Self-Calibrating Optical Motion Tracking for Articulated Bodies

Alexander Hornung*
Computer Graphics Group,
RWTH Aachen University

Sandip Sar-Dessai†
Computer Graphics Group,
RWTH Aachen University

Leif Kobbelt‡
Computer Graphics Group,
RWTH Aachen University

ABSTRACT

Building intuitive user-interfaces for Virtual Reality applications is a difficult task, as one of the main purposes is to provide a "natural", yet efficient input device to interact with the virtual environment. One particularly interesting approach is to track and retarget the complete motion of a subject. Established techniques for full body motion capture like optical motion tracking exist. However, due to their computational complexity and their reliance on pre-specified models, they fail to meet the demanding requirements of Virtual Reality environments such as real-time response, immersion, and ad hoc configurability.

Our goal is to support the use of motion capture as a general input device for Virtual Reality applications. In this paper we present a self-calibrating framework for optical motion capture, enabling the reconstruction and tracking of arbitrary articulated objects in real-time. Our method automatically estimates all relevant model parameters on-the-fly without any information on the initial tracking setup or the marker distribution, and computes the geometry and topology of multiple tracked skeletons. Moreover, we show how the model can make the motion capture phase robust against marker occlusions by exploiting the redundancy in the skeleton model and by reconstructing missing inner limbs and joints of the subject from partial information. Meeting the above requirements our system is well applicable to a wide range of Virtual Reality based applications, where unconstrained tracking and flexible retargeting of motion data is desirable.

CR Categories: I.3.1 [Computer Graphics]: Graphics Utilities—Virtual device interfaces; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

Keywords: Virtual Reality Interfaces, Robust Optical Motion Tracking, Automatic Self-Calibration, Skeleton Reconstruction, Retargeting, Animation

1 INTRODUCTION

Capturing the motion of a subject or other articulated bodies plays an important role in Virtual Reality applications and computer animation as well as in motion analysis for medicine or sport science. Tracking the position and orientation of the subject's limbs allows the realistic reproduction and transfer of this motion to virtual characters with the same skeleton topology, enabling an intuitive interaction with the virtual environment. By this, position and motion tracking can be considered the very basic requirement for immersive interfaces. Further applications requiring motion analysis can be found in areas like medical research, sport science, or rehabilitation. However, tracking complex, articulated bodies is still a difficult problem, since it generally requires an involved calibration phase and an adequate pre-specified model, which strongly reduces the available degrees of freedom. While there exist quite efficient and practical solutions for tracking simple objects such as head-position or single limbs, the above mentioned restrictions make general motion tracking still an offline process, which compromises its usability as a flexible and robust input device for Virtual Reality.

The two major approaches to track the motion of a subject are based on contour finding [4] and marker-based methods, where the trajectory of markers attached to the subject's limbs are tracked magnetically [13] or optically [8]. In this paper we consider the use of common infrared based optical tracking systems for the use in virtual environments. These systems generally suffer from two fundamental problems. First, retroreflective optical markers are visually indistinguishable. Hence we need appropriate methods to *identify* markers based on other criteria in order to associate detected markers with their respective limbs. The second fundamental problem is *occlusion*. To reconstruct the three dimensional position of a marker, it has to be visible from at least two cameras. This cannot be ensured for an actor moving freely. Hence we need methods to compensate for missing markers in order to reconstruct the position and orientation of the actor's limbs, even if a significant number of markers is occluded.

Current tracking systems for recording the motion of the complete body [20] focus on setups in which only a small number of markers, in general one or two, is attached to every limb. These markers are then tracked using sophisticated motion prediction and a pre-specified skeleton model. While these methods are sufficient for typical tracking applications like motion generation for computer animation or movies, they do not automatically extract the complete underlying skeleton geometry and topology for arbitrary articulated bodies. In particular, the missing degrees of freedom due to the limited number of markers can only be resolved based on the pre-specified skeleton model, and additional manual postprocessing of the tracked data is often inevitable. This clearly restricts the range of possible applications for ad hoc tracking of arbitrary subjects, which cannot be described by a standard model. It also renders this method unusable for complex tracking tasks in real-time Virtual Reality environments.

An alternative approach is to assemble several markers into a rigid clique (often called "body") and to attach these cliques to limbs of the tracked subject (Fig. 3). By tracking a rigid clique of markers, one can identify temporarily occluded markers based on characteristic fixed inter-marker distances. Furthermore, it is possible to reconstruct the complete orientation of the tracked limb from such a clique. These cliques of markers are often used for optical head- or object tracking in Virtual Reality applications [1].

In general, both methods require a considerable amount of time for the manual calibration of the tracking system. Once such a system is calibrated for a specific tracking setup, it allows reliable and robust marker recognition and tracking. However, the range of possible applications is obviously restricted by the above mentioned issues. In contrast, our work focuses on methods to make optical motion tracking a completely automated real-time pipeline without the need for auxiliary information about the tracking setup or the

*e-mail: hornung@cs.rwth-aachen.de
†e-mail: sandip@saw.rwth-aachen.de
‡e-mail: kobbelt@cs.rwth-aachen.de

marker distribution, which makes it available as a general input device for Virtual Reality.

Our contribution lies in a self-calibrating, real-time system initialization, which uniquely identifies rigid cliques of markers from tracked data. From these we automatically compute the underlying skeleton geometry and topology of potentially several distinct subjects. In particular, we do not constrain the degrees of freedom of the underlying model in any way and thus are able to track arbitrary articulated bodies. Moreover, we reconstruct the position and orientation of limbs completely in contrast to other methods, which often have unset degrees of freedom concerning the orientation of limbs, or which require constraining the skeleton topology beforehand. During the motion recording we take advantage of this information to compensate for occluded markers, which helps us to reconstruct the position and orientation of limbs and joints in an accurate and robust way for real-time environments.

2 Related Work

Our work builds on a number of previous solutions to some of the occurring subproblems in optical motion capture. We integrate and extend some of these techniques to satisfy the above mentioned requirements for interaction devices in Virtual Reality.

Several partial solutions increasing automation and robustness in optical motion tracking have been proposed. To overcome the problem of marker identification, methods based on active light-emitting markers or pattern recognition like [9] were developed. However, infrared systems based on passive, retroreflective markers have advantages in terms of sensitivity to external influences like ambient light and are a commonly used method throughout the VR community.

