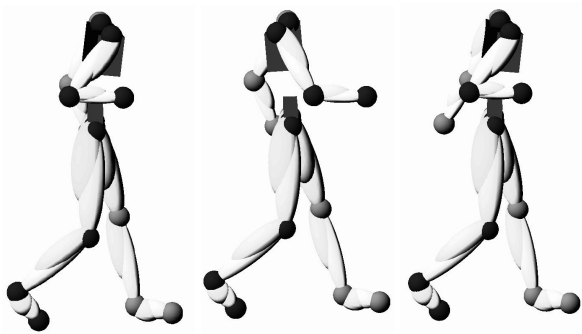
This is an interesting question, and also the essence of this chapter. What is wrong with using another perhaps more intuitive measure to rate the likelihood of a configuration? Such a measure could be the sum-of-squared distances between the hypothesized pose and all the other corresponding poses in the training shapes. Generally, it turns out that with fairly noise free data, such a euclidian measure does virtually the same job as the qualitative approach. This situation is illustrated in fig. 5.6. In this example, the qualitative constraints were computed using the poses in fig. 5.5. One pose, not used for computing qualitative constraints, was projected down to 2D, as illustrated in the figure. The most likely poses according to the qualitative measure as well as to the euclidian measure are shown. By visual inspection, most poses seem to be quite possible for a walking person. In fact, the



(a) One walking pose was projected using the synthetic camera shown in the middle. The skeleton to the left shows the resulting 2D projection. To the right is shown the original frame viewed from the side.



(b) 3 of the best poses according to the qualitative measure.



(c) 3 of best poses according to the euclidian measure

Figure 5.6: One walking pose was projected to a 2D point set. All possible reconstructions (assuming known limb lengths) were constructed. The best (most likely) pose was then selected using a qualitative measure (b) and a euclidian measure (c). Both measures render realistic solutions. No measure picked the exact correct pose, that is shown in (a). This is because the walk to be reconstructed was quite different from the walks in the training sequences.

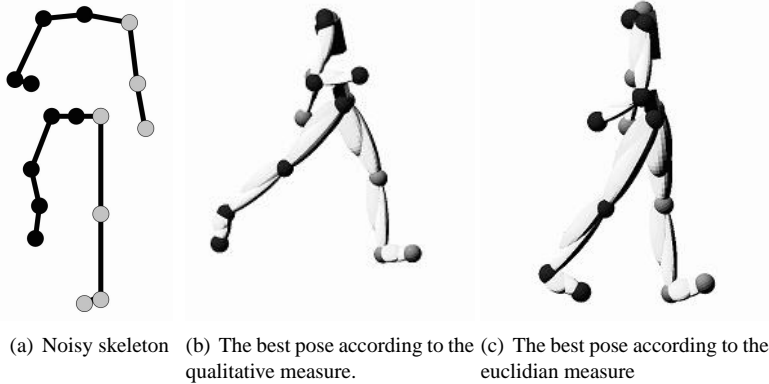


Figure 5.7: In this example, the projected pose has been prepared with some obvious noise. The resulting pose is quite far from the poses in the training sequences. In such a scenario, the qualitative measure is more successful in selecting a pose that intuitively seems to be more correct. When using a euclidian measure, the pose selected contained obvious errors, such as the arm bending backwards.

qualitative poses are generally more similar to the true pose (also shown in the figure) than are the euclidian selected poses - however, this can probably be accounted to pure luck. Now, what happens if we manipulate the projected joints, and manually introduce an obvious error? This is illustrated in fig. 5.7, where the right leg (assuming the person is walking towards the camera) is behaving funny. In this case, the qualitative measure selects a much more likely pose than the pose selected by using the euclidian measure. In the latter case, we can see that one arm bends backwards. Also interesting to note is that the legs of the reconstructions using the euclidian criterion tend to be very similar to the legs of the reconstructions in the unperturbed case. By using a qualitative selection criteria, the behavior of the lower body does not look like a regular walking pose; however, in this case the 2D data does not represent a normal pose according to a euclidian measure, which means that the reconstruction should not do so either. In principle, the euclidian approach tries very hard to make everything look similar to the examples without paying any attention to qualitative constraints. According to euclidian measure, bending the arm backwards is a perfectly acceptable trade-off to make the lower body look as it usually does in the examples. For a human observer, this trade-off is generally not acceptable.

Another way of illustrating the strength of a qualitative measure is to analyze what happens to the dissimilarity value when transforming one pose (the reference pose) into a significantly different pose. Even though the transition will change the pose significantly in a euclidian sense, the qualitative measure will not necessarily be affected, unless the change is in fact a qualitative change. This is illustrated by the following two shape metamorphoses:

- Transition from a walking pose to a running pose (qualitatively similar poses).
- Transition from a walking pose with the left foot in front of right foot to the mirrored pose (the right foot in front of the left foot).

Both these metamorphoses are shown in sub-sampled form in fig. 5.8. In the first shape metamorphosis, all the shapes are intuitively similar in a qualitative sense, since they all correspond to various styles of a walking pose, as running can be considered as a sort of walk. In the second metamorphosis, the start and end pose are considerably different, since the end pose is the complete opposite to the start pose (left foot and right arm in front in the start pose, and the opposite at the end pose). Those correspond to strong qualitative changes throughout the metamorphosis. In fig. 5.9 the result of a euclidian and qualitative dissimilarity measure is shown respectively, for the two metamorphoses. The duration of both metamorphoses were 100 frames. The dissimilarity between each pose in the metamorphoses and the reference pose, shown inside the graphs. As can be seen, the change in euclidian similarity is rather similar throughout both metamorphoses. When using the qualitative measure, on the other hand, the dissimilarity of the beginning and end pose is very small in the first metamorphosis, and very large in the second. This should indicate that the automatically identified qualitative constraints do in fact capture some of the qualitative properties of the motion.

## 5.6 Forming the final reconstruction

After implementing the previous step, we now have a relatively small number of hypothesized poses for each frame. Each of these poses are qualitatively correct, and they also obey the image data perfectly. Randomly selecting one hypothesis from each frame would thus result in a reconstruction that is correct at each frame; however, such a sequence may not be very smooth. In other words, we still need to select a path through the search space of fig. 5.3. In order to generate a smooth motion, the last selection criterion is to simply chose the sequence (path) with the smallest euclidian change between poses. At this stage, all configurations are qualitatively correct (or at least as correct as possible), which means that a euclidian measure no longer can cause violation of crucial constraints. The cost of a sequence can thus be given by

$$C(X_1, X_2, \dots, X_F) = \sum_{f=2}^N \|X_f - X_{f-1}\|^2 \quad (5.9)$$

In fig. 5.10 the results from a reconstruction, where the 2D data was acquired by orthographically projecting a walking sequence using the same camera setup as in fig. 5.6 is shown.

### Bias towards examples

Another interesting aspect when evaluating a reconstruction is how well it reflects individual differences between sequences. In other words, a good reconstruction method should

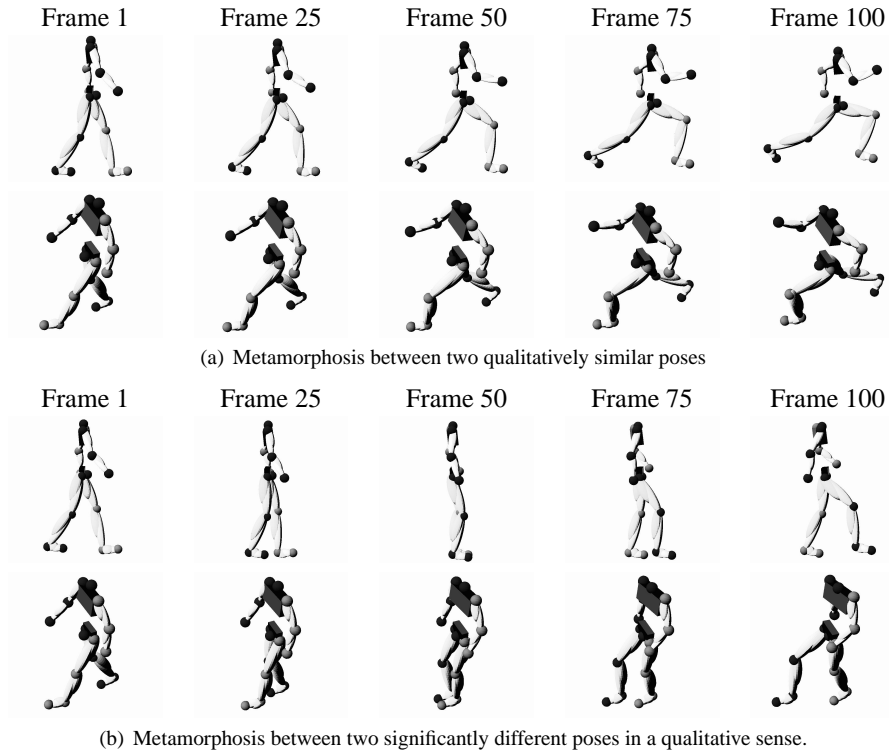
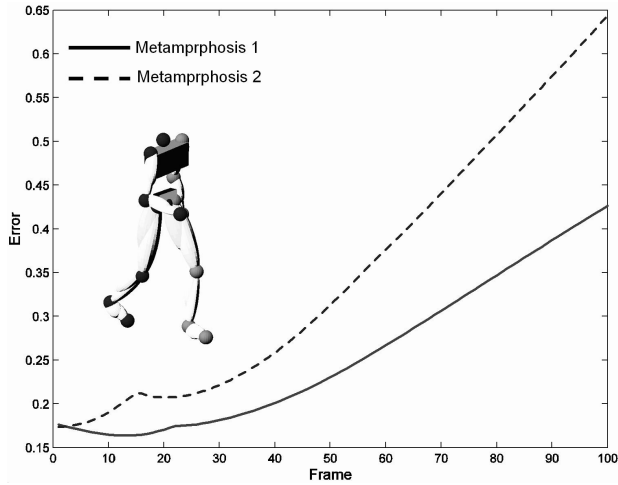
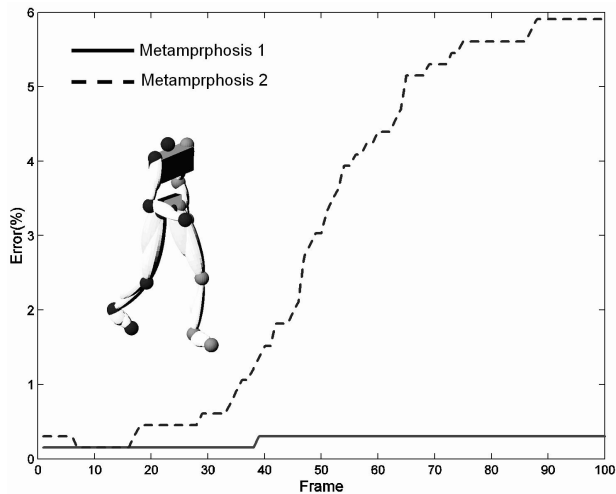


Figure 5.8: Metamorphosis of one pose into another. Each metamorphosis is shown from two different viewpoints.(a) shows a smooth transition between a quite normal walking pose into a quite unusual, but yet, walking pose. In (b) a transition between two poses that differ strongly in a qualitative sense is shown. This metamorphosis will show that the qualitative measure easily identifies such differences. Each metamorphosis is shown from two angles.



(a) Euclidian distances



(b) Qualitative distances

Figure 5.9: Distances from the reference pose using two different measures. In (a) the euclidian dissimilarity of each pose in each metamorphosis sequence with respect to the reference pose is shown. In (b) the qualitative dissimilarity is shown. The unit of the qualitative dissimilarity is percentage of violating point quadruples, with respect to the example pose. From the euclidian distances, we can see that the dissimilarity increases in both metamorphoses, while the qualitative measure indicates strong similarity between all poses in the first metamorphosis, but an increasing error in the second. This indicates that our identified measure is able to distinguish between shapes in a qualitative sense.



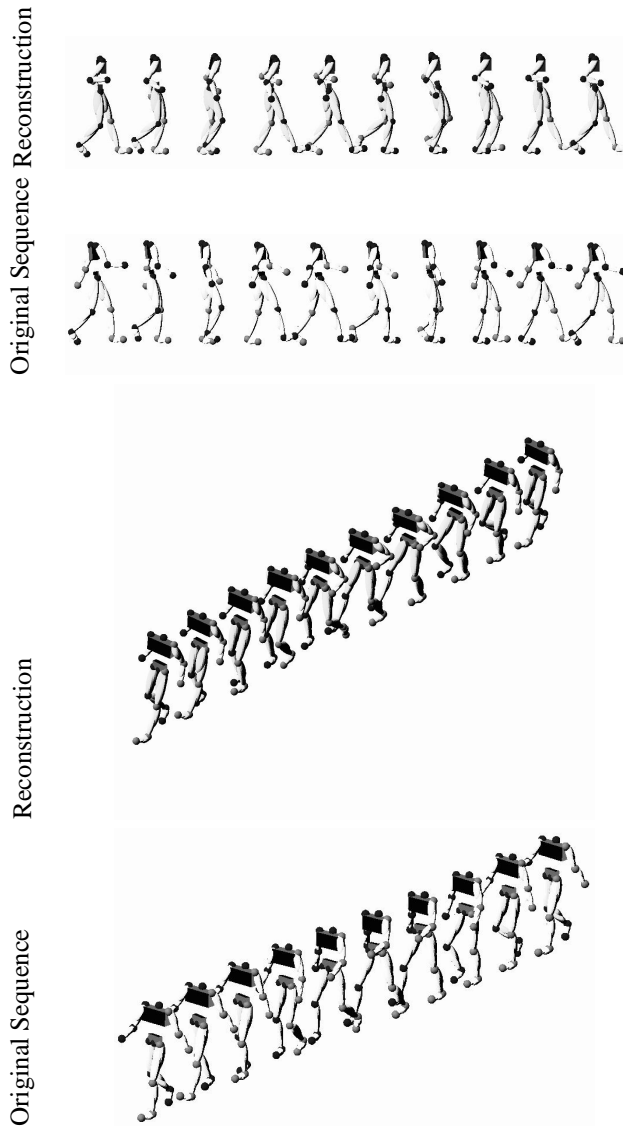


Figure 5.10: The result of one monocular reconstruction from two viewpoints, together with the original sequence. The original sequence was projected to 2D by using a virtual camera in front of the person, on which the reconstruction was performed. Each pose in the reconstructed sequence appears to be correct. However, the reconstruction differs from the original sequence in many respects - for example the bending of the right arm.

<b>1.4848</b>	2.5998	2.5665	2.4362	2.4830	2.3053	2.7052	2.6800	2.2361	2.4768	2.5724
2.5593	<b>1.9543</b>	2.1859	2.5595	2.1107	3.1863	2.5156	2.5644	2.2524	2.4351	2.3067
2.6016	2.2955	<b>1.8980</b>	2.2543	2.1730	2.9472	2.4263	2.3898	2.1383	2.1492	1.8550
2.0721	1.8950	1.8228	<b>0.9418</b>	1.9154	2.6997	1.5355	1.5902	1.4562	1.4165	1.6546
1.9278	1.3163	1.5787	1.7533	<b>0.8247</b>	2.4335	1.8190	1.8857	1.6508	1.6358	1.6502
1.5263	2.3397	2.0908	2.2475	2.1580	<b>0.6308</b>	2.3563	2.1949	1.9813	1.9673	2.0941
2.2115	2.1214	1.8918	1.5338	1.8845	2.5593	<b>1.2107</b>	1.7721	1.7354	1.5888	1.8248
2.2752	2.1689	2.1121	1.6496	2.0094	2.6827	1.7583	<b>1.3933</b>	2.0260	1.7527	2.0236
1.9088	1.8291	1.8298	1.5965	2.0086	2.6199	1.9965	2.2000	<b>1.2693</b>	1.7782	1.7982
2.5408	2.2901	2.3326	1.9475	2.4055	2.9726	2.3009	2.2823	2.0549	<b>2.0338</b>	2.2197
2.0918	1.8719	1.2568	1.5749	1.8406	2.4887	1.9042	1.9676	1.3932	1.5651	<b>1.0762</b>

Figure 5.11: The similarity matrix. The bold numbers are generally the smallest numbers in the row, indicating an unbiased reconstruction.

be unbiased to the training data. In order to verify how biased the reconstructions obtained from the method explained in this chapter are, the experiment was repeated 11 times; one time for each of the projected sequences from the set of examples, in order to reconstruct each sequence. The sequence that was being reconstructed was of course never used as example for that experiment. The reconstruction was compared to the original sequence on a frame by frame basis. We used a least squares distance between each frame to compare two sequences. A low value means similar shapes. The results are summarized in the table in fig. 5.11. Each row indicates the similarity between a reconstruction and all the original shapes (one shape per column). If we consider one row at a time, we can see that the best match is generally obtained when the reconstruction is compared to the sequence we used to generate the 2D data. These values are the diagonal elements of the matrix, and are shown in bold. This indicates that the reconstruction method is, to some extent, able to identify individual variations. In the reconstructions shown, the limb lengths have been computed as the average of the length in the example sequences.

