# Full Body Motion Capture

## A Flexible Marker-Based Solution

Christian Schönauer, Thomas Pintaric, Hannes Kaufmann
schoenauer@ims.tuwien.ac.at, pintaric@ims.tuwien.ac.at, kaufmann@ims.tuwien.ac.at
Interactive Media Systems Group
Institute of Software Technology and Interactive Systems
Vienna University of Technology,
Vienna Austria

## Abstract

Full body motion capture (MoCap) plays an increasingly important role in many fields from entertainment to medicine. However, accurate motion capture systems have not been used in some application areas due to reasons like cost, inflexibility and complexity of operation. We developed a complete marker based optical MoCap system, which targets these issues. In this paper we describe how a flexible calibration method, robust skeleton tracking and interfaces to third party software have been developed. To demonstrate robustness, reliability and accuracy, our system has been successfully employed to provide input to a serious game targeting rehabilitation of patients with chronic pain of the lower back and neck, a group that has previously been neglected by serious games. A brief overview of the application and preliminary results of a medical evaluation with chronic pain patients is provided.

**CR Categories:** H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Input devices and strategies*; J.3 [Computer Applications]: Life and Medical Sciences—*Health, Medical information systems*; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—*Animations, Artificial, augmented, and virtual realities*

**Keywords:** Motion Capture, Full Body Interaction, Serious Game

## 1 Introduction

Marker based optical full body Motion Capture (MoCap) has become a de facto standard capture technology in professional applications in the last decade. Thereby, one or more actors wearing motion suits with retro-reflective markers attached to them, are tracked by a number of cameras in real time and millimeter accuracy [MotionAnalysis 2011; Robertson 2006]. In medical applications and evaluations these systems have been applied to determine movement parameters. Usually this is done for a specific body part with a subset of the markers required for full body MoCap e.g. in gait analysis. While unmatched in accuracy and update rate, marker based full body MoCap systems have not seen use in many other fields due to numerous reasons like cost, inflexibility and complexity of operation. We have developed our own marker based optical tracking system, which is targeting these issues.

In general a MoCap system has to solve two main problems in order to determine the posture of a user. First a model, matching the current user's physical properties, has to be created. Then, this model must be matched to the marker positions in each frame in order to determine the user's posture. Our system solves both problems with a minimum of user interaction, while providing posture information in real time.

Not much information is available on the intrinsic functioning of commercially available systems and how they create or adapt a skeleton model for a specific user and match it with marker positions during the tracking process. In this paper we aim to fill this gap. We describe the applied algorithms and methods that were used to achieve these goals in order to develop a highly accurate, commercial grade, robust optical motion capture system.

Serious games and especially their use in healthcare applications are an active and rapidly growing area of research [Holden 2005; Rego et al. 2010]. A key aspect of games in rehabilitation is 3D input, which can be used to control a game and provide feedback to the patient. We have developed a serious game specifically designed for rehabilitation of patients with chronic pain of the lower back and neck. This game contains a number of exercise programs integrated in a motivational environment. In this context an accurate MoCap system is required, because the capabilities and progress of the patients over weeks have to be determined for medical evaluation. Finally, a study has been conducted to get insights in patients' expectations and experiences with the system.

Our contributions are:

- Development of a new flexible, full body motion capture (MoCap) system and description of all required algorithms.
- Demonstration of its use within a serious game targeting rehabilitation of patients with chronic pain of the lower back and neck.
- Evaluation of a serious game targeting rehabilitation of patients with chronic pain.

## 2 Related Work

### 2.1 Full Body MoCap Systems

In this context, full body MoCap, is defined as capturing the user's body in a way that allows for the reconstruction of six degrees of freedom (DOF) for every bone in a simplified skeleton model, which is available in a complete kinematic chain.

Professional systems are usually marker based infrared optical solutions, such as those from Motion Analysis [MotionAnalysis 2011] or Vicon [Vicon 2011]. These setups can be scaled to cover huge areas with multiple users by using large numbers of cameras. All motions are computed in real time (with millimeter accuracy) and are available for further processing e.g. recording, analyzing, motion transfer to a virtual character and more.

Xsense [Xsense 2011] and other companies, offer solutions for MoCap using inertial, mechanical or magnetic tracking. When using inertial sensors or mechanical tracking, larger volumes are possible, while the tracking quality is bound by the used technology e.g. subject to drift of inertial sensors.

Furthermore, systems using other tracking technologies like markerless optical tracking or magnetic tracking exist. However, they either do not deliver tracking data in real time, are restricted to specifically prepared environments or do not achieve the same accuracy as marker based solutions and are therefore neglected in this discussion.

### 2.2 Skeleton Calibration

Current commercially available full body MoCap systems (e.g. Vicon) usually rely on a predefined skeleton model with a small number of markers, which have fixed positions on the tracked subjects. This reduces the flexibility of adaptation to different patients and use for more specific applications only needing input from a part of the body.

Several methods have been proposed for the automatic calculation of a skeleton model from motion capture data. Often, they rely on the calculation of the center of rotation between bones defined in their position and rotation by three or more optical markers [Cameron et al. 2005; Ringer et al. 2002; Silaghi et al. 1998], or magnetic tracking [O'Brien et al. 2000]. In these approaches using two adjacent bones with six DOF the joint positions are calculated using a least squares fit. However, these methods require many markers or (inaccurate) magnetic tracking (with one sensor attached to each bone). The method, described in [Kirk et al. 2005], suggests using a computationally more expensive algorithm, which requires less markers. Details on how we have improved the latter to be more robust and reliable can be found in section 4.

### 2.3 Skeleton Tracking

Skeleton tracking or fitting [Silaghi et al. 1998] describes the process of adapting the position and the joint rotations of the skeleton model in a way, that it fits the recorded data. Little information is published on the intrinsic functionality of skeleton tracking in commercially available full body MoCap systems. Some algorithms, like the global skeleton fitting described in [Silaghi et al. 1998], rely on the identity of the markers to be known. However, this is a constraint, which for reasons of stability and robustness, we do not want to depend on.