Ringer et al. [14] present an automatic method to identify indistuingishable markers based on cliques. However, they need an explicitly occlusion-free training sequence which is processed offline to determine marker cliques and model parameters like the skeleton structure. Kurihara et al. [12] present a system for realtime motion capture and marker labeling, but rely on carefully chosen, asymmetrical marker distributions and a predefined model. Our methods do not impose constraints on the initial training sequence. Due to its online calibration our system provides permanent information about the calibration quality, so that potential errors can be detected and corrected early.

Our work on marker tracking and the dynamic identification of rigid marker cliques by formulating them as instances of a generic correspondence estimation problem is based on the work of Scott et al. [15]. They present an elegant algorithm to associate the features in two images for applications in computer vision. Transferring this method to our domain of optical motion tracking enables us to solve several tracking-related problems in a flexible and unified manner.

O'Brien et al. [13] show how to recover the structure and geometry of an unknown skeleton model. They describe a least squares fit of input motion data of individual limbs to a rotary joint model. Silaghi et al. [16] compute joints by estimating the rotation center of markers and their associated limbs. We use the technique presented in [13] since it results in higher accuracy and robustness concerning noise.

Approaches to make motion data recording more robust include predicting future marker positions using a Kalman filter [6, 21], search space reductions based on other prediction quality measures [19], or resolving occlusions based on the skeletal model of the tracked person as described in [8]. Our method does not try to identify or reconstruct markers based on predictions of future states but focuses on their robust recognition based on generated marker-signatures. This ensures a reliable identification even after occlusions during several frames, where prediction models possibly fail due to unconstrained movements of the tracked subject. We improve the actual tracking quality in the case of missing markers by applying methods of inverse kinematics to the computed skeleton as presented in the work of Tolani et al. [17]. They show how to reconstruct missing inner limbs of a skeleton up to one degree of freedom based on adjacent limbs in real-time. We extend their solution to determine the remaining degree of freedom if at least one additional marker on the lost limb is known.

In a recent work Zordan et al. [22] use a force-based forward dynamic model to map optical motion tracking data to a body model. Their technique does not estimate the skeleton of the tracked subject and is also running offline. However they explicitly mention benefits of a real-time system for motion capture, which is achieved by our method.

Commercial systems like Vicon [20] provide software tools for all phases of the tracking pipeline. However, such systems focus on setups with single markers attached to limbs, resulting in the above mentioned restrictions. Other systems like the one of A.R.T. [1] provide only low-level tracking and marker recognition without methods for automatic calibration, skeleton estimation or robust tracking of articulated bodies.

Besides the extended applicability of flexible, ad hoc motion tracking in Virtual Reality environments, we are also investigating its applications to fields associated with medical research, like physiologically correct retargeting [11], extraction of motion characteristics [3], or motion analysis [18] and VR-based patient training [2].

3 Overview

Consider the following setup to record the motion of an arbitrary articulated body like a human actor. We equip the subject with a set $\tilde{M}$ of spherical markers $\tilde{m}_1, \ldots, \tilde{m}_k$. The system has no information about the tracking setup, the number or distribution of markers, and the geometry or topology of the tracked subject other than the total number of tracked limbs. In principle, we want a subject equipped with markers to simply walk into the field of view of the tracking cameras, and an automatic initialization phase takes care of computing all model parameters in real-time.

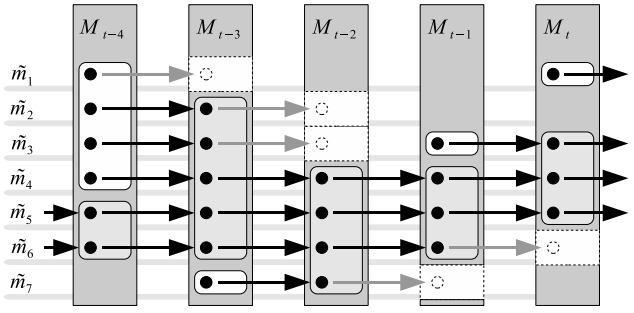

Figure 1: In this figure a subject equipped with seven markers $\tilde{m}_i$ is tracked for a few frames. For every frame F_t, we have a set M_t of reconstructed markers. Markers in white boxes correspond to newly detected markers, light gray boxes stand for continuously tracked markers, and dashed boxes for lost markers. One of the major problems during tracking is visualized in the first line corresponding to the physical marker $\tilde{m}_1$. This marker is lost for the tracking system in frame F_{t-3}. Until frame F_t, additional markers get lost, while other, previously untracked markers are found. Finally in frame F_t, the original marker $\tilde{m}_1$ is available again, but since it is visually indistinguishable from previously detected markers, it gets a completely new id from the tracking system. The main difficulty lies in detecting these correspondences between new and previously tracked markers and to associate each of them with a fixed global ID $\tilde{m}_i$.

Our A.R.T. optical motion tracking system [1] reconstructs the 3D position $\mathbf{p}_i^t$ of a marker m_i^t at time t when it is seen by at least two cameras in the corresponding frame F_t. It also provides low level marker tracking by establishing correspondences between the markers of two successive frames F_{t-1} and F_t. However, the system can only track a marker's trajectory as long as it is not occluded. Since markers cannot be distinguished the system does not know whether a new appearing marker is really a new one or was lost previously. So if a marker is temporarily occluded, it is permanently lost for the system, and every new appearing marker gets a new global system id.

Hence, while tracking the subject, we know a number $k(t) < k$ of markers M_t for every time frame F_t, their respective 3D positions P_t and a sequence $m_i^{t-n} \rightarrow \ldots \rightarrow m_j^{t-1} \rightarrow m_l^t$ of corresponding markers in previous frames. To compute meaningful marker-to-limb associations, we have to find a global mapping for every tracked marker m_l^t to a globally unique identity (the actual physical marker) $\tilde{m}_i$. Figure 1 depicts this situation in a compact form.

