Real-time Estimation of Missing Markers for Reconstruction of Human Motion

Qiong Wu
University of Alberta
Department of Computing Science
Edmonton, Alberta T6G 2R3 Canada
qiong@ualberta.ca

Pierre Boulanger
University of Alberta
Department of Computing Science
Edmonton, Alberta T6G 2R3
pierreb@ualberta.ca

Abstract

Optical motion capture is a prevalent technique for capturing and analyzing movement. However, a common problem in optical motion capture is the missing marker problem due to occlusions or ambiguities. Most methods for resolving this problem either require extensive post-processing efforts or become ineffective when a significant portion of markers are missing for extended periods of time. In this paper, we present an approach to reconstruct human motion corrupted in the presence of missing or mis-tracking markers. We propose a data-driven, piecewise linear predicting kalman filter framework to estimate missing marker position, and reconstruct human motion in real time by rigid body tracking solver. It allows us to accurately and effectively reconstruct human motion within a simple extrapolation framework. We demonstrate the effectiveness of our method on real motion data captured using OptiTrack. Our experimental results demonstrate that our method is efficient in recovering human motion.

1. Introduction

Motion capture is the techniques of recording movement of an subject in real life and translating it into digital data. It is widely used in movie and video game [21] industry, as well as in other areas such as sports [13] [1], medical [8] and performing art etc. For example, in the movie *Titanic*, the movements of all computer simulated characters are created from motion capture data.

Amongst all motion capture techniques, optical motion capture (e.g. Vicon [2]) is the most commonly used in various applications. A set of markers are attached onto the object and tracked by a array of cameras. A passive system usually uses infrared illuminator collocated with each camera and markers are retro-reflective material to reflect infrared back to camera. An active system use LED markers instead. Both design enables cameras to track position of markers, and therefore to infer time-spatial varying markers by triangulation of markers position projected onto several camera's image plane. The captured data can be used to compute the movements of the skeleton and transformed into movements of digital characters.

However, even with expensive motion capture equipment, data from optical motion tracking may contain noise, outliers or missing of data for a time period. A major cause of missing markers is occlusion, that is markers may be occluded by props, body parts or other object. Outliers may be caused when system confuses one marker with its neighborhood markers, and therefore mis-tracks marker's position. Because of the presence of these corrupted data in recording, it requires significant amount of manual editing in the post processing, which is not only time consuming, but also error prone to diverse motions.

This paper presents an automatic method to predict missing or correct corrupted data and reconstruct the skeleton movement during motion capture session in real time. The method starts with data-driven prediction on missing or corrupted marker position, consider the rigid body tracking defined by marker set, as well as constraints between body parts, to reconstruct valid human poses from corrupted data. We focus on optical motion capture device and articulated figure motion (human motion), demonstrate our method on a low-cost motion capture device OptiTrack [3] and its tracking software Arena [4].

The following section presents related work, followed by a section showing our methodology. Then come to empirical section showing the experimental results. This is followed by a conclusion.

2. Related Work

Either caused by occlusion or ambiguities, it is all called missing marker problem. In order to overcome this common problem and recover data in optical motion capture system, there are various existing approaches, mainly falling into two categories: off-line and real-time approaches.

Some typical off-line methods interpolate data using linear or non-linear methods [10, 22, 28]. However, interpolation requires future measurements, therefore it causes noticeable latencies in processing, and can only be used in a post-processing step. Some offline approach learns a series of spatial-temportal filter bases from pre-captured human motion data and use them to filter corrupted human motion data [20]. Some uses a pre-trained classifier to identify local linear model for each frame to recover missing data in a new data sequence [19]. Some Mocap systems [5] [2] also use interpolation techniques with kinematic information as recovery solution. Off-line approaches also include model based methods. Rhijn and Mulder [25] proposed a geometric skeleton based approach to bridge existing gaps in the measured data series.