In an alternative approach, as used in [Ringer et al. 2002], a rigid clique (or body) is placed on each bone-segment. These cliques, consisting of three or more markers with fixed distances, can be used to derive position and orientation of the bone. As described in [Hornung et al. 2005] these can also be trained dynamically. However, this approach depends on the markers to be fixated on rigid structures and four markers per bone, limiting applicability.

Furthermore, kinematic models for different joints and body parts have been developed and analyzed (e.g. for the knee [Andersen et al. 2010]), but little is published on generic models for the tracking of a human body.

## 3 System Overview

Our setup comprises of an infrared optical tracking system that we developed ourselves, and a number of software components performing different tasks in the MoCap and data processing pipeline.

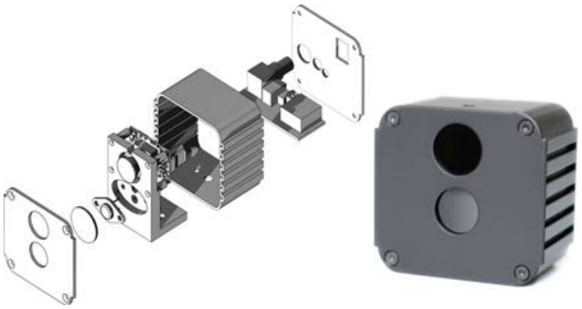

**Figure 1.** Explosion chart and picture of the camera used in our system.

### 3.1 Hardware Setup

The tracker used for our motion capture system is an iotracker [Pintaric et al. 2007], which is a passive marker based infrared optical motion tracking system. It was primarily built for collaborative virtual reality and augmented reality applications, but is also well suited for motion capture. Commodity hardware is used to minimize the cost. All calculations are performed on PC workstations, while no additional electronics is required. Iotracker cameras are shutter synchronized and stream digital video to the tracking workstation at an update rate of 60 Hz. They have a very compact form factor of about 7x7x4 centimeters. To make (only) the passive markers visible in the camera images the cameras are equipped with infrared strobe lights and optical band pass filters. Figure 1 shows the camera with all components. The application "iotracker server" calculates the 3D positions from camera images. Measurements indicate low latency (20-40ms), minimal jitter (RMS less than 0.05mm), submillimeter location resolution and an absolute accuracy of ±0.5cm. For a more detailed description of the marker tracking performance please refer to [Pintaric et al. 2007].

### 3.2 Software Workflow

In the first step a skeleton model has to be generated and adapted for each new user. Using a separate tool, the "skeleton calibration assistant" (Figure 2), from the marker positions alone, a skeleton model is calibrated fully automatically. To execute the algorithm the user has to perform some initial calibration movements often

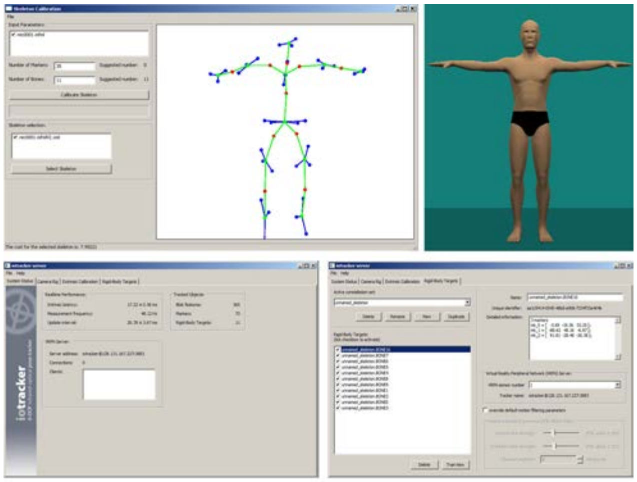

**Figure 2.** Skeleton calibration assistant (upper row), iotracker server (lower row)

referred to as the “Gym motion” [Herda et al. 2000; Silaghi et al. 1998]. For a successful calibration the user moves the joints of interest to their full extent for about 15-20 seconds. For practical use with patients, the skeleton calibration assistant shows movement sequences, which have proven to work well. However, no restrictions are imposed on the Gym motion except that it has to start in the T-Pose [Geroch 2004]. Out of all the skeleton models generated from the sequences, the assistant automatically selects the best for further processing; however, the therapist is able to override this selection manually. In practice a maximum of 2-3 repetitions is sufficient to acquire a high quality skeleton model. The evaluation of the skeleton is based on the joint optimization cost of a skeleton. Finally, the choice of a skeleton is acknowledged and is automatically labeled with bone names by comparison with a predefined template. This is important in order to map the joint angles to the avatar in the game.

The labeled skeleton is then handed to the tracking software and the calibration assistant is being closed. The MoCap module integrated within the iotracker server (Figure 2) is responsible for fitting the skeleton to the tracked 3D positions of the markers in each frame. First it loads the skeleton and optimizes the internal representation. Then it starts the MoCap process producing angles for all joints of the skeleton model plus the position and orientation of the skeleton in 3D space. The iotracker server implements a number of interfaces to external applications, which can retrieve the postural information from the MoCap module as detailed in section 5.5.