To resolve these ambiguities, one attaches not only one but several markers to every limb of the tracked subject. Markers located on the same limb form rigid *cliques* with characteristic invariant inter-marker distances, while distances to markers on other limbs change over time. We can think of attaching a string between each pair of markers. As we traverse the frames, we record the length variation of each string. If a string is stretched too much it rips (Fig. 2). In the end only strings between rigid cliques remain. These constant distances to other markers within the same clique form a unique *signature* Sig_i for every marker $\tilde{m}_i$ and make it possible to identify an unknown marker m_j^t by computing its distance to all other markers found in the same frame F_t. If there are enough correspondences between these distances and some signature Sig_i, m_j^t can be identified as $\tilde{m}_i$. For example, the lengths of the emphasized edges in Figure 2 form the signature for the rightmost marker. If we do not find a correspondence for marker m_j^t, we have to assume that it is a new marker which has not been tracked previously.

Accordingly the basic outline of our self-calibrating algorithm works as follows. First it is initialized with all visible markers in the first frame by creating global marker ID's $\tilde{m}_i$ and corresponding signatures Sig_i of pairwise marker distances. Then, for every frame F_t the following steps are performed:

1. *Continuous Tracking:* Continuously track markers $m_i^{t-1} \rightarrow m_j^t$ between the previous and current frame.
2. *Marker Recognition:* Recognize temporarily occluded markers by their previously generated signature.
3. *Marker Registration:* For all unrecognized markers m_i^t we have to assume that they are new and create a new global identity $\tilde{m}_j$ for them.
4. *Marker Cleanup:* In case of frequent marker occlusions, step 2 can fail to recognize a marker, which was already tracked in previous frames. This step corrects these errors and deletes other types of unreliable markers.
5. *Signature Generation:* Record the pairwise distance variation between all markers in a global distance matrix $\tilde{\mathbf{D}}$ and extract signatures based on constant inter-marker distances for every marker.

We formulate the central steps of this algorithm as instances of a general correspondence estimation problem, which allows us to use a general, unified framework for such matching problems without the need for specialized solutions for every single step. Our approach based on [15] is described in section 4.

The above algorithm iteratively creates and refines marker signatures in a self-calibrating manner until it detects a target number of rigid cliques defined by the number of limbs in the scene, or until it is interrupted by the user. At this point we have created stable signatures of fixed inter-marker distances for every marker within a rigid clique, which allows a robust recognition of temporarily occluded markers. In contrast to systems like [1], this initialization can be done completely automatic in real-time with a permanent feedback on the calibration quality to the user. The details of this algorithm are explained in more detail in section 5.

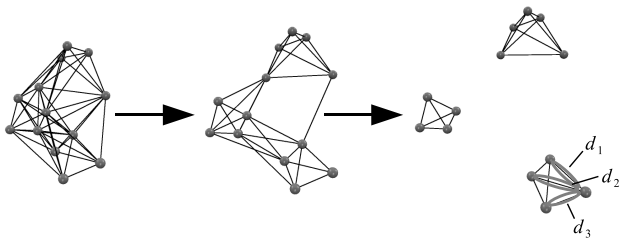

Figure 2: "Ripping strings". Initially all markers are connected to each other. By moving the respective cliques around, varying edges between different cliques are identified, until only the final rigid cliques remain. The invariant inter-marker distances within a clique form a unique signature for every marker, as depicted for the rightmost marker with the signature $Sig_i = \{d_1, d_2, d_3\}$.

The positions and orientations of the limbs can now easily be computed by embedding a local coordinate frame into every rigid clique of markers. These coordinate frames are redundantly defined if a clique consists of more than three markers making tracking even more robust against marker occlusions. To compute the underlying skeleton geometry and topology, we use the method of [13]. However, our algorithm also detects the presence of multiple skeletons, such that several independent subjects can be tracked at once without the need for separate system calibration. These methods are described in section 6.

Once all model parameters are estimated, we can use the gathered information to make the actual motion recording phase more robust to marker occlusions or to even reconstruct lost limbs and joints, as described in section 7.

Finally, we present some results and discuss potential applications of this work to Virtual Reality environments and motion analysis.

4 Correspondence Estimation

Our signature-based algorithm to identify markers is related to graph- or correspondence-matching problems and in particular motivated by the work of Scott et al. [15]. They present an elegant approach to find a partial mapping between two sets S_a and S_b of objects minimizing the overall squared sum of some inter-object measurements based on the singular value decomposition of a proximity matrix. Scott et al. use this method to find an assignment between corresponding feature points in two images. However, using their method in a more general sense by formulating the subproblems of our algorithm as a matching problem between two partially corresponding sets of objects, we can solve the signature comparisons and modifications in an unified manner.

The original approach of Scott et al. for correspondence estimation between two sets of 2D image feature points P_{t-1} and P_t for two images F_{t-1} and F_t starts by creating a proximity matrix $\mathbf{G} \in R^{m \times n}, m = |P_{t-1}|, n = |P_t|$, in which the entries g_{ij} represent some kind of *similarity measure* for the relation between two 2D image features $\mathbf{p}_i^{t-1}$ and $\mathbf{p}_j^t$. A positive, high value g_{ij} represents a strong relation, whereas a positive value close to zero indicates a weak relation. In [15] this is a function of the Euclidean distance $g_{ij} = e^{-\delta_{ij}^2/2\sigma^2}$ between two image feature points with

$\delta_{ij} = \|\mathbf{p}_i^{t-1} - \mathbf{p}_j^t\|$. σ describes the "locality" of the interaction between the image features, where a higher σ allows for a more global interaction. The second step is to perform a singular value decomposition $\mathbf{G} = \mathbf{TDU}$, with $\mathbf{T}$ and $\mathbf{U}$ being orthogonal matrices and $\mathbf{D}$ a diagonal matrix containing the singular values. The idea is to change $\mathbf{D}$ into a matrix $\mathbf{E}$ by simply replacing all singular values with one, and to compute a new association matrix $\mathbf{A} = \mathbf{TEU}$.

As shown in [15], in the ideal setting (e.g. static image feature locations) $\mathbf{A}$ is a permutation matrix which maps features of image F_{t-1} to features of image F_t. In a real setting with moving feature locations, however, $\mathbf{A}$ has entries a_{ij} ranging from zero to one representing matching probabilities between feature positions $\mathbf{p}_i^{t-1}$ and $\mathbf{p}_j^t$. A one-to-one mapping between $\mathbf{p}_i^{t-1}$ and $\mathbf{p}_j^t$ is found if a_{ij} is the maximum in its respective row *and* column. In contrast to greedy methods working directly on $\mathbf{G}$ this method finds an optimal partial mapping under consideration of the given global distance measure.

