

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235974158>

# Predicting Missing Markers in Real-Time Optical Motion Capture

Conference Paper *in* Lecture Notes in Computer Science · November 2009

DOI: 10.1007/978-3-642-10470-1\_11 · Source: DBLP

CITATIONS

13

READS

132

4 authors, including:



[Andreas Kunz](#)

ETH Zurich

214 PUBLICATIONS 1,522 CITATIONS

[SEE PROFILE](#)



[Morten Fjeld](#)

Chalmers University of Technology

165 PUBLICATIONS 1,683 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PhD Ralf Moryson [View project](#)



Cognitive Complexity [View project](#)

# Predicting Missing Markers in Real-Time Optical Motion Capture

Tommaso Piazza<sup>1</sup>, Johan Lundström<sup>1</sup>, Andreas Kunz<sup>2</sup>, and Morten Fjeld<sup>1</sup>

<sup>1</sup>t2i Lab, CSE Chalmers University of Technology, Rännvägen 6B  
412 96 Göteborg, Sweden  
{piazza,fjeld}@chalmers.se  
www.t2i.se

<sup>2</sup>ICVR, Swiss Federal Institute of Technology, Tannenstrasse 3  
8092 Zurich, Switzerland  
kunz@iwf.mavt.ethz.ch  
www.icvr.ethz.ch

**Abstract.** A common problem in optical motion capture of human-body movement is the so-called missing marker problem. The occlusion of markers can lead to significant problems in tracking accuracy unless a continuous flow of data is guaranteed by interpolation or extrapolation algorithms. Since interpolation algorithms require data sampled before and after an occlusion, they cannot be used for real-time applications. Extrapolation algorithms only require data sampled before an occlusion. Other algorithms require statistical data and are designed for post-processing. In order to bridge sampling gaps caused by occluded markers and hence to improve 3D real-time motion capture, we suggest a computationally cost-efficient extrapolation algorithm partly combined with a so-called constraint matrix. The realization of this prediction algorithm does not require statistical data nor does it rely on an underlying kinematic human model with pre-defined marker distances. Under the assumption that human motion can be linear, circular, or a linear combination of both, a prediction method is realized. The paper presents measurements of a circular movement wherein a marker is briefly lost. The suggested extrapolation method behaves well for a reasonable number of frames, not exceeding around two seconds of time.

**Keywords:** Extrapolation, linear approximation, nonlinear approximation.

## 1 Introduction

Motion capture is the process of recording movement and translating sampled data onto a digital model. It is typically used in military, entertainment [1], sports [2], and medical applications [3]. For instance, in real-time computer-aided surgical systems with graphics, image-forming devices can provide closed-loop feedback between the real environment and the tracking system [17]. However, the captured data can also be transformed into characters and modified completely during post-production. In this paper we focus on improving the usage of real-time tracked data of human-body movement. Hence, we propose an algorithm that requires neither a simplified geometry [19] of a human (kinematics) nor a huge amount of statistical data.

Motion capture can either be inertial, mechanical, magnetic, or optical. In optical motion capture - which is the focus of this work - cameras are used together with active or passive infrared-reflecting markers. Among the advantages of optical motion capture is that the performer feels free to move since no cables connect the body to the equipment. Also large physical spaces can be covered, more performers can be included, and clean, detailed data is yielded [15]. Deficits of optical tracking are that it is prone to light interference and that reflective dots can be blocked by performers or other structures, causing temporary loss of data, also called occlusion. In order to overcome this problem, various approaches to close the data gaps in which no measurements are available were proposed. While some approaches only work off-line, others have real-time capability. These approaches typically use statistical data or some simplified rigid body assumptions for approximating human behavior [4].

The following section presents related work in the area of off-line and real-time methods, followed by a section showing the methodological basis for our experimental investigations. Then come two empirical sections each showing the experimental set-up and presenting gathered data from a circular movement. This is followed by a short section on applications in the area of human-body movement, and a final conclusion and outlook section.

## 2 Related Work and Requirements

Some researchers make use of knowledge about the tracked objects. Molet et al. [16] suggest a method to achieve real-time conversion of sensor measurements from anatomical human rotations. Their method requires only simple calibration, allows for sensors slippage, and takes advantage of knowledge of the type of motion being performed. As we aim for a method that can be benchmarked with various types of motion, we prefer not to incorporate knowledge about those types of motion into our method. This is the first requirement (R1).