Real-time approaches to recover motion data include predicting measurement state using Kalman filter [23, 27], extended Kalman filter (EKF) [11] [26], or unscented Kalman filter [24]. Information processed by Kalman filter ranges from marker's position, velocity, to limb's rotation and angular velocity. However, process model of rotation does not obey linear process. In addition, Euler angle representation has the singularity problem and Quaternion does not satisfy angular velocity requirement. For example, Fang et al. [12] had to maintain unity of quaternion in order to apply low-pass filter to the estimated angular velocity in quaternions. Filtering non-linear orientation space is also not computationally effective as the linear filter, therefore many researchers explore ways to convert non-linear process into linear filter process. Lee and Shin [18] formulated filtering non-linear orientation data in a linear time-invariant filtering framework, by transforming orientation data into a vector space and then transforming the results back to orientation space after applying filter.

Some real-time approaches made prediction based on other quality measurements such as fixed distance between markers [7]. This type of approach usually does not consider kinematic information or internal skeleton model. Because it assumes markers on a given limb segment has constant inter-marker distance, it becomes ineffective when all markers on one limb segment are missing. In addition, it uses active marker system which has problem of sensitivity to external influences such as ambient light. Method proposed by Hornung et al. [17] also takes advantage of inter-marker signature, as well as uses rigid body tracking and inverse kinematics to improve tracking quality. However, they do not identify or reconstruct markers based on prediction of future states like our method does. Their method can compensate missing markers due to occlusion but may fail in case of ambiguities that cause jitter and rapid tracking changes.

Others resolve occlusions based on sophisticated human model [15] [14] alone. They identify marker's position or disambiguate its 3D location by replacing it by expected position on the skeleton. However, because it does not consider historical statistical information, prediction from skeleton alone may become ineffective when markers are occluded or confused with neighborhood markers for a short period of time.

Unlike previous approaches in real-time category, our work first avoids non-linear orientation space by focusing on position space. Building on the success of applying Kalman filter to remove jitter, rapid changes in recorded marker positions that do not correspond to actual movements and filling the missing data gapes, we receive a continuous stream of 3d data that prevents wrong tracking and recover missing data. We then perform rigid body tracking solver. With kinematic information, fixing the tracking of a parent rigid body may lead to correct tracking of the connected limbs. This ensures a reliable tracking even all markers on a rigid body is occluded or corrupted over long period of time. We demonstrate our method with a practical optical motion capture device, OptiTrack, and its software Arena, showing the ease but effectiveness of our method. Our method could significantly increase the robustness of the marker tracking, and drastically reducing or even eliminating the need for human intervention during the 3D reconstruction process.

3. Human Motion Reconstruction

Our method is an improvement based on the data acquired from a low-cost optical motion capture device, OptiTrack [3], and its skeleton model computed from its associated software, Arena [4]. 3D tracking of each marker is computed from 2D images of six tracking cameras of the OptiTrack. However, despite the fact that OptiTrack produces reliable tracking results most of time, occlusions, ambiguities or sudden changes in motion still often cause erroneous motion tracking or missing markers in the tracking trajectory. For this reason, we explain in this section how we can improve tracking with the practical motion capture device. We focus on predicting marker's position, where constraints from rigid body tracking and inverse kinematics (IK) are then applied for a more reliable tracking.

3.1. Prediction of Missing Markers

We call it missing marker problem no matter it is caused by occlusion or ambiguities. Occlusions result in missing gaps in data series and ambiguities result in rapid change in data curves. We examine such a situation by monitoring the tracking data, that is the 3D position of each marker reported from the motion capture server of the OptiTrack.

The state of each marker in current frame x_t, is compared with its state in the previous frame x_{t-1}, and the problem is identified if the difference is a rapid change, represented as $|x_t - x_{t-1}| > \delta$, where δ is the threshold.

Prediction is made based on the Taylor series:

$$x(t+\Delta t) = x(t) + \dot{x} \cdot \Delta t + \frac{1}{2}\ddot{x}(t) \cdot \Delta t^2 + \frac{1}{6}\dddot{x}(t) \cdot \Delta t^3 + \cdots \quad (1)$$

and the Kalman filter. The process model that updates the state of the Kalman filter is given by

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1} \quad (2)$$

where A is the state transition matrix, B is the control input matrix, u is the control vector, and w is the process noise with zero mean and covariance Q, represented as $P(w) \sim N(0, Q)$. The measurement model is given by

$$z_t = Hx_t + v_t \quad (3)$$

where H is the measurement matrix, and v is the measurement noise with zero mean and covariance R, represented as $P(v) \sim N(0, R)$. Q and R represent systematic adjustment according to statistical knowledge about the noise in the process model and the measurement model respectively.