## 4 Skeleton Calibration

In order for the system to easily and quickly adapt to different users, we have implemented the skeleton calibration assistant (Figure 2), which automatically generates an approximate skeleton model. There is no need to take measurements by hand or place markers on precisely predefined positions, which are both time consuming matters. Furthermore, the skeleton calibration algorithm is flexible enough to cope with arbitrary articulated structures. This is especially interesting with a view towards universal accessibility. A skeleton can easily be calibrated for people with a significantly different body model. For wheel chair users, for example, the skeleton can be calibrated for upper-body only. For amputees the suit can be adjusted with Velcro strips and the skeleton calibrated for the required limbs. Furthermore, skeleton models for generic motion capture input can easily be generated. The applied algorithms follow the general approach described in [Kirk et al. 2005]. A detailled description of the basic implementation can be found in [Schönauer 2007]. However, several improvements were made, especially to make skeleton calibration more robust when using imperfect tracking data. Furthermore, mechanisms allow for more flexible marker placement and increased speed of the calibration calculations. First, data is recorded by the tracker during calibration sequences and processed in the following main steps:

***Step 1:* *Marker labeling (assigning the markers ids).***

Passive markers in an infrared optical tracking system cannot be uniquely identified per se. The skeleton calibration, however, is depending on the markers being unambiguously labeled during the whole sequence. Therefore, in a first step double exponential smoothing [LaViola 2003] is used to predict the label of a marker in the next frame(s) instead of the linear prediction used by [Kirk et al. 2005]. During longer occlusions this method is not always sufficient. In a second step we are therefore using a method similar to [Kirk et al. 2005] to match labeled sequences of markers with the support of spectral clustering.

***Step 2:* *Marker clustering (group markers on a bone).***

The assumption for the clustering is, that the distance between markers situated on one bone does change only very little. Instead of spectral clustering as suggested by [Kirk et al. 2005] , we are using Kmeans++ clustering [Arthur et al. 2007], which for many of our data sets performed better. Kmeans++ takes into account the different spatial distribution of points in clusters of different sizes (e.g. in our application markers on the chest are further apart than markers on a forearm).

***Step 3:* *Joint position calculation.***

Assuming that the joints are approximately rotational joints, the markers of two bones connected to a joint should ideally stay in the same distance to the joint during all movements. As suggested in [Silaghi et al. 1998] and [Kirk et al. 2005] this is formulated as a least squares fit problem and optimized.

***Step 4:* *Computation of skeleton parameters*.**

Once the joint positions are calculated for every frame, the offsets between markers and joints can be calculated and averaged after removing outliers using a variation of RANSAC. These offsets can then be used in skeleton tracking to identify markers and correctly position bones and joints in the marker point cloud. Skeleton calibration is an offline procedure during which skeleton models are calculated sequentially from one or multiple calibration sequences. Due to the offline nature of this processing step, sophisticated algorithms such as non-linear optimization or spectral clustering can be used. However, the MoCap process itself is performed in real time.

To improve robustness of the labeling, in addition to the basic algorithm described in [Kirk et al. 2005], we are using the results of the second step (clustering). We have defined and minimized an intra-cluster distance variation value in order to reduce negative influence of tracking errors (e.g. "ghost markers"). Also marker identification after occlusions is improved. In the third step, before joint positions are calculated for all frames the skeleton structure has to be determined. This is done by pseudo randomly selecting a limited amount of about 20 frames and computing the joint positions/cost between all possible combinations of bones. The combinations with the lowest cost are then interpreted as the actual joints. Since nonlinear optimization, such as nonlinear conjugate gradient method depend very much on good initialization values, we are using the results of these samples in a local optimization to speed up the calculation on all frames as a further improvement to [Kirk et al. 2005]. We have also tried to discard frames with similar limb transformations and maximize diversity among the sample set. However, this has brought no significant enhancement in our tests. Other smaller improvements have been made to improve performance of the calibration algorithm, which are beyond the scope of this publication.

## 5 SKELETON TRACKING

### 5.1 Overview

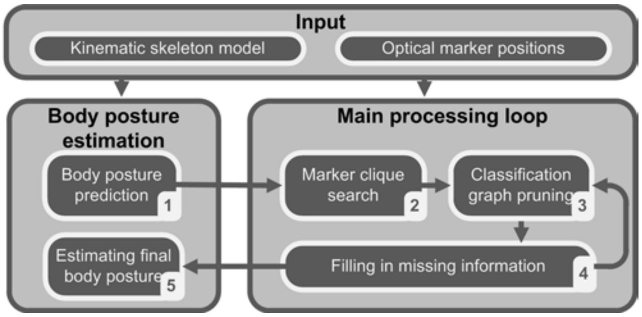

**Figure 3.** Overview of the skeleton tracking pipeline

Using the skeleton calibration assistant described in the previous section, we are able to reconstruct a full body kinematic constraint model of the user's skeleton. This kinematic model is stored as a hierarchical chain of rigid bones, connected by rotational joints. Every bone is associated with one or more optical markers, the position of which relative to the bone remains static, building a rigid clique. The skeleton model is then fitted to the marker positions as generated by iotracker to obtain a skeleton pose for every frame.

Our skeleton model fitting algorithm is composed of the following steps (see Figure 3). Algorithmic and implementation details are given in the next subsections.

***Step 1:*** ***Prediction of the expected body posture.*** Upon first run or re-initialization no previous body posture estimates exist. Therefore, in these cases we are using a generic one (a standing pose or the so-called T-pose) as initialization. Otherwise, we use predictive forward kinematics to compute the expected body posture from the last-known one as described in section 5.2.

***Step 2:*** ***Rigid marker clique search, generation of a classification graph.*** We try to identify every limb, with a rigid clique of two or more optical markers, in the point cloud, by using inter marker distances. Usually, this will result in multiple hypotheses how a clique could be fitted to marker positions in the current frame. From the differences between the actual relative distances and the ones stored in the skeleton model, a model alignment error is calculated. Every hypothesis, whose model alignment error lies below a certain threshold, is stored in the vertex of a classification graph, instead of immediately deciding on a unique solution. Edges represent geometric transformations between different cliques and are computed upon insertion of new vertices. Details on this step are described in sections 5.3 and 5.4.

***Step 3:*** ***Classification graph pruning.***

From the hypotheses in the classification graph, we first compute the absolute pose of the associated limb in as many degrees of freedom as possible. We then test this pose against the expected posture (determined in Step 1). If the difference between both poses exceeds a certain (experimentally determined) threshold, the vertex and all of its edges are removed from the graph.