The drawback of this method is that if both sets of features contain a significant number of actually uncorrelated elements like lost image features in F_{t-1} and completely new ones in F_t, the described method still finds correspondences between such elements. By following a simple thresholding-approach we can set the similarity values g_{ij} between two elements to zero if δ_{ij} is above a certain threshold τ to express that they cannot be correlated at all. In the case of image features, this threshold would naturally be defined by a maximally allowed Euclidean distance between feature points.

When applying the above algorithm, some of the singular values will be zero. To compensate for numerical errors, we set singular values to zero which are below a certain threshold ε. By setting only those singular values to one which are greater than ε, we effectively compensate the problem of matching uncorrelated elements, since the corresponding rows in the resulting matrix $\mathbf{A}$ of elements without a matching partner will be zero and thus do not contain a maximum element. Please note that the overall algorithm is not sensitive to the choice of ε, but that this threshold merely compensates for numerical errors. In all our experiments we use a default threshold $\varepsilon = 1^{-10}$.

We use the same procedure in our more general context by replacing the Euclidean distance with an appropriate measure to express the correlation between elements of two sets S_a and S_b. By formulating each problem in an appropriate way, we can use the same mapping algorithm to solve different tracking related subproblems.

5 Self-Calibration

In this section we will present the steps of our self-calibrating algorithm in more detail.

The central data structure of our algorithm is a (symmetric) global distance matrix $\tilde{\mathbf{D}}$. Each row (column) i corresponds to a specific marker identity $\tilde{m}_i \in \tilde{M}$. Every entry $\tilde{d}_{ij}$ of $\tilde{\mathbf{D}}$ stores the Euclidean distance and its variation over time between two markers $\tilde{m}_i$ and $\tilde{m}_j$ up to the current frame. Inter-marker distances with a small variation form the signature Sig_i for marker $\tilde{m}_i$.

Suppose we are at the first frame F_1 of our tracking sequence, where the first $k(1)$ markers become visible to the tracking system. Initially all tracked markers in M_1 are new to the system, so we assign a global ID $\tilde{m}_i$ to every marker m_i^1 and create an initial global distance matrix $\tilde{\mathbf{D}}$ of size $k(1) \times k(1)$ with the current Euclidean distance and zero variation $\tilde{d}_{ij} \leftarrow (\,\|\mathbf{p}_i^1 - \mathbf{p}_j^1\|, 0\,)$ for each pair of markers. This first iteration corresponds to steps 3 and 5 of our algorithm and mainly serves to set up initial data structures. For all successive frames F_t the following steps are processed.

Continuous Tracking: In the first step, the algorithm identifies continuously tracked markers based on the tracking systems' data. Since low-level frame-to-frame marker tracking is described in previous research [6, 19] and already solved by the tracking system [1] we will omit the details of this step in the following discussion. Each marker m_i^t which was tracked from frame F_{t-1} to F_t can be trivially associated with its previously generated global ID $\tilde{m}_j$.

Marker Recognition: For the remaining detected markers, which could not be identified in step 1, we first assume that they correspond to formerly continuously tracked markers, for which we already generated signatures and which became occluded for some reason. Hence, we try to find a matching global identity $\tilde{m}_i$ of a previously continuously tracked marker by comparing its signature to the Euclidean distance pattern of the unrecognized marker in the current frame. As mentioned before, we formulated the involved computational steps as instances of a general correspondence problem and apply the algorithm described in section 4.

More specifically, for each new, unrecognized marker m_n^t we compute a distance vector $V_n^t = \{d_1^t, \ldots, d_{nk(t)}^t\}$ with distances to all other detected markers in M_t. Our aim is now to find a signature Sig_m which has a sufficient number of elements in common with V_n^t, meaning that m_n^t and $\tilde{m}_m$ can be considered identical. Here our correspondence matching algorithm is applied on two levels. In a first step *a)*, we find the best assignment of *entries* in V_n^t to respective entries of each signature Sig_m. In the second step *b)* we then find the best matching among all signatures.

To define a similarity measure between a signature Sig_m and V_n^t, we first have to determine which (if any) elements from both sets match one another. Thus we first have to find the best matching of distances $d_{ni}^t \in V_n^t$ to $\tilde{d}_{mj} \in Sig_m$. This correspondence problem is solved in step *a)*. Please recall, that the correspondence algorithm of section 4 needs two thresholds as input, one for the maximally allowed dissimilarity between matches, and one for the standard deviation. We define our similarity measure $g_{ij} = e^{-\delta_{ij}^2/2\sigma^2}$ with $\delta_{ij} = |V_n^t(i) - Sig_m(j)|$ being the difference between values of V_n^t and Sig_m. The maximum threshold τ for allowed differences δ_{ij} and the expected deviation σ both depend on the precision of the tracking system. In our experiments we achieved stable correspondence estimation for reasonable values $\tau = 5mm$ and $\sigma = 1mm$. This correspondence estimation must be computed for every unrecognized pair of marker $m_n^t \in M_t$ and signature Sig_m. The result is the best possible permutation $\xi(i)$ of distances $V_n^t(i)$ to distances $Sig_m(\xi(i))$.

Based on the discrepancy between elements $V_n^t(i)$ and $Sig_m(\xi(i))$ we know how good V_n^t matches Sig_m. The larger the difference between them, the less V_n^t corresponds to Sig_m and the less likely is it that m_n^t corresponds to $\tilde{m}_m$. In step *b)* we use this information to find likely assignments of a global marker identity $\tilde{m}_m$ to the untracked marker m_n^t, which is equivalent to finding the best corresponding matches between all V_n^t and Sig_m. Once this is done, we have identified some of our new markers as actually reappearing markers.

The final similarity measure between distance vectors V_n^t and signatures Sig_m is again computed by the original proximity value g_{nm} for every pair of vectors and signatures based on the average difference $\delta_{nm} = \frac{1}{|Sig_m|}\sum_i |V_n^t(i) - Sig_m(\xi(i))|$. Again, the two thresholds for the correspondence algorithm can be chosen quite intuitively. For our setup, we again use $\sigma = 1mm$ and $\tau = 3mm$, meaning that we do not want to match V_n^t to Sig_m if the average difference between their elements is larger than $3mm$. It should be noted that these parameters depend on the precision of the tracking system, not on a tracking session. Hence they have to be specified only once for a tracking system.