Our method should be sufficiently generic to work with most real-time motion capture systems. We are aware that some researchers offer related solutions utilizing a specific camera. Aliaga et al. [18] suggest the Occlusion-Resistant Camera (ORC) to acquire active environments despite the presence of so-called moving interfering occluders. While real-time solutions to the missing marker problem may include the camera, we do not deem camera-integration to be feasible at this stage of our project. That is, our method should be sufficiently generic to work with most real-time motion capture systems. This is the second requirement (R2).

Most companies make real-time motion capture systems with passive targets and reflective markers. The markers are illuminated by infrared light synchronized with high-speed 2D digital imaging systems that can simultaneously image multiple targets. In the work presented here, we used a Qualisys Motion Capture System. Since the issue of missing markers is relatively well researched, a number of methods have been proposed. However, most of these methods are either off-line procedures to be used in post-processing, or they have unsatisfactory prediction accuracy.

Some typical off-line methods interpolate data using linear or non-linear approaches [5] [6] [7]. However, these approaches cause noticeable latencies and cannot be used in real-time applications. Another off-line approach is presented in [8], in which a geometric skeleton structure is automatically generated in order to bridge

existing gaps in the measurement series. Herda et al. [9] describe a real-time method in which an anatomical human model is used to predict the position of the markers. However, this model is too complex for envisioned integration into the human modeling and simulation programs, e.g. ‘Jack’ [10], which already offer a sophisticated anatomical human model. Hornung et al. [11] rely on the fact that the markers on a limb are separated by fixed distances. Finally, [4] uses Kalman filters for predicting marker positions, which are relatively simple and take measurement noise into consideration. However, they require statistical data about the noise and its covariance. As we consider the use of Kalman filters to be a well-examined approach, we will not research it any further here.

### 3 Methodology

In principle, there are two approaches to closing the gaps resulting from missing marker positions: interpolation and extrapolation. For the interpolation, special algorithms such as Catmull-Rom splines [12] must be used since they go through all measurement points being used as control points. However, such an interpolation requires future measurements also, otherwise it can only be used in a post-processing step after all data is available. Extrapolation, on the other hand, does not depend on future information, but the accuracy of an extrapolated point strongly depends on how it is determined from the existing measurements.

In this paper, we present a method of extrapolating a missing point based on previously measured points assuming the most common motion behaviors, such as linear or circular movement. Using a so-called matching factor (see equations 11–12) and the acquired measurements, our algorithm determines whether a missing point should be on a circular or linear extrapolated curve. After describing the algorithm, the paper concludes with first measurement results indicating the performance of the algorithm.

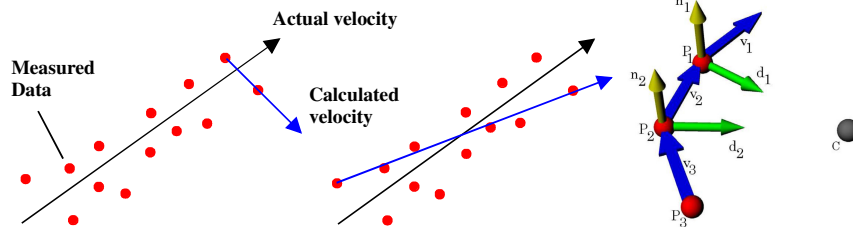
#### 3.1 Prediction

All prediction is typically based on the Taylor series:

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

The further the prediction goes into the future, the more derivatives are available. Typically, only the position is available at discrete points in time, and thus the derivatives can only be obtained by a numerical differentiation. However, the higher derivatives become too noisy to be used without any additional filtering, so we use position and velocity only. To further improve accuracy, we do not utilize consecutive measurements, but, rather, from a distance of 30 frames in the system’s backlog (Fig. 1). This means that the velocity is calculated over the last 250 ms.

However, using equation (1) with the terms for position and velocity would only allow a linear extrapolation. In order to predict more complex motions as well, a simple but effective approach was used. The assumption is that for short time intervals – and thus also for short periods in which a marker is not visible – a human motion can be approximated by linearly combining two predefined movements: linear and circular. Curvatures are also possible, but these are not covered by the truncated Taylor series.



**Figs. 1 (left and center) and 2 (right).** Fig. 1. Velocity calculated from the last two measured points (left) and from the beginning and the end of a data set (center). Red dots represent measured coordinates (positions) of a marker over time, while the arrow marked “actual velocity” shows the actual path of the marker. When using measured coordinates with a very short difference in time, the calculated velocity can differ from the true velocity greatly (left). To improve the accuracy of the calculated velocity, measured coordinates with a greater time difference were used (center). Fig. 2. Calculation of the center of rotation (C).