We then test the remaining edges of the classification graph against the kinematic constraints of the skeletal model by pair wise examination of adjacent limbs. When we find an edge that does not satisfy the kinematic constraints (within a certain threshold), we remove both connected vertices from the classification graph.

***Step 4:*** ***Filling in missing information.*** In the event that occlusions prevent parts of the inner kinematic chain (e.g. upper or lower arm) from being unambiguously reconstructed, but the end-effector pose (hands, feet) is known, we attempt to solve for the middle joint pose by an iterative inverse kinematics approach [Buss et al. 2005].

From here, we repeat Step 3 and 4 until all ambiguities are resolved.

***Step 5:*** ***Posture estimation.*** In most cases, all ambiguities are resolved in Step 3 and only one vertex remains for each clique (containing the hypotheses, how it best matches the data). In the rare event that there are multiple hypotheses remaining, we assign scores to the hypotheses based on the model alignment error. We then select the highest-scoring hypothesis from the classification graph and remove the remaining hypotheses. We label all markers according to the clique and associated bone they are matched with. If necessary, occluded optical markers are substituted with "virtual markers".

Based on the uniquely labeled optical marker set, we iteratively compute the full skeleton parameterization (using forward kinematics) with the goal of minimizing the model-alignment error.

## 5.2 Prediction of the expected body posture

The full parameterization of the skeletal model is given by the set of all joint angles plus the 6-DOF transformation (position and orientation) of the root limb. The torso, as the center of the body, is usually chosen as the root of the kinematic tree. During the calibration procedure, the Skeleton Calibration Assistant will automatically identify the torso (or any other bone of choice) and label it as the root bone, which will be named bone ***0*** in the following.

For a kinematic chain of ***n*** bones, let $\boldsymbol{\theta_i}$ be the joint-angle between bone ***i*** and bone ***i-1***. Given $\boldsymbol{\theta_1}$ ***...*** $\boldsymbol{\theta_n}$, the frame of bone ***n*** relative to bone ***0*** is:

$$^{0}T_n = \prod_{i=1}^{n} {}^{i-1}T_i(\theta_i)$$

Where $\boldsymbol{^{i-1}T_i(\theta_i)}$ is the transformation matrix from the frame of bone ***i*** to bone ***i–1***. In absence of a valid measurement for $\boldsymbol{\theta_k}$ (k = 1..n), it is not possible to compute $\boldsymbol{^{i-1}T_i(\theta_i)}$ for any i≥k.

We solve this problem by employing a double-exponential smoothing-based prediction filter (DESP) [LaViola 2003] for every joint angle $\boldsymbol{\theta_i}$, which provides us with estimates of the joint angle in absence of actual measurements.

## 5.3 Rigid Clique Fitting Algorithm

Our system attempts to uniquely identify markers of each rigid clique within a larger, unstructured set of observed markers by comparing the Euclidean distances between all pairs of observed markers with the known distances between markers of each clique. To account for the presence of noise and different types of process errors, two distances $\mathbf{d_i}$ and $\mathbf{d_j}$ are considered equal if

$$| \mathbf{d_i} - \mathbf{d_j} | < \varepsilon.$$

We formulate this combinatorial optimization problem in a way that it represents an instance of the maximum clique problem. Maximum clique is a well-studied problem for which fast algorithms exist [Östergard 2002]. Nevertheless, it is NP-hard, which is why we apply a polynomial-time complexity-reduction heuristic that is based on a form of geometric hashing. At startup, we compute the Euclidean marker distance matrix for each clique and store its entries in a one-dimensional look-up table (Figure 4). These tables are used at runtime to rule out unlikely clique-to-marker matches before the maximum clique problem is solved.

Measurement uncertainties are assumed to follow a Gaussian distribution whose standard deviation is chosen to coincide with the tracker's absolute accuracy. We refer to the entries stored in this look-up table as *correlation scores*, because they represent the probability with which any given marker-pair (or more precisely, their separating distance) is part of the same target.

At runtime, we calculate the Euclidean distance between every pair of observed 3D markers for a given frame. For every entry in the Euclidean distance matrix (EDM), we look up its corresponding *correlation scores* and store it in a separate Correlation score matrix (CSM). From the CSM, we compute a vector containing the accumulated correlation scores for every observed marker through row- or column-wise summation.

Depending on the number **n** of model-markers, we also expect every member of the solution-set to have an accumulated correlation score of at least $\mathbf{T_n = t * (n\text{-}1)}$, where **t € [0.5,1]**. The easiest

