Radboud University, Nijmegen

Bachelorthesis

DigiCoach

An exploration in the field of Human Motion Analysis

Author:
Erik Crombag

Supervisor:
dr. Tom Heskes

January 27, 2008

Contents

List of Figures

List of Tables

Foreword

In our world the computer has rapidly become integrated in our every day life. Where would we be without it? Together with this rapid integration, many technological advancements are achieved and new ways of using the computer are explored. This thesis addresses the use of visual information and more specifically estimating the pose of a human in three dimensions using only two dimensional video footage.

The subject was inspired by my other hobby, track and field athletics. Being a sprinter, your technique is very important and the difference between a good and a bad race is a matter of details. Seeing the flaws in an athletes technique is not that easy. It all happens very fast and you often wish that there were means to review your performance and receive feedback on what is going wrong. This gave me the idea to dive into the field of Human Motion Analysis and try to design a system which can do exactly that. Unfortunately such a system is still far away, but the road I have taken up to this point is described in this thesis.

I would like to thank my supervisor Dr. Tom Heskes for taking the time and patience to guide me through the process, because it took some time. Furthermore I would like to thank my coach Michael for the advice he gave on how to approach the subject from the point of view of an athletics coach. Finally I would like to thank my family and friends for supporting me and pushing me forward. Without you all, I probably would not have succeeded.

I hope you enjoy reading the report on my journey into the field of Human Motion Analysis.

Chapter 1

Introduction

1.1 Human Motion Analysis

The field of computer vision has always drawn the attention of many researchers. As shown by Oviatt in [15] it is important to look for different ways in which computers can interact with their surroundings and vision is certainly one of them, often a picture tells more than a thousand words.

A speciality within the field of computer vision is called "Human Motion Capture", "Looking at People", or "Human Motion Analysis". It concerns itself with the movement of humans and the interpretation of those movements by a computer.

The applications for systems which are able to interpret human motion are numerous. Table 1.1 gives a general overview of the different areas in which such systems can be deployed.

1.2 Problem Definition

1.2.1 General Problem

Making a computer see is not that easy. Humans are able to recognize and classify different objects within the blink of an eye, but for a computer recognizing and classifying unconstrained visual information is nearly impossible. Among others, problems which have to be solved concern:

Changes in illumination During the process of capturing the data lighting conditions may change. Furthermore there is the problem of shadows being cast.

Background dynamics Not only our subject moves, but in the background objects may pass by or the wind moves the plants and trees.

Object motion Everyone moves in a different way and the direction and speed of movement related to the camera is not guaranteed.

Different individuals Each person is unique. We all have different looks and move in different ways.

Constraints are made in order to reduce the problem space. However these constraints result in a less general approach. Depending on these constraints various methods have been developed,

General domain	Specific Area
Virtual reality	- interactive virtual worlds - games - virtual studios - character animation - teleconferencing (e.g.film, advertising, home-use)
"Smart" surveillance systems	- access control - parking lots - supermarkets, department stores - vending machines, ATM's - traffic
Advanced user interfaces	- social interfaces - sign language translation - gesture driven control - signaling in high noise environments (airports, factories)
Motion analysis	- content based indexing of sports video footage - personalized training in golf, tennis etc. - choreography in dance and ballet - clinical studies of orthopedic patients
Model based coding	- very low bitrate compression

Table 1.1: Applications of "Looking at people". Taken from [5].

each for a specific application. Unfortunately there is no 'best practice' and if one wants to develop a system which is able to interpret human motion one needs to see whether there is a method available.

1.2.2 Research Aim

In this thesis I will look into the field of human motion analysis and try to propose a model for a system which is able to play the role of a digital coach, called DigiCoach[1]. The system will be able to analyse video-footage shot by a single camera, translate this footage into a 3D-model representing the motion of the athlete and be able to analyse this model and compare it to a representation of the perfect motion in order to improve the performance of the athlete. A more comprehensive description can be found in appendix A.

With this application I will try to see whether the research field is able to provide solutions which could aid me solving the problem of creating the system described. It is not my aim to develop new methods, but to adapt current methods in order to create my own model which can be used to solve the problem.

[1]This system will fit within the general domain of *motion analysis* from table 1.1

1.3 Research Questions

1.3.1 Main Question

The aim of my research is to develop a system which is able to convert a sequence of 2D images into a 3D model representing the motion of the athlete. In general I am interested to see:

How can human motion be captured and tracked by a computer?

The answer to the main question will consist of a small literature survey of the field of human motion analysis, a model of a system which is able to translate human motion in a sequence of 2D images to a 3D representation, and an implementation of the model.

1.3.2 Subquestions

The main question is quite broad and provides few leads how it should be answered. Given the contents of the final answer, I came up with a number of subquestions which will provide some lead how to conduct my research. These questions are:

1. What difficulties arise when trying to track human motion?
2. What methods have been developed in order to track human motion?
3. Which difficulties does each method try to solve?
4. What are the strong and weak points of each solution?
5. Which solution is best suited for my purpose?
6. How can this solution be modeled?
7. How can this solution be implemented?

1.4 Structure

This thesis contains the following chapters:

Introduction This chapter.

Human Motion Analysis A study of the field of human motion analysis and a comparison of the different methods proposed. In this chapter subquestions 1 to 4 will be answered.

Probabilistic Framework The analysis of one specific solution and the description of the model I am going to use. Subquestions 5,6 and 7 will be answered in this chapter.

Temporal Prior An elaboration on how the transition model or temporal prior should be defined.

Likelihood The definition of the likelihood or sensor model.

Discussion Some remarks on issues which are unaddressed in the previous chapters.

Conclusions Some concluding remarks.

Chapter 2

Human Motion Analysis

Now that the research is introduced, we can look into the field of Human Motion Analysis. First a general overview is given, defining the research field, the general structure and introducing some common assumptions concerning HMA-systems. Next we look into the functional decomposition of an HMA-system and the methods developed to solve the problems which arise in each different building block. This chapter does not provide a complete survey of the research field, only methods relevant to the intended system are discussed.

2.1 General Overview

2.1.1 Definition

Human motion analysis (HMA) is the field of research which attempts to capture and analyse human motion within a sequence of images and tries to translate it into a (mathematical) model. In recent years many different scientists have studied this problem and many different methods have been proposed [1, 5, 13, 14]. Moeslund and Granum give it the following formal definition [13]:

Definition 1
Human motion analysis is the process of capturing the large scale body movements, of a subject at some resolution.

They added the words *"at some resolution"* to emphasise that the field covers everything between the tracking of the subject as a whole to the tracking of its different limbs. It does not cover the tracking of small scale movements such as facial expressions and hand gestures.

2.1.2 General Structure

Moeslund and Granum describe an overall structure of a comprehensive HMA-system [13]. This structure consists of four separate components, shown in figure 2.1 (on page 9). Before the system can process the acquired data it needs to be *initialized*. In this component some general features of the model are initiated, e.g. specific information about the subject's size, kinematic structure and appearance in order to constraint tracking and pose estimation [14]. After the

Related to Movements	Related to Appearance
1. The subject remains inside the workspace 2. None or constant camera motion 3. Only one person in the workspace at the time 4. The subject faces the camera at all times 5. Movements parallel to the camera plane 6. No occlusion 7. Slow and continuous movements 8. Only move one or a few limbs 9. The motion pattern of the subject is known 10. Subject moves on a flat ground plane	**Environment** 1. Constant lightning 2. Static background 3. Uniform background 4. Known camera parameters 5. Special hardware **Subject** 1. Known start pose 2. Known subject 3. Markers placed on the subject 4. Special coloured clothes 5. Tight-fitting clothes

Table 2.1: Typical assumptions made by HMA-systems. They are ranked in order of frequency. From [13].

initialisation *tracking* of the subject's motion can take place. This component tries, for each image in the sequence, to segment the subject from the background in a single image and tries to relate the current image with previous frame(s). *Pose estimation* tries to estimate the positions of the joints in the kinematic structure or skeleton of the model in such a way that they match the segments found in the tracking process. This component may provide the output for the system when we are only interested in the pose of the subject, e.g. controlling an avatar in a virtual environment. However there is a fourth component, *recognition*. In this component low level kinematic information is mapped onto some higher level conceptual information. For example gesture recognition or the recognition of suspicious behaviour by a surveillance system. In section 2.3 each separate component is described in more detail.

A system does not need to implement all four components. This holds for most systems described in the scientific literature, because most research focuses on one specific component, they only demonstrate the functionality of that specific component.

2.1.3 Assumptions

Creating an HMA-system means solving a great number of problems and if we do not constrain ourselves this number of problems becomes too large. An easy way to limit ourselves is to make basic assumptions about various variables in our environment. These will provide some basic foundation and characterise our system. Within the area of human motion analysis there are some assumptions which are very common. They may be divided into two classes, *movement* and *appearance assumptions*, concerning either the movement of the subject or the environment and the appearance of the subject. In table 2.1 the most common assumptions are listed ranked by occurrence.

The first three assumptions about movement are used in almost every system. The fourth assumption concerns mainly advanced user interfaces. The fifth assumption can be used to reduce the problem from a 3D to a 2D problem eliminating depth. The next two assumptions greatly simplify the tracking process, since all limbs are visible at all time and there are no sudden or jerky movements. The eighth and ninth assumption are used to reduce the problem space by focusing on part of the subject or on a fixed trajectory and finally the tenth assumption allows calculation between the camera and subject using basic geometry and the size of the subject.