## 5.7 Summary

In this chapter a method to systematically reduce the enormous search space involving all possible configurations given the projection of an articulated chain has been outlined. Euclidian measures have been avoided in order to rate the likelihood of a hypothesized pose. This is done to prevent the solution from ending up in physically impossible configurations, or configurations significantly deviating from the actual motion. By using the qualitative constraints, the essence of a class of shapes has been captured by using a small set of example shapes. By doing this, the need to manually code any constraints, such as joint limits or other specific constraints known from the shape has been eliminated. All constraints are automatically learned from the examples. The problem of optimizing the limb lengths has not been addressed in this chapter. Doing this should lead to great visual improvements in the final reconstruction, and it would be a nice continuation of this work. However, this was not needed in order to demonstrate the power of qualitative constraints. Also, it must be emphasized that the experiments have intentionally been executed on perfect data, in order to evaluate the potential of the qualitative constraints. In reality, 2D

data used as input will be somewhat perturbed with noise, especially when dealing with automatically tracked data. The work presented here is not powerful enough by itself in the quest for nice looking reconstructions given noisy input data. However, the results may become very useful as part of coming reconstruction systems. Further, the principle of qualitative constraints may possess power whose applicability extends far beyond the problem of human motion reconstruction.



## Chapter 6

# 2D vs. 3D data for motion based recognition and classification

The motion pattern of specific body locations of a human performing a specific action contains information about the identity of the human and the nature and quality of the action (Moeslund and Granum, 2001). Analyzing motion patterns therefore have potential applications such as:

- gait based identification
- sports action coaching
- biomedical analysis

These motions take place in 3D space and the analysis should ideally be based on 3D information. If this is done using video it will require multiple cameras with relative calibration and establishment of correspondence between cameras. Using just a single camera would imply a much more flexible system for analysis. The question then arises: What information is lost if human motion patterns are analyzed in a single 2D view instead of in 3D ? It can be expected that this depends on the problem and the kind of analysis that is performed and it is not always obvious how to measure “information content” in a pattern. In this chapter a general measure of motion similarity is used in order to demonstrate that similarity structure is to a large extent preserved when projected 2D data is used instead of original 3D data. Using sets of similar motions in 3D acquired from motion capture systems, the similarity structure of the set in 3D in terms of distances between motions is measured. This produces a matrix of dissimilarity values for the 3D motion. By repeating the same procedure for the body locations projected to an image from a certain viewpoint , we get a corresponding matrix in 2D, as shown in fig. 6.1. It turns out that if a proper viewpoint is selected, these matrices have a very similar structure, indicating that they encode the same information about the motion pattern. In the next chapter, it is shown that using automatically tracked body locations similarity structure is also preserved, although to a less extent, indicating that automatic tracking of body locations in a single view is a

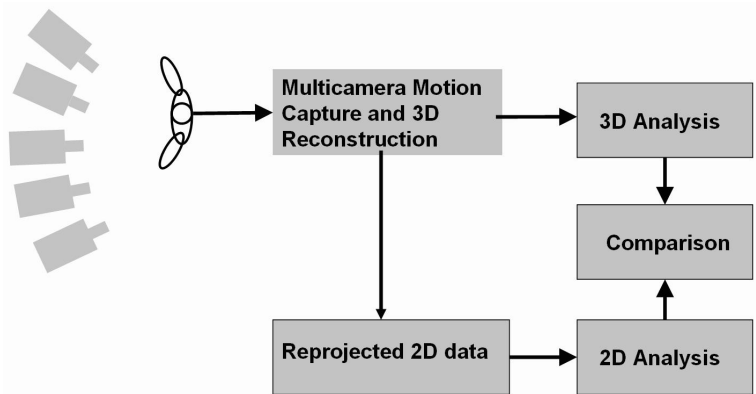


Figure 6.1: The purpose of this work is to investigate to what degree 2D data of human motion delivers the same information as the original 3D motion acquired by a motion capture system

potential tool for analyzing and classifying human motion. The relationship between 2D and 3D motions has been studied before, in general terms (Brand, 1999), as well as for specific applications (Yacooob and Black, 1998; BenAbdelkader and Cutler, 2002).

## 6.1 Comparing motion

Comparing motions is a very task dependent activity. Usually, we are looking at a particular, well defined, feature that in some sense classifies the motion. In the case of athletic coaching, we also have some general idea of the appearance of this feature in an optimally performed trial. For example can we compare sprinters by their knee drive, and use the knowledge that world class sprinters generally lift the knees higher than recreational runners. We also know that a gymnast performing a routine in the high bar strives to maintain a straight pose of the body, in order to score high points. Another way to rate an exercise is by the "general appearance". This is a very soft measure, but nevertheless the most commonly used in everyday life. When we recognize a friend from a distance by his or her gait, we rely solely upon general appearance. This is also the case for the general audience when watching a ballet performance. The reason why we perceive one motion more aesthetic than another is still somewhat an open problem. When developing computer systems for action recognition or automatic performance assessment, the similarity measure has to be defined properly, so that the computer can evaluate it.

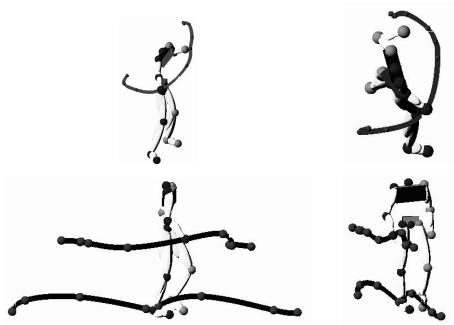


Figure 6.2: Examples of trajectories for particular joints. In the tennis sequence (top) the trajectory of the right hand is shown from two different viewpoints. In the walking sequence (bottom) the trajectories of the right ankle and left hand are shown.

### Similarity measure

The purpose of this chapter is not to develop a universal motion similarity measure, but rather to use a very generic measure, and evaluate how well similarities are carried over when going from 3D to 2D. Therefore a sum-of-squared differences between corresponding points in the two motions is used. As before, a motion is a sequence of poses through time. By using this representation, it is possible to regard the motion as a set of trajectories, where a trajectory is the motion of one joint. Such examples are depicted in fig. 6.2. The dissimilarity between motion  $X_A$  and  $X_B$  is given by:

$$d(X_A, X_B) = \min_S \{\|X_A^T - SX_B^T\|^2\} \quad (6.1)$$

where  $S$  is a similarity transformation, minimizing the dissimilarity. This measure requires that the motions being compared are temporally aligned. There are various approaches to achieve this automatically. In the experiments in this chapter, though, the sequences are manually aligned, in order to maintain focus on the task at hand. In the previous chapter, it was shown that such a euclidian comparison sometimes comes up with counterintuitive results. Generally, this occurs when the motions being compared are rather dissimilar. In this chapter, the motions are rather similar and a euclidian measure will work well.

### Dissimilarity matrix

Given a set of 3D motions,

$$\mathcal{T} = \{X_1, X_2, \dots, X_n\}$$

we need a compact way to represent the relative distances between each pair of motions. This is accomplished by constructing the *dissimilarity matrix*  $A_{\mathcal{T}}$ , which is a zero-diagonal symmetric  $n \times n$  matrix, where

$$A_{\mathcal{T}}(i, j) = d(X_i, X_j)$$

Further, for the set  $\mathcal{T}$ , the set of projected 3D motions is defined by

$$\mathcal{T}_C = \{(\mathcal{C}X_1^T)^T, (\mathcal{C}X_2^T)^T, \dots, (\mathcal{C}X_n^T)^T\}$$

where  $\mathcal{C}$  is a  $2 \times 3$  orthonormal projection matrix defined by. This set is used to generate the *projected dissimilarity matrix*:

$$A_{\mathcal{T}_C}(i, j) = d(\mathcal{C}X_i^T, \mathcal{C}X_j^T)$$

### Comparing dissimilarity matrices

What does  $A_{\mathcal{T}}$  tell us about the properties of  $\mathcal{T}$ ? By simply looking at the table it is easy to identify very dissimilar shapes. It is also easy to identify very distinctive shapes, by identifying rows (or columns) containing only high values. The purpose of this study is to investigate how well classification of motions remains invariant to an orthographic projection. In other words, in order to justify 2D analysis from a certain viewpoint of a motion as a good estimate of the contents of the original 3D motion, the dissimilarity matrices  $A_{\mathcal{T}}$  and  $A_{\mathcal{T}_C}$  should be similar in some sense, from a certain viewpoint  $\mathcal{C}$  while a large discrepancy indicates inaccurate analysis. The question then arises how to compare dissimilarity matrices. Again, this is rather task dependent.

- Is 2D analysis accurate in order to identify distinctive clusters among the motions?
- Is 2D analysis accurate in order to identify the overall structure of the relative similarities?

In order to maintain maximal generality, a measure suitable for the second criteria is used. This is accomplished by using a *rank inconsistency* measure (where rank has nothing to do with the matrix operation). In principle, rank consistency is obtained by choosing one motion in the set as a reference motion, and then find the motion most similar to the reference. That motion will have rank 1 with respect to the reference motion. The second most similar motion will have rank 2, and so on. The ranking of an entire matrix is obtained using each motion as reference in turn. More specifically, given two sets of motions,  $\mathcal{T}$  and  $\mathcal{T}_C$ , with corresponding dissimilarity matrices  $A_{\mathcal{T}}$  and  $A_{\mathcal{T}_C}$ , a pair of distances,  $(i, j)$  and  $(k, l)$ , are considered *inconsequent* if the following inequality holds:

$$(A_{\mathcal{T}}(i, j) - A_{\mathcal{T}}(k, l))(A_{\mathcal{T}_C}(i, j) - A_{\mathcal{T}_C}(k, l)) < 0 \quad (6.2)$$

The rank inconsistency is denoted by  $D(\mathcal{T}, \mathcal{T}_C)$ , and the definition is given as:

$$D(\mathcal{T}, \mathcal{T}_C) = \text{ration of inconsequent pairs with respect to } \mathcal{T} \text{ and } \mathcal{T}_C$$



**Example:** Consider the hypothetical dissimilarity matrices below:

$$A_1 = \begin{pmatrix} 0 & 4 & 7 \\ 4 & 0 & 5 \\ 7 & 5 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 5 & 3 \\ 5 & 0 & 6 \\ 3 & 6 & 0 \end{pmatrix}$$

In this case there are three distance pairs to be checked:

(1, 2), (1, 3):  $(A_1(1, 2) - A_1(1, 3))(A_2(1, 2) - A_2(1, 3)) < 0 \Rightarrow$ inconsequent.

(1, 2), (2, 3):  $(A_1(1, 2) - A_1(2, 3))(A_2(1, 2) - A_2(2, 3)) > 0 \Rightarrow$ consequent.

(1, 3), (2, 3):  $(A_1(1, 3) - A_1(2, 3))(A_2(1, 3) - A_2(2, 3)) < 0 \Rightarrow$ inconsequent.

This means that 66% of the distance pairs in this trivial example are inconsequent.

Thus,  $D(A_1, A_2) = 0.66$

### Visualizing dissimilarities by multidimensional scaling (MDS)

One way to get an idea about how the relative distances are related between two sets, is by using *multidimensional scaling* (MDS). This is achieved by representing each sequence by a point in a 2D graph, and organize these points so that the euclidian distance between two points in the graph is as similar as possible to the value in the dissimilarity matrix. The MDS graph will never be exact, since dimensionality is lost when going from the high dimensional similarity measure; however, it is generally good enough to give an idea about the correlations. In the presentation of the results, MDS is used in order to give an intuitive feel for what is going on.

## 6.2 Test scenarios

As has already been discussed, the purposes for comparing motions vary, but three main categories of motion comparison problems can be listed in increasing order of complexity.

1. Comparing *different people* performing *different motions*.
2. Comparing *different people* performing *the same motion*.
3. Compare *different trials* of a motion performed by *the same person*.

The first task (action recognition) is useful in order to get a coarse idea about what someone is doing. This is a common issue in automatic surveillance systems. One objective of such a system is to decide if somebody is doing something suspicious. This task is generally much less precise than the other two, and primarily concerns interpretation of image data from a psychological viewpoint. For example, how do we classify an activity as "suspicious"? Another useful area for action recognition is in automatic retrieval of video data. Examples of useful functions of such a system could be to extract all forehands in a tennis match, or extract all dancing sequences in a movie, etc. In principle, systems designed for action recognition generally focus on similarities in 2D, and the classes being investigated of such systems are defined in terms of properties in the video flow. By this definition, 2D

analysis is of high relevance in such system, and will not be explored further in this chapter. The next issue is of more interest in automatic identification. For example, is it possible to identify a person by his or her gait? As mentioned before, it is a relatively easy task for a human, and some successful systems have been developed to solve this problem to some extent. The last issue is the toughest since it provides the smallest variations between the samples. Two trials of the same person are usually very similar. It generally requires a well-trained eye in order to distinguish between, for example, two different long jumps performed by the same person. A successful system, solving this problem, has a broad class of potential applications, the primary two being:

1. Athletic performance evaluation. How good was a golf swing? The measure of "goodness" can be distance (according to some measure) to a reference swing known to be good.
2. Orthopaedics. A patient may be interested in quantitatively monitor the improvements of his or her walking.

The experiments demonstrated here are designed to illustrate the three situations. The motion sequences used are acquired from different motion capture systems.

### 6.3 Finding the optimal angle

The relevance of 2D analysis of human motion is extremely sensitive to the orientation and position of the camera. In the real world, the optimal viewpoint is usually selected by using common sense. For example, in order to classify sprinters, the intuitive choice is probably to place the camera perpendicular to the direction of the run (side view). Whether this placement is optimal or not, of course depends on the similarity measure. If the similarity measure for the classification is the height of the knee lift of the runners, side view footage is probably the most accurate. In the case of the similarity measure used in this study (least-squared distances between corresponding points), the optimal camera orientation is much less obvious. To find the best viewpoint for our synthetic camera the rank inconsistency measure is used to systematically test a large number of orientations, in order to find a good estimate of the *optimal projection matrix*,  $\tilde{C}$ , minimizing  $D(A_{\mathcal{T}}, A_{\mathcal{T}_C})$ . This is illustrated in fig. 6.3. The roll angle of the camera (the rotation around the focal axis) does not affect  $A_{\mathcal{T}_C}$ , since the distance measure in 2D is rotation invariant. Therefore, we iterate over the pitch and the yaw angles, in the interval  $[0, \pi]$  respectively. In each of the experiments, the results correspond to a synthetic camera from this estimate of the optimal viewpoint, if nothing else is stated.

### 6.4 Different people doing the same action

The first investigation involves the second item in the complexity list; classification of different people performing the same action. Two experiments were carried out in this category. The first experiment involved comparing 11 different people walking. In the second example, three different strokes from three different tennis players were used.