Instead of estimating a line and a circle that best describe the path of the marker using the least square methods, a missing marker’s position is calculated using three evenly spaced samples from the tracking system’s backlog. Index 1 denotes the latest (newest) frame in the backlog, Index 2 the intermediate one, and, Index 3, the older one.  $P_1$ ,  $P_2$ , and  $P_3$  denote the positions, while  $\vec{v}_1$ ,  $\vec{v}_2$ , and  $\vec{v}_3$  denote the velocities.

**Linear case.** The algorithm approximates a line into the historical data that would be extended to predict future coordinates. For this case, it is assumed that a missing marker will move with the same velocity as the latest/newest one. Thus, it is:

$$P = P_0 + \vec{v}_1 * dt \quad (2)$$

Where:

$$P_0 = (P_1 + P_2 + P_3) / 3 \quad (3)$$

**Circular case.** For a circular movement, four variables must be known:

1. Center of rotation (Center of the circle)
2. Axis of rotation (normal of a plane that is defined by two vectors)
3. Radius of the rotation
4. Rotational velocity  $\omega$

Assuming that the motion is non-linear, the vectors  $\vec{v}_1$ ,  $\vec{v}_2$ , and  $\vec{v}_3$  are not parallel (Fig. 2). Each pair of vectors defines a plane with the normal vectors:

$$\vec{n}_1 = \vec{v}_1 \times \vec{v}_2 \quad \text{and} \quad \vec{n}_2 = \vec{v}_2 \times \vec{v}_3 \quad (4)$$

Better approximations to the actual velocities in  $P_1$  and  $P_2$  are  $(\vec{v}_1 + \vec{v}_2)/2$  and  $(\vec{v}_2 + \vec{v}_3)/2$ . Using these, we can find two vectors,  $\vec{d}_1$  and  $\vec{d}_2$ , both pointing approximately towards the center of rotation (Fig. 2).

$$\vec{d}_1 = \left( \frac{\vec{v}_1}{2} + \frac{\vec{v}_2}{2} \right) \times \vec{n}_1 \quad \text{and} \quad \vec{d}_2 = \left( \frac{\vec{v}_2}{2} + \frac{\vec{v}_3}{2} \right) \times \vec{n}_2 \quad (5)$$

The center of rotation,  $C$ , can be found at the point where the two vectors intersect. Using the line distance algorithm, we find two points,  $H$  and  $Q$  that approximate  $C$ :

$$C = (H + Q) / 2 \quad (6)$$

Finally, the radius,  $r$ , of the rotation is obtained by:

$$r = |P_1 - C| \quad (7)$$

**Linear or circular case?** In this next step, it has to be determined how a line or a circle approximates the measured points in the backlog. Using the least square method, the resulting errors for the line or the circle are calculated.

In the case of a line, the deviation is the shortest distance between one of the samples,  $P_1, P_2$ , or  $P_3$ , and the line,  $P_0 + \bar{v}_1 * t$  as found using the following equation:

$$d_i = \left| (P_i - P_0) - \left( \frac{\bar{v}_1}{|\bar{v}_1|} * (P_i - P_0) \cdot \frac{\bar{v}_1}{|\bar{v}_1|} \right) \right|, \quad i \in [1, 3] \quad (8)$$

From this, the ‘line error’ can be calculated as:

$$e_L = \sum_{i=1}^3 \left[ P_i - P_0 - \frac{\bar{v}_1}{|\bar{v}_1|} * (P_i - P_0) \cdot \frac{\bar{v}_1}{|\bar{v}_1|} \right]^2 \quad (9)$$

In the case of the circle, the resulting ‘circle error’ is:

$$e_C = \sum_{i=1}^3 \left[ |P_i - C| - r \right]^2 \quad (10)$$

To find out which approximation fits best, a matching factor,  $\alpha$ , is defined to be:

$$\alpha = \frac{e_L}{e_L + e_C} \quad (11)$$

Note that:

$$0 \leq \alpha \leq 1 \quad (12)$$

Also,  $\alpha=0$  can be interpreted as ‘100% linear’ and  $\alpha=1$  as ‘100% circular’.