Environmental assumptions are made to ease the process of segmentation. The first and second assumption make sure that the image only changes as an effect of movement of the subject. The third assumption makes it very easy to segment the subject from the background using simple thresholding. The fourth assumption is used if it is necessary to know the exact measures of the subject and the final assumption concerns the use of special hardware like IR-cameras.

The first assumption about the subject greatly simplifies the initialisation problem since the system knows where and how the movement cycle starts. The second assumption is about basic model parameters of the subject, e.g. number of limbs, degrees of freedom, width and height, etc. The last three assumptions all concern the segmentation process, making it easier to segment the subject from the background. The last assumption helps estimating the pose, since tight clothing nicely follows the body contours.

Of course these are not all assumptions which can be made, but these are the most common ones. The assumptions made depend on the specific purpose and (external) constraints of the system.

2.2 Applied Assumptions

In order to determine which assumptions apply for the DigiCoach-system we first need to specify some requirements. These can be found in appendix A. From these requirements the assumptions given in table 2.2 can be deduced. The first assumption is a bit odd since it does not reduce the problem space. The fact that footage shot in any environment should be possible only broadens the problem. However it reduces the amount of methods available, since some require a strictly controlled environment. The fact that only a single camera is used eliminates all multi-ocular approaches and knowing that the camera (and as such most of the background) remains static, makes it easier to distinguish our athlete from the background. Furthermore we assume that the subject remains within the workspace[1]. That is, the subject may move in and out of the workspace, but tracking will start when the subject enters the workspace and stops when it moves out. There is no need to predict the movements of the subject once it has left the workspace. We also assume that only one person is in front of the camera, also reducing the problem of separating our subject and the background since we expect only one object to make large movements.

All remaining assumptions concern the movements or appearance of our subject. First we assume that the subject has a humanoid shape and sticks to his activity which is running and does not make any sudden 'jerky' movements. This combined with the fact that we know the subject will be running allows us to guess what the pose in the next frame will be, allowing an informed search. The known start pose will ease the initialisation of the tracking. In practice the

[1] In this case workspace equals the camera's view

Applied Assumptions	
1.	Any environment
2.	Single camera
3.	No camera motion
4.	Subject remains within workspace
5.	Only one person within the workspace
6.	No jerky movements
7.	Known subject (human)
8.	Known activity (running)
9.	Known start pose
10.	Tight-fitting clothes

Table 2.2: Assumptions which apply to the DigiCoach-system, given the requirements in appendix A.

system will ask the user to estimate the pose for the first frame the subject entered the workspace. Finally tight-fitting clothes will increase the accuracy of the pose estimation since it is important that the position of the limbs is estimated correctly, not hampered by any excess clothing, in order to provide good recommendations.

2.3 Functional Decomposition

In this section the four functional components, plus the preliminary step of *data acquisition*, are described in more detail. For each I give the functionality, which data are processed in this component and for what purpose. Furthermore I will try to give insight into the difficulties that arise, when we try to process the data and discuss methods which address these problems.

2.3.1 Data Acquisition

Moeslund and Granum do not include the process of *data acquisition* in their functional decomposition. Although not part of the actual system it has great influence on the external constraints, since the way the data is captured largely determines which kind of and how much data is available. For example active sensing will probably produce a series of coordinates or angles while passive sensing produces a sequence of images and if multiple cameras are used occlusion might be eliminated. The process can also be reversed, allowing the constraints to determine how the data acquisition takes place.

In order to capture the body movements of a subject, two approaches can be used. They are called *active* and *passive sensing*. Active sensing allows for the placement of sensors on the subject, tracking its motion. Passive sensing uses only natural signal sources, e.g. visual or infrared light, and there is no need for any sensors worn by the subject. A mixture between the two approaches is also possible, e.g. the use of visual markers.

The great benefit of passive sensing is that it is not as intrusive as active sensing. “It allows in principle for touch-free and more discrete “pure” motion capture systems” [13]. Another benefit is that it is far more easy to acquire the required data when using passive sensing. No need for a strictly controlled environment and wearable devices. Some light and a camera will suffice. This

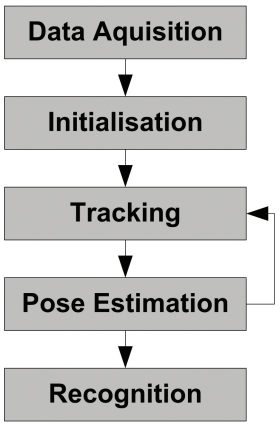


Figure 2.1: Functional decomposition of a HMA-system.

simplicity also comes with a drawback since the acquisition of information is so simple, it will be harder to process it.

2.3.2 Initialisation

Before we can actually process the images recorded and track our subject, we first need to *initialize* our system. Within this component two problems are addressed: finding a model which fits the subject and finding the subject's initial pose. A correct model and start pose improve tracking and pose estimation as they constraint the search space of these problems.

The first major problem in initialisation is finding a correct model to match the subject. This may seem straightforward, all humans have the same skeletal structure, but since no two humans are exactly alike there is the need to adapt for instance limb lengths, or the appearance of the body and face. Finding the model can be broken down into different subject's: kinematic structure, 3D shape and appearance.

Kinematic Structure

By defining the kinematic structure we specify which parts of our model move, where the joint locations are, how many degrees-of-freedom each joint has and we specify the limb length between each joint. The majority of HMA-systems uses a-priori knowledge about the human body to specify the various joints and their degrees-of-freedom, leaving only the estimation of limb lengths. These lengths can then either be specified by the user, with regard to anthropomorphic constraints [2] or can be detected by the system [10, 17].

Allowing the user to specify an initial estimation of the position of some of the subject's joints-locations provides a good basis for an estimation of the limb lengths and starting pose, especially when it is combined with a database of information about the constraints and ratios of a human body [2]. A potential problem for this approach is the fact that not every viewpoint can be initialised. There have to be some limbs which are oriented in a plane which is almost parallel to the viewing plane of the camera. Otherwise no accurate length estimate can be made.

Methods which allow for autonomous recognition of the subject's kinematic structure infer the

joints and corresponding limbs from the movement of the subject [10]. These systems allow for autonomous operation, but it is not guaranteed that the subject is modeled correctly. Furthermore these systems cannot handle occlusions very well.

Shape and Appearance

Some systems may require an estimation of the shape or appearance of the tracked subject. This may be for output purposes (consider a 3D-avatar), but it may also assist the tracking process [20]. Sidenbladh for instance maps each image at time t onto the limbs of the estimated 3D-model in order to predict the input-image at $t+1$. There are also systems which model the full appearance of the subject [3], but this is beyond the scope of the DigiCoach-system.

2.3.3 Tracking

The function of the *tracking*-phase is two-fold. The first processing step taken is *figure-ground segmentation* - determining which pixels in the source image describe the subject and which describe the background. The second process is called *temporal correspondences* - a process which establishes a relationship between the current and the previous image, describing in which way the pose of the subject has altered between the two images. By combining all these relationships over the total number of images we get a description of the subject's motion. Another useful feature of tracking is, once such a motion-description has been established this can be used to predict how the next frame will look like reducing the search space for figure-ground segmentation.

Figure-Ground Segmentation

By applying *figure-ground segmentation* one divides each pixel in the image into one of two categories. The ones that belong to the (human) subject we are tracking and the ones that belong to the background.

The most common technique for figure-ground segmentation uses the motion of the subject as its main cue. Pixels which stay constant over time belong to the background and when a pixel changes due to the movement of the subject it is classified as a foreground-pixel. Before the late 1990's figure-ground segmentation depended mostly on non-adaptive models. Models in which, once established, the background model never becomes updated. These models are not robust. They cannot cope with changes due to lighting conditions or other arbitrary changes in the scene [21]. An example of a non adaptive approach is taking the mean value for each pixel of a series of initialisation images. If the difference between a pixel value in the current image and the mean value exceeds a certain threshold the pixel belongs to the foreground, otherwise it is a background pixel.

A solution to the problem of non-adaptive background models is presented by Stauffer and Grimson [21]. They proposed the idea to represent each pixel not by its value, but by a *'Mixture of Gaussians'* (MoG). Each pixel in the image is represented by a number of Gaussian distributions, representing a different color-value and these Gaussians are updated each image allowing the background-model to adapt. The problem using a MoG-representation however is that it takes a long time to initialise and it takes long for false positives to sink in [9].

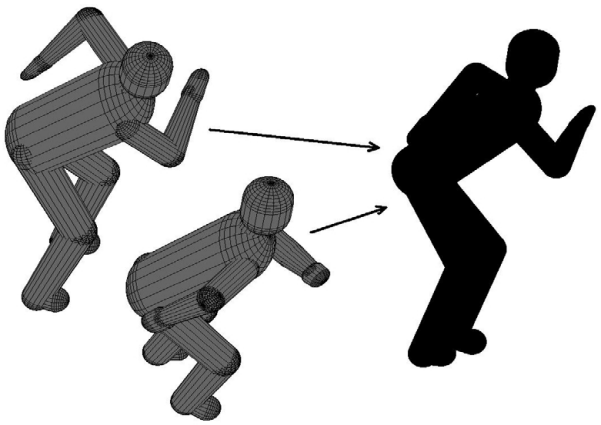


Figure 2.2: Left-right ambiguity. This figure illustrate one of the problems of pose estimation as both models, which look different in a three dimensional view, map to the same silhouette in a two dimensional image when looked at from a specific viewpoint, in this case from the side under orthographic projection [7].