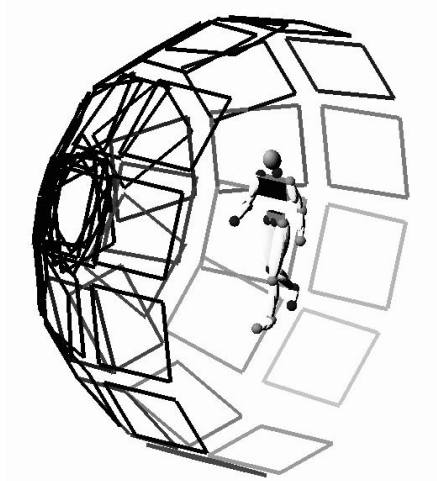


Figure 6.3: In order to find the angle that maximizes the validity of 2D analysis, a large number of projection planes are evaluated. Each square in the figure represents a projection plane.

### Experiment 1 - Different people walking

In this experiment, a set of motion sequences of regular walks were used, in order to form the set  $\mathcal{T}^w$ . The walks were acquired from a commercial motion capture system, and the walk of eleven different people were studied<sup>1</sup>. Each trial covered one stride (two steps) of the walk, and the sequences were sampled to 10 frames each. They were also temporally aligned, so that each sequence begins and ends with the touchdown of the left foot. All sequences are shown in fig. 6.4.

Using the method described in section 6.3, the discrepancy between 2D and 3D dissimilarities were computed using a large number of camera orientations. In figure 6.6(a),  $D(\mathcal{T}^w, \mathcal{T}_{\mathcal{C}}^w)$  is plotted as a function of the pitch and yaw angles used to generate the projection matrix,  $\mathcal{C}$ . By locating the global minima on this surface, the optimal camera,  $\mathcal{C}$  was selected. The resulting dissimilarity matrices,  $A_{\mathcal{T}^w}$  and  $A_{\mathcal{C}}^w$  are shown in compact form in fig. 6.5.

The left value in each cell is the 2D dissimilarity ( $D_{\mathcal{T}_{\mathcal{C}}^w}$ ), and the right value is the 3D dissimilarity ( $D_{\mathcal{T}^w}$ ). All values are normalized. As can be seen, most values are rather similar, indicating that the 2D projections reflect the original actions well. Quantitatively, the result was rather encouraging:  $D(\mathcal{T}^w, \mathcal{T}_{\mathcal{C}}^w) = 0.015$  indicating, that by placing the camera at the optimal viewpoint only 1.5% of the distance pairs were inconsequential. Fig. 6.7(a) shows the orthographic projection plane corresponding to the optimal orientation. In fig. 6.7(c) one of the walking sequence is shown from the optimal viewpoint. As can be seen from fig. 6.6(a), there is an obvious maxima in the number of inconsequential pairs,

<sup>1</sup>The sequences were taken from the CMU motion database

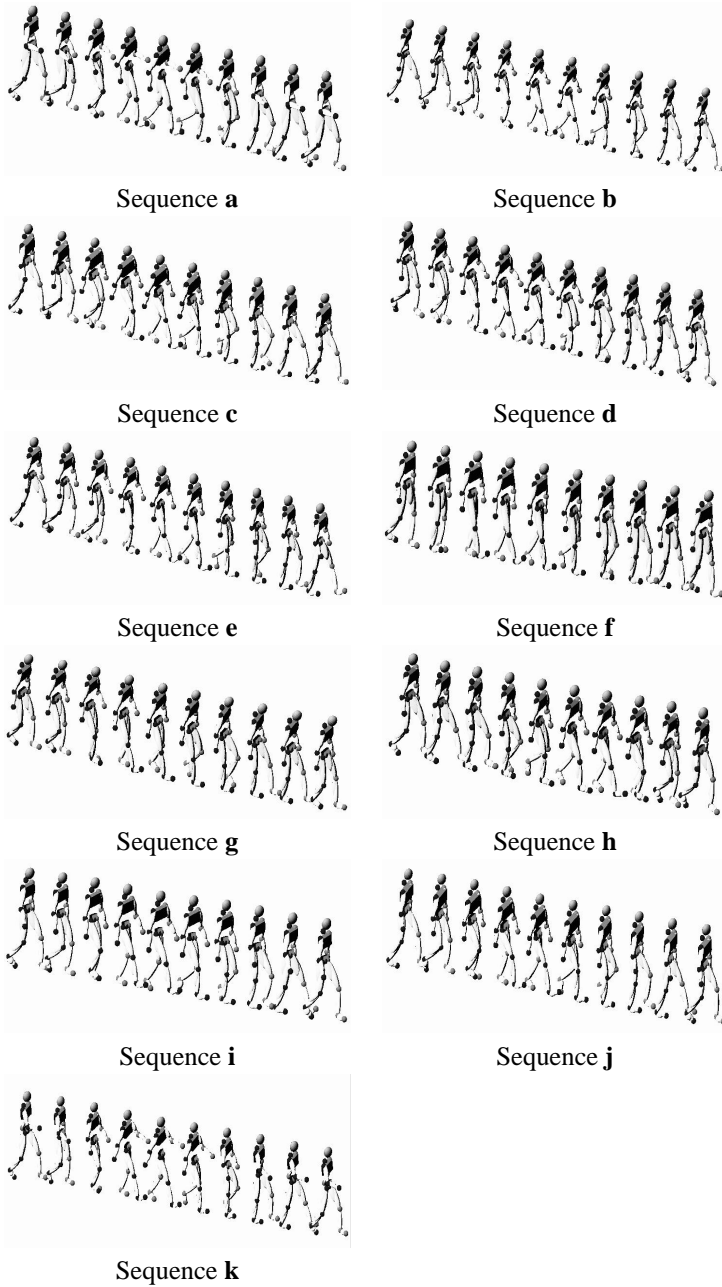


Figure 6.4: The eleven sequences used in experiment. Each sequence, which consists of one gait cycle of a walking person, is sampled to 10 frames. They are aligned temporally, beginning with the first contact of the left foot with the ground.

Sequence	1	2	3	4	5	6	7	8	9	10	11
1	.0,.0	.54,.54	.44,.48	.65,.66	.72,.74	.88,.87	.89,.89	.33,.34	.37,.43	.34,.39	.51,.54
2	.54,.54	.0,0	.39,.42	.74,.72	.56,.57	.95,.93	.94,.91	.42,.44	.43,.48	.71,.70	.63,.65
3	.44,.48	.39,.42	.0,0	.57,.57	.53,.54	.77,.75	.78,.75	.35,.39	.20,.20	.57,.58	.42,.44
4	.65,.66	.74,.72	.57,.57	.0,0	.57,.56	.41,.41	.39,.39	.61,.61	.54,.55	.78,.80	.27,.31
5	.72,.74	.56,.57	.53,.54	.57,.56	.0,0	.60,.58	.59,.57	.74,.75	.63,.64	.87,.88	.51,.53
6	.88,.87	.95,.93	.77,.75	.41,.41	.60,.58	.0,0	.37,.39	.89,.88	.77,.76	.100,.100	.49,.52
7	.89,.89	.94,.91	.78,.75	.39,.39	.59,.57	.37,.39	.0,0	.91,.90	.79,.77	.99,.98	.48,.47
8	.33,.34	.42,.44	.35,.39	.61,.61	.74,.75	.89,.88	.91,.90	.0,0	.24,.29	.43,.44	.48,.50
9	.37,.43	.43,.48	.20,.20	.54,.55	.63,.64	.77,.76	.79,.77	.24,.29	.0,0	.50,.51	.39,.40
10	.34,.39	.71,.70	.57,.58	.78,.80	.87,.88	.100,.100	.99,.98	.43,.44	.50,.51	.0,0	.61,.63
11	.51,.54	.63,.65	.42,.44	.27,.31	.51,.53	.49,.52	.48,.47	.48,.50	.39,.40	.61,.63	.0,0

Figure 6.5: The dissimilarity matrices for the synthetic walking example, shown in compact form. The left and right value in each cell correspond to the 2D and 3D dissimilarities, respectively. The values are normalized, and indicate the percentage of the largest dissimilarity. The values are generally very similar, indicating good correlation.

implying a poor camera orientation. This poor orientation is approximately straight in front (or behind) the person, as shown in fig. 6.7(b), and yielded 36 % inconsequent pairs. In fig. 6.6(b) and 6.6(c), the MDS graphs of  $T^w$  and  $T_C^w$  are shown. Note that the MDS graph is invariant to rotations and reflections. It becomes evident, however, that the distances between the points in the two point sets are quite similar, supporting the hypothesis that 2D analysis is in fact useful, provided the given similarity measure. Also, in fig. 6.6(d) the MDS graph of the projected sequences are shown when the least optimal camera angle was used. This graph is significantly different from the graph of the original 3D motions, emphasizing the importance of choosing a proper orientation of the camera.

## Experiment 2 - Clusters of different actors

The previous experiment showed that the overall relations between the sequences are preserved fairly well, when projected to the plane, from the optimal angle without presence of noise. Another illustrative experiment would be to see how *apparent clusters* of motions are preserved in projected data. In order to investigate this, motion data from three different tennis players were used. Three fore hand strokes from each player were registered by motion capture systems. These sequences are shown in fig. 6.8. The exact same procedure as in the previous experiment was repeated on these sequences. From the optimal camera position, as shown in fig. 6.10(a), 3.0% of the distance pairs were found to be inconsistent. The MDS graphs of the original 3D structures and the 2D projections from the optimal angle is shown in fig. 6.9(b) and 6.9(c). The clusters of the three players are very well preserved when going from 3D to 2D.

## 6.5 Individual variation

In this experiment, the purpose is to determine the possibility of using 2D analysis in order to identify personal variation between different trials of a certain action, performed by the same person.

## Experiment 1 - MDS

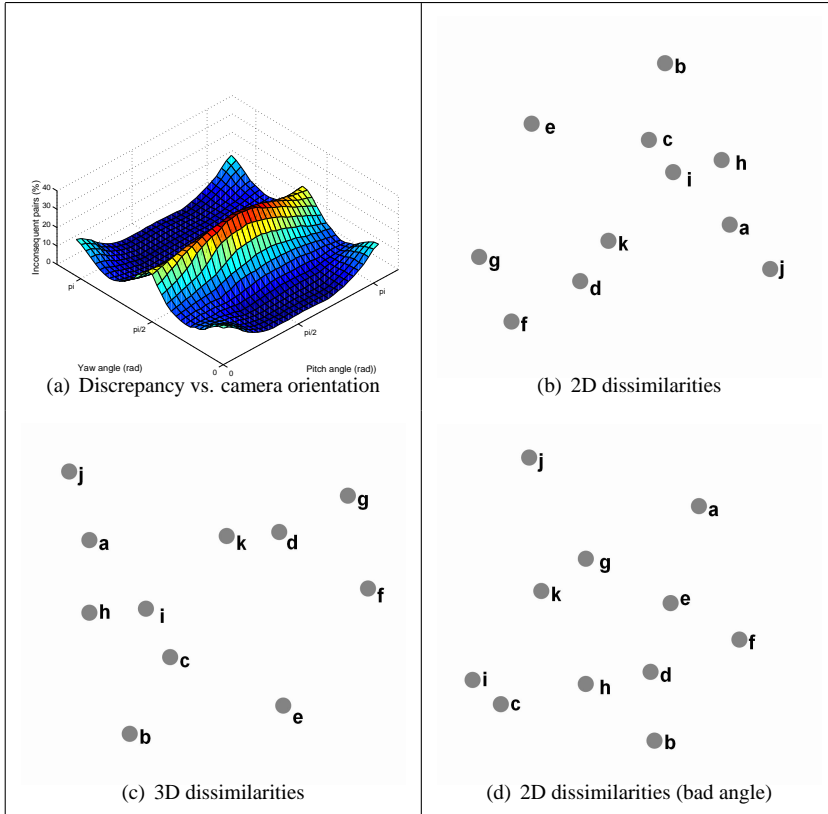


Figure 6.6: Walking example. Multidimensional scaling on the dissimilarity matrices corresponding to 11 walking sequences. In (a), the height of the surface indicates the percentage of inconsequent pairs between 2D and 3D dissimilarities, from a certain camera orientation. The surface is relatively smooth, with two peaks, indicating poor positions of the cameras. In (b), the MDS plot of the dissimilarities of the projected walking sequences. The camera is oriented according the most discriminating angle. In (c), the MDS plot of the dissimilarities of the original 3D walking sequences. Since the MDS plot is invariant to rotations and reflections, the absolute positions of each point is not interesting. The interesting observation is that sequences close to each other in the projected space are generally close to each other in the original, 3D space, indicating that 2D analysis reflects the 3D properties well; at least for the similarity measure used in the experiment. (d) The dissimilarities of the projected sequences from one of the bad angles. Apparently, this graph indicates significantly different distances than the original 3D dissimilarities.

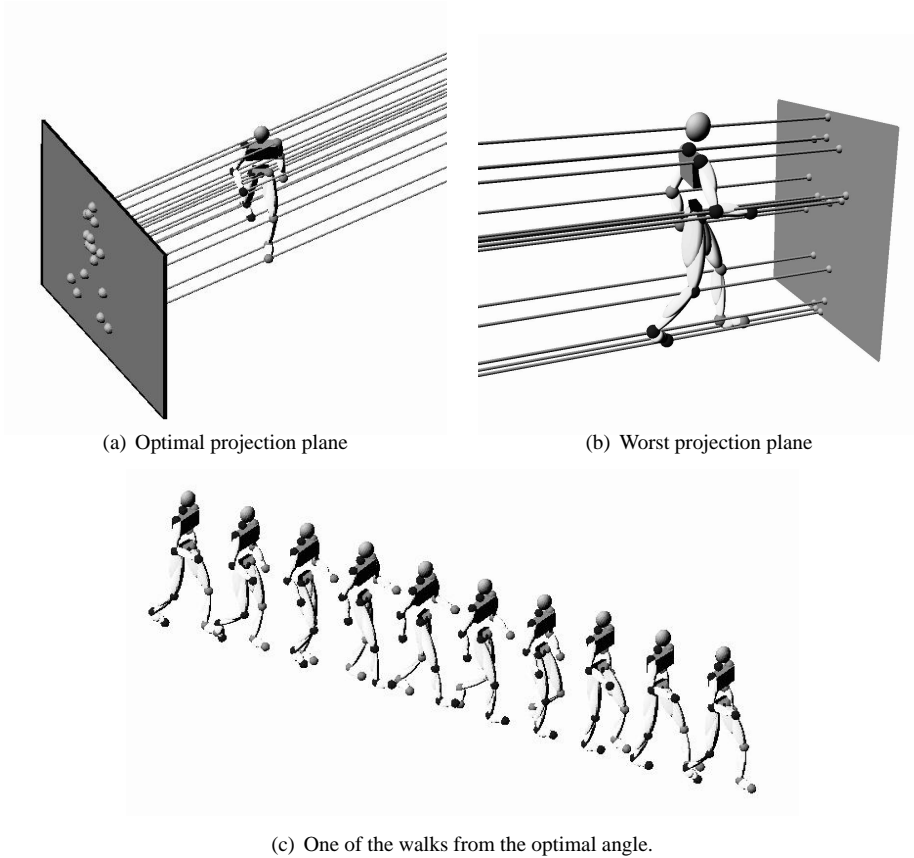
**Experiment 1 - Viewpoints**

Figure 6.7: Walking sequences. **(a)** The orientation of the projection plane at the optimal viewpoint. **(b)** The orientation of the projection plane from the worst orientation. **(c)** One of the sequences viewed from the optimal camera angle.