After this step we have markers recognized by the continuous tracking approach as well as based on the currently available signatures. For the remaining unrecognized markers we have to assume that they are either completely new, or that the recognition algo-

rithm failed for some reason, like insufficient surrounding markers to generate a distance pattern V_n^t.

Marker Registration: In step 3 we simply create a new identity for all unrecognized markers by assigning them a global ID $\tilde{m}_i$ and by adding a corresponding row and column to the global distance matrix $\tilde{\mathbf{D}}$. Initially these new markers are added to a clique with all other markers below a certain distance, which corresponds to connecting them recursively by "strings". This threshold is naturally given by the maximum distance of markers within a single clique. We currently use a value of $180mm$. However, the algorithm is not sensitive to the choice of this value.

Marker Cleanup: Since quite a number of cases exist in which lost markers cannot be assigned to their corresponding global identity, and because of strings erroneously ripped due to noise or errors in the continuous tracking step, we have to apply some kind of error correction and cleanup of the data structures. For the first problem of multiple global marker identities for the very same actual marker, we try to merge entries of the global distance matrix $\tilde{\mathbf{D}}$ by examining the signatures of all $\tilde{m}_n$. The method to find corresponding $\tilde{m}_n$ and $\tilde{m}_m$ is identical to the marker recognition described above. The only difference is that it works on pairs of signatures Sig_n and Sig_m instead of distance patterns V_n^t for unrecognized markers and signatures Sig_m. To eliminate further erroneous marker identities, we apply some simple but effective checks. For example, we throw away currently untracked markers $\tilde{m}_i$ which have been tracked continuously only for a very small number of frames and are therefore probably unreliable. The same is true for markers with empty signatures, not having any connections left to other markers. Markers with a small signature containing less than three entries, which were not tracked for a longer period of time can also be deleted, since it is unlikely that they will suddenly become visible again with a complete clique of more than two markers. Finally, the tracking system can often produce false virtual markers for example caused by reflections or ambiguous marker configurations. These can effectively be eliminated by assuming a minimal distance between markers within a clique, e.g. $20mm$.

Signature Generation: In the last step we record the pairwise distance between all markers and its variation in the global distance matrix $\tilde{\mathbf{D}}$. The "strings" between markers with high length variation are ripped. To measure the variation we use a simple time adaptive average of the variance to smooth out sudden marker jumps due to errors in the tracking system. Finally, from the remaining connections we generate the signatures Sig_i for every marker $\tilde{m}_i$. From these signatures, we can compute the current estimates for marker cliques.

This last step concludes one iteration of the self-calibration procedure. As mentioned in section 3, the algorithm generates and refines marker signatures until it automatically detects a given target number of cliques, or until the user manually stops the calibration. During the auto-calibration, the tracking results provided by the system continuously improve and can already be used by an underlying Virtual Reality application.

Although in case of frequent marker occlusions a potentially high number of singular value decompositions is necessary for the correspondence estimation, we still achieve real-time performance by several simple but effective improvements. For example, we restrict the connections between markers to those lying within a reasonable distance of currently $180mm$, since markers further apart cannot form a single clique on a limb. This greatly reduces the number of necessary correspondence tests. During the recognition algorithm in step 2, which assigns an untracked global marker identity $\tilde{m}_i$ to a currently unknown marker m_j^t, we only allow assignments that do not destroy any existing "strings" of $\tilde{m}_i$ to other tracked markers. This reduces the number of actually needed singular value decompositions for the correspondence estimation to a level where the self-calibration can be done in real-time.

6 Skeleton Estimation

The next step in the motion capturing pipeline is to reconstruct the underlying skeleton model automatically. The extracted skeleton geometry and topology will allow us to retarget the captured motion data to arbitrary objects with the same skeleton structure. Moreover, the computed model helps us to make the motion capturing phase more robust in cases of occluded markers.

Several methods were proposed to automatically reconstruct the underlying skeleton structure from motion tracked data. Under the assumption of a skeleton model with rigid bones and rotational joints, O'Brien et al. [13] show how to robustly compute precise joint positions by solving a least squares system of motion measurements. We will describe this method shortly for completeness and provide a simple extension which allows us to reconstruct separate, disconnected skeletons for multiple tracked subjects at the same time.

Each of the identified rigid cliques corresponds to a limb of the tracked subject. We compute a local coordinate frame for each limb by defining the center of gravity of the marker clique as the local origin, while the orientation can be derived from the edges between markers of a clique. The minimum number of markers per clique to define such a coordinate frame is naturally three. By including a higher number of markers per clique, one can create several representations for this local frame based on every 3-subclique. This redundant information can be used to recover the orientation in a more robust manner, e.g., to resolve problems with occluded markers or with noisy measurements of the tracking system.

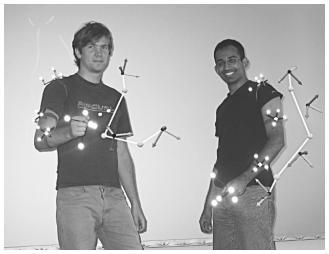

Figure 3: This figure shows a motion capture session with two simultaneously tracked subjects. The estimation routine correctly identified the two separate skeletons. We overlayed the computed limbs and joints to a photograph of the tracking session.

When each limb l_i is associated with a time-varying local coordinate system, there is a transform $\mathbf{L}_i^t = \left[\mathbf{R}_i^t | \mathbf{t}_i^t\right]$ which maps from l_i's current local coordinates to world coordinates. The joint between two limbs l_i and l_j has constant local coordinates $\mathbf{c}_i$ with respect to l_i and constant local coordinates $\mathbf{c}_j$ with respect to l_j. The coordinates $\mathbf{c}_i$ and $\mathbf{c}_j$ are related to each other by the fact that they map to the same position in world coordinates, i.e., $\mathbf{L}_i^t \mathbf{c}_i = \mathbf{L}_j^t \mathbf{c}_j$ for every frame F_t. For every possible pair of limbs and measurements in n frames this leads to an overdetermined system which we can solve for the local joint coordinates in the least squares sense:

$$\begin{bmatrix} \mathbf{R}_i^0 & -\mathbf{R}_j^0 \\ \vdots & \vdots \\ \mathbf{R}_i^{n-1} & -\mathbf{R}_j^{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{c}_i \\ \mathbf{c}_j \end{bmatrix} = \begin{bmatrix} \mathbf{t}_j^0 - \mathbf{t}_i^0 \\ \vdots \\ \mathbf{t}_j^{n-1} - \mathbf{t}_i^{n-1} \end{bmatrix} \tag{1}$$

For a reliable joint reconstruction, the tracked subject should perform motions which exert the available degrees of freedom for each joint, e.g., bending and stretching the knees or rotating the arms. Generally the quality of the joints does not so much depend on the duration of the motion but on the range of performed movements, so that this step is generally finished after only a few seconds to minutes, depending on the number of tracked limbs.