Temporal Correspondences

The next task in the tracking phase is to find the temporal correspondences. That is, finding a function $t = f(t-1)$ which maps the state-change of the subject from time $t-1$ to t. In other words, finding a function which describes the temporal trajectory through the state-space.

Finding these correspondences is often closely linked to pose estimation as most generative systems use a model to keep track of the current location in the state-space, based on the information gained from all previous locations. For example Sidenbladh [20] keeps track of a velocity-vector describing the temporal dynamics of the tracked subject over time. Methods like the Kalman-filter can then be deployed to estimate the state-space position of the next frame. This information is generated during *pose-estimation*, hence the feedback-arrow in figure 2.1.

2.3.4 Pose Estimation

The term *pose estimation* refers to the process of estimating the position of the elements of the subject's kinematic structure for each frame. There are several challenges when addressing this problem. First there is the fact that given a 2D image taken from a given angle there are always two or more 3D configurations which lead to that specific 2D image. This is called left-right ambiguity [7] (see figure 2.2). Another problem is handling occluded body-parts. Both problems are of greater concern for systems which are monocular, since there is no other angle which can be reviewed.

There are various classes of methods for pose estimation, but given the assumptions of our system only the 'direct model use'-class remains as others require a multi-ocular approach.

Direct Model Use

Within this class of methods the model plays an active role in the recognition process [12]. The parameters of the model are matched against the image data, and if needed adapted, in order to find a configuration which fits the data. These methods rely on the synthesis of model parameters and verification against data. Therefore this class is also referred to as *analysis-by-synthesis*.

The majority of the older approaches (prior to 2000) used some form of deterministic gradient descent technique to iteratively estimate changes in pose [14]. A weakness of this approach is that only one model is used during the whole tracking period. This model gets updated each frame, however if ambiguities occur or the subject moves rapidly this updating can go horribly wrong. In order to counter this problem researchers have turned more recently toward deterministic or stochastic search methods or multiple hypothesis tracking. They turn, among others, to multiple Kalman filters [4] or sampling methods [8, 20].

2.3.5 Recognition

The field of action and activity representation and recognition is relatively old, yet still immature. This area is presently subject to intense investigation which is also reflected by the large number of different ideas and approaches [14].

Most research in this area focuses on the classification of human movement. For example separating 'irregular' activities from ordinary movement patterns. As stated before many different methods have been developed employing various techniques and focusing on different abstraction levels. From full scene analysis to investigating the subtle movements of an individual. However this more 'surveillance-like' approach does not fit within the context of this thesis. Therefore this subject will not be further investigated.

The final aim of the digital coach is to interpret the movements of the subject, but at a very low abstraction level. For example we are interested to see whether the knee-angle reaches 90 degrees. Perhaps it is better to speak about constraint satisfaction instead of recognition.

Chapter 3

Probabilistic Framework

Now that we have seen a general overview of the field of Human Motion Analysis it is time to look more specifically at the problem introduced in the first chapter, the creation of a digital coach. First and foremost it is important to establish a foundation for our tracking algorithm by specifying a probabilistic framework.

3.1 Various Approaches

When tracking a human model three approaches can be taken. One way of approaching this problem is to extract local image information like edges and corners, and combine them into higher order information like limbs. This approach is called the bottom-up approach as you start at the lowest level of information possible (individual pixels) and work your way up towards higher order structures. However this approach experiences problems when it encounters occlusions, the computer does not know when a limb gets occluded and becomes invisible [19].

A second method is to evaluate a known database of poses and corresponding silhouettes against the found image. This approach is computationally the least expensive, but it requires a large database of poses and will fail if no match can be found.

3.2 Bayesian Inference

But what if we could generate the needed database on the fly? If we take a model of a human which can be bent into various poses by modifying its joint parameters ϕ_t for each time step t. Using a top-down method we could start with a high-order structure, a set of joint parameters, and turn this into a low-order structure, a silhouette of the model from a certain viewpoint. This method is called analysis-by-synthesis, since you synthesize your database of known poses on the fly.

It can then be projected onto the image plane and compared to the image data. We should try to find the best fit. Finding the best fit can be seen as a search or optimisation task. Trying out every possible configuration for each frame would be an impossible task. Therefore we should seek means to incorporate information from previous frames. We can approach this problem using

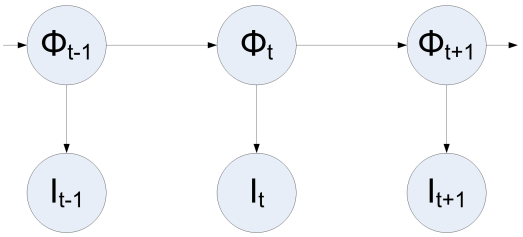


Figure 3.1: Bayesian network structure describing our state space. The transition model is expressed as $p(\phi_t|\phi_{t-1})$ and the sensor model as $p(\mathbf{I}_t|\phi_t)$.

Bayesian inference. It provides methods for incorporating prior information on size, shape and possible configurations. It is impossible to learn this information [19].

A drawback of choosing Bayesian inference is the high computational load. A complex human model has to be compared directly with the image data, which leads to a higher amount of computations then a bottom-up approach, where the image data is segmented first and then used in higher level operations. However since the intended system is not used with online data or in real time, computational complexity is less of an issue.

3.3 Mathematical formulation

The general problem we are trying to solve in this thesis is: Given a series of images, what is the corresponding pose for each image? We can rephrase this as: For each image, given a series of previous images and the corresponding poses, what is the pose corresponding to the last image? It should be noted that this description does not include the initialisation step of finding the pose corresponding to the first image, without any previous information. This is because initialisation is a challenge requiring a different approach.

If we formulate the problem using Bayesian inference we would need to find a formula which yields the following *posterior distribution*: $p(\phi_t|\vec{\mathbf{I}}_t)$, where ϕ_t equals the pose of the human model and $\vec{\mathbf{I}}_t$ is a vector containing all previous and the current image $[\mathbf{I}_0 \dots \mathbf{I}_t]$, expressing the probability that a configuration ϕ_t matches image $\mathbf{I}_t$. We seek the configuration with the highest probability.

We may assume a first-order Markov process - the configuration ϕ_t depends only on ϕ_{t-1} and not on any earlier states [20]. We now can model the problem as shown in figure 3.1. The transition model, propagating the states in time can be expressed as $p(\phi_t|\phi_{t-1})$ and the sensor model, linking the internal state space to observable images can be expressed as $p(\mathbf{I}_t|\phi_t)$. The posterior distribution $p(\phi_t|\vec{\mathbf{I}}_t)$ can now be expressed in terms of the transition and sensor model.

posterior distribution at t = sensor model at t × transition model between t-1 and t × posterior distribution at t-1

Because the posterior distribution at $t-1$ is not a discrete value but a probability distribution we have to take the integrand over all possible configurations ϕ_{t-1}.

Using all of the above we can express the posterior distribution $p(\phi_t|\vec{\mathbf{I}}_t)$ at time t as:

$$p(\phi_t|\vec{I}_t) = \kappa p(I_t|\phi_t) \int p(\phi_t|\phi_{t-1}) p(\phi_{t-1}|\vec{I}_{t-1}) d\phi_{t-1} \tag{3.1}$$

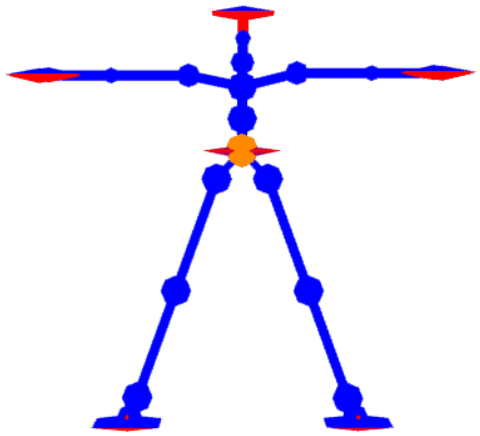


Figure 3.2: The human model used in this system. It consists of 31 joints, each able to rotate in 3 dimensions.

where I_t is the image data at time t and κ an independent normalizing constant [19].

The posterior distribution $p(\phi_t|\vec{I}_t)$ represents all knowledge extracted about the model configuration and can be used for further processing, like recognition or motion reconstruction. The sensor model $p(I_t|\phi_t)$ is also called the *likelihood* of observing image I_t given a model configuration ϕ_t. The transition model $p(\phi_t|\phi_{t-1})$ is also referred to as the *temporal prior* or temporal correspondence. The integral as a whole can be regarded as a *prediction*, since it is a multiplication of the posterior distribution at $t-1$ and an estimation how the model will change between $t-1$ and t.

Now that we have a found a formal definition of our problem, we should seek for formulas which yield the likelihood (sensor model) and the temporal prior (transition model).

3.4 Human model

Before we can start looking for the likelihood and temporal prior, we first need to define a human model and its corresponding parameters ϕ.

The human model used in this system consists of a stick figure with 31 joints, each able to rotate around all 3 angles. This model is derived from the data obtained from Carnegie Mellon's motion capture library[1] and is encoded using the BioVision bvh-file format[2]. A more detailed description is given in appendix B. Together with the global position of the model, our parameter vector ϕ would contain 96 dimensions (the translation in 3 dimensions plus 93 joint angles).