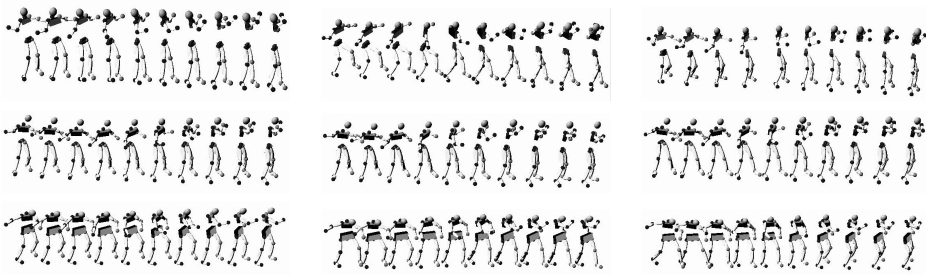


Figure 6.8: The forehand strokes used in the clustering experiment.

Experiment 2 - MDS

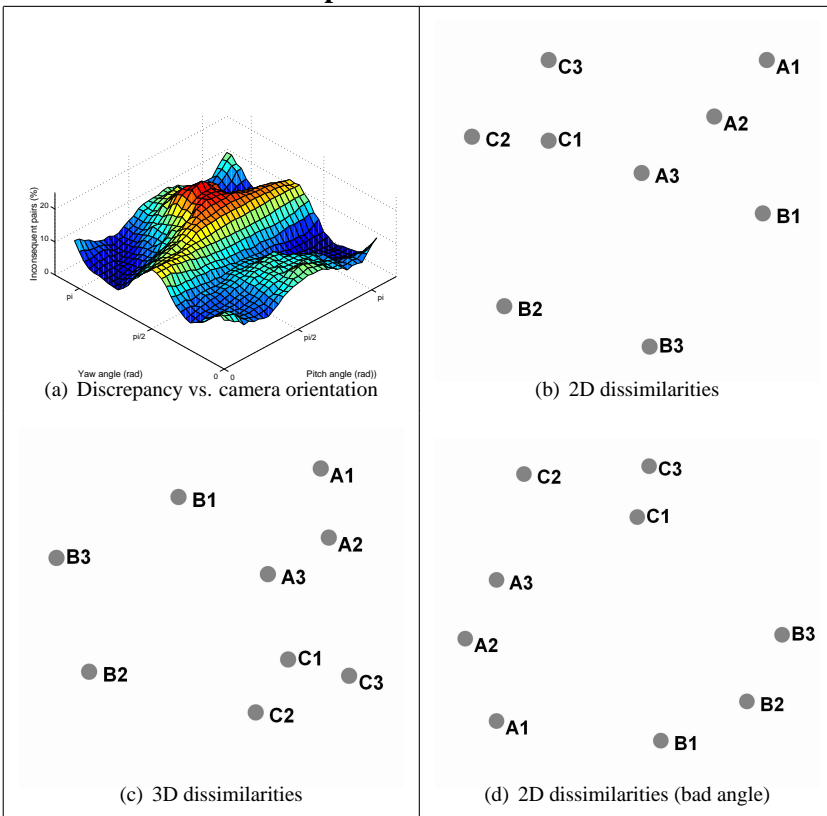


Figure 6.9: 3 different players. In this case, it is interesting to observe that the clusters are not affected very much by selecting a poor camera angle. This indicates large differences in personal style of the players.



### Experiment 2 - Viewpoints

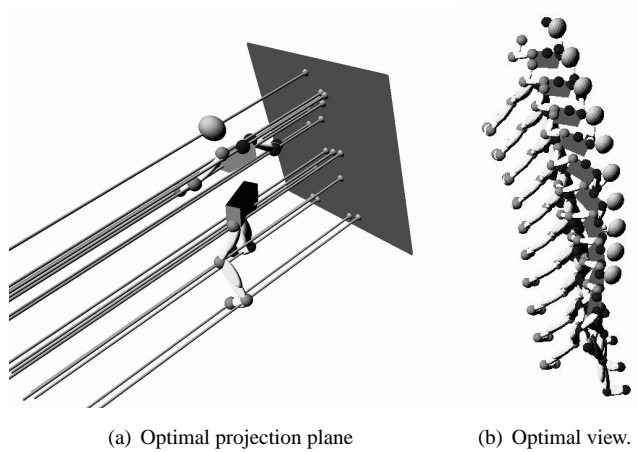


Figure 6.10: (a) The projection plane according to the most discriminating angle in the clustering experiment. (b) One of the sequences viewed from the optimal angle.

In this experiment eleven forehand strokes (all from the same person) were used to form the set  $\mathcal{T}^t$  (fig.6.11). From the optimal viewpoint, 4.4% of the distance pairs were inconsequent. The orientation of the camera plane from this optimal orientation is shown in fig. 6.13(a). Again, from the graph in fig. 6.12(a) there appears to be one particularly bad angle to place the camera, that should be avoided. This camera placement results in no less than 33 % inconsequent distance pairs. The resulting MDS graphs are shown in fig. 6.12.

## 6.6 Intermediate discussion

The percentages of inconsequent pairs of the three experiments are summarized in the table below.

Experiment	Inconsequent pairs (%)
11 walks, 11 different persons	1.6
9 forehands, 3 different persons	3.0
11 forehands, the same person	4.4

The work presented here is an effort to rate the validity of analyzing human motion in 2D. It should be regarded as a theoretical fundament in order to analyze the relevance of classification of 2D data. It should not be regarded as an attempt to actually perform any classification. Only one similarity measure has been used in this study. Future studies will involve other measures, more specific to a certain task.

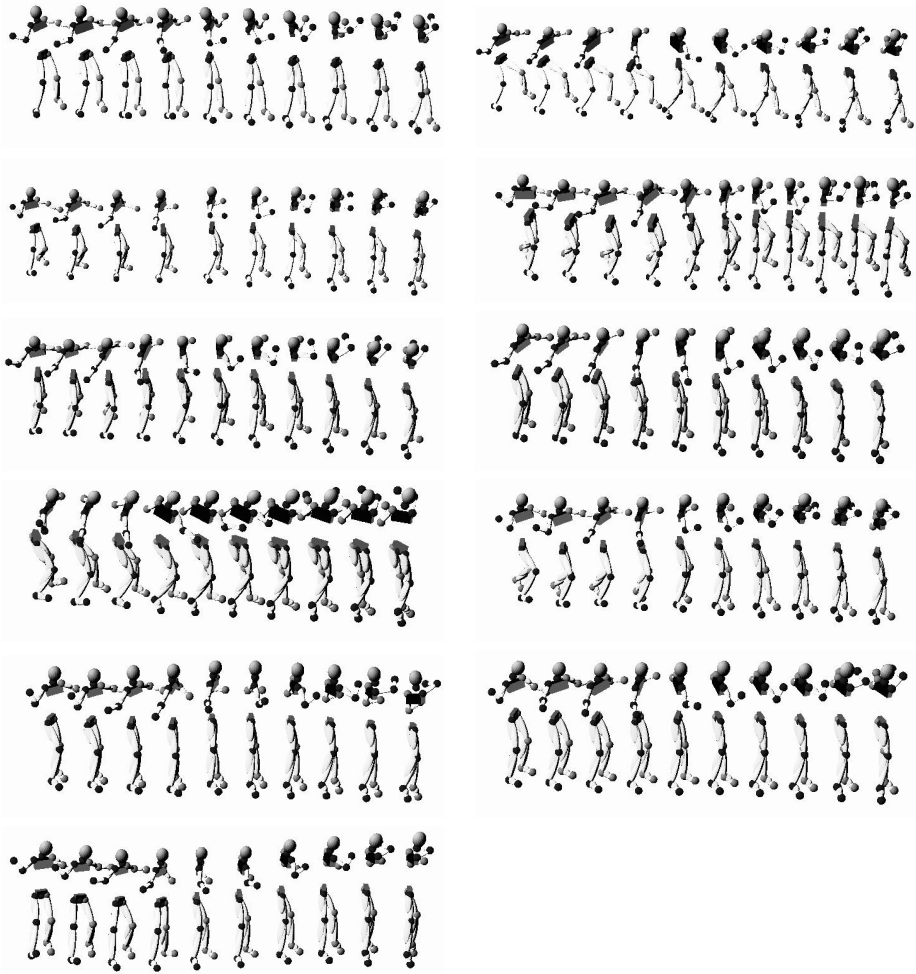


Figure 6.11: The forehand strokes used in the experiments.

Experiment 3 - MDS

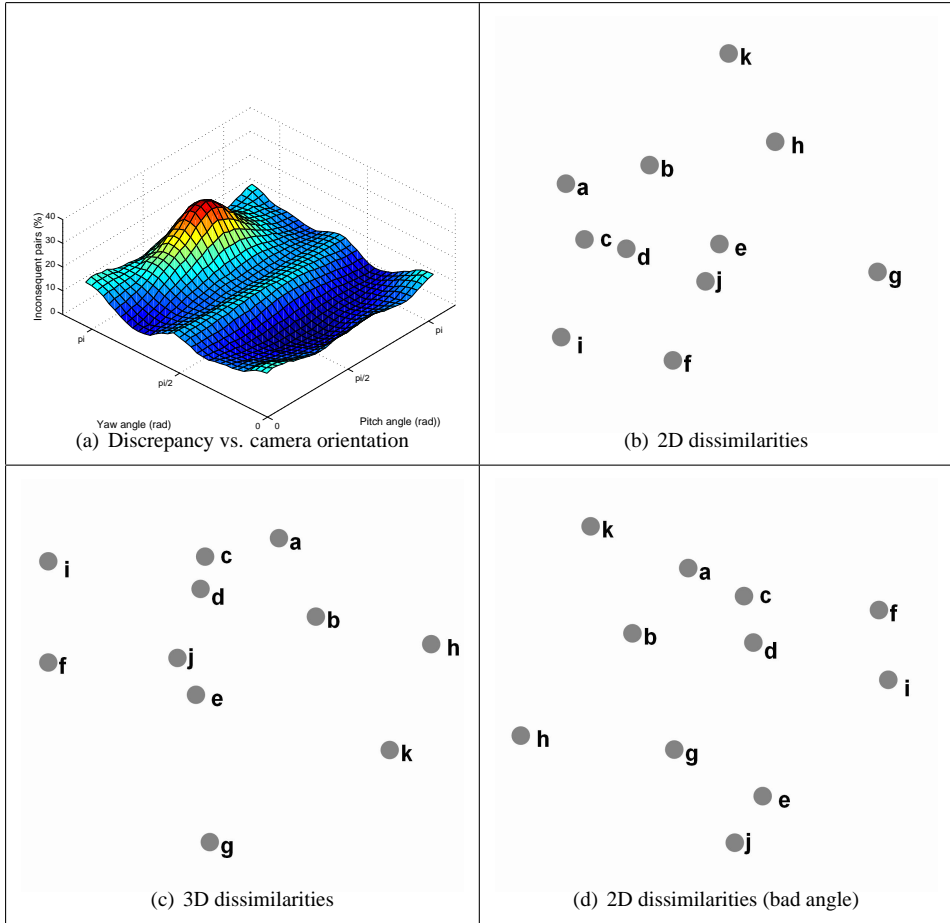


Figure 6.12: Eleven forehand strokes of the same player.

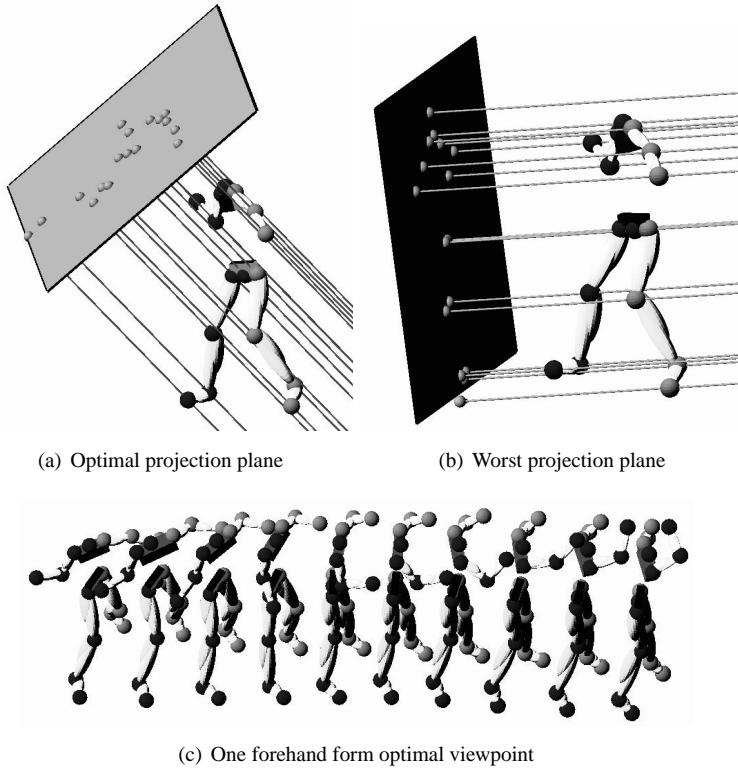
**Experiment 3 - Viewpoints**

Figure 6.13: Best and worst projections in order to perform 2D analysis.

**6.7 Analyzing monocular reconstructions**

Up to this point, the consistency between a 2D projection and the original 3D motion has been studied. As one of the main objectives of this thesis is to investigate how a 3D motion can be reconstructed given its 2D projection, it is natural to ask the question: Can the relevance of the 2D analysis be enhanced by first reconstructing the 3D motion? Of course, a perfect 3D reconstruction of the motion should yield the theoretical maximum of the classification. The problem is that a 3D reconstruction from monocular data can never be perfect. Even if the input 2D data is completely noise-free, we still need to rely on a model based on prior information in order to resolve the image depth of the 2D features. Computing the effect of small errors in the model is necessary in order to rate the reliability of the classification. The outline of the experiments is shown in fig. 6.14. The same similarity measure as in the previous experiments is used. Again, as we assume

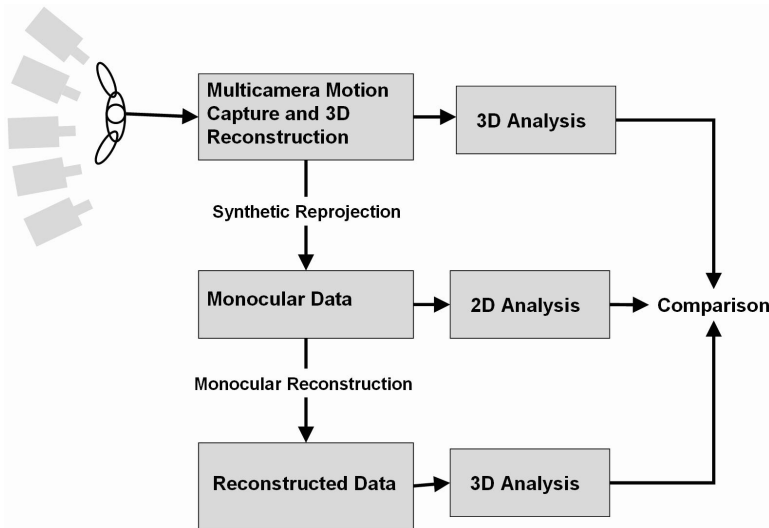


Figure 6.14: Outline of the experiment.

to have temporally aligned sequences, all we have to do is comparing 3D point sets with known point-to-point correspondences. The same walking sequences that were used in the previous experiments are used here as well. They are again projected down to 2D. At this point, however, the 2D data is used to reconstruct the original 3D sequences. Classification is then performed on the reconstructed sequences in order to see how much validity is lost in the reconstruction process. In order to evaluate the effect of errors in the model used for reconstruction, increasing amount of errors will deliberately be added to the model, in order to see how a poor model affects the reliability of the classification.