According to Kalman filter, it is a feedback control process and mainly contains two processes: prediction and correction. In the prediction step, the predicted state $\hat{x}_t^-$ and its error P_t^- can be computed as:

$$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_{t-1} \quad (4)$$

$$P_t^- = AP_{t-1}A^T + Q \quad (5)$$

The correction stage is to compute the Kalman gain, and uses Kalman gain and new measurement to correct predicted state and predicted error. Kalman gain is computed as:

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1} \quad (6)$$

The new state $\hat{x}_t$ at time t is corrected as a combination of the prediction $\hat{x}_t^-$ and the correction from observation, given by:

$$\hat{x}_t = \hat{x}_t^- + K_t(Z_t - H\hat{x}_t^-) \quad (7)$$

The error covariance matrix of the updated prediction is:

$$P_t = (I - K_t H)P_t^- \quad (8)$$

Our goal is to predict the current state of the marker when the marker is occluded, or correct current state of the marker when it is a wrong tracking. We use a constant velocity model involving position and velocity in the Kalman filter, because the higher derivatives the more noisy and higher computational cost it becomes. According to constant velocity model,

$$x_t = x_{t-1} + \dot{x}_{t-1}dt \quad (9)$$

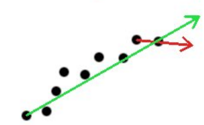

Figure 1. Comparison of velocity from consecutive measurements and sampling measurements

where x_t and $\dot{x}_t$ is the position and velocity of the marker respectively at time t. The predicted state in the Kalman filter can be rewritten as:

$$\begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{bmatrix} \quad (10)$$

This model does not have control terms B and u, and the transition matrix A becomes:

$$A = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \quad (11)$$

In order to improve accuracy, we do not use consecutive measurements to compute velocity, but sampling in a certain rate to compute historical information about the velocity. This means we compute piecewise velocity. Suppose our sampling rate is at δt [1], equation (10) becomes:

$$\begin{bmatrix} x_t \\ \dot{x}_{\delta t} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{\delta t-1} \end{bmatrix} \quad (12)$$

Figure 1 illustrates the difference of velocity information before and after sampling. Because each measurement has white noise, it does not reflect desired historical information and thus cannot be directly used in the filter model.

3.2. Rigid Body Tracking

3D coordinates of the marker are computed using stereo triangulation from 2D projection images of at least two cameras that can see the marker. Once the marker's 3D position is reconstructed, tracking of the marker from one frame to the next frame is called 3D tracking. With 3D marker tracking, a skeleton can be inferred. The process of fitting the skeleton to the subject's anatomy by scaling bones length is called skeleton calibration. Essential steps to establish a skeleton that fits in the marker cloud mainly include:

[1] δt is smaller than the motion capture rate, in our case, 100 frames per second from OptiTrack

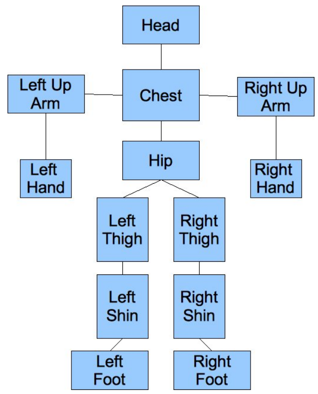

Figure 2. Rigid body definition in Arena

1. Segment markers into groups that define body segments.
2. Compute joint position of the skeleton and determine skeleton topology.
3. Adjust bones (or segments) length of the skeleton.

The reason to segment markers into groups is that bone motion can be well simulated by a rigid body transformation (including translation and rotation), and marker group attached on the rigid body can define rigid body motion. Given the number of segments, say n, of the skeleton to define, markers can be segmented into n groups. Clusters can be determined by minimizing the sum standard deviation of distances over all marker pairs. Usually markers attached on the same limb will be grouped together. With markers partitioned into rigid segments, it can compute fixed inter-marker distance and fixed marker's location with respect to joints on the segment, so called marker model. With this marker model, the skeleton's topology and locations of the joints connecting segments can be determined. 3D location of the centre of rotation (CoR) between adjacent segments corresponds to the joint location. A general skeleton fitting technique by estimating the CoR of markers and their associated limbs is presented in [6]. After marker segmentation and skeleton fitting, each marker will be labeled a unique identity to mark which group it belongs to or which limb it is attached to. The identity will help track the marker and the limb it is attached to during the motion capture. Not every limb needs to be defined as a rigid body, because the transformation of an inner limb can be determined if the connected joint positions are available based on the neighboring limbs. For example, Figure 2 shows a typical rigid body definition and skeleton topology, where lower arm is not defined.