[1] `http://mocap.cs.cmu.edu/`
[2] `http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html`

Chapter 4

Temporal Prior

Now that we have obtained a formal definition of the problem, we need to find a formal definition for the transition model or temporal prior. This is done by extracting information from known examples.

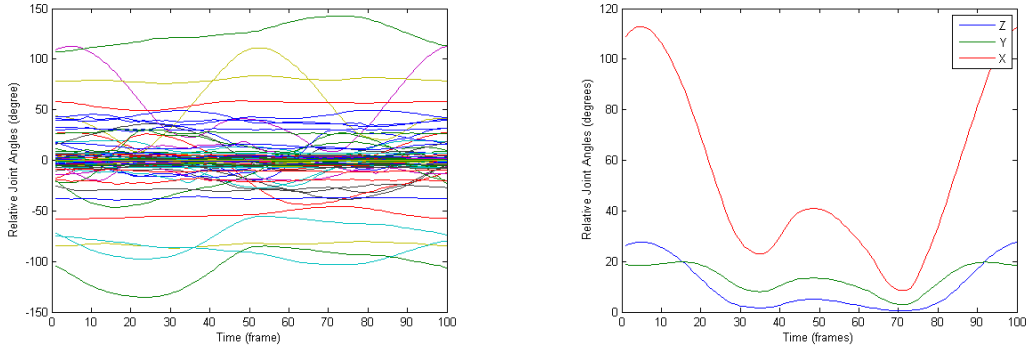
4.1 Parameterized Model

Finding a model configuration ϕ_t that fits the image data $\mathbf{I}_t$ using the human model described in the previous chapter and taking all 96 dimensions into account, would lead to an intractable search space. However in most of this search space no data points will ever reside, since they lead to physically impossible configurations. In order to reduce our search space we can use the fact that human motion is often repetitive, symmetrical and highly constrained. If we focus ourselves on running these facts certainly apply.

4.2 Learning the Motion Model

In order to reduce the search space we could use the fact that we obtained a plausible configuration in the previous frame ϕ_{t-1}. If we could propagate this configuration in a sensible way we could guide our search process, since we already have a clue what the current configuration ϕ_t might look like. If we look at the integrand in equation 3.1 we see that the new posterior distribution depends on the old distribution multiplied by the temporal prior $p(\phi_t|\phi_{t-1})$. If we could specify a function which returns the temporal prior, the job is done. Unfortunately due to the fact that human motion is quite complex, it is hard to specify this function analytically [19]. It might, however, be possible to "learn" this function from data obtained by motion capturing systems.

Data is taken from $n = 11$ trials using data found in Carnegie Mellon's motion capture library. Each trial is manually segmented so that it contains one motion cycle and is scaled to the same size of t_{max} frames. Now we have obtained $\mathbf{M} = \{\mathbf{M}_1 \cdots \mathbf{M}_n\}$ containing the motion data of all trials. Each matrix $\mathbf{M}_i$ contains a series of column vectors $\vec{\theta}_{i,j}$ $(j\epsilon[1, d_{max}])$ each containing the relative angles of a single joint (dimension) during the whole cycle and $\theta_{i,t,j}$ $(t\epsilon[1, t_{max}])$ is defined as the joint angle of the j'th joint at time t of the i'th trial. A mean $\mathbf{M}_{mean}$ is derived by taking the mean of each separate joint for each t of each trial:



(a) All 93 dimensions.

(b) Relative angles of the left knee.

Figure 4.1: Relative joint angles of the full body (a) or just the left knee (b).

$$\tilde{\theta}_{t,j} = \frac{(\sum_{i=1}^{n} \theta_{i,j,t})}{n} \tag{4.1}$$

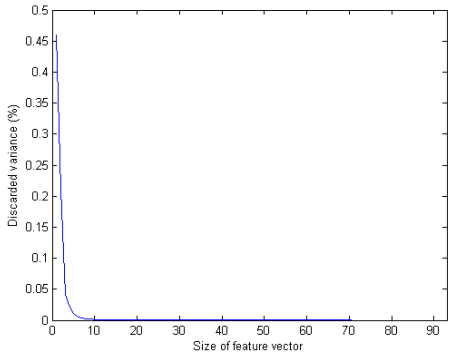
$$\mathbf{M}_{mean} = \begin{pmatrix} \tilde{\theta}_{1,1} & \dots & \tilde{\theta}_{1,d_{max}} \\ \vdots & \ddots & \vdots \\ \tilde{\theta}_{t_{max},1} & \dots & \tilde{\theta}_{t_{max},d_{max}} \end{pmatrix} \tag{4.2}$$

This mean, however, has the same dimensionality as our previous search space. In order to reduce this space we use the fact that we might expect some correlation between the data points. In order to find these similarities we apply Principal Component Analysis. First for each of the dimensions a mean value $\bar{\theta}_j$ is calculated by taking the average across each dimension. By subtracting the mean from each data point we obtain $\alpha_{t,j} = \tilde{\theta}_{t,j} - \bar{\theta}_j$ and $\mathbf{A}$, the matrix containing all values of α. Then we calculate the covariance matrix $\mathbf{A}_{cov}$ from $\mathbf{A}$. Once we obtained the covariance matrix we can calculate its eigenvectors u_j and eigenvalues λ_j. The eigenvector tell us in which direction our data variates for each dimension and the corresponding eigenvalue tells us how large this variation is. By joining the top-N eigenvectors with the highest eigenvalues we obtain a feature vector $\vec{F}$. Finally a transformed dataset is calculated by multiplying $\vec{F}^T$ with $\mathbf{A}^T$, $\mathbf{MT}^T = \vec{F}^T \times \mathbf{A}^T$. $\mathbf{MT}$ now only contains $b = N$ dimensions, but is still representing some part of the original dataset. As the feature vector becomes bigger, more of the original dataset is represented by our new, transformed, dataset, but leads to less dimensionality reduction. In this case the dataset containing the joint angles consists of 93 dimensions and 100 frames. After PCA has been applied, taking only the top-3 eigenvectors will suffice, since they represent already 95% of the original dataset, so in our case $b = 3$ (also see figure 4.2).

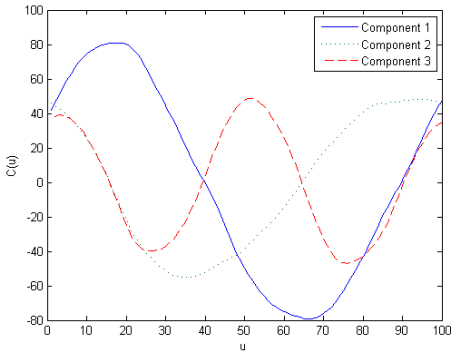
4.3 Propagating Parameters in Time

In order to adapt the model to changes during tracking we need to adapt the parameters in time.

A tracked walking cycle does not need to be of the same size as our learned cycle. Therefore the parameter μ_t is introduced, indicating the current position in the model's walking cycle. The



(a) Contribution of each eigenvector ordered by decreasing eigenvalue. The amount of variance lost is plotted against the size of the feature vector $\vec{F}$



(b) The obtained principal components. This graph shows the values of the components **MT** plotted against parameter μ.

Figure 4.2: Graphs showing the results of PCA application.

parameter T_t denotes the global translation of the model in the workspace in x,y and z-direction, $T_t = [x_t, y_t, z_t]$. The global rotation of the model is already included in the learned model.

We can now express our model configuration as $\{\vec{c}, \mu_t, T_t\}$, where $\vec{c}$ is a vector containing row μ_t of **MT**. A lookup function $f(\mu_t) : \mathbf{MT} \rightarrow c$ is defined as a function which returns row μ_t of the matrix **MT**. $f(\mu_t) = [\mathbf{MT}_{\mu_t,1}, \ldots, \mathbf{MT}_{\mu_t,b}]$.

We also introduce one additional function and one constant. The parameter $v(t)$ describes the velocity function which returns the speed of the subject at time t. Furthermore we introduce a constant δ which describes the amount by which parameter μ_t should be increased.

$$p(\mu_t|\mu_{t-1}) = G(\mu_t, (\mu_{t-1} + \delta), \sigma_\mu) \tag{4.3}$$

$$p(\vec{c}_t|\vec{c}_{t-1}) = G(f(\mu_t), \vec{c}_{t-1}, I_3\sigma_{\vec{c}}) \tag{4.4}$$

$$p(T_t|T_{t-1}, v_{t-1}) = G(T_t, T_{t-1} + v_{t-1}, \sigma_T) \tag{4.5}$$

where

$$G(x, z, \sigma) = \alpha e^{\frac{-(x-z)^2}{2\sigma^2}}$$

and I_3 is a 3×3 identity matrix.

The values of σ_μ and σ_T are empirically determined. The value of $\sigma_{c_j} = \epsilon\lambda_j$, where $j \in [1, b]$ and ϵ is a small normalizing constant.

4.4 Obtaining the Joint Angles

From $\{\vec{c}, \mu_t, T_t\}$ the original parameter configuration ϕ_t representing the various joint angles can be reconstructed.

$$\phi_t = [T_t, (\vec{F} * \vec{c}^T)^T + [\bar{\theta}_1, \ldots, \bar{\theta}_{d_{max}}]] \tag{4.6}$$

The transformed joint angles are concatenated with the translation T_t. The Gaussian distribution

over all parameters implies a Gaussian distribution over ϕ_t so the temporal prior $p(\phi_t|\phi_{t-1})$ of equation 3.1 is obtained.