### Monocular reconstruction

We already know that given the projected 2D points of an articulated chain together with the limb lengths, the method presented by (Taylor, 2000) will be able to retrieve the 3D coordinates of the points. In addition to the limb lengths of the person, the method requires knowledge about the binary configuration of the articulated chain at each time frame of the motion. Computing the binary configuration of a 2D point set automatically is impossible without prior information about the motion. In chapters 3,4 and 5 we have also seen some other methods, based on priors, in order to resolve this problem. In this chapter, however, we are allowed to cheat a bit. Since the 2D points are acquired by projecting 3D sequence using an orthographic camera of known orientation, we can simply use the binary configuration of the original sequence, with respect to the camera plane. This means that *no errors will occur due to bad selection of binary configuration*. Thus, the relevance of the

reconstructions obtained here is again a *theoretical maximum*. Knowing the configuration of the articulated chain, the difference in depth between two connected points,  $p_a$  and  $p_b$ , is given by:

$$\Delta Z = \sqrt{L_{a,b}^2 - \|p_a - p_b\|^2} \quad (6.3)$$

where  $L_{a,b}$  is the known 3D length of the limb connecting  $p_a$  and  $p_b$ .

As this study allows us to dodge the problem of disambiguation of the configuration, we focus all the attention on the sensitivity of the model - more specifically the limb length estimation.

### Simulated errors and camera orientations

Earlier, we saw that each class of motions has an optimal angle in order to perform classification on projected data. In this experiment, a synthetic camera was again applied in different orientations. The first orientation is straight in front of the person, and the other view is from the side. The reason for these choices is simply because these viewpoints are commonly used in literature. They are also interesting, since the side view is near the optimal angle, and the front view is, as we have seen, a very bad angle. These orientations are depicted in fig. 6.15 and 6.19. The 3D walks are then reconstructed using the projections and the binary configuration of the original shape, and the original limb lengths multiplied by a scale factor. If this scale factor is set to 1.0, the reconstruction will be exactly the same as the original motion. Sometimes, the estimated limb length of a limb is too short. This happens when the length of the projected limb is larger than the estimated 3D limb length. In such a case, the 2D limb length of the projection is used. This means that by setting the scale factor to zero (or near zero) the reconstruction will be virtually a flat structure. The classification in such a case will perform exactly the same as the 2D projections.

### Presentation and results.

At one given scale factor, each sequence is reconstructed. The dissimilarity matrix of the reconstructions  $D_R(i, j)$  is computed, as well as the dissimilarity matrix of the original shapes,  $D_S(i, j)$ . Plotting  $D_S(i, j)$  versus  $D_R(i, j)$  for each  $i, j$  will form a straight line if the two dissimilarity matrices are perfectly rank consistent. As the matrices become less rank consistent, the points will form a cluster with increasing variance.

In the first case, the walks were projected with the camera straight from the side. The curve in fig. 6.16 shows the number of inconsequent pairs versus the scale factor. The scale factor is given as percentage of the correct limb length. Fig. 6.17 shows the plots explained above for six choices of scale factors. Fig. 6.18 shows reconstructions for one pose from one sequence using the given scale factor. Apparently, as the scale factor becomes large, the reconstruction looks very peculiar. This is indicated by the plot of inconsistent pairs - if the scale factor is chosen small, the consistency between reconstructions and original shapes is rather good. However, as the limb length estimates gets too large, the consistency severely drops (a large increase in number of inconsistent pairs).

In fig. 6.20, 6.21 and 6.22, the same things are shown when the synthetic camera is placed

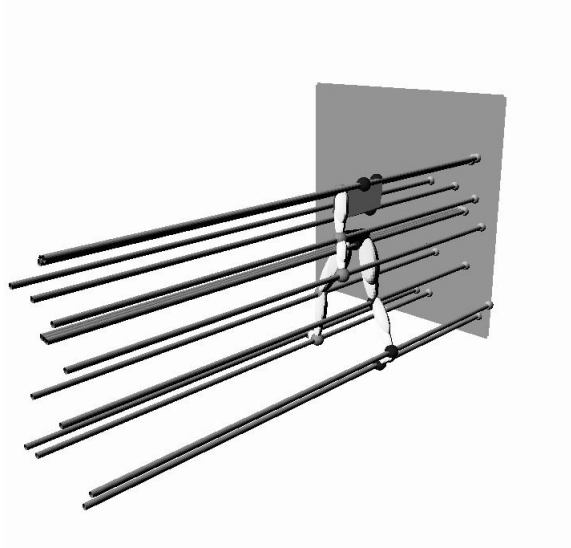


Figure 6.15: The orientation of the synthetic camera used in the first experiment.

in front of the person walking. In this case, it is interesting to see how the number of inconsequent pairs actually drops initially, reaching zero at 100%, before starting to rapidly rise. Recall from the earlier experiments that front view camera was very poor in order to classify motions based on 2D projections. This is the reason that the number of inconsequent pairs is very high when the scale factor is set too small.

## 6.8 Conclusions of monocular reconstruction

In this chapter, the performance of a very generic classification measure with perturbed input data has been investigated. Specifically, the performance of a classifier using 3D monocular reconstructions and the performance of a classifier using the projected 2D data has been compared, in order to see under what conditions monocular reconstruction actually improves the analysis. From the experiments, it turns out that 3D reconstruction can be justified if the camera viewpoint is poorly chosen, as long as the limb lengths are very accurately estimated. Otherwise, analyzing the original 2D data yields better results.

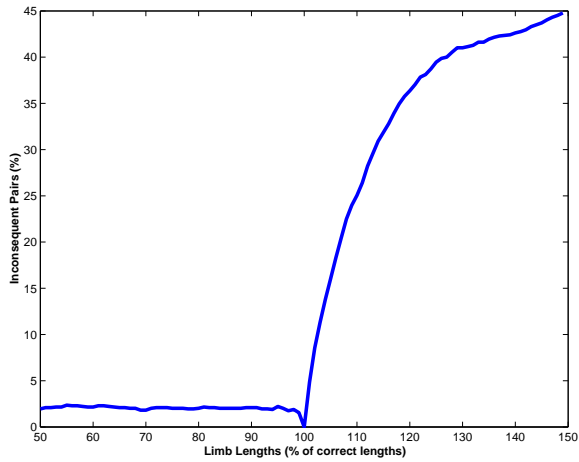


Figure 6.16: The number of inconsequent pairs plotted versus the error in limb length estimation. The horizontal axis shows the correctness in limb lengths estimation. Thus, at 100 the reconstructions are identical to the original motions, which yields perfect classification.



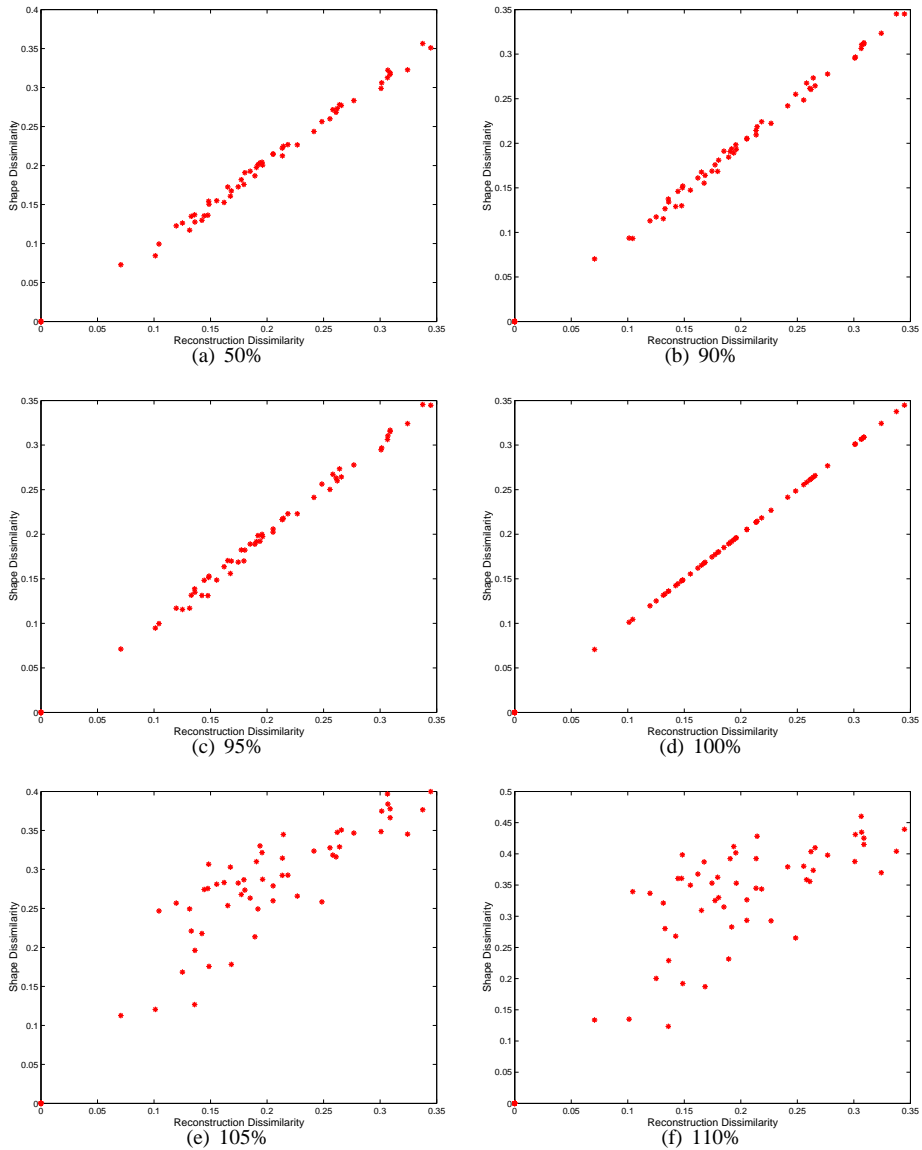


Figure 6.17: The dissimilarities from the reconstructed sequences plotted versus the dissimilarities of the original sequences. If the two sets of motions are perfectly consistent, a large dissimilarity between two original shapes should yield a large dissimilarity between corresponding reconstructions. If the limb length estimate is good, the consistency is good, as is shown in **(d)** where the points reside on a straight line.

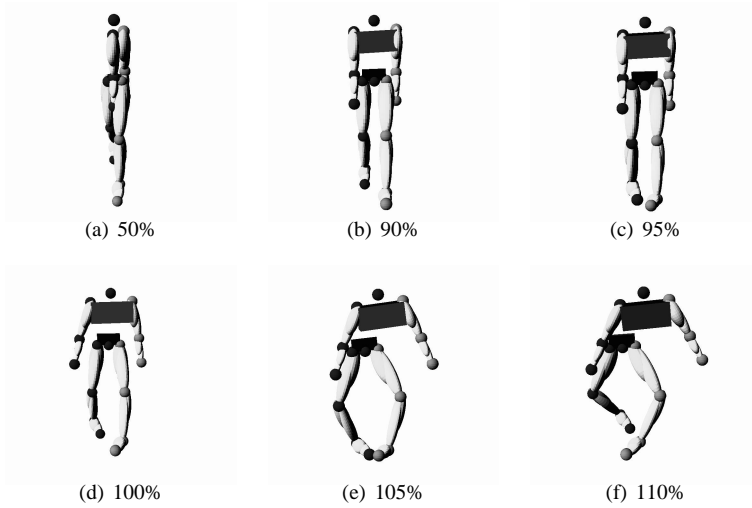


Figure 6.18: Examples of what one pose of the reconstructed sequence will look like using the given estimation in limb lengths.

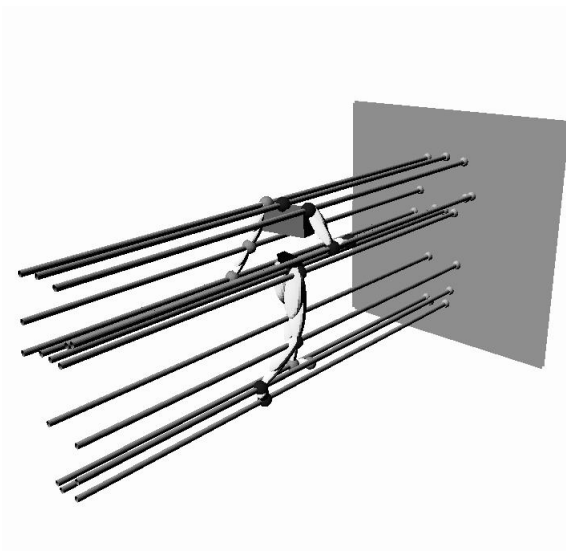


Figure 6.19: The orientation of the synthetic camera used in the second experiment.

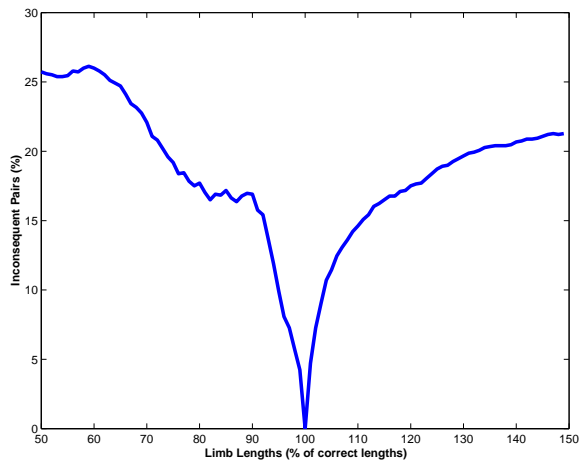


Figure 6.20: The number of inconsequent pairs plotted versus the error in limb length estimation.

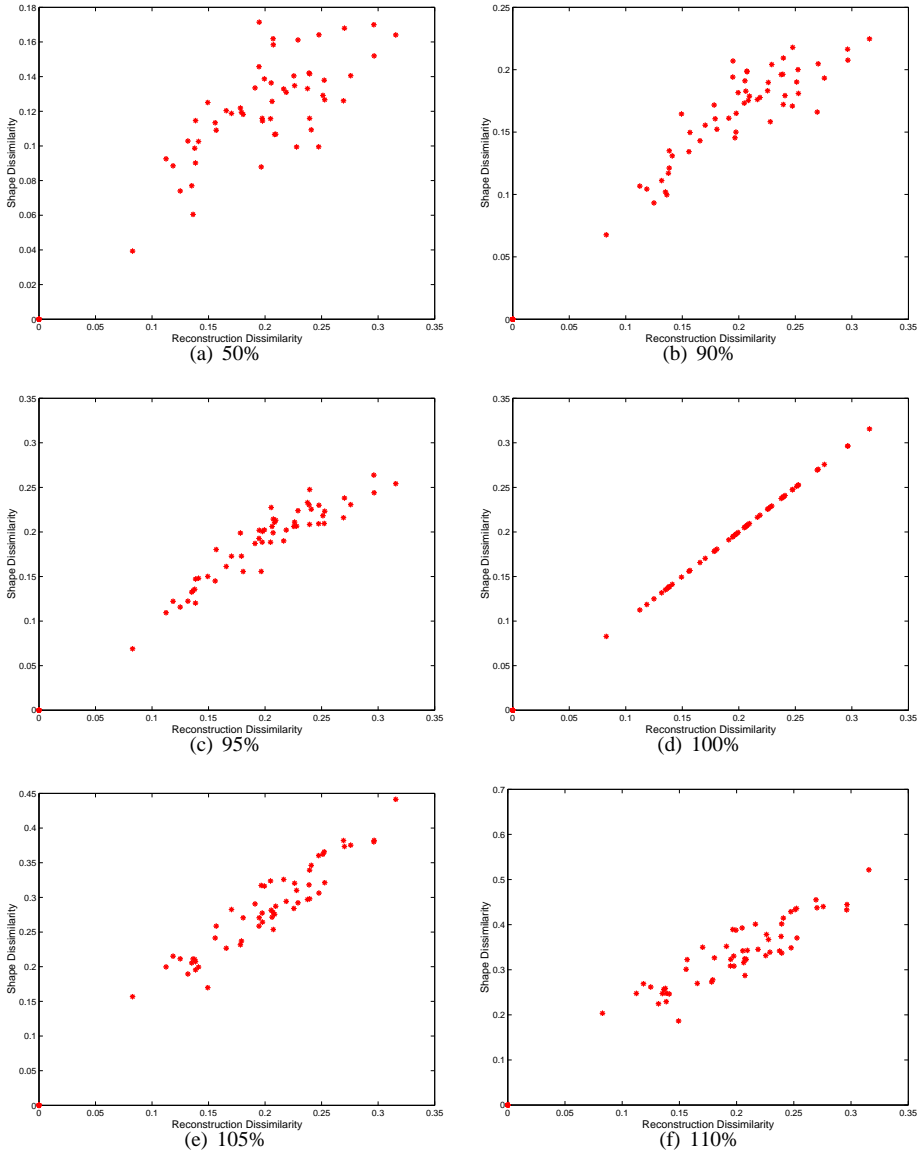


Figure 6.21: The dissimilarities from the reconstructed sequences plotted versus the dissimilarities of the original sequences.