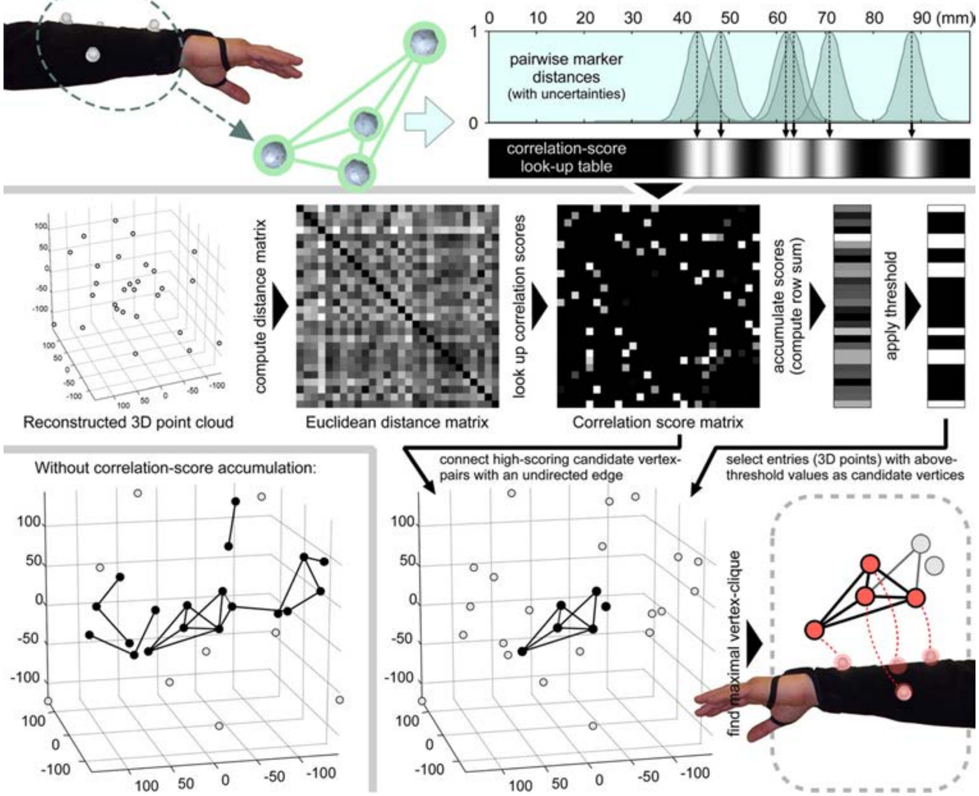
0 10 20 30 40 50 60 70 80 90 (mm)
1
0
pairwise marker distances
(with uncertainties)
correlation-score look-up table
compute distance matrix
look up correlation scores
accumulate scores (compute row sum)
apply threshold
Reconstructed 3D point cloud
Euclidean distance matrix
Correlation score matrix
Without correlation-score accumulation:
connect high-scoring candidate vertex-pairs with an undirected edge
select entries (3D points) with above-threshold values as candidate vertices
find maximal vertex-clique
100
50
0
-50
-100
100
0
-100
100 50 0 -50 -100

**Figure 4.** Quadratic-time complexity reduction (top).
Formulation of the model-fitting problem as maximum-clique search (bottom).

way to enforce this constraint is to apply threshold $\mathbf{T_n}$ to the accumulation-vector. Afterwards, every marker whose thresholded entry is zero can be eliminated as a candidate.

Our heuristic can be calculated very efficiently on multiple processor cores. According to our observations, this elimination procedure will reliably reduce the number of candidate markers by up to 90% in a real-life scenario.

Once the maximum clique problem is solved the clique can be aligned with the data by minimizing the least squares distance between observed markers and the clique's idealized markers as stored in the skeleton model. This distance value is also used as error function (model alignment error) in the other steps.

## 5.4 Algorithmic Extensions for Kinematic Chains of Rigid Cliques

In an unstructured point-cloud, unique marker identities can only be established by inspecting known pair-wise distances between markers. Often, multiple similar distances between markers can be observed in the entire point-cloud, which leads to labeling ambiguities. Due to the relatively large number of markers in a motion-capture configuration, it is not enough to just use the known pair-wise distances between markers on the same bone in order to establish unique marker labels. Therefore, we extended the complexity reduction algorithm described in the previous section from a single rigid clique of markers to kinematic constraint models by pre-computing additional multi-dimensional marker-to-marker distance lookup tables from the calibrated skeleton. This computation is performed as a separate offline step. Figure 5 shows a simple two-bone skeleton fragment connected by a spherical joint, where each bone has exactly one rigidly attached marker ("marker1" and "marker2"). In an unstructured point-cloud, only markers that are found to lie at a distance between $\mathbf{d_{min}}$ and $\mathbf{d_{max}}$ from Marker1 should be considered likely to be labeled Marker2. The value-pair [ $\mathbf{d_{min}}$, $\mathbf{d_{max}}$ ] is computed a-priori for every pair-wise combination of markers on adjacent bones.

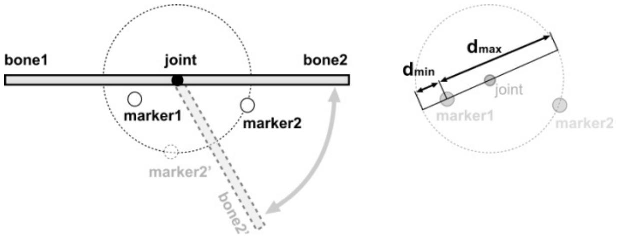

**Figure 5.** Distance limits for markers on adjacent bones.

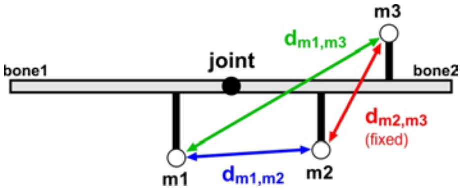

**Figure 6.** Pair-wise distances of multiple markers on adjacent bones.

Distance lookup-tables can be extended to span multiple markers on adjacent bones: <u>Example:</u> For any given distance $\mathbf{d_{m1,m2}}$ (where $\mathbf{d_{min}} <= \mathbf{d_{m1,m2}} <= \mathbf{d_{max}}$) between marker **m1** (on bone1) and marker **m2** (on bone2), distance $\mathbf{d_{m1,m3}}$ between marker **m1** and marker **m3** (on bone2) can be stored in a lookup table. Figure 6 illustrates this example. For reasons of efficiency and memory

**Table 1.** Mini games, their medical rationale and the MoCap data used

| Game name | Mini games overview | | |
|---|---|---|---|
| | ***Game description*** | ***Clinical goal*** | ***Input data*** |
| Temple of Magupta | The player runs through an ancient temple, collects artifacts and avoids obstacles. | Physical reconditioning, increase walking speed and walking time | Movement rate |
| Face of Chronos | The patient climbs a mountain by extending the arms upwards until the next hold is reached and collects artifacts placed on holds. | Increase reaching ability, velocity and smoothness of the motion, relaxation of the trapezius muscle after reaching | Position of the hand, movement characteristics of the arm (path, velocity), muscle activity of the left and the right trapezius muscles |
| Three Wind Gods | The player imitates a series of head movements executed by fictive characters. | Increase cervical ROM, velocity and smoothness of cervical motion | Measures of cervical ROM and current rotation (Flexion/extension, Right/left bending, Left/right rotation), movement velocities, movement acceleration |

usage, distance values are discretized with step sizes of multiple millimeters. Using the distance lookup tables described above, we iteratively inspect the distance relationships between markers on adjacent bones. This allows for significant improvements in marker labeling accuracy, with minimal computational overhead.