Once the skeleton and rigid body model is established, at any time, the orientation of a rigid body can be computed as described by Horn [16], that best aligns the marker and joint locations for that segment. The location of joint at any time can be computed using [9] by taking advantage of markers attached on a limb are approximated by a rigid body.

There are many advantages of employing rigid body tracking in the motion tracking:

1. In ideal cases, markers defining the rigid body do not move with respect to each other. In other words, inter-marker distance on a rigid body is fixed. However, markers are actually attached to skin or tight clothes that may move relative to the underlying bone structure, so constant inter-marker distance alone may cause problem in marker identification process.
2. If only one marker is missing on the rigid body, by tracking a rigid body of markers, it can predict position of missing markers based on characteristics of fixed inter-marker distances, so that the system can continue tracking.
3. In passive optical tracking system, when markers appear again after period of occlusion, the system cannot identify their identity but leaving them as anonymous markers. Rigid body tracking will attempt to identify anonymous markers. All rigid bodies that are missing one or more markers will be identified, and associate anonymous 3D marker to the limb by the distance closest to the marker-to-joint distance defined by the marker model.

3.3. Inverse Kinematics

Inverse Kinematic (IK) solver is a simulation mechanism that situates limbs according to their known end effector position. The IK technique requires the position and orientations of certain joints, named end effector, to configure the remaining DoFs. FABRIK (Forward And Backward Reaching Inverse Kinematics) is one of the popular algorithms that uses points and lines to solve the IK problem. For example, once a rigid body skeleton is fitted to the data, rotations can be found using this algorithm. It has also been used for marker prediction and CoR estimation.

Computing joint rotation from 3D tracking of marker group is error prone to marker noise and require at least three markers per segment. Using IK can improve rotation computation once a rigid body skeleton is fitted to the data. Since the rigid body skeleton and the marker model is known, IK can find the optimal rotations that minimize the distance from the marker position on the segment and the input data.

4. Result

The experiments were carried out using OptiTrack motion capture system with 6 cameras. 34 markers are attached

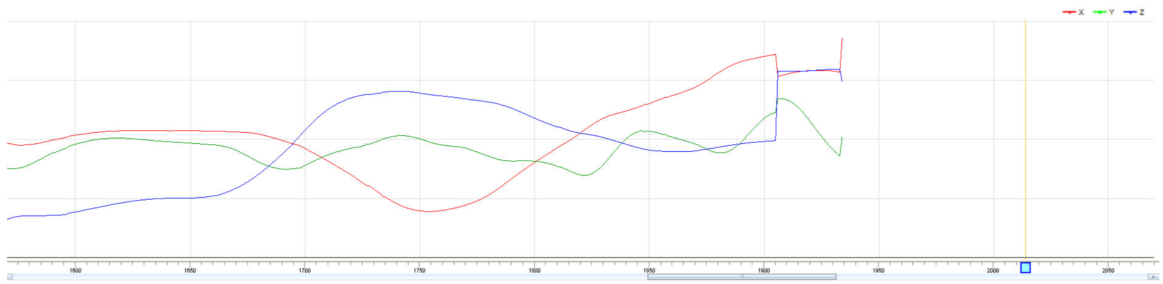

Figure 3. Motion curve from Arena's motion curve editor: position of the first marker on the right upper arm, RUArm1, between frames 1550-2050. The Arena failed in tracking the marker's position starting around the frame 1935.