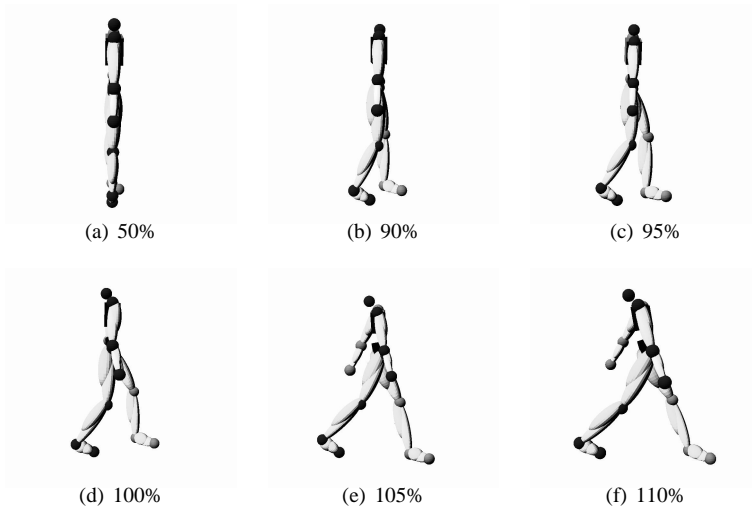


Figure 6.22: Example of one reconstructed pose from one sequence using the given limb length estimation.



## Chapter 7

# Action quality assessment from automatic motion capture

This chapter will close the loop that was initiated in the introduction and deals with an "automatic virtual coach". The chapter will discuss how well some of the methods described and developed in the thesis can be used in order to provide useful information about a motion *from the viewpoint of a coach or athlete*. In other words, how far will computer vision come in the quest of capturing minute details of a certain class of motions. Most prior work in this area deals with comparing motions for the purpose of identification - the most common task being gait recognition. The intent of this chapter is to promote the efforts of combining biomechanics and computer vision in order to develop specific, non-intrusive tools for coaching. This will include a discussion about what problems typically occur in the field of athletics, and what kind of information that is generally requested. The results are based solely upon computer vision primitives. By improving such primitives, for example by developing better edge detectors, it is most likely possible to improve upon the results. However, the purpose of the chapter is to present tracking techniques in order to solve a given task, not to solve the vision problem. It is demonstrated that the methods will indeed be useful in order to perform some automatic quality assessment. This is done by showing that an automatic tracking method is able to rate certain, manually selected, properties of a motion almost as well as a human observer.

### What is athletic coaching

Most individual sports involve performing a motion that looks *almost* the same every time. In earlier chapters, we have seen how pole vaults can form a class of motions in which different trials are very similar to each other - even if they are performed by different people. However, a well-trained eye can tell not only fine differences between two jumpers, but also differences between two trials of the same jumper. This is primarily due to the fact that the coach (or any trained observer) is very familiar with the motion and knows exactly where the differences between trials generally occur. For example, in the case of pole

vaulting, it is important that the vaulter reaches high with the right hand at takeoff. In a high bar routine in gymnastics it is important that the gymnast maintains a straight and tall posture throughout the routine. The point is that there are specific cues that a coach looks at that are very important for the result of the trial. *By exploiting these cues, it is possible to develop a system that is able to identify exactly the visual cues that are of interest for the specific exercise.* Rather than extracting a general indicator of the entire motion, as is generally done in gait recognition, the system should provide functionality that allows the user to select what should be analyzed.

## Virtual coach functionality

What is the main difference between a "coaching software system" and other systems able to track and compare motion? Part of the answer is the evaluation criterion. One method of rating the quality of a trial of a certain exercise could be to capture some very generic parameters from a "good" trial. Later trials are then rated by comparing them to the good reference, in generic ways. Such generic comparisons could be to extract the joint centers, as has been the case throughout this thesis, and compare the poses in point set representations. If the trial and the reference are similar, that was a good trial, and vice versa. This approach is not always useful, since large variations in some part of the motion may not necessarily affect the result. For example, the motion of the legs may not be very discriminating when rating a javelin throw, where the success may depend more on small differences in arm positions. Information about what the most discriminating factor is, has to be provided by an expert (coach).

## Preliminary experiment

We will first look at an example that gives an idea about how comparing trials to a reference motion can perform. This is done by considering the automatically tracked data from chapter 4. In the previous chapter, the rank consistency between dissimilarity matrices corresponding to 3D motions and their projected 2D motions were computed. In this example, the rank consistency between the dissimilarity matrix of the automatically tracked motions and that of the motions obtained by manual labelling of joints are compared. The tracking data from one of the forehands is shown in fig. 7.1. Using the same measure for rank inconsistency as in chapter 6, it turned out that 14 % of the distance pairs were inconsequent. If this is good enough for automatic coaching depends on the requirements. It is important to keep in mind that the similarity measure used here is rather blunt, and some knowledge about the game of tennis would make it possible to come up with a more sophisticated measure. The MDS graphs of the autotracked motions as well as the hand labelled motions are shown in fig. 7.2.



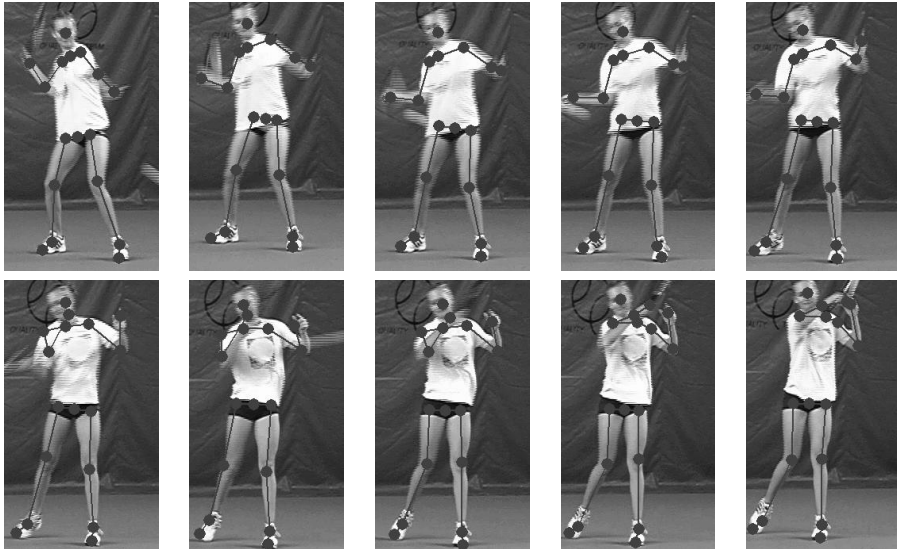


Figure 7.1: Automatically extracted feature points in one of the 11 forehand strokes used in the experiment.

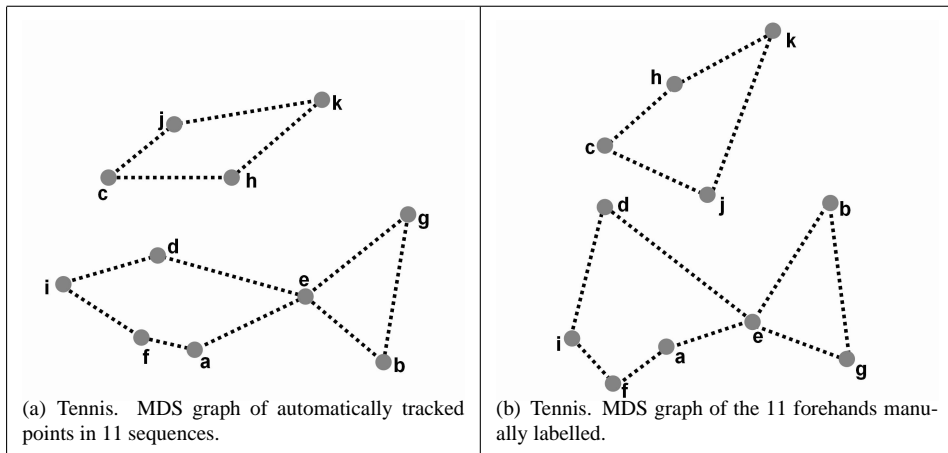


Figure 7.2: Autotracked vs. hand clicked. These MDS graphs show how well automatically tracked data reflects the ground truth, which in this case was manually marked joint locations. The correlation between the two MDS graphs is of course not as good as when the synthetic cameras from the previous chapter were used. Obviously, some correlation between the two graphs exist. The dashed lines show structures that are similar between the two point sets.

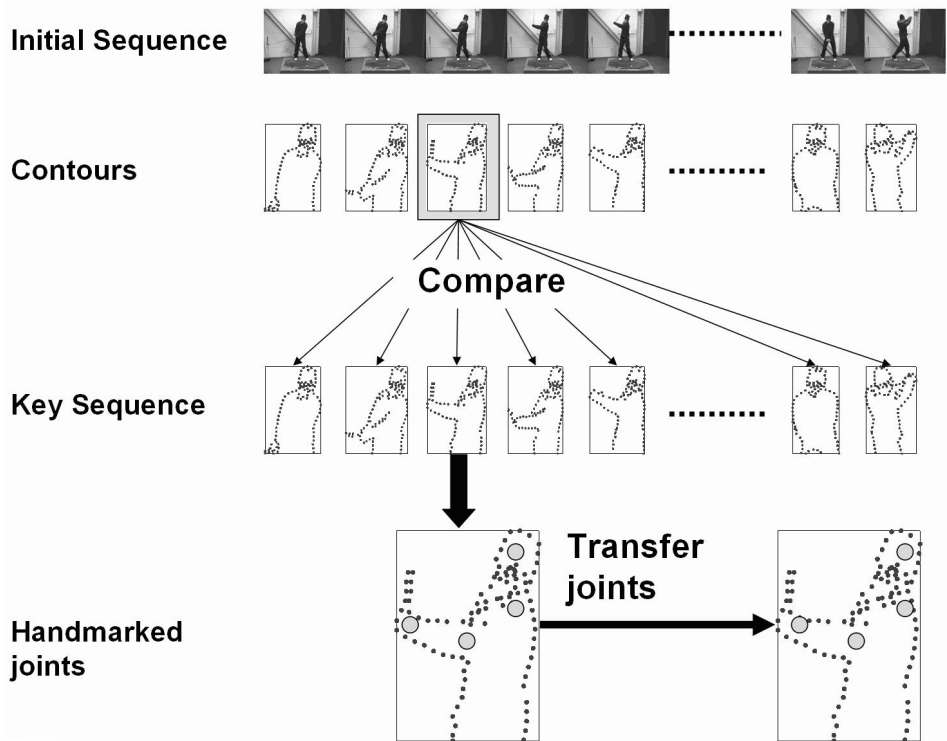


Figure 7.3: Outline of the joint localization.

## 7.1 Motion tracking revisited

Even though motions are represented as point sets, it turns out that the *silhouettes* of the person can provide quite good information, without having to first find the joint centers. Actually, this falls quite naturally, since the location of the joint centers are primarily located based on the silhouette.

The principles for tracking in this chapter are similar to the approach used earlier, in terms of using key frames, where the joint centers of a reference motion have been manually labelled. The difference is that we are now interested in identifying very small differences between two almost identical trials. This calls for a larger number of key frames. The goal is to see how far the key frame approach will take us. The joint location procedure strongly resembles the method described in chapter 4 (Sullivan and Carlsson, 2002), with some slight variations in the point matching scheme. The procedure is outlined in fig. 7.3 In key framed joint localization, joint locations in one key sequence are manually marked. Given a new frame from a new sequence, this new frame is compared to all the silhouettes

of the key sequence in order to find the most similar silhouette. After this silhouette is identified, the manually marked joints need to be transferred to the new frame. Assuming we have a relatively stable silhouette extractor, this procedure needs to solve:

1. How to compare two silhouettes
2. How to transfer joints from the key frame to the new frame.

These two aspects will be discussed next.

## Comparing silhouettes

The silhouettes are obtained by performing a canny edge detection in RGB-space followed by a background subtraction of the edges from the pre-recorded background. The silhouettes are then sampled to approximately 200 edge elements. The word silhouette is perhaps a bit miss-leading, since the point sets may include interior edge elements.

Comparing two silhouettes requires finding a matching between the points in the two silhouettes. In other words, given two point sets

$$P = p_1, p_2, \dots, p_n$$

and

$$Q = q_1, q_2, \dots, q_m$$

we need to find the mapping  $k = C(i)$  where,  $1 \leq k \leq n$  and  $1 \leq i \leq m$  so that the distance between the two point sets

$$d(P, Q) = \sum_{i=1}^n \|p_i - q_{C(i)}\| \quad (7.1)$$

reflects the intuitive dissimilarity. There are numerous methods in order to determine point matches. One common tool for this is to compare the shape context (Mori and Malik, 2002) of two points. This method tries to compute every point's relation to the other points in the set. Such an approach exploits global as well as local context of a point. Another method is to use the Hausdorff distance in combination with the image gradient of the points. In other words, two points that are close to each other and have relatively equal gradient are assigned a good match. The drawback with this approach is that the point sets need to be properly aligned beforehand in order to get good results. For the results in this chapter, shape context matching and extended Hausdorff matching turned out to yield very similar results.

Given a similarity measure between two points on a silhouette (hausdorff, shape context, or some other measure), a dissimilarity matrix  $D(n, m)$  is computed. The method used here to determine the matches, given the  $D(m, n)$  is a purely greedy approach iterating over the following steps:

1. Find  $(r, c)$  s.t.  $s = D(r, c)$  is the smallest value in  $D$ .

2. Set  $C(c) = r$
3. Set  $D(r, j) = \infty$  for  $1 \leq j < n$  and  $D(i, c) = \infty$  for  $1 \leq i < m$ .
4. Repeat until  $s > t$  where  $t$  is a manually set threshold.

Fig. 7.4 shows the matching of points in two cases: One case when the silhouettes are very similar, and one case when they differ remarkably.

In order to compare two silhouettes, we can now use equation 7.1. In addition to this measure, the similarity measure should penalize the comparison of two silhouettes where there are a large number of unmatched points. The final similarity measure between two silhouettes is given by

$$d(X_1, X_2) = \alpha \sum_{i=1}^n \|p_i - q_{C(i)}\| \quad (7.2)$$

where  $\alpha$  is the ratio of unmatched and matched points.