Chapter 5

Likelihood

The likelihood or sensor-model is a function which returns the probability that a configuration matches the current frame. Before we can specify the function we first need to pre-process our image data, subtracting the background from the subject.

5.1 Background Subtraction

Before we can address the issue of matching a given 3D-configuration to a subject in the 2D-image plane we first need to determine where the subject resides within the image plane. We need to apply *figure-ground segmentation*, also referred to as *background subtraction*.

The term background subtraction is a bit old fashioned and refers back to the early days of Human Motion Analysis. The current image was pixel-wise subtracted from a static (empty) background image. This subtraction will yield greater distances between pixels where a new subject or non-stationary object resides. Classification between foreground and background was based on a fixed threshold value. This method has been used in many systems and yields quite fair results (see figure 5.1).

The biggest drawback of using background subtraction is the fact that it is a static method, it cannot adapt to any changes in the environment. It only works if the background remains static, that means there should be no camera movement and all objects in the background should also remain still. Furthermore it is unable to cope with illumination changes. To address these problems various other methods have been developed using a background-model which is adaptive [11, 16].

A drawback using adaptive methods is the fact that these methods use a background model which needs to be initialised. In the beginning such a model will yield a high number of false positives and as time goes on it adapts itself to the current situation and the performance increases. Using video-footage of only several seconds you want good performance from the first frame onward, because there is no time for initialising the model. Therefore basic background subtraction is applied using a non-adaptive model. We may expect this model to perform reasonably well, since we assume a static background and only a single subject in the scene. Furthermore dramatic illumination changes are not expected in a sequence which only lasts several seconds.

As a preprocessing step, each image is smoothed using a Gaussian convolution filter. This

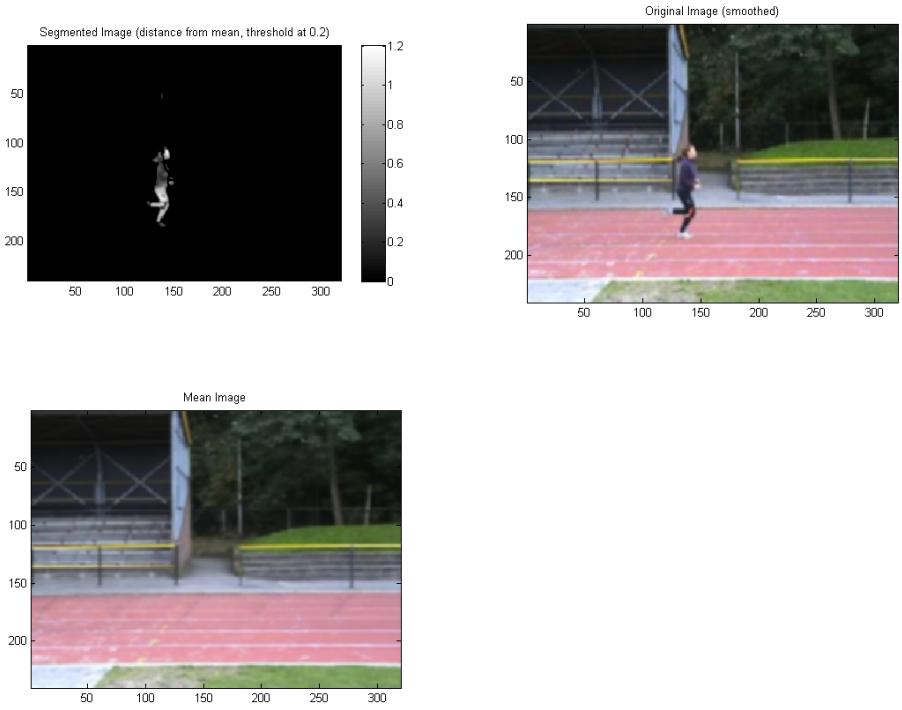
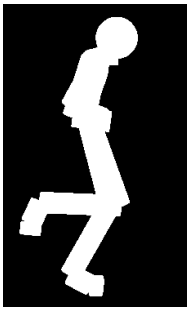


Figure 5.1: A result of background subtraction. Each image in the sequence was smoothed using a Gaussian convolution filter. The reference image was created by taking the mean over the full sequence. The top-left image is the result of subtracting each of the RGB-channels from the mean and calculating the Euclidean distance between the reference and current image. A brighter pixel indicates a larger distance.

eliminates a great deal of noise in the input data. The background model is constructed by taking the mean of every image in the sequence. For each pixel of the reference image, the difference in the color space between the reference image and the background is calculated and if the distance exceeds a threshold it is classified as a foreground pixel. A more sophisticated method [6] can be applied eliminating the classification of shadows as foreground pixels.

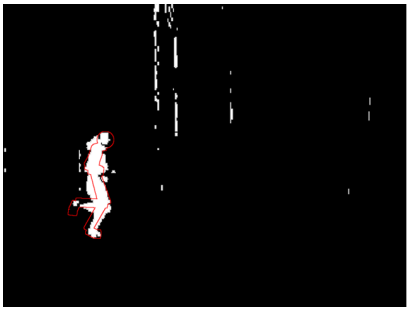
5.2 Defining the Likelihood

The likelihood or sensor model is defined as the probability of a specific sensor response given a hidden internal state of the system. In the system proposed it indicates the probability that the pose of a subject extracted from a frame of the input video sequence at time t matches the pose of our human model ϕ_t, $p(\mathbf{I}_t|\phi_t)$. To define this probability distribution we seek a function which yields a measure how well a subject's pose in image $\mathbf{I}_t$ and a configuration ϕ_t match. A perfect match should yield a result of 1, and the more the two drift apart the lower the result.



(a) Generated silhouette

(b) Reference image



(c) Subtracted silhouette with generated silhouette as an overlay

Figure 5.2: Silhouette comparison. Although manually generated, these images should provide a visual explanation of silhouette comparison. The silhouette of the model (a) is placed on the subtracted image-silhouette (c) and the amount of match is determined.

5.2.1 Comparison

At the moment our human model is defined as a set of 96-parameters ϕ indicating the global position of the model within the environment and the different angles of each joint. To compare a configuration ϕ with image data $\mathbf{I}$ various approaches can be taken. One can try to seek the edges of the subject with the image and match a model to these edges [19]. Drawback of using this method is that the system has to learn an appearance model using a large database of manually segmented images in order to see whether an edge may belong to a human or not. Once learned however one has obtained a very robust method to segment and compare images. A more simple approach can be taken by comparing silhouettes [7]. Once extracted we can take the outline of our human subject and turn it into a silhouette. If we overlay this silhouette with a silhouette generated from ϕ we can determine the amount of correspondence between the two. A possible problem using this model is left-right ambiguity, because depth information is eliminated from the model before it is compared with the image data. This is, once successfully initialised, not longer a problem because of the temporal prior. It is highly unlikely that it will propose a transition between two states nearly opposite from each other.

Up until now the human model was visualised by means of a stick figure. This, however, is unsuited if we try to generate a silhouette. Therefore we represent each limb, except for the torso and head, as a cylinder. A cylinder is the geometric figure closest to the actual appearance of our limbs. If we would use a cylinder for the torso, we would obtain a human which is as thick as he is wide, unless we want to model a human with a serious amount of overweight this is not a good approximation. Therefore the torso is modeled as a box. The head is modelled like a sphere. Assuming that the lengths and widths of each limb are specified in such a way that we obtain a model that matches our subject in terms of length and limb-ratios, we now have the means to specify the likelihood-function.

5.2.2 Likelihood-function

Using silhouette-matching we can define function $l(\mathbf{S}^\phi, \mathbf{S})$ - the probability $p(\mathbf{I}_t|\phi_t)$ as the amount of correspondence between the silhouette extracted from image $\mathbf{I}_t$, $\mathbf{S}_t$ and the silhouette generated from the model configuration ϕ_t, $\mathbf{S}_t^\phi$. We define the matrices $\mathbf{S}$ and $\mathbf{S}^\phi$ as a matrix of the same dimensions as $\mathbf{I}$ and with each element either 1 or 0, representing the subject or the background. Thus, $\mathbf{S}_{i,j}, \mathbf{S}_{i,j}^\phi \in [0, 1]$.

We can now define the match between both silhouettes as:

$$l(\mathbf{S}^\phi, \mathbf{S}) = 1 - \frac{\sum_{i=0}^{w} \sum_{j=0}^{h} \mathbf{S}_{i,j}^\phi \times (1 - \mathbf{S}_{i,j})}{\sum_{i=0}^{w} \sum_{j=0}^{h} \mathbf{S}_{i,j}^\phi} \tag{5.1}$$

We compare both silhouettes on a pixel, or element, basis and subtract - if we encounter a pixel which belongs to the generated silhouette $\mathbf{S}^\phi$ - the value of the corresponding pixel in the image-silhouette $\mathbf{S}$. The sum over the full image yields the amount of pixels within the generated silhouette that do not have a counterpart in the image-silhouette, this is then normalized by the total amount of pixels in the generated silhouette. Now we have the percentage of mismatch between the two. Subtracting this from 1 returns the percentage of match between the two silhouettes. This value is between 0 and 1, so we also obtain our probability $p(\mathbf{I}_t|\phi_t)$.