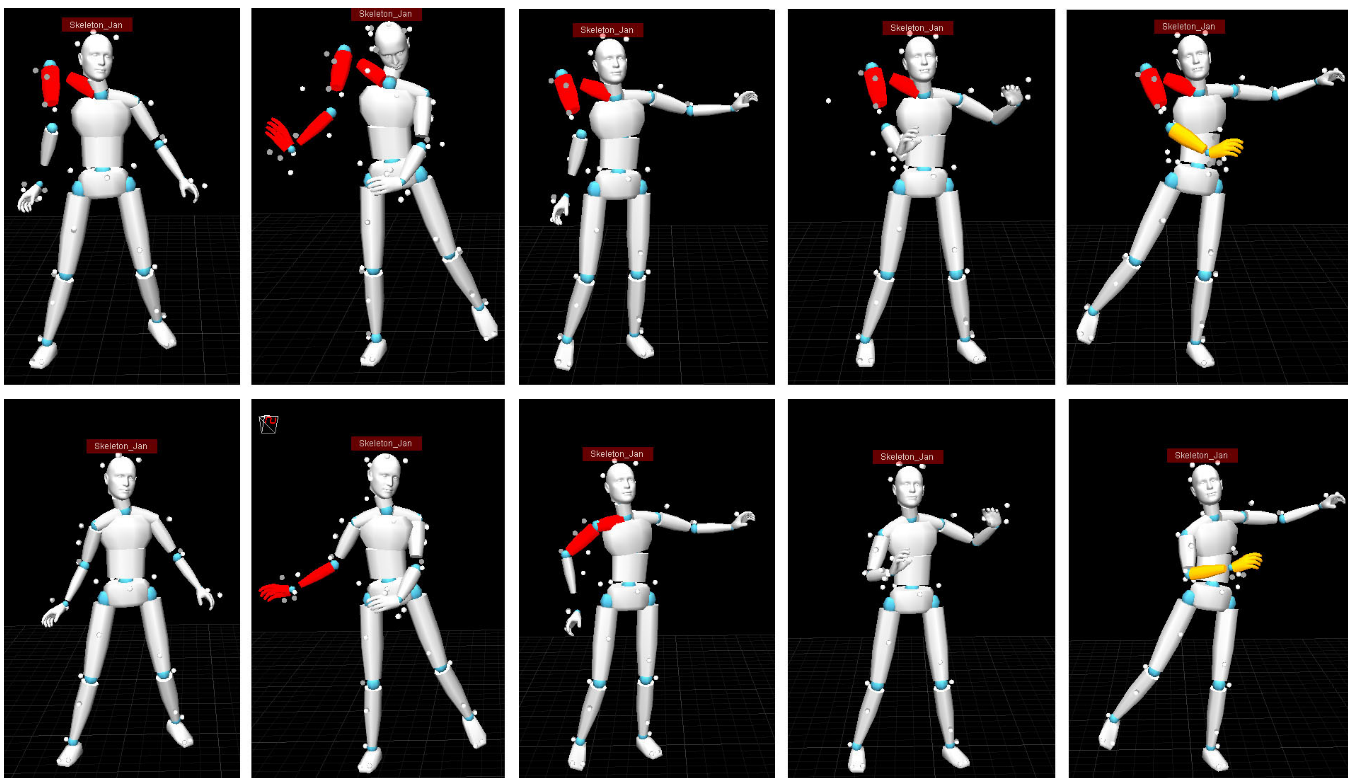

Figure 4. Comparison of motion before and after reconstruction at the frame 1984, 1956, 2023, 2126 and 2200, in a situation where the right upper and lower arm are lost during tracking due to missing markers on the right upper arm. Above panel: Motion from original marker's information. Rigid body engine will make the rigid segment flex (segment in red color) to match its defined point cloud and inverse kinematics model may cause connected limbs flex as well. Bottom panel: Motion reconstructed from marker's state computed using our method. Only markers located on the right upper arm (indexed as RUArm1, RUArm2, RUArm3) are identified missing marker problem and filtered for prediction. The motion curves before and after filtering of three markers defining rigid body of the right upper arm are shown in Figure 4.