### 5.5 External Interfaces

In order to make use of the generated body posture, interfaces have been implemented to external tools. This way, data from our system can be used for character animation (*Autodesk MotionBuilder*), medical evaluation (*C-Motion Visual3d*) and a game engine (*Unity3d*). For all three tools we have implemented plugins that can be used to utilize the data. In *Unity3d* and *Visual3d* the plugins are communicating with the tracker over a network interface utilizing a customized XML protocol, while *MotionBuilder* is connected to the iotracker using VRPN [RUSSELL et al. 2001]. To transform and extract data (e.g. information about velocities, specific positions etc.) from the skeleton pose data produced by the iotracker server, we are using an additional software layer. For this layer OpenTracker [Reitmayr et al. 2001] is used, which is a multi-modal data flow framework. It can serve as a center piece between the plugins and the iotracker, by implementing the required communication protocols. Figure 7 shows screenshots of the first two applications receiving the tracking data from our system in real time.

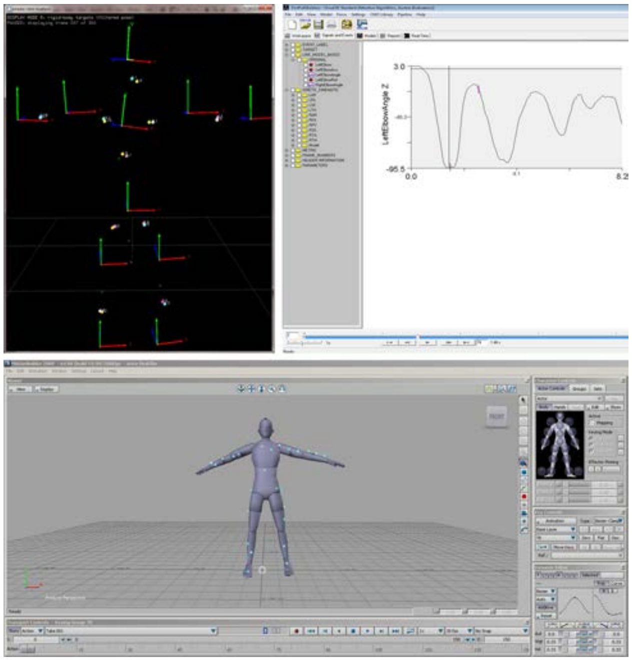

**Figure 7.** Visualization of the skeleton model's local coordinate systems (upper left), joint angle curve after the data was streamed into Visual3d (upper right), Motion Builder (bottom) showing animated avatar with marker positions.

## 6 A Serious game for chronic pain rehabilitation

### 6.1 The Game

Finally, we want to introduce the prototype of a serious game that has been implemented within the project PLAYMANCER to assist rehabilitation of patients with chronic pain of the lower back and the neck. An overview of the work done within the project can be seen in [Euronews 2011]. Also a video showing our system and game in action is provided there. Chronic pain patients are in a vicious circle of maladaptive pain-related cognitions (i.e. fear of movement), coping strategies (i.e. avoiding physical activities) and physical disuse. Therefore, therapeutic activities performed in pain rehabilitation practice, focus on changing these cognitions and increasing physical functioning. An important way to do this is by letting patients experience and execute physical exercises. Two important aspects that exercises commonly used in chronic pain rehabilitation focus on:

- Mobility and coordination by activities that focus on increased range of motion, increased velocity and smoothness of the motion.
- Improving a patient’s physical conditioning by endurance exercises like walking.

Our full body MoCap system is employed to track these exercises and works as main source of input data for the serious game. To incorporate these exercises into the game three separate mini games have been developed. An overview of the mini games, their clinical goals and input data used is listed in Table 1. They are embedded within an adventure setting, linking the games with a common story line. In that story the player arrives with his ship at a deserted island and discovers the remains of an ancient civilization. Pictures of test users playing the mini games can be seen in Figure 8.

The ship is the base of the patient. While on the ship, the game can be configured individually to each patient's needs and abilities by calibrating goals and baselines for the mini games. During the mini games the player can collect items, which can be seen in an inventory. When the player has collected enough items, he is rewarded by unlocking the next part of the story, which should provide additional motivation.

During game play the three mini games provide game feedback (scores, collected items) as well as visual, auditory and textual feedback on the patient's performance and results. Furthermore, after every session, therapist and patient view the patient’s progress, by having a look at the patient's profile. In this profile