The skeleton structure can be computed by a minimum spanning tree (MST) for the graph connecting all the limbs. Joints are the connecting edges, weighted by the residual of the solution for the

above equation system. Solutions with a small residual correspond to consistent joint positions for two limbs, while all other solutions indicate unconnected limbs. This approach can easily be extended to allow the reconstruction of multiple skeletons, because all left edges within this MST with a high residual correspond to false connections between actually distinct skeletons (Fig. 3). This enables independent tracking and retargeting of several persons in a Virtual Reality scenario.

During the actual tracking phase, joints can be computed by averaging the two positions $\mathbf{L}_i^t\mathbf{c}_i$ and $\mathbf{L}_j^t\mathbf{c}_j$. However, even if one limb is completely lost, all joint positions are still explicitly defined. The geometry of the bones is given by the distance between adjacent joints.

Similarly to the signatures and local coordinate frames based on fixed inter-marker distances we can exploit the fact that every marker has an invariant distance to the joints associated with its corresponding limb. Therefore, two joints plus a marker of the associated limb define an additional signature for marker recognition, as well as an unique transformation to the coordinate frame of the actual limb. The same is true for two markers plus one joint of the same limb. It should be clear that in this way marker signatures, limbs, and joints are redundantly defined. While this is a valuable method to make the marker recognition itself more stable, it is also a necessary technique for the recovery of lost limbs and joints, as described in the next section.

7 Robust Motion Capture

During the motion capturing phase we can use several methods to make the tracking robust. We have two robust methods to identify formerly lost markers, clique-based marker recognition and joint-based marker recognition as described in section 5 and 6 respectively. Position and orientation are redundantly given for each limb, and finally the computed skeleton reduces the degrees of freedom for every limb by imposing constraints of adjacent limbs.

In this section we will consider cases of multiple unseen markers, which lead to completely lost limbs or joints. For example, if the position of the upper *and* lower arm are unknown, the shoulder and hand positions allow the system only to compute a circle in space on which the elbow must lie [17]. We will show how the remaining degree of freedom can be determined by using partial information from a single marker and by exploiting the geometric constraints of the skeleton. In fact such cases occur quite frequently since a whole clique is quite easily lost during tracking while scattered marker data remains.

In the following discussion we will distinguish between the so-called inner limbs with at least two adjacent limbs, and outer limbs having only one single joint attached. Similarly, inner joints lie between two inner limbs, while outer joints have at least one incident outer limb.

7.1 Lost Inner Limbs

When the marker clique of an inner limb gets occluded or lost during tracking, the corresponding joint positions are still available based on the surrounding limbs. But unfortunately, the missing orientation information of the lost marker clique leaves one rotational degree of freedom open. However, some markers of the lost limb might still be visible, since in most cases marker cliques are only partially occluded. Although we cannot identify such a marker using our clique-based signatures of section 5, we can use the methods presented in the previous section to identify the marker by taking its distance to both associated joints into account. Knowing which marker of the lost clique is currently tracked allows us to put up a coordinate system using the markers and the two joints positions. The obtained coordinate system is then transformed to fit the limb orientation using a transformation matrix which has been pre-computed for the identified marker. If a limb has more than two joints they already define a coordinate system themselves which can be transformed into the coordinate system of this limb's marker-clique.

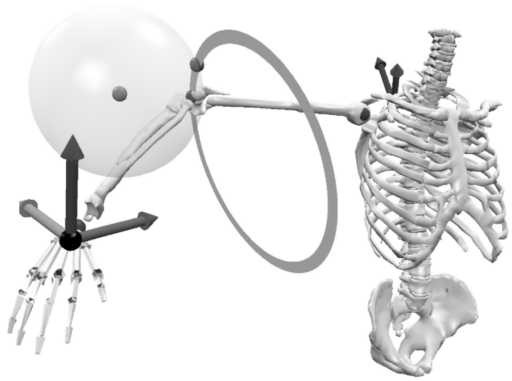

Figure 4: This figure shows a situation where the upper and lower arm are lost during tracking. The lost inner joint can be reconstructed by intersecting three spheres as described below. The circle visualizes the intersection of two of them. The third sphere intersects this circle in two points, yielding the lost inner joint.

7.2 Lost Inner Joints

If two inner limbs like that of a human arm like chain (HAL-chain) are lost during tracking, the position of the missing inner joint can still be computed up to one degree of freedom. Tolani et al. [17] show that it has to lie on a circle defined by the intersection of two spheres (see Fig. 4) given by the outer joint positions $\mathbf{j}_1$ and $\mathbf{j}_2$ of the HAL-chain and both inner limb lengths l_1 and l_2. Fortunately we will see, only one single marker of the inner limbs already helps resolving this issue. Like in the above case it is very unlikely that both cliques of the lost limbs are completely occluded. In most cases there will be at least one additional marker position $\mathbf{p}$ available. Again, this marker can be identified by its rigid distance to the corresponding joints. Knowing its constant distance d to the missing inner joint position, it is possible to define a third sphere centered at the marker's position $\mathbf{p}$ with radius d. The lost inner joint position $\mathbf{j}$ has to be the intersection $\mathbf{j} \in S(\mathbf{j}_1, l_1) \cap S(\mathbf{j}_2, l_2) \cap S(\mathbf{p}, d)$ of these three spheres.

Noisy measurements can lead to more than one or no solution. However, additional markers constrain the solution even more. Otherwise we choose the most plausible solutions, according to continuity assumptions or other heuristics.