on the moving person, and the skeleton model as well as 13 rigid segments of the skeleton are defined in Arena and the motion is computed in real time. This skeleton and rigid body model has a marker model associated with it, that is one of the segments connected to the joint have at least two markers, and the other at least one marker. The motion capture software Arena associated with OptiTrack is used as the server sending motion capture data, including marker's position information, rigid body's position and orientation, as well as skeleton's information (6 degrees of freedom for each joint). We implemented the prediction part. The algorithm is implemented in C# and run on a 3-GHZ Pentium PC. The system can process up to 100 frames per second, which is the speed of the motion capture system. Because the rigid body tracking and inverse kinematics are well implemented in Arena, this avoids us to implement this part by ourselves. Arena can also visualize motion on a virtual character in real time, so we visualized the reconstructed motion results in Arena, by reconstructing the marker's motion curve in Arena's motion curve editor based on our computation along with its rigid body engine and inverse kinematics model. The data set used in the experiment is a real data sequence captured using OptiTrack with natural occlusions and wrong tracking, containing more than 5000 frames in the data sequence.

As shown in Figure 3, the tracking of the position (x, y and z) of the RUArm1[2] over time, Arena is failed in tracking marker's position when there is a rapid change in marker's position that may be caused by ambiguities. Images in the above panel of the Figure 4, show the constructed motions resulted from losing track of markers on the right upper arm. What Arena does in this situation is that the rigid body tracking engine will make an assumption of where that marker should be, and predicted positions are markers in grey color. Most of time, markers on the rigid body are predicted based on its fixed position on the rigid body, or its position relative to the rest of markers in this marker group that defines the rigid body (called point cloud), so that Arena can continue to track the rigid body even when some markers are occluded. In ideal cases, predicted marker's position should be the same as measured position, that means markers in grey color should overlay with markers in white color. However, when the markers are still visible but for some reason the point cloud markers are out of original shape, e.g. the last image in the bottom panel of Figure 4, the rigid body will flex to match that location up to a threshold (tint the segment to yellow). Arena has a threshold about how much the markers can be flexed from their ideal location. When they distort above the threshold, the rigid body will tint its line to red. The body tracking engine in Arena places that rigid body to the best of its ability amongest the point cloud makres, but it is not fitting exactly.

[2]It is the index of the first marker on the right upper arm.

The proposed method processes the data received from the Arena server and computes in real time. When the algorithm identifies a missing marker problem it can predict the position of the marker and replace the original measured value. Figure 5 shows the motion curve before and after filtering of three markers that define the right upper arm, indexed as RUArm1, RUArm2, and RUArm3. The rapid changes in the motion curve are filtered, therefore jitter changes are removed. According to the new motion curves produced by our method, we can repair the motion curves of these markers in Arena, such as the one shown in Figure 3, to visualize how it affects the motion tracking along with the rigid body engine and inverse kinematics (IK) model in real time. Images in the bottom panel of Figure 4 show the reconstructed motion of our method, comparing to the original recorded motion in the first panel of Figure 4. One can notice that after flittering, the point cloud defining the right upper arm are positioned better than before flittering, because predicted marker's position (in grey color) computed by the rigid body engine overlays with our computation result, as shown in the motion images at the frame 1984 and 1956. With inverse kinematic model embedded in the Arena, the orientation of the parent joint will affect the orientation of the children joints. Therefore, when the joint of the right upper arm is corrected due to the new motion curve and the rigid body matching, the joint of the right lower arm reduces its degrees of freedom due to IK, meanwhile it is also constrained to match the point cloud due to the rigid body engine. With more constraints, the children joints have better chance of finding the right orientation due to the correct orientation of the parent joints. An example is the motion image at the frame 2126 in Figure 4. In the original tracking, because one point on the right upper arm is far to the left, it results the defined right upper arm hanging over, hence the connected lower arm is handed up too. After reconstruction, the right upper arm is located to match our prediction, hence the connected lower arm matches its point cloud better as well without modifying any motion curves of markers on the right lower arm.

5. Conclusion

This paper describes a method related to the problem of missing marker in the optical motion capture system. We take advantage of our motion capture device OptiTrack and its software Arena which can automatically compute the skeleton and track motions from markers attached on the human body. We preset a method of predicting marker's position, along with rigid body tracking and inverse kinematic model implemented in Arena, we are able to reconstruct motion from new marker's state. In the experiment section we demonstrate how our method can effectively predict motion tracking. Our method are computed in real time using

Kalman filter and constant velocity model. Rigid body engine and inverse kinematic model is also implemented to compute in real time in Arena. This approach works effectively on motion reconstruction when there is occlusion or jigger, rapid changes in marker's tracking. By reconstructing motion curves in Arena along with its rigid body engine and inverse kinematic model, we are able to show that reconstructed motion can reasonable reflect real motion and correct erroneous tracking in the Arena. Future work will introduce more constraints to the rigid body tracking, instead of using rigid body engine in Arena, we will work on our own rigid body engine to improve the rigid body tracking results.