Chapter 6

Discussion

6.1 A full HMA-system

A comprehensive HMA-system consists of all the different building blocks described in the second chapter. In this thesis only the blocks 'tracking' and 'pose-estimation' are described in detail. The others are addressed briefly in this section.

6.1.1 Initialisation

In the previous chapters the tracking and pose estimation of a human subject throughout a video sequence was described. The temporal prior and likelihood combined provide means to model the state-change of the system from time t to time $t + 1$. This description, however, does not cover the initialisation of the process - what should be done at time $t = 0$, when there is no previous state to update? How can the initial pose of our human model ϕ_0 be retrieved?

The initialisation of the model, defining the limb ratios, the initial position and the scale of the model, is a whole different subject. It requires different techniques, since there is no temporal context. You have only a single frame and all information should be extracted from that frame. Barron and Kakadiaris [2] have designed a method which is able to, after the user supplied some initial points as input, estimate the pose of a subject in a single frame. The method is able to both recover the pose of the subject and estimate all limb lengths using anthropomorphic constraints. This method might be a good starting point for tracking to commence. However, this method only works if some limbs are situated in a plane which is parallel to the image plane of the camera. This constraint violates the assumption made that the subject may be captured under any angle. Other methods investigated assume that the system is initialised using either model parameters which are manually set [20] or that the subject assumes a couple of initial poses isolating different joint locations [10]. These methods are more robust, but cannot be used in an automated system.

A first thought on how initialisation would be implemented in the DigiCoach system is to combine all three methods. The system keeps track of a user model containing the athlete's history (see appendix A). This user model can be used to store kinematic information of the subject - including limb lengths, but also information like physical disabilities. This information can either be obtained by manual input or some form of visual input, e.g. a photograph of the

subject taken from the front or behind with no occlusion. The method of Barron and Kakadiaris can then be used to estimate the initial pose of the subject, with the difference that the need for a limb in a plane parallel to the image plane of camera is not needed anymore, because the limb ratios are already known.

6.1.2 Estimating the prediction

In the description of the probabilistic framework introduced in the third chapter, the implementation of the integrand or 'estimation' has not been addressed. If one would implement the system, one should seek means to estimate the multi-dimensional, non-Gaussian, non-linear, probability distribution which is described by the integrand.

An estimation could be made using a particle filter. A set of particles is generated from the original probability distribution. Then each sample is propagated forward in time using the temporal prior given its current value for ϕ_t. Each sample is weighed by the likelihood it assigns to the new evidence and as a final step the population is resampled to generate a new population of samples. The old population is discarded and from the old population N samples are drawn with a probability equal to their weight, so there is a high chance that samples are drawn with the highest likelihoods (the best fit). The new set of particles then provides a good estimation of the posterior distribution at time $t + 1$. A particle filter aimed at visual recognition is the CONDENSATION algorithm [8].

6.1.3 Recommendation

After the pose is estimated the DigiCoach system will provide recommendations on how the subject can improve his or her technique. As an indication of what it takes to improve an athlete, track and field coach Michael Snijders was consulted. For each of his pupils he keeps track of three mental pictures: The most recent performance, a global picture of the current performance level of the athlete and a goal picture, the best performance an athlete may reach given his or her physique. Recommendations are given based on the current level and aim to bring this level closer towards the goal. The current performance of the athlete provides feedback whether an instruction had the desired effect, e.g. in order to reach maximum speed an athlete is required to hold his torso upright. Many people however tend to lean forward while running, so the coach tells the athlete to focus on leaning backward. Most people are capable of running upright, so the goal picture will contain an athlete who is running perfectly upright. Currently he or she is leaning forward so the mental picture of the current level will include the athlete running forward. The current performance depends on whether the instruction yields a positive result. If so the athlete is told to try and keep his torso this way, otherwise another instruction is given. As the athlete performs better in this area the mental picture of the current level gets updated and the coach turns his attention to other flaws.

If we would implement the story above, we should keep track of two models. A model of a perfect athlete. The computer will be unable to differentiate between different individuals, but a general example will do, since we may expect that an athlete will never reach perfection. There is always room for improvement. The current level can be tracked by updating the second model after each trial. We start off with an initial estimation, e.g. the average of N-trials, and thereafter

each time the model is updated with the current performance. Perhaps it would be recommended that a user may give some feedback on the update process, since a person focusing on a single instruction (for example running upright) may get sloppy in other parts of the motion. Once the current level-model is updated, new instructions can be given by comparing both models and consult a database with instructions in order to correct differences between the two.

The above gives a general notion of a concept which could be used to solve the problem of how to recommend improvements. It may not be expected to be flawless as it is a research by itself to develop such a system.

6.2 Robustness

When designing AI-systems, robustness is always an issue. Is your system able to cope with unseen situations? How much does it depend on correct sensor inputs and is it able to recover from errors?

The methods described in this thesis certainly have robustness issues. The acquired data must be of high quality in order to perform good figure-ground segmentation. Noisy or low-contrast images will prove difficult to segment, decreasing the overall performance of the system.

Furthermore the system does not provide a general model of motion, but an example based one. So it can only be used to track a running person, not a tennis player or even a walking person. This makes it very limited, although other models of motion can be learned and deployed in exactly the same way as described in chapter 4. For now we have to stick to these methods as a general model of motion is computationally not feasible [19].

Finally, we still depend on lots of assumptions specified in the second chapter[1]. If one of these assumptions is violated, the system is likely to fail.

6.3 Precision

A final point of concern is the precision of the estimated poses. If we want to draw conclusions based on the found poses we must be sure that they correspond with reality. This largely depends on the result of the background subtraction as this determines the shape of the silhouette which the poses are matched against. If a lot of detail is preserved and the edges are crisp it is much easier to make a good estimation. But the precision also depends on the used motion model. If the model describes a motion pattern which differs too much from the motion pattern of our subject it is very likely that a pose estimation is made which differs from the data as the motion pattern constraints the search space. So poses corresponding to input data lying outside this search space will not be found.

The exact amount of precision required depends on the user of the system and the recommendations the system will give. If they depend on subleties, the estimations made must be very precise. If the recommendations depend on constraints which are more coarse, precision is less of an issue.

[1] See table 2.2.

Chapter 7

Conclusions

7.1 Conclusions

Creating a system which is able to track human motion is a difficult task. When I started off I thought that within the six months available, it should be possible to design and build a system which should at least be able to estimate the pose of a subject, implementing the first four blocks of figure 2.1.

Unfortunately the field of HMA is very broad. Many different methods are proposed and they all have their different strenghts and weaknesses. They are often targeted at a specific application or only demonstrate very specific functionality. I was unable to find a paper which described a comprehensive system. This made it difficult to select suitable methods for various parts of the system and I was forced during the process to lower the bar. I decided to concentrate myself on the core of the system: tracking and pose-estimation. Creating a framework on which a comprehensive system can be build.

7.1.1 Questions

The main question was: *How can human motion be captured and tracked by a computer?* Using a number of sub-questions I have tried to give an overview of what it takes to develop a system which is able to recover the 3D pose of a human subject using 2D video footage.

What difficulties arise when trying to track human motion?

Moeslund and Granum have defined a functional decomposition of a comprehensive Human Motion Analysis-system which contains five different building blocks (figure 2.1). Each of those blocks comes with their own set of challenges.

What methods have been developed in order to track human motion, which difficulties do they address and what are their strenghts and weaknesses?

In the second chapter an overview is given of various methods proposed in the scientific literature for each of the different building blocks of an HMA-system. This overview is based on a number of assumptions, which in turn are based on the requirements of the DigiCoach-system.

How the data is acquired largely depends on the available hardware. The only suitable method remaining is passive sensing, because of the assumption that the system can be used in an every day environment using only natural sources.

Initialisation concerns itself with initialising the human model and finding the correct starting pose. The human model is initialised either through estimation within an image of the input data, aided by the user, or through a series of images in which the subject takes poses which isolate different joints. The latter is more robust, but cannot be used unless one operates in a situation in which the user can be requested to take the required poses.

In order to track the human through the scene one first needs to classify each pixel as either belonging to the subject or the background. This can be done using background subtraction techniques which learn a non-adaptive background model. A pixel is classified as a background pixel if its values are near the ones of the background model. These methods are simple and do not need any initialisation time. However they may fail when confronted with illumination changes, noisy sensors and dynamic backgrounds. In order to counter these problems, methods using an adaptive background-model were developed. They are able, although some are better than others, to cope with the problems of a non-adaptive model. However, these systems often need a long initialisation time and can only be used in surveillance-like situations.

Once the subject is recognised its pose needs to be recovered. The only suited class of methods using only a single camera is the 'direct model use'-class, as all other classes badly cope with occlusions. Finding the correct pose of the model is done either through a gradient descent technique or through stochastic search methods. The latter has the ability to recover when it gets off track, in return for increased computational complexity.

Most research in the field of recognition focusses on the classfication of human movement. Most methods proposed are aimed towards the surveillance systems and do not suit the aim of this thesis. Therefore this subject is not explored.

Which solution is best suited for my purpose and how can it be modeled?

In this thesis the work has focussed on finding a solution for tracking and estimating the pose of the subject. A probabilistic framework is defined which is described in the third chapter. In the fourth and fifth chapter two different parts of the framework are highlighted, the 'temporal prior' and the 'likelihood'. An example based method is used which allows to estimate the pose at the current time, using a parameterized model of our subject, and predict the pose for the next frame reducing the search space.

Initialisation and recognition are only briefly discussed in the sixth chapter.