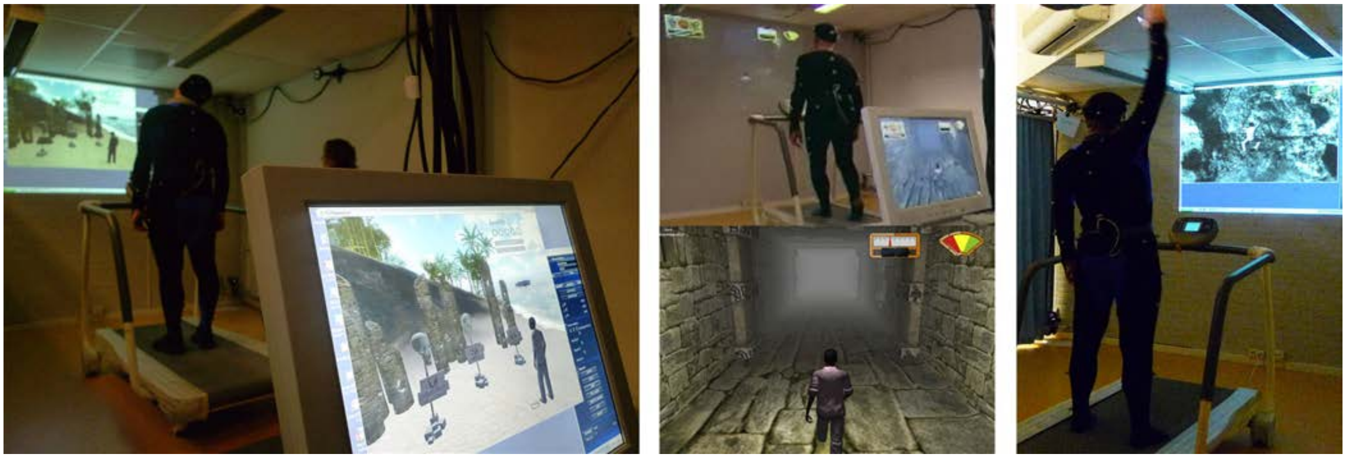

**Figure 8.** Our MoCap system and two of the mini games during the preliminary testing. "Three Wind Gods" (left), "Temple of Magupta" (center), "Face of Chronos" (right).

objective data, recorded during gaming, is presented (e.g. reaching ability, cervical range of movement.

### 6.2 Results

The "iotracker server" application runs on a quad-core CPU (Intel Core2Quad Q9300 with 2,5 GHz with around 75% CPU load). The skeleton tracking is implemented in a highly parallelized way using OpenMP and utilizes all cores. Especially step 2, which is the most computationally intensive, is well suited for parallelization. During our evaluations we were running the iotracker server and the game (developed in Unity3D) along with other input modalities (automatic speech recognition and electromyography) on the same PC. Therefore, we used a Workstation with two Intel Xeon DP X5355 processors (a total of 8 cores). CPU load was at around 85% during active gameplay.

The latency of the MoCap system consists of Firewire data transfer latency of 18ms (in case of 60 Hz tracking), since all camera images are transferred to the PC for processing. Additionally, overall latency is determined by the time needed for skeleton tracking, which is typically between 5 and 10ms for one skeleton on the above CPUs.

An evaluation of the serious game with eight patients has been conducted. Each patient was participating in the game interventions in a period of four weeks with one or two game sessions per week. During the evaluation the usability of the skeleton calibration procedure was rated good. All patients were capable to perform the requested skeleton calibration exercise and the calibration assistant produced and selected a functional skeleton model for each individual in almost all cases. Furthermore, the MoCap system was working stable and produced robust tracking data throughout the evaluation. Due to the lack of ground truth no absolute measurements of the system's accuracy have been made. Measuring human motion data with non-intrusive means (e.g. a goniometer) is usually highly inaccurate and depends on the tester and articulation used. Acquiring kinematic data of a human body by intrusive means on the other hand requires an enormous effort (e.g. radiostereometry [Tranberg et al. 2011] or bone pins [Andersen et al. 2010]) for which we lack the means. However, in the future we plan to compare our system's accuracy with that of another commercially available MoCap system. Nevertheless, visual inspection and subjective evaluation of the motion data showed smooth movements with little jitter. Evaluation of the joint angles showed also little jitter and high relative accuracy (smooth curves of angles and graduation in the sub-degree domain). Plots of unfiltered tracking data produced by our system can be seen in Figure 9. We have measured the rotation of the head relative to the torso on the up-axis. Even during fast motions like shaking of the head our system produced a smooth curve of measurements and even the inferred rotation-velocity introduced little jitter. When looking at slower motions our system produces values with sub-angular accuracy, which exhibit high plausibility. In addition, movement parameters calculated from positional information show little jitter in position and velocity. Finally, measurements regarding range of motion of assessed exercises show good repeatability.

The clinical changes induced by the gaming interventions are promising. The pain intensity measured on a visual analogue

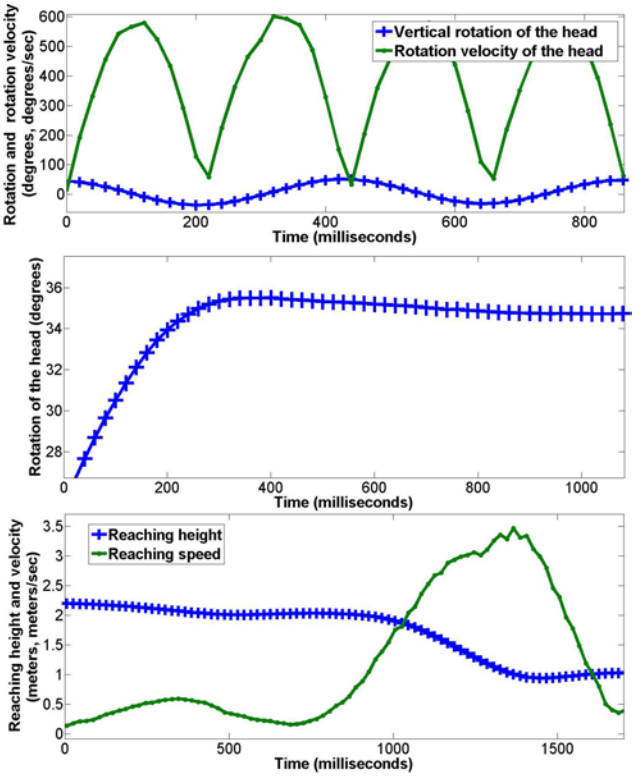

**Figure 9.** Rotation angle and velocity of the head relative to the torso around the up-axis during fast movement, i.e. head shaking (top), rotation of the head in the sub millimeter domain (middle), reaching height and velocity of a hand (bottom)

score decreased after gaming by an average of 11 millimeters. Also, physical fitness increased for many patients, especially for subjects with low exercise capacity. Furthermore, an increase of cervical range of motion of 7-10% for all three degrees of freedom has been determined. In addition a small increase in the patients reaching height has been observed. Also, usability was rated good. Finally, the patients were positive about the Playmancer game and enjoyed playing.