References

[1] Motion Reality golf systems, `http://www.motionrealitygolf.com`.

[2] Vicon Motion System Ltd, `http://www.vicon.com`.

[3] Optical Motion Capture solutions, `http://www.naturalpoint.com/optitrack/`.

[4] ARENA Motion Capture Software, `http://www.naturalpoint.com/optitrack/products/full-body-mocap.html`.

[5] Peak Performance Inc, `http://www.peakperform.com`.

[6]

[7] A. Aristidou, J. Cameron, and J. Lasenby. Real-time estimation of missing markers in human motion capture. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, May 2008.

[8] J. Broeren, K. Sunnerhagen, and M. Rydmark. A kinematic analysis of a haptic handheld stylus in a virtual environment: a study in healthy subjects. *Journal of NeuroEngineering and Rehabilitation*, 4(13), 2007.

[9] J. Cameron and J. Lasenby. A real-time sequential algorithm for human joint localization. In *ACM SIGGRAPH 2005 Posters*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[10] J. christophe Nebel. Keyframe interpolation with self-collision avoidance. In *Eurographics, Springer Computer Science*, pages 77–86. Springer, 1999.

[11] K. Dorfmller-Ulhaas. Robust optical user motion tracking using a kalman filter. In *10th ACM Symposium on Virtual Reality Software and Technology*, 2003.

[12] Y. Fangt, C. Hsieh, M. Kim, J. Chang, and T. Woo. Real time motion fairing with unit quaternions. *Computer-Aided Design*, 30(3):191 – 198, 1998.

[13] J. Hashiguchi, H. Nivomiya, H. Tanaka, M. Nakamura, and K. Nobuhara. Biomechanical analysis of a golf swing using motion capture system. *Proceedings of Annual Meeting of Japanese Society for Orthopaedic Biomechanics*, 27:325–330, 2006.

[14] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *In Proceedings of Computer Animation 2000. IEEE CS*. Press, 2000.

[15] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science*, 20(3):313–41, 2001.

[16] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. A*, 4(4):629–642, Apr 1987.

[17] A. Hornung, S. Sar-Dessai, and L. Kobbelt. Self-calibrating optical motion tracking for articulated bodies. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 75 –82, 2005.

[18] J. Lee and S. Shin. General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics*, 8:119–128, 2002.

[19] G. Liu and L. McMillan. Estimation of missing markers in human motion capture. *Vis. Comput.*, 22:721–728, September 2006.

[20] H. Lou and J. Chai. Example-based human motion denoising. *Visualization and Computer Graphics, IEEE Transactions on*, 16(5):870 –879, 2010.

[21] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 1999.

[22] C. Rose, B. Bodenheimer, and M. F. Cohen. Verbs and adverbs: Multidimensional motion interpolation using radial basis functions. *IEEE Computer Graphics and Applications*, 18:32–40, 1998.

[23] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher. Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20:67–94, April 2001.

[24] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. *ACM Trans. Graph.*, 24:98–117, January 2005.

[25] A. Van Rhijn and J. D. Mulder. Optical tracking and automatic model estimation of composite interaction devices. In *Proceedings of the IEEE conference on Virtual Reality*, VR '06, pages 135–142, 2006.

[26] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The hiball tracker: high-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '99, pages 1–10, New York, NY, USA, 1999. ACM.

[27] G. F. Welch. History: The use of the kalman filter for human motion tracking in virtual reality. *Presence*, 18(1):72–91, Feb 2009.

[28] D. Wiley and J. Hahn. Interpolation synthesis of articulated figure motion. *Computer Graphics and Applications, IEEE*, 17(6):39 –45, 1997.