## Transferring joint locations

Once the correct key frame is identified, the manually labelled joints are transferred from the key frame to the actual frame. The tracking presented here is designed to only use information from the silhouette. No intensity information from the image is used. This is done by computing the homography,  $H$ , between the points on the key frame silhouette that are spatially close to the hand marked joint, and the corresponding points on the actual frame (using the correspondences  $C$ ), as illustrated in fig. 7.5. A perfect homography would satisfy

$$Hp_i = q_{C(i)}$$

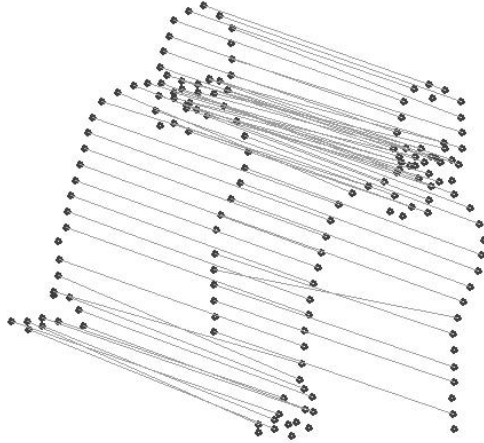
where  $p_i$  are points in the key frame silhouette in the neighborhood of the manually marked joint. The least-squares solution is obtained by seeking for the  $H$  minimizing

$$q_{C(i)} \times Hp_i = \mathbf{0}$$

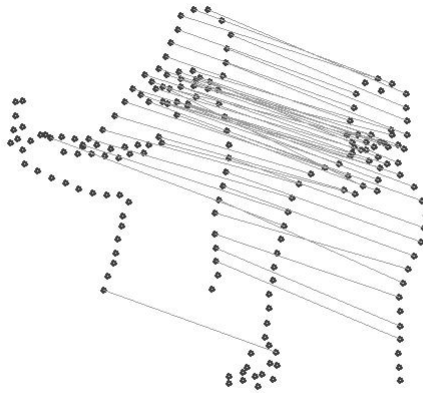
One example of tracking a golf stroke is shown in fig. 7.6. The question now is: How good are these results? They are definitely good enough to produce a nice looking animation, using techniques from the previous chapters. However, are they powerful enough to be useful for the virtual coach? This will be investigated next.

## 7.2 Temporal assessment

One of the most important aspects of an athletic trial is the *pace* of its execution. One important feedback to receive immediately after an attempt is the timing of the trial compared to a reference. Here, golf will be used as an example, but the principle is crucial in most sports. A golf stroke is an extremely repetitive motion. This means that the differences between a key pose and an actual pose is rather small. Thus, a big part of the information can



(a) Point matches between two similar silhouettes



(b) Point matches between two rather different silhouettes.

Figure 7.4: In (a) the silhouettes are quite similar, yielding rather good matches. In (b) on the other hand, the silhouettes differ quite significantly. This results in almost no matches at all between the arms.

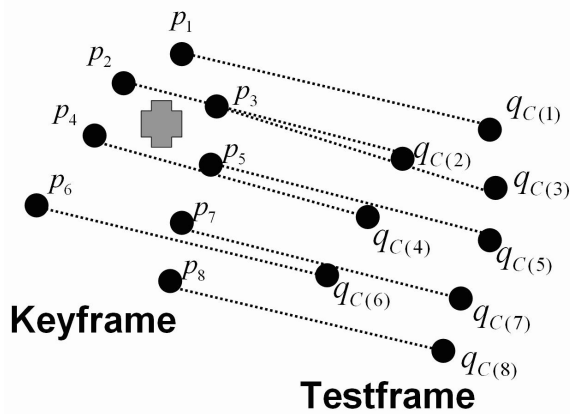


Figure 7.5: Point correspondences between two silhouettes. The homography between the corresponding points is computed in order to transfer the manually marked joint location (the big cross) from the key frame to the test frame.

be obtained from the actual key frame assignments, which will reveal the pace at which a stroke is carried out. This is illustrated in fig. 7.7, where the frames of one stroke is placed on top of the other. The strokes are initially aligned, but as we can see, one of the strokes is carried out faster than the other.

By comparing the silhouettes of one stroke to the silhouettes of the key frames, using the similarity measure described earlier, it is possible to follow how the stroke is executed temporally, in comparison to the key stroke. Another method is to select one feature that is changing significantly over time, and use this feature as references. In the case of golf it is natural to select the motion of the hands to represent the temporal aspects of the stroke. In fig. 7.8, the vertical motion of the hands in two strokes are shown. The distance in the graph is the distance from the top of the image. The strokes measured are shown in fig. 7.7. The dashed curve represents the stroke using an iron club (the stroke on top in fig. 7.7). The solid curve represents the bottom stroke, which is executed using a driver. Notice that when an iron club is used, the hands are initially at a lower position with respect to the rest of the body, since this club is shorter. One question that could be interesting is whether or not the temporal differences are due to the club selection. In fig. 7.9 six curves, similar to the ones in the previous example are shown. Three of the curves (the dashed curves) correspond to iron clubs, and the remaining three correspond to strokes using a driver. By looking at the curves, there is no reason to believe that the club selection plays a major role in the temporal execution of the stroke. At least not for this golfer.

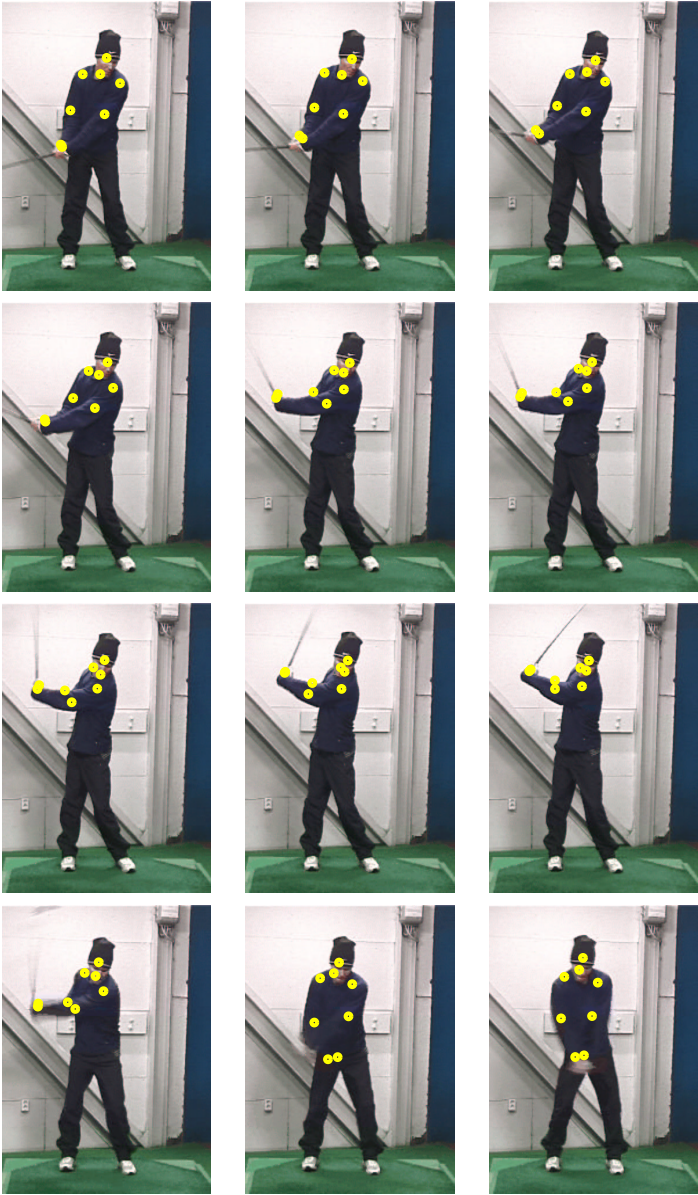


Figure 7.6: Example of automatically tracked joints in one of the strokes

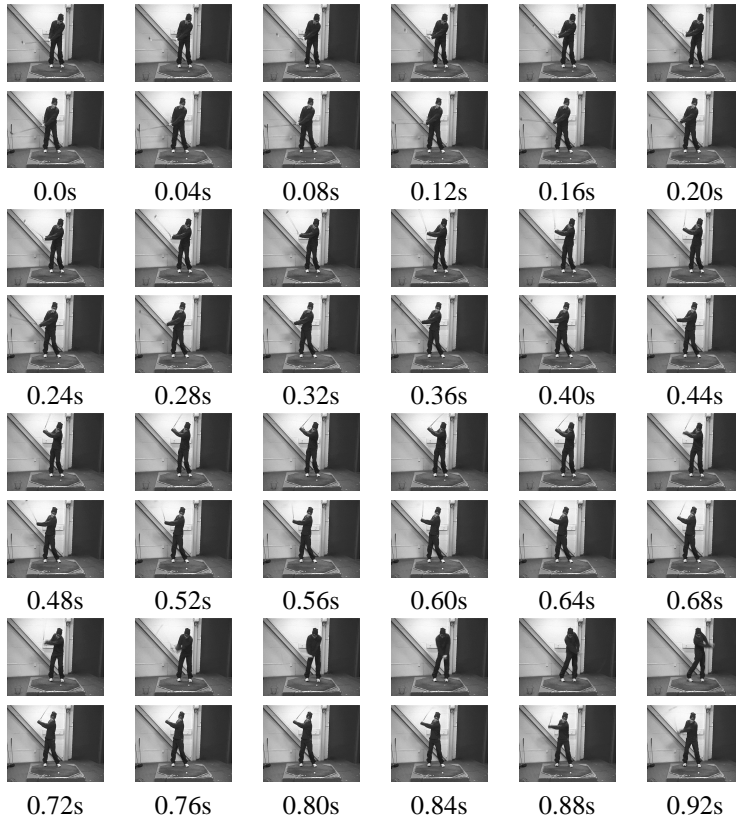


Figure 7.7: Two golf strokes executed at different paces.

### 7.3 Spatial assessment

The next step is to identify to what extent we can use the automatically tracked data in order to decide quantitative differences between a number of strokes. Rather than considering the entire stroke, this study focuses on one important pose of the stroke, namely at the very end of the back-swing. By tracking the joint locations of 18 strokes, the pose where the hands reached the maximum height was used. Three frames from this pose are shown in fig. 7.12. In order to investigate the usefulness of the automatically extracted joint locations, the joints of this pose was manually marked as well in all 18 strokes, to be used as a reference. Again, two dissimilarity matrices are computed. One corresponding to the dissimilarities of the automatically located joints, and one that corresponds to the dissimilarities of the manually marked joints. These matrices can be visualized by using MDS as shown in fig. 7.10. It is very interesting to see that the clusters appearing in the MDS of the manually located joints, appear in the MDS of the automatically tracked



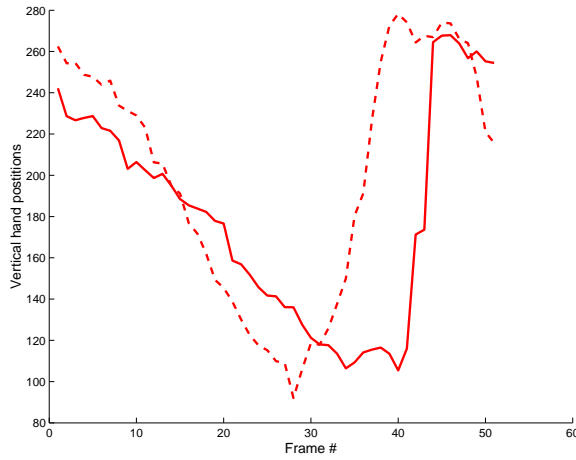


Figure 7.8: The vertical displacement of the hands with respect to the rest of the body of two different strokes. Obviously, the timing of the two strokes differ significantly.

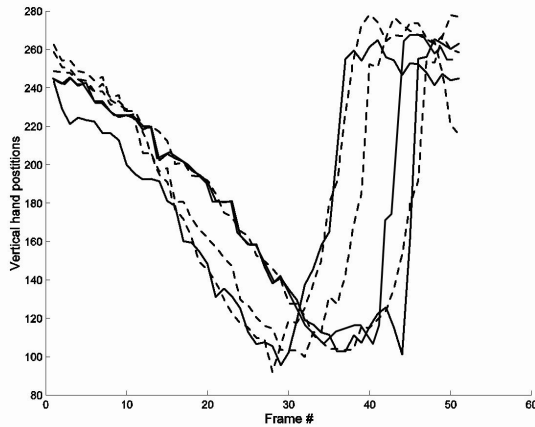


Figure 7.9: The vertical displacement of the hands of six different strokes. The dashed, curves correspond to strokes using an iron club, and the solid curves correspond to strokes using a driver. The choice of club does not seem to strongly affect the pace of the stroke.

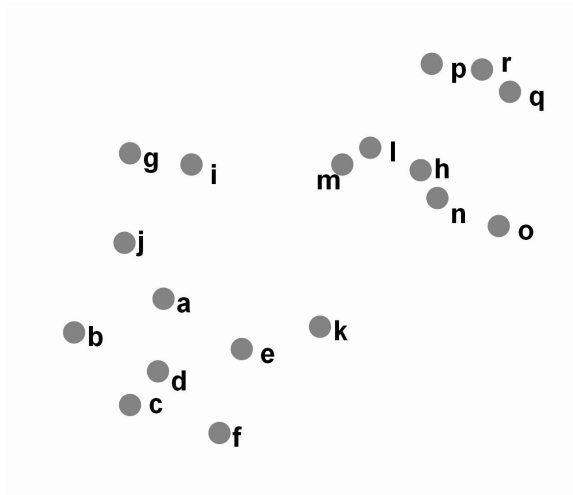
joints as well. These clusters are actually quite interesting; the cluster in the "lower left" corner of the diagrams correspond to strokes using an iron club, while the remainder of the strokes were executed using the driver. It is obvious that, despite the fact that all strokes are almost identical, there are slight variations between the strokes. Further, these variations can be identified by using key framed tracking. As can be seen in the MDS graphs, the clusters of the automatically tracked data do not show the same variance within the clusters, as can be observed when the joint centers are manually located. This is due to the fact that there still exist a slight bias to the reference motion.

### Interpretation

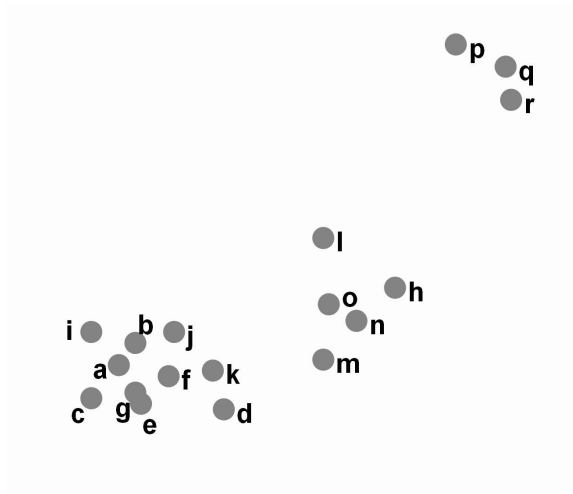
The MDS graphs help us to conclude that some strokes are more similar than others. However, it is impossible to identify and explain exactly *what* the difference was by looking at the MDS graph. In a coach-athlete relationship, the coach must of course be able to communicate to the athlete what the major flaw was in a certain trial. In this case it is possible by looking at the video, to notice that in some strokes (generally when the iron club was used) the player has a slightly "higher" backswing. By comparing the difference between the height (y-value) of the hands in the backswing, and the center of mass of all joints, we can quantitatively observe the height of the backswing. The diagram in fig. 7.11 show that a backswing that is "low" according to the manually labelled joints is also low according to the automatically tracked data. However, the differences are not as significant in the automatic case.

## 7.4 Conclusion

This chapter has given a brief introduction to some aspects of interest in the field of automatic coaching. The main difference from previous chapters is that we have now been dealing with a motion class that is highly repetitive (almost all golf strokes look very similar) in order to see how powerful computer vision can be for the purpose of quality assessment. Apparently, some conclusions can be made, both regarding temporal properties of a motion, as well as some spatial qualities. It is important to keep in mind, though, that an automatic coach can never possess any intelligence about the activity; at least not without very advanced learning. This means that the computer vision aspects of automatic coaching must be complemented with specific knowledge from an expert in the field. The two illustrations in this chapter (the temporal and spatial analysis) are only *examples* on what is important. In order to make a system interesting for commercial purposes, a very good interface needs to be implemented, that allows the user to tell the system how to interpret the results.



(a) Manually marked joints



(b) Automatically located joints

Figure 7.10: Multidimensional scaling. Poses from different strokes, corresponding to the maximum backswing are compared. (a) shows the diagram where the joints are manually located and (b) shows the diagram using automatically located joints. They look qualitatively similar. Of particular interest is that all strokes in the "lower left corner" of the diagrams were executed with an iron club, and the other with a driver. Automatic tracking can at least distinguish between details to a certain level.