## 7 CONCLUSION AND FUTURE WORK

We have implemented a system for full body motion capture. Its high flexibility, accuracy and a moderate price makes it suitable for a large number of applications. We have introduced the applied algorithms and software components of the system. In addition, a serious game for the rehabilitation of patients suffering from chronic pain has been developed and evaluated in a clinical setting. Tests have shown promising results for our system and the game.

Patients suffering from chronic pain are a group that has previously been neglected by serious games. Therefore, the study had an experimental character, but the results could be very well used to define a test protocol for a controlled clinical trial, which might be conducted in the future. The evaluation has shown that patients suffering from chronic pain are a very heterogeneous group. Therefore, in future studies more mini-games and more specific inclusion criteria have to be employed. Furthermore, it is assumed that an increase in intensity, frequency and duration would have a positive effect on the outcome.

## 8 Acknowledgements

This work was funded by the European Union DG information Society Networked Media Systems unit within the PLAYMANCER project (FP7-ICT-215839). We want to thank Stephanie Kosterink, Michiel Minten, Miriam Vollenbroek, and Leendert Schaake from RRD for their cooperation and contribution to the results. Furthermore, we want to extend our thanks to Cris Voldum, Jens Juul Jacobsen and Jeppe H. Nielsen from Serious Games Interactive for the implementation of the game.

## 9 REFERENCES

ANDERSEN, M.S. et al. 2010. Do kinematic models reduce the effects of soft tissue artefacts in skin marker-based motion analysis? An in vivo study of knee kinematics. *Journal of Biomechanics 43*, 268-273.

ARTHUR, D. AND VASSILVITSKII, S. 2007. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, PA, USA, 1027-1035.

BUSS, S.R. AND KIM, J.S. 2005. Selectively damped least squares for inverse kinematics. *Journal of Graphics, GPU, & Game Tools 10*, 37-49.

CAMERON, J. AND LASENBY, J. 2005. A real-time sequential algorithm for human joint localization. In *Proceedings of the ACM SIGGRAPH 2005 Posters,* 2005, 107.

EURONEWS 2011. Playing for health http://www.euronews.net/2011/01/12/playing-for-health/

GEROCH, M.S. 2004. Motion capture for the rest of us. *Journal of Comp. Sciences in Colleges 19*, 157-164.

HERDA, L., FUA, P., PLÄNKERS, R., BOULIC, R. AND THALMANN, D. 2000. Skeleton-based motion capture for robust reconstruction of human motion. *Computer Animation 2000 (CA'00)*, 77.

HOLDEN, M.K. 2005. Virtual environments for motor rehabilitation: review. *Cyberpsychology & behavior 8*, 187-211.

HORNUNG, A., SAR-DESSAI, S. AND KOBBELT, L. 2005. Self-calibrating optical motion tracking for articulated bodies. In *Virtual Reality, 2005. Proceedings. IEEE VR 2005.*, 75-82.

KIRK, A.G., O'BRIEN, J.F. AND FORSYTH, D.A. 2005. Skeletal Parameter Estimation from Optical Motion Capture Data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, 2005*.

LAVIOLA, J.J. 2003. Double exponential smoothing: an alternative to Kalman filter-based predictive tracking. In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, New York, USA, 199-206.

MOTIONANALYSIS 2011. Motion Analysis: Passive Optical Motion Capture http://www.motionanalysis.com/.

O'BRIEN, J.F., BODENHEIMER JR, R.E., BROSTOW, G.J. AND HODGINS, J.K. 2000. Automatic joint parameter estimation from magnetic motion capture data. *In Proceedings of Graphics Interface*, 53-60.

ÖSTERGARD, P.R.J. 2002. A fast algorithm for the maximum clique problem. *Journal of Discrete Applied Mathematics 120*, 197–207.

PINTARIC, T. AND KAUFMANN, H. 2007. Affordable Infrared-Optical Pose-Tracking for Virtual and Augmented Reality. In *Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop*.

REGO, P., MOREIRA, P.M. AND REIS, L.P. 2010. Serious games for rehabilitation: A survey and a classification towards a taxonomy. *Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on*, 1-6.

REITMAYR, G. AND SCHMALSTIEG, D. 2001. An open software architecture for virtual reality interaction. In *Proceedings of the ACM symposium on Virtual reality software and technology*, 47-54.

RINGER, M. AND LASENBY, J. 2002. A Procedure for Automatically Estimating Model Parameters in Optical Motion Capture. *Image and Vis. Comp. 22*.

ROBERTSON, B. 2006. Big moves. *Computer Graphics World 29*.

RUSSELL, M., HUDSON, T.C., SEEGER, A., WEBER, H., JULIANO, J. AND HELSER, A.T. 2001. VRPN: a device-independent, network-transparent VR peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, Baniff, Alberta, Canada2001 ACM, 55-61.

SCHÖNAUER, C. 2007. Skeletal Structure Generation for Optical Motion Capture. Vienna University of Technology, Vienna.

SILAGHI, M.-C., PLÄNKERS, R., BOULIC, R., FUA, P. AND THALMANN, D. 1998. Local and Global Skeleton Fitting Techniques for Optical Motion Capture. *IFIP CapTech 98, Geneva 1537*, 26-40.

TRANBERG, R., SAARI, T., ZÜGNER, R. AND KÄRRHOLM, J. 2011. Simultaneous measurements of knee motion using an optical tracking system and radiostereometric analysis (RSA). *Acta Orthopaedica 82*, 171-176.

VICON 2011. Vicon motion capture system http://www.vicon.com.

XSENSE 2011. Wireless Inertial Motion Capture http://www.xsens.com/.