7.3 Lost Outer Limbs and Joints

In the case of lost outer limbs and joints one can apply similar methods for reconstruction. For instance, a lost outer limb is constrained by one additional marker to lie on a circle in space. A second marker again allows us to define a coordinate system, of which the limb position can be derived. For lost outer joints which are caused by a lost inner and outer limb, every additional marker can be used to reduce the degrees of freedom in an analogous way.

Since markers are generally occluded only during a few frames, alternative marker or limb recovery methods based on movement prediction are of course also applicable [6]. However, explicit solutions for lost limbs and joints can provide much more consistent results in real-time applications like Virtual Reality scenarios, where instant visualization of a tracked subject is necessary.

Please note that all presented methods for the skeleton estimation and lost limb reconstruction are not restricted to humans, but work for arbitrary articulated bodies. This renders these techniques applicable to a much wider range of applications than specialized tracking solutions.

8 RESULTS

We tested our implementation in two different setups with four and six ARTTrack1 cameras [1] respectively. During all experiments, the tracking rate was set to 50Hz and the cameras' fields of view covered an area of $4 \times 6 \times 3$ meters within a rectangular room. Generally we used cliques composed of four markers to keep the number of necessary markers small while ensuring a redundant definition for each clique. However, we assembled these cliques quite arbitrarily without optimizing their inter-marker distances for the best possible signature distinction to simulate environments where a perfect assembly is not always possible. Typical inter-marker distances ranged from 40*mm* to 120*mm*.

In the first setup we placed four cameras in the upper corners of the room. A subject was equipped with 40 markers / 10 cliques to record a full body motion sequence (Fig. 6). While this setting allows us to track quite unconstrained movements, it also results in very frequent occlusions. If a clique is attached to the limb of a subject, rarely more than two cameras can see the corresponding markers since the body of the person occludes the view of the opposite cameras. Moreover, using cliques of size four or larger it is very likely that two markers lie approximately on the same viewing-ray for one of the cameras and cannot be reconstructed.

The movements for the self-calibration phase were performed as described before. In the following the subject performed some arbitrary movements such as walking or running. In the average the tracking hardware detected 95% (38) markers per frame. However, this means in the worst case that after one second of tracking only $40 * 0.95^{50} = 3$ of the original markers are left (see Figure 1).

The continuous marker tracking performed by the tracking hardware succeeded only in 29% of the recorded frames to track all of these 95% markers, such that in 71% of the frames one or more new markers appeared and our signature based marker recognition was activated. In these cases, the average number of identified markers increased from 89% (35.6) of continuously tracked markers to 93% (37.2) overall recognized markers. Although the per-frame improvement may not seem large, the detected markers play a crucial role for the tracking process for the above mentioned reasons. Once a continuously tracked marker is lost by the system, it cannot be identified in later frames. Hence almost no initially known marker would be left already after only a few seconds of tracking without our recognition procedure.

Please note that due to the suboptimal assembled marker cliques we needed all four markers of a clique to recognize it after a complete occlusion. The number of recognized cliques should improve for setups with carefully assembled cliques, where the signatures are distinct enough for three markers to suffice for recognition.

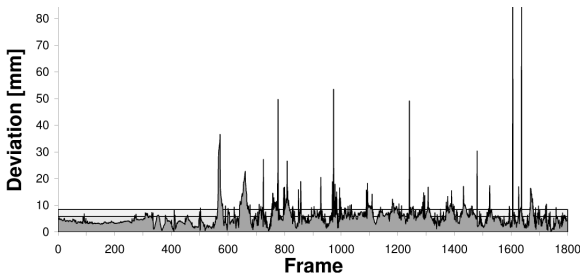

Figure 5: The deviation between the computed joint position using the inverse kinematic method and the actual joint position for the elbow.

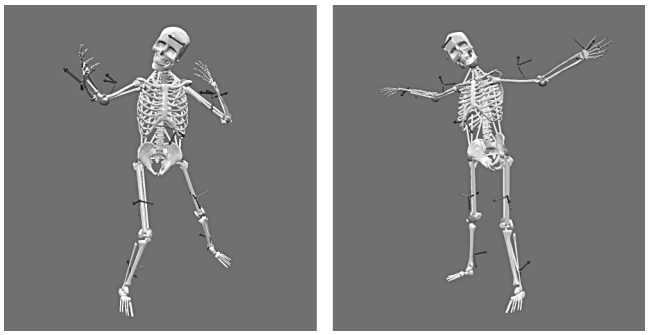

Figure 6: This figure shows two frames of a full body motion capture session. The limbs of the automatically computed skeleton have been manually augmented with bone meshes.

If the number of unseen markers or cliques becomes too high due to problems of the tracking hardware and therefore there is almost no continuous tracking possible between frames, e.g. during very fast running or jumping motions, the self-calibration procedure cannot generate stable marker signatures and cliques. In such cases, the SVD based correspondence estimation can take up to 300*ms* to recognize all ten cliques at once. However, we observed these problems only in extreme cases where the tracked subject is in permanent complex motion right from the beginning. In practically relevant situations we did not encounter these problems. Furthermore the setup with only four cameras can be considered a worst case for full body motion tracking due to the inevitable occlusions as described above. In general, already a few seconds of slow to moderate motion with a low number of occlusions suffice for the self-calibration to terminate successfully.

The standard deviation of the inter-marker distances for the signature creation during the self-calibration phase was 0.98 *mm* for a relatively still standing person up to 2.15 *mm* for moderate motions.

Figure 3 shows an example situation with two simultaneously tracked persons equipped with 34 markers / 8 cliques. Our software correctly calibrates and tracks the motion of the two separate arms from shoulder to the hand. While our tracking hardware had a limitation of approximately sixty simultaneously tracked markers during our experiments, this restriction will surely be alleviated in the future, allowing for a higher number of interacting persons.

As an example for the inner joint estimation (section 7) we measured the deviation of a reconstructed elbow joint (Fig. 4) from its exact position (Fig. 5). The average deviation of the reconstructed joint from the actual joint position is only about 5 *mm* with a standard deviation of 2.5 *mm*. The high peaks result from noisy low-level tracking or wrongly identified single markers, in which case the sphere $S(\mathbf{p}, d)$ is of wrong size and the computed circle intersections result in wrong marker positions. However, such errors can be identified easily by assuming a continuously moving subject. The wrong positions can be eliminated by enforcing physically plausible movements of the joints.