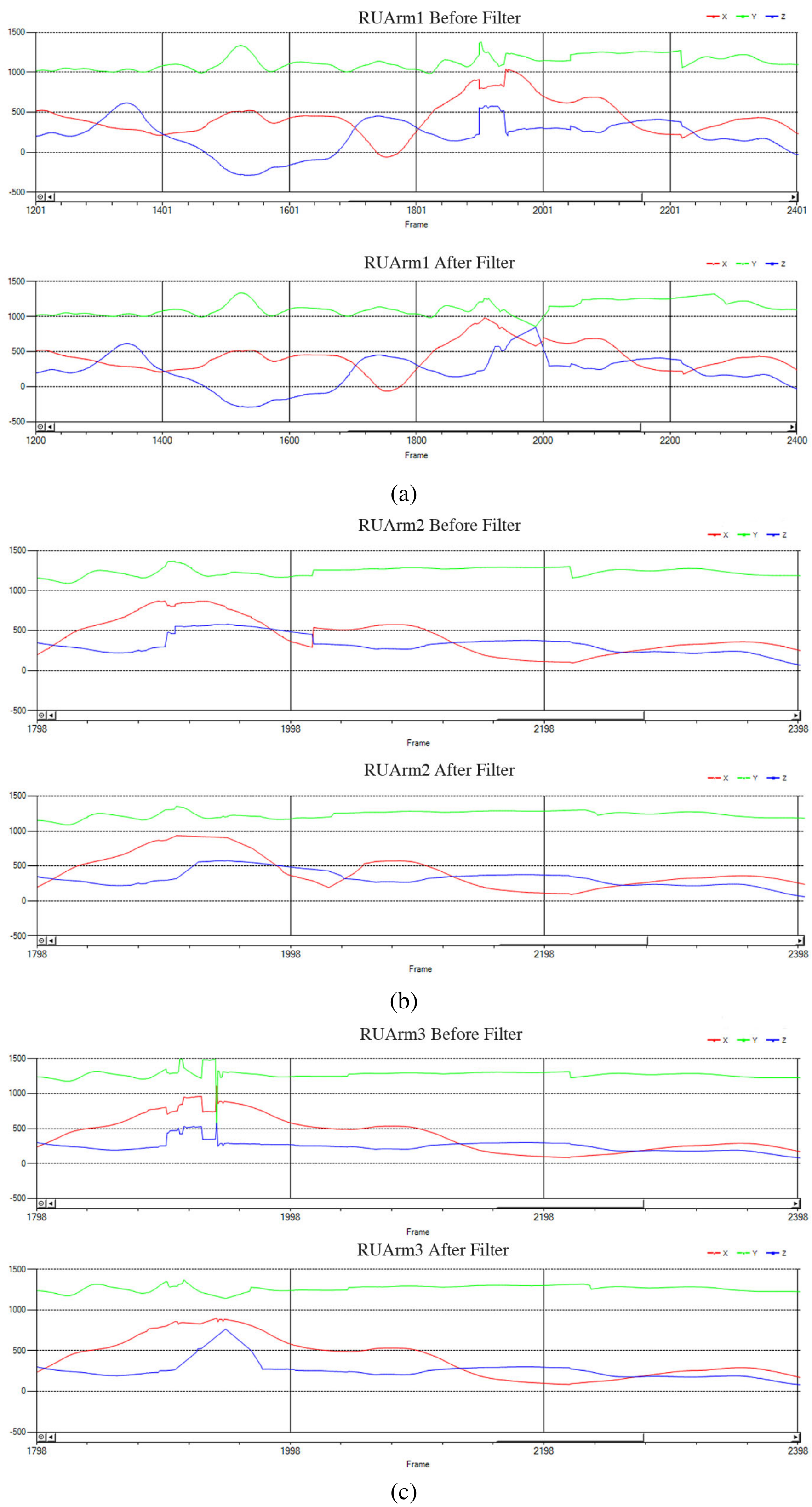

Figure 5. Comparison of marker's tracking before and after filtering. (a) RUArm1: the 1st marker on the right upper arm. The marker is missing between frames 1800-2000. (b) RUArm2: the 2nd marker on the right upper arm. The marker is missing around the frame 1900 and right after the frame 2000. (c) RUArm3 : the 3rd marker on the right upper arm. The marker is missing between frame 1880-1960.