How can this solution be implemented?

For now bits of code have been implemented in Matlab, but any language can be used to implement the final system. A first proposal for the architecture of the system is done in appendix A.

7.1.2 The field of Human Motion Analysis

Although good progress has been made, many issues are still unaddressed. At the moment it is not possible to robustly initialise a model using only the footage shot during the activity. One

either needs to specify it by hand or have the subject taking various poses. This is a problem because if we want to draw conclusions based on the poses found we need to achieve a high degree of precision. It even remains to be seen whether the required amount of precision can be achieved using only the current methods and hardware. A final point of concern is the fact that the proposed method is still example based. It highly depends on the learned motion model and therefore also might influence the outcome of the pose estimation.

At the moment the field of Human Motion Analysis is unable to provide solutions for a robust and general applicable HMA-system. A lot of effort is still needed in order to create more robust methods. Next to that technological advances are needed to provide extra computational power allowing for more general approaches.

7.2 Future work

The framework described in this thesis provides a good basis in order to complete the DigiCoach system. Future research will be done on the initialisation of the model and the propagation of the model through time using a particle filtering. Another research will address the issues of providing recommendations to the user. Once the system is completed research may be directed towards increasing robustness and precision.

Appendix A

Requirements and Architecture

A.1 Requirements

This appendix provides a short summary of the requirements for the DigiCoach system, although it would be recommended to make a full requirements analysis of the system. This, however, is beyond the scope of this thesis.

A.1.1 System description

The purpose of the DigiCoach system is to analyse the running of a track-athlete and to give an advise on how his style can be improved in order to reach better performance. The system uses video-footage of the athlete as its main input and produces both a 3D-representation of the athlete's pose at each timeframe as well as a list of recommendations how the motion can be improved.

The system uses only a single camera and there is no need for a strictly controlled environment. Data can be acquired in an outdoor environment using various lighting conditions and a plain camera will suffice. There is no need for markers or other special equipment. The system assumes the camera is in a stationary position and there is only one subject within the range of the camera. See figure A.1. It is not necessary for the subject to move in a plane which is tangential to the camera as is shown in the figure.

The system should also be able to keep track of an individual model for each athlete so that previous sessions can be compared with the current footage and a mean can be produced, highlighting recurring errors and surpassing accidental mistakes.

A.1.2 System functionality

In order to fulfill its purpose, the system needs to be able to do the following:

Input The system should be able to read the video of the subject and uncompress this to a series of separate images.

Extraction It should be able to extract the subject from the various images, eliminating all background information.

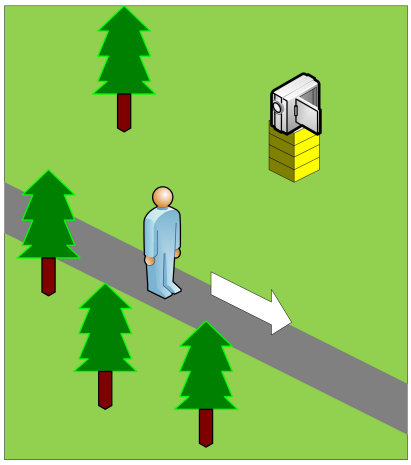


Figure A.1: Schematic representation of data acquisition.

Translation The system must be able to translate a series of 2D images into a sequence of 3D poses.

Recommendation The system can recommend improvements in order to reach better performance. This is achieved by comparing the athlete's personal history with a model of the 'perfect' athlete.

User Interface The system does not have to operate unsupervised. It should be possible for the user to correct false recognitions and the user is required to estimate the starting pose of the subject.

A.1.3 Intended users

The system is intended for users who have basic knowledge about track and field athletics and have basic knowledge about operating a computer system.

A.2 System architecture

The final system will consist of four main blocks each shown in figure A.2. The user controls the system through the *interface*. The interface presents the user with information regarding the processing of the data and results. Furthermore it allows the user to specify which data should be used, specify various parameters and assist the system when it fails to recognise the subject.

The *data reader* is able to translate the video file compressed with the AVI-codec into a series of images suited for processing by the *model generator*, which is able to turn each separate image into a 3D-representation of the subject for that specific timeframe. When put together these 3D-representations show the movement of the subject.

All 3D-representations together will serve as input for the *recommender* who adds the new information gained from the model generator to an already known model of the athlete. This model is then compared to a model of the perfect movement and differences are fed back to the user.

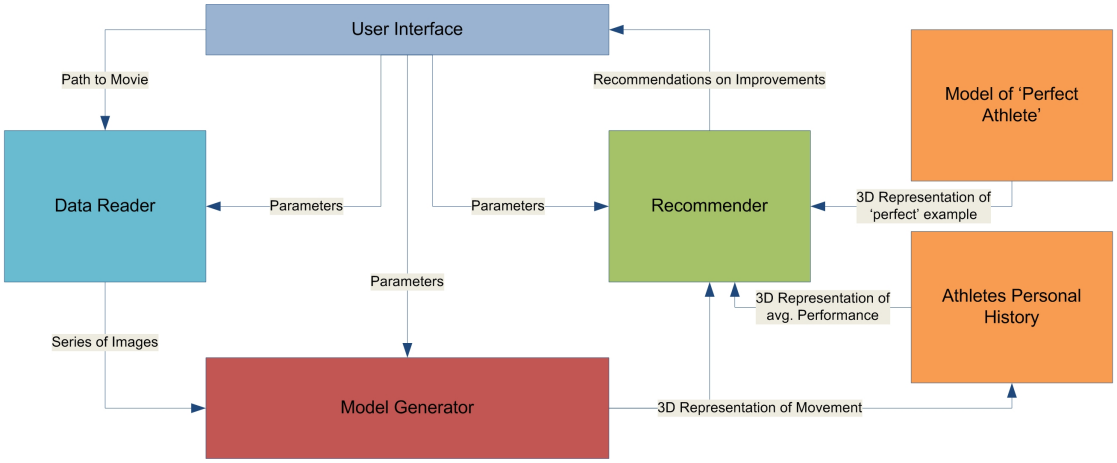


Figure A.2: System architecture.

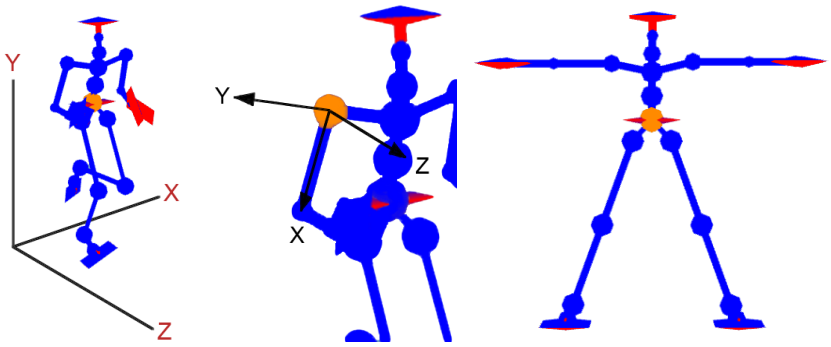
Appendix B

Human Model

B.1 Joints

The human model used in this thesis is derived from motion data downloaded from Carnegie Mellon's online motion capture library. It has 31 joints, although some of them do not actually rotate, these are not used when expressing a running motion, but may be needed when extending to other types of motion. They are indicated with a * in table B.1. Each joint is able to rotate in all three dimensions. This is not consistent with a real human being, but it proposes no problems since the learning data is obtained from real humans. Thus the model will be 96-dimensional, 31×3 joint angles, although some dimensions will remain 0, and three dimensions specifying the global translation of the model.

The joints are organised in a hierarchy with each joint having an offset in X,Y and Z-direction compared to its ancestor. Each joint has a local coordinate system which depends on the direction its ancestor is facing. For a graphical example, see figure B.1. A joint with all angles equal to 0 will be directed in the offset direction towards the next joint. The global coordinate system is defined as a right-handed system with the X-coordinate facing horizontal, the Y-coordinate facing vertical and the Z-coordinate facing outward.



(a) Global coordinates (b) Local coordinates (c) Human Model

Figure B.1: Graphic representation of the used model. Both (a) and (b) show the coordinate system, either global or local. The last figure shows the model with all joint angles set to 0.