In the second setup we placed two additional cameras in the lower corners of the room to decrease the number of occluded markers for recording more complex motions such as running, boxing, or jumping. This setup dramatically reduces the number of unseen markers, so that we had approximately 97% (38.8) identified markers with ten cliques. We compared the reconstructed skeletons of five different recording sessions and found an average / maximal variation of only 2.1% / 2.9% in the computed limb lenghts.

Due to the direct feedback on the calibration quality one can

easily correct potential errors early during the motion recording pipeline and therefore improve the overall quality for the actual motion recording. With the computation of the skeleton geometry and topology being a matter of seconds, the whole calibration procedure is generally finished within a few minutes.

Finally, Figure 6 shows two frames of a full body tracking session.

9 Conclusions

We presented a self-calibrating optical motion tracking framework for arbitrary articulated bodies, which allows us to estimate all relevant model parameters without any auxiliary information on the tracking setup, enabling a much wider range of applications in comparison to current practice.

Since our method is designed to respond to highly dynamic environments, it is well suited for different types of Virtual Reality scenarios involving varying setups for tracking and retargeting articulated bodies, where pre-specified assumptions would constrain the range of possible applications. Being able to distinguish multiple simultaneously tracked skeletons in a virtual environment enables on-the-fly retargeting and interfacing for multi-user scenarios. This opens up new possibilities for direct manipulation or interaction metaphors in virtual environments. The requirement of immersion is intrinsically fulfilled by using the motion of a subject as the input for virtual interaction devices.

It should be mentioned that our system also preserves the properties of systems like [1], since once a marker-clique is calibrated and known to the system, it does not have to be recalibrated for further tracking sessions. However, using retroreflective stickers instead of spherical-markers would also allow us to apply this method to highly dynamic environments by just attaching a number of these stickers to the limbs of an arbitrary person.

In the future we plan to apply the presented self-calibrating methods to setups with a reduced number of markers per limb, which could be a benefit for current practice. A further reason would be the restricted number of trackable markers for current tracking hardware, which currently forbids to track two complete skeletons simultaneously using our method. Furthermore we will intensify our research on extending our self-calibrating pipeline to create a higher order analysis of the tracked data beyond the skeleton geometry, such as constraints on the degrees of freedom at joints, or statistical distributions of limb positions. In this context we are also working on automatic parameterizations of the reconstructed model and the integration of musculoskeletal models like the one of Delp et al. [5].

Once these steps have been taken one can analyse the movements of a subject based on these higher order models. Such models would enable the retargeting of the tracked motion to quite different target models, evaluate the validity of performed motions in the context of the new target model, and apply suitable changes (see also [11]). The resulting movements can be expected to be very realistic, even if the target character or the target environment differ significantly from the recorded data.

We think that combining techniques for easily accessible and configurable tracking of arbitrary subjects could also contribute to medical disciplines like real-time muscle force computation [10], gait analysis [18], extraction of motion characteristics [3], or virtual training scenarios [2].

10 Acknowledgements

We thank A.R.T. [1] for kindly supporting this project with additional tracking equipment.

References

[1] A.R.T. advanced realtime tracking GmbH, http://www.ar-tracking.de/. *ARTtrack1 & DTrack*.

[2] Seongmin Baek, Seungyong Lee, and Gerard Jounghyun Kim. Motion retargeting and evaluation for vr-based training of free motions. *The Visual Computer*, 19(4):222–242, July 2003.

[3] Thomas Beth, Ingo Boesnach, Martin Haimerl, Jörg Moldenhauer, Klaus Bös, and Veit Wank. Characteristics in human motion - from aquisition to analysis, October 2003.

[4] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *ACM Transactions on Graphics*, volume 22, pages 569–577, 2003.

[5] Scott L. Delp, J. Peter Loan, Melissa G. Hoy, and Felix E. Zajac. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering*, 37(8):757–767, 1990.

[6] Klaus Dorfmüller-Ulhaas. Robust optical user motion tracking using a Kalman filter. In *10th ACM Symposium on Virtual Reality Software and Technology*, 2003.

[7] Anthony C. Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. In *ACM Transactions on Graphics*, pages 417–426, 2003.

[8] Lorna Herda, Pascal Fua, Ralf Plänkers, Ronan Boulic, and Daniel Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proc. Computer Animation*, 2000.

[9] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR)*, 1999.

[10] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Calculation and visualization of the dynamic ability of the human body. *The Journal of Visualization and Computer Animation*, (10):57–78, 1999.

[11] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer*, (16):254–270, 2000.

[12] Kazutaka Kurihara, Shin'ichiro Hoshino, Katsu Yamane, and Yoshihiko Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. In *IEEE International Conference on Robotics and Automation*, 2002.

[13] James F. O'Brien, Robert E. Bodenheimer, Gabriel J. Brostow, and Jessica K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proc. Graphics Interface*, 2000.

[14] Maurice Ringer and Joan Lasenby. A procedure for automatically estimating model parameters in optical motion capture. In *British Machine Vision Conference*, pages 747–756, 2002.

[15] Guy L. Scott and H. Christopher Longuet-Higgins. An algorithm for associating the features of two images. In *Proc. R. Soc. London*, volume 244, pages 21–26, 1991.

[16] Marius-Calin Silaghi, Ralf Plänkers, Ronan Boulic, Pascal Fua, and Daniel Thalmann. Local and global skeleton fitting techniques for optical motion capture. *Lecture Notes in Computer Science*, 1537:26–40, 1998.

[17] Deepak Tolani, Ambarish Goswami, and Norman I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, (62):353–388, 1999.

[18] Raquel Urtasun and Pascal Fua. 3D tracking for gait characterization and recognition. Technical Report 200404, Computer Vision Lab, Swiss Federal Institute of Technology (EPFL).

[19] Robert van Liere and Arjen van Rhijn. Search space reduction in optical tracking. In *Ninth Eurographics Workshop on Virtual Environments*, number 9, 2003.

[20] Vicon Motion System Ltd, http://www.vicon.com/. *Vicon iQ*.

[21] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments. In *Symposium on Virtual Reality Software and Technology*, 1999.

[22] Victor B. Zordan and Nicholas C. Van Der Horst. Mapping optical motion capture data to skeletal motion using a physical model. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.