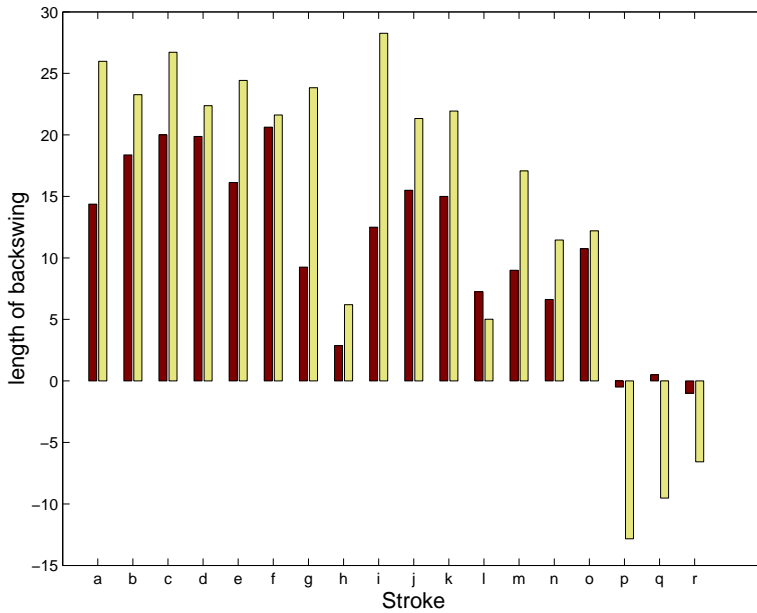


Figure 7.11: The "height" of the backswing of the golfer. The dark bars indicate the height of the backswing according to the manually tracked data, while the dark bars show the same thing for automatically located joint centers. Even though the variance between the automatically tracked data is lower, there is a strong correlation between automatic and manual joint localization.

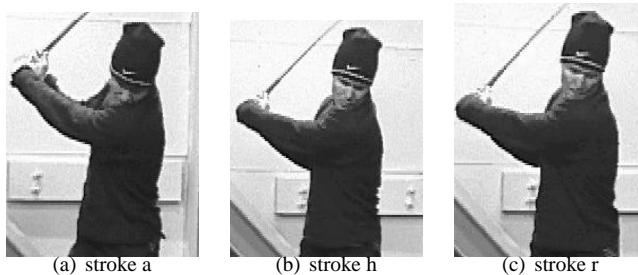


Figure 7.12: Three examples of backswing poses. Each pose is taken from one of each clusters that are formed in the MDS diagrams.

## Chapter 8

# Summary and future directions

In this thesis, visualization and analysis of human action has been investigated from various aspects. While many of the results are very promising, there is still much to do. This closure tries to summarize the thesis, as well as giving an outlook to where the continued research is going.

### 8.1 Summary

The objective has been to present a number of interesting ideas, and also to put these ideas into a working system. A system for visualization differ remarkably from a system intended for motion analysis. This is one of the fundamental points. In order for computer vision techniques to be useful, the requirements must be very specific. Creating a nice looking animation of a live tennis game must be one hundred percent non-intrusive. When developing a system for accurate motion analysis this is not the case. Compromises can be made in order to help the vision algorithms to capture the details requested by the user.

### Visualization

Concerning visualization, the focus of the thesis has been on identifying constraints in order to compute 3D motions from a 2D tracker that occasionally delivers erroneous data. As the purpose of visualization is to obtain nice-looking motion sequences, it is reasonable to sometimes revert to prior knowledge about the motion being reconstructed. The use of qualitative constraints has turned out to be very promising. By automatically learning certain constraints from a set of examples, really bad reconstructions are avoided. The qualitative constraints have been designed in order to avoid excessive biasing of the solution towards the training data. It has also been shown that qualitative properties are sometimes better than a euclidian measure in order to compare two actions. Further, the use of 3D key frames has been shown useful, in the case reconstruction is to be done when no training data is available. A toolkit of methods to use in order to make a sequence look good when the input is contaminated has also been presented.

## **Analysis**

Several important issues for motion analysis have been presented. In order to develop an automatic analysis system based on computer vision, one has to be very clear about the purpose of the system. It is difficult to develop the general "motion analysis machine", since analyzing a golf stroke is rather different from analyzing the gait. It has been showed that computer vision techniques work best if the task is very specific. By using this line of thinking it has been possible to extract some key properties from athletics, that could be useful for professional as well as recreational athletes.

The validity of analysis in 2D has also been investigated. It has been shown that by carefully selecting the camera placement, properties of the original 3D motion still hold when the motion is projected down to 2D. It has also been shown that techniques in monocular reconstruction are generally not useful for the purpose of analysis.

## **8.2 Future research**

Now is the time to decide what to do with the presented ideas. One important thing is to step away from the scientific aspects, and look at the problem from a system point of view. Many tools have been presented, and most of the tools have been implemented in a visualization/analysis system. This system has to be adapted to the user, in order to learn what information coaches, athletes, orthopaedics and other people are looking for.

## **Visualization**

The methods presented in the thesis are very promising. One important future direction is to improve the computational efficiency of the qualitative constraints. When these constraints are applied on long sequences, the number of constraints becomes extremely large, and a more sophisticated method is required to resolve violations. The qualitative constraints can also be exploited in other application domains of statistical learning.

The term "carving priors" has been used. This is of course a never ending direction for future research in this field. Motions in this thesis have exclusively been regarded as point sets in time-space. By using an angular representation of the human model, it is possible that qualitative priors can be designed in joint-space, yielding more efficient algorithms.

## **Analysis**

The most interesting thing to do next would be to design a system that allows an uneducated end-user to guide the system regarding what to analyze. When truly useful data is delivered, such a system is actually interesting for a coach or athlete. One important aspect is that some compromises concerning the "non-intrusiveness" of the system are allowed.

Athletes may for instance be willing to put on a specially designed suit for a practice session, if this is required for the vision algorithm. However, such solutions should be avoided as much as possible, which means that the research in joint extraction on regular video also should go on.





# Bibliography

- Agarwal, A. and Triggs, B. (2004). Tracking articulated motion with piecewise learned dynamical models, *European Conference on Computer Vision*.
- Aichholzer, O., Aurenhammer, F. and Krasser, H. (2002). Points and combinatorics, *Special Issue on Foundations of Information Processing of TELEMATIK* 1(7): 12–17.
- BenAbdelkader, C. and Cutler, R. (2002). Person identification using automatic height and stride estimation, *IEEE International Conference on Pattern Recognition*.
- BenAbkelkader, C. (2002). *Gait as a Biometric for Person Identification in Video*, PhD thesis, University of Maryland.
- Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 257–267.
- Bowden, R. (2000). Learning statistical models of human motion, *IEEE Workshop on Human Modeling, Analysis and Synthesis, CVPR*.
- Bradski, G. and Davis, J. W. (2002). Motion segmentation and pose recognition with motion history gradients, *International Journal of Machine Vision and Applications* 13(3): 174–184.
- Brand, M. (1999). Shadow puppetry, *IEEE International Conference on Computer Vision*, pp. 1237–1244.
- Bregler, C. (1997). Learning and recognizing human dynamics in video-sequences, *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 568–574.
- Bregler, C. and Malik, J. (1998). Tracking people with twists and exponential maps, *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 8–15.
- Bregler, C. and Omohundro, S. M. (1994). Surface learning with applications to lipreading, in J. D. Cowan, G. Tesauro and J. Alspector (eds), *Advances in Neural Information Processing Systems*, Vol. 6, Morgan Kaufmann Publishers, Inc., pp. 43–50.
- Carlsson, S. (2000). Recognizing walking people, *European Conference on Computer Vision*, pp. I: 472–486.

- Cham, T. and Rehg, J. (1999). A multiple hypothesis approach to figure tracking, *IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 239–245.
- Choo, K. and Fleet, D. (2001). People tracking using hybrid monte carlo filtering, *IEEE International Conference on Computer Vision*.
- Deutcher, J., Blake, A. and Reid, I. (2000). Motion capture by annealed particle filtering, *IEEE International Conference on Computer Vision and Pattern Recognition*.
- Drummond, T. and Cipolla, R. (2000). Real-time tracking of multiple articulated structures in multiple views, *European Conference on Computer Vision*, pp. 20–36.
- Efros, A. A., Berg, A. C., Mori, G. and Malik, J. (2001). Recognizing action at a distance, *IEEE International Conference on Computer Vision*.
- Eriksson, M. and Carlsson, S. (2004). Monocular reconstruction of human motion by qualitative selection, *International Conference on Face and Gesture Recognition*.
- Gavrila, D. (1999). The visual analysis of human movement: A survey, *Computer Vision and Image Understanding* **71**(1): 82–98.
- Gavrila, D. and Davis, L. (1996). 3-d model-based tracking of humans in action: a multi-view approach, *IEEE Conference on Computer Vision and Pattern Recognition, 1996*, pp. 73–80.
- Gleicher, M. (1998). Retargetting motion to new characters, *Proceedings of SIGGRAPH*, pp. 33–42.
- Gomes, J. and Mojsilovic, A. (2002). A variational approach to recovering a manifold from sample points, *European Conference on Computer Vision*, pp. 3–17.
- Guo, S. and Robergé, J. (1996). A high-level control mechanism for human locomotion based on parametric frame space interpolation, *Proceedings of the Eurographics workshop on Computer animation and simulation '96*, Springer-Verlag New York, Inc., pp. 95–107.
- Halvorsen, K. (2002). *Model-based Methods in Motion Capture*, PhD thesis, Uppsala Universitet.
- Heap, T. and Hogg, D. (1998). Wormholes in shape space: Tracking through discontinuous changes in shape, *IEEE International Conference on Computer Vision*, pp. 344–349.
- Herda, L., Fua, P., Plänkers, R., Boulic, R. and Thalmann, D. (2002). Using skeleton-based tracking to increase the reliability of optical motion capture, *International Conference on Computer Graphics and Interactive Techniques*, pp. 612–619.
- Hodgins, J. and Pollard, N. (1997). Adapting simulated behaviors for new characters, *SIGGRAPH '97*.

- Hogg, D. (1983). Model-based vision: a program to see a walking person, *Image and Vision Computing* 1(1): 5–20.
- Holden, E. and Owens, R. (2003). Recognising moving hand shapes, *International Conference on Image Analysis and Processing*.
- Howe, N. R., Leventon, M. E. and Freeman, W. T. (1999). Bayesian reconstruction of 3d human motion from single-camera video, *Technical Report TR-99-37*, MERL-A Mitsubishi Research Laboratory.
- Ioffe, S. and Forsyth, D. (2001). Human tracking with mixtures of trees, *IEEE International Conference on Computer Vision*, pp. 690–695.
- Isard, M. and Blake, A. (1998). Condensation - conditional density propagation for visual tracking, *International Journal of Computer Vision* 29(1): 5–28.
- Ju, S. X., Black, M. J. and Yacoob, Y. (1996). Cardboard people: A parameterized model of articulated motion, *International Conference on Automatic Face and Gesture Recognition*, pp. 38–44.
- Kakadiaris, I. A. and Metaxas, D. (1996). Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection, *IEEE Conference Computer Vision and Pattern Recognition*, pp. 81–87.
- Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K. and Pollard, N. S. (2002). Interactive control of avatars animated with human motion data, *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, pp. 491–500.
- Lee, J. and Shin, S. Y. (1999). A hierarchical approach to interactive motion editing for human-like figures, *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 39–48.
- Lee, L. and Grimson, W. (2002). Gait analysis for recognition and classification, *IEEE Conference on Face and Gesture Recognition*, pp. 155–161.
- Leventon, M. E. and Freeman, W. T. (1998). Bayesian estimation of 3d human motion from an image sequence, *Technical Report TR-98-06*, MERL-A Mitsubishi Research Laboratory.
- Liebowitz, D. and Carlsson, S. (2001). Un-calibrated motion capture exploiting articulated structure constraints, *IEEE International Conference on Computer Vision*.
- Little, J. and Boyd, J. (1998). Recognizing people by their gait: the shape of motion., *Videre* 1(2).
- Loy, G., Eriksson, M., Sullivan, J. and Carlsson, S. (2004). Monocular 3d reconstruction of human motion in long action sequences, *European Conference on Computer Vision*.

- Mittal, A., Zhao, L. and Davis, L. (2003). Human body pose estimation using silhouette shape analysis, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 263–270.
- Moeslund, T. B. and Granum, E. (2001). A survey of computer vision-based human motion capture, *Computer Vision and Image Understanding: CVIU* **81**(3): 231–268.
- Mori, G. and Malik, J. (2002). Estimating human body configurations using shape context matching, *European Conference on Computer Vision*.
- Ngo, J. T. and Marks, J. (1993). Spacetime constraints revisited, *Computer Graphics* **27**(Annual Conference Series): 343–350.
- Park, M. J., Choi, M. G. and Shin, S. Y. (2002). Human motion reconstruction from inter-frame feature correspondences of a single video stream using a motion library, *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 113–120.
- Pavlovic, V. and Rehg, J. M. (2000). Impact of dynamic model learning on classification of human motion, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 788–795.
- Remondino, F. and Roditakis, A. (2003). Human figure reconstruction and modelling from single image or monocular video sequences, *International Conference on 3D Imaging and Modeling*, pp. 116–123.
- Rohr, K. (1994). Towards model-based recognition of human movements in image sequences, *CVGIP:Image Vision and Understanding* **59**(1): 94–115.
- Rose, C., Cohen, M. and Bodenheimer, B. (1998). Verbs and adverbs:multidimensional motion interpolation, *IEEE Computer Graphics and Applications* **18**(5): 32–40.
- Rose, C. F., Sloan, P.-P. J. and Cohen, M. F. (2001). Artist-directed inverse-kinematics using radial basis function interpolation, *Eurographics*, Vol. 20.
- Shoemake, K. (1985). Animating rotation with quaternion curves, *Proceedings of SIGGRAPH*, pp. 245–254.
- Sidenbladh, H. and Black, M. J. (2002). Implicit probabilistic models of human motion for synthesis and human tracking, *European Conference on Computer Vision*.
- Sidenbladh, H., Black, M. J. and Fleet, D. J. (2000). Stochastic tracking of 3d human figures using 2d image motion, *European Conference on Computer Vision*, pp. 702–718.
- Sidenbladh, H., la Torre, F. D. and Black, M. J. (2000). A framework for modeling the appearance of 3d articulated figures, *International Conference on Automatic Face and Gesture Recognition*, pp. 368–375,.

- Sminchisescu, C. and Triggs, B. (2001). Covariance scale sampling for monocular 3d body tracking, *IEEE Conference on Computer Vision and Pattern Recognition*.
- Sminchisescu, C. and Triggs, B. (2002). Building roadmaps of local minima of visual models, *European Conference on Computer Vision*, Vol. 1, pp. 566–582.
- Sminchisescu, C. and Triggs, B. (2003). Kinematic jump processes for monocular 3d human tracking, *IEEE International Conference on Computer Vision*.
- Sullivan, J. and Carlsson, S. (2002). Recognizing and tracking human action, *European Conference on Computer Vision*, Vol. 1, pp. 629–644.
- Taylor, C. J. (2000). Reconstruction of articulated objects from point correspondences in a single image, *Computer Vision and Image Understanding: CVIU* **81**(3): 349–363.
- Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method, *International Journal of Computer Vision* **9**(2): 137–154.
- Wachter, S. and Nagel, H. (1999). Tracking persons in monocular image sequences, *Computer Vision and Image Understanding* **74**(3): 174–192.
- Witkin, A. and Kass, M. (1988). Spacetime constraints, *Computer Graphics* **22**(4): 159–168.
- Wren, C. R., Azarbayejani, A., Darrell, T. and Pentland, A. P. (1997). Pfnder:real-time tracking of the human body, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7): 780–785.
- Yacoob, Y. and Black, M. (1998). Parameterized modeling and recognition of activities, *IEEE International Conference on Computer Vision*, pp. 120–127.