Dim.	Name	Description
1-3	-	Translation
4-6	hip	Main hip joint
7-9	lhipjoint*	Left hip center
10-12	lfemur	Left hip outside
13-15	ltibia	Left knee
16-18	lfoot	Left ankle
19-21	ltoes	Left toes
22-24	rhipjoint*	Right hip center
25-27	rfemur	Right hip outside
28-30	rtibia	Right knee
31-33	rfoot	Right ankle
34-36	rtoes	Right toes
37-39	lowerback*	Lower back
40-42	upperback	Center back
43-45	thorax	Center shoulders
46-48	lowerneck*	Lower neck
49-51	upperneck	Upper neck
52-54	head	Center head
55-57	lclavicle*	Left shoulder cent.
58-60	lhumerus	Left shoulder out.
61-63	lradius	Left elbow
64-66	lwrist	Left wrist
67-69	lhand	Left hand
70-72	lfingers	Left fingers
73-75	lthumb	Left thumb
76-78	rclavicle*	Right shoulder cent.
79-81	rhumerus	Right shoulder out.
82-84	rradius	Right elbow
85-87	rwrist	Right wrist
88-90	rhand	Right hand
91-93	rfingers	Right fingers
94-96	rthumb	Right thumb

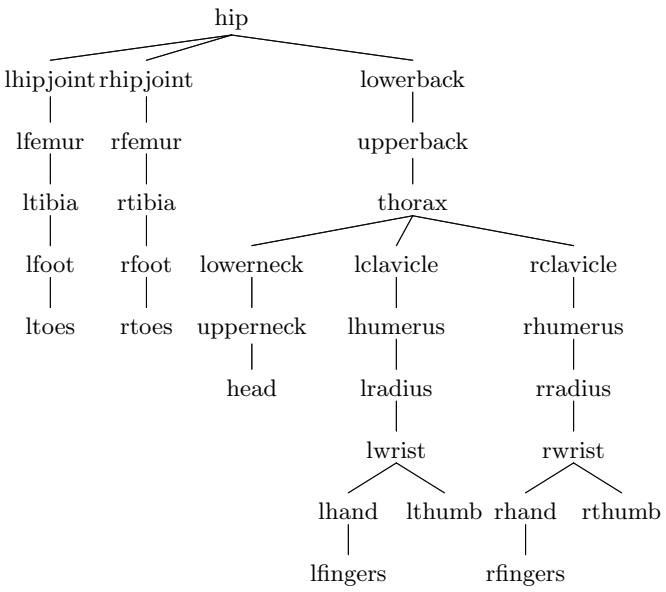


Table B.1: The various dimensions (joints) used in the model and their hierarchical structure. Joints which angles remain 0 are indicated by *.

B.2 BVH File Format

A BVH-file consists of two parts. The first part starting with the keyword `HIERARCHY` gives a definition of the model used. Each model consists of one `ROOT` joint and several other joints specified by `JOINT`. Each joint can contain one or several sub joints creating a tree-like hierarchy. This hierarchy is parsed using depth-first recursion. For each separate joint it is specified how it is located compared to its parent using the keyword `OFFSET` and then specifying the offset in X,Y and Z direction. Furthermore it is specified how many degrees of freedom, or `CHANNELS` each model has. The offset and rotation of the root-joint indicate the global offset and rotation.

The second part of the file starts with the keyword `MOTION` and contains all motion information. First is specified how many frames there are and how long each frame takes (in seconds), using `Frames` and `Frame Time`. This is followed by a large matrix of numbers, sized $frames \times dimensions$. Each number represents a rotation of a specific joint in a specific direction for a certain frame. This is expressed in degrees and ranges from 0 to 359, except the first three numbers of each row which represent the global translation [1].

Below is the corresponding BVH-structure of the model described in the previous section. The motion data itself is discarded, because this would not fit the paper.

```
HIERARCHY
ROOT hip
{
  OFFSET 0.000000 0.000000 0.000000
  CHANNELS 6 Xposition Yposition Zposition Zrotation Yrotation Xrotation
  JOINT lhipjoint
  {
    OFFSET 0.000000 0.000000 0.000000
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT lfemur
    {
      OFFSET 1.656740 -1.802820 0.624770
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT ltibia
      {
        OFFSET 2.597200 -7.135760 0.000000
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT lfoot
        {
          OFFSET 2.492360 -6.847700 0.000000
          CHANNELS 3 Zrotation Yrotation Xrotation
          JOINT ltoes
          {
            OFFSET 0.197040 -0.541360 2.145810
```

[1] For a more elaborate description see: `http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html`

```
          CHANNELS 3 Zrotation Yrotation Xrotation
          End Site
          {
            OFFSET 0.000000 -0.000000 1.112490
          }
        }
      }
    }
  }
}
JOINT rhipjoint
{
  OFFSET 0.000000 0.000000 0.000000
  CHANNELS 3 Zrotation Yrotation Xrotation
  JOINT rfemur
  {
    OFFSET -1.610700 -1.802820 0.624760
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT rtibia
    {
      OFFSET -2.595020 -7.129770 0.000000
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT rfoot
      {
        OFFSET -2.467800 -6.780240 0.000000
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT rtoes
        {
          OFFSET -0.230240 -0.632580 2.133680
          CHANNELS 3 Zrotation Yrotation Xrotation
          End Site
          {
            OFFSET -0.000000 -0.000000 1.115690
          }
        }
      }
    }
  }
}
JOINT lowerback
{
  OFFSET 0.000000 0.000000 0.000000
  CHANNELS 3 Zrotation Yrotation Xrotation
```

```
JOINT upperback
{
  OFFSET 0.019610 2.054500 -0.141120
  CHANNELS 3 Zrotation Yrotation Xrotation
  JOINT thorax
  {
    OFFSET 0.010210 2.064360 -0.059210
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT lowerneck
    {
      OFFSET 0.000000 0.000000 0.000000
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT upperneck
      {
        OFFSET 0.007130 1.567110 0.149680
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT head
        {
          OFFSET 0.034290 1.560410 -0.100060
          CHANNELS 3 Zrotation Yrotation Xrotation
          End Site
          {
            OFFSET 0.013050 1.625600 -0.052650
          }
        }
      }
    }
    JOINT lclavicle
    {
      OFFSET 0.000000 0.000000 0.000000
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT lhumerus
      {
        OFFSET 3.542050 0.904360 -0.173640
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT lradius
        {
          OFFSET 4.865130 -0.000000 -0.000000
          CHANNELS 3 Zrotation Yrotation Xrotation
          JOINT lwrist
          {
            OFFSET 3.355540 -0.000000 0.000000
            CHANNELS 3 Zrotation Yrotation Xrotation
```

```
        JOINT lhand
        {
          OFFSET 0.000000 0.000000 0.000000
          CHANNELS 3 Zrotation Yrotation Xrotation
          JOINT lfingers
          {
            OFFSET 0.661170 -0.000000 0.000000
            CHANNELS 3 Zrotation Yrotation Xrotation
            End Site
            {
              OFFSET 0.533060 -0.000000 0.000000
            }
          }
        }
        JOINT lthumb
        {
          OFFSET 0.000000 0.000000 0.000000
          CHANNELS 3 Zrotation Yrotation Xrotation
          End Site
          {
            OFFSET 0.541200 -0.000000 0.541200
          }
        }
      }
    }
  }
}
JOINT rclavicle
{
  OFFSET 0.000000 0.000000 0.000000
  CHANNELS 3 Zrotation Yrotation Xrotation
  JOINT rhumerus
  {
    OFFSET -3.498020 0.759940 -0.326160
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT rradius
    {
      OFFSET -5.026490 -0.000000 0.000000
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT rwrist
      {
        OFFSET -3.364310 -0.000000 0.000000
        CHANNELS 3 Zrotation Yrotation Xrotation
```

```
                JOINT rhand
                {
                  OFFSET 0.000000 0.000000 0.000000
                  CHANNELS 3 Zrotation Yrotation Xrotation
                  JOINT rfingers
                  {
                    OFFSET -0.730410 -0.000000 0.000000
                    CHANNELS 3 Zrotation Yrotation Xrotation
                    End Site
                    {
                      OFFSET -0.588870 -0.000000 0.000000
                    }
                  }
                }
                JOINT rthumb
                {
                  OFFSET 0.000000 0.000000 0.000000
                  CHANNELS 3 Zrotation Yrotation Xrotation
                  End Site
                  {
                    OFFSET -0.597860 -0.000000 0.597860
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

MOTION
Frames: 100
Frame Time: 0.033333
trans11 trans12 trans13 angle11 angle12 ... angle1n
trans21 trans22 trans23 angle21 angle22 ... angle2n
  .       .       .
  .       .       .
  .       .       .
transm1 transm2 trans m3 anglem1 anglem2 ... anglemn
```

Bibliography

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.

[2] Carlos Barron and Ioannis A. Kakadiaris. Estimating anthropometry and pose from a single image. *Computer Vision and Pattern Recognition*, 01:1669, 2000.

[3] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22(3):569–577, 2003.

[4] Tat-Jen Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 1999.

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

[6] Thanarat Horprasert, David Harwood, and Larry S. Davis. A robust background substraction and shadow detection. In *ACCV'2000*, Taipei, Taiwan, 2000.

[7] Nicholas R. Howe. Silhouette lookup for monocular 3d pose tracking. *Image Vision Comput.*, 25(3):331–341, 2007.

[8] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[9] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection, 2001.

[10] N. Krahnstoever and R. Sharma. Articulated models from video. *cvpr*, 01:894–901, 2004.

[11] A. McIvor. Background subtraction techniques. In *Image and Vision Computing*, Auckland, New Zealand, 2000.

[12] Thomas B. Moeslund. The analysis-by-synthesis approach in human motion capture: A review. In *Proceedings of the 8th Danish conference on pattern recognition and image analysis*, 1999.

[13] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

[14] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding: CVIU*, 104(2):90–126, 2006.

[15] Sharon Oviatt and Philip Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Commun. ACM*, 43(3):45–53, 2000.

[16] F. Porikli. Achieving real-time object detection and tracking under extreme conditions. *Journal of Real-time Image Processing*, 1(1):33–40, October 2006.

[17] B. Rosenhahn, U. Kersting, L. He, A. Smith, T. Brox, R. Klette, and H.P. Seidel. A silhouette based human motion tracking system. Technical Report CITR-TR-164, Centre for Image Technology and Robotics (CITR), 2005.

[18] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.

[19] Hedvig Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences*. PhD thesis, Stockholm University, 2001.

[20] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.

[21] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 1999.