**Estimate of position.** Let  $P_1$  denote the last known position from the backlog, and let  $P'$  denote the extrapolated position that we want to find. The linear estimate is:

$$L' = P_1 + \bar{v}_1 * dt \quad (13)$$

In order to find the circular estimate, we calculate the angular velocity,  $\omega$ .

$$\bar{c}_i = P_i - C \quad (14)$$

The vector:

$$\bar{c}_N = \frac{P_1 - C}{|P_1 - C|} \quad (15)$$

gives the direction,  $P_1$ , relative to  $C$ . Next, an approximation of the rotation axis can be calculated by:

$$\bar{n} = \frac{\bar{n}_1 + \bar{n}_2}{|\bar{n}_1 + \bar{n}_2|} \quad (16)$$

Now, the angular velocity,  $\omega$ , can be calculated by:

$$\omega_1 = a \cos \left( \frac{\bar{c}_2 \cdot \bar{c}_1}{|\bar{c}_2| |\bar{c}_1|} \right) \quad (17)$$

$$\omega_2 = a \cos \left( \frac{\bar{c}_3 \cdot \bar{c}_2}{|\bar{c}_3| |\bar{c}_2|} \right) \quad (18)$$

$$\omega = \left( \frac{\omega_1 + \omega_2}{2} \right) * \text{sgn}((\bar{c}_N \times \bar{n}) \times \bar{v}_1) \quad (19)$$

*sgn* stands for the sign function. The cross product results in a tangent vector of the circle. The dot product in combination with the sign function shows whether the rotation orientation is clockwise or counterclockwise.

Now, the circular estimate can be found using quaternions [14]:

$$q = \left( \cos \frac{\omega}{2}, \bar{n} \sin \frac{\omega}{2} \right); \quad p = (0, P_1 - C); \quad p' = (0, C' - C) = q p \bar{q} \quad (20)$$

where  $q$ ,  $p$ , and  $p'$  are quaternions,  $\bar{q}$  denotes the quaternion conjugate of  $q$ , and  $C'$  is the circular estimate. Finally, the estimated position  $P'$  is the result of a linear interpolation between  $L'$  and  $C'$  using  $\alpha$ :

$$P' = L'(1 - \alpha) + C' \cdot \alpha \quad (21)$$

To further improve the predicted results, a corrector is introduced. It consists of a constraint matrix (CM) (or inter-marker distance matrix) which stores the minimal and maximal recorded pairwise distances between all markers. All markers ( $n$ ) are identified by an index  $1 \dots n$ . The record of two markers, say  $a$  and  $b$ , can be found by looking up the corresponding row and column. Note that the entry ( $a$ ,  $b$ ) describes the same relation as entry ( $b$ ,  $a$ ) and thus the matrix is symmetric.

Another optimization was to remove all correlations between markers that have large differences between their minimum and maximum measured distances. This was done by setting *min* and *max* to negative and positive infinity, while the difference  $\text{max} - \text{min} > \text{threshold}$ . An appropriate value of this threshold is 10 to 20 centimeters.

Hence, the matrix stores all rigid relations (like a marker on the elbow compared to one on the shoulder) and ignores all non-rigid relations.

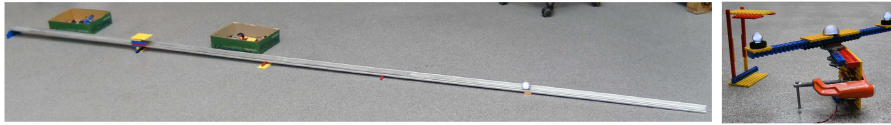
## 4 Measurements without Constraint Matrix (WOM)

The system is built upon the assumption that the trajectories are either linear, circular, or a combination of both. The prediction is a linear combination of both, as shown in (21). After an experiment with a linear test, we set up a circular test. The data presented here is based on the circular test. The setup consisted of the Qualisys Motion Capture System [13]. The frequency of the cameras was 120 frames per second (FPS).

### 4.1 Linear and Circular Setup

The linear test used a sloped groove, in which a spherical marker was rolled, thus providing an almost linear trajectory (Fig. 3). At one point along the groove, the marker was hidden from the camera in order to apply our algorithm. Note that the sloped groove also allowed some unwanted movements to the side in the  $y$ -direction.

We also realized a setup to measure a circular motion of the marker. The marker was hidden from the camera at a certain point of the circle (Fig. 4). Due to the simplified setup, the movement is not an ideal circular motion in the horizontal plane only.

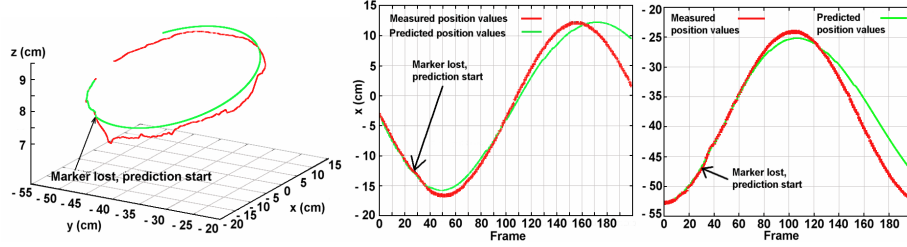


**Figs. 3 (left) and 4 (right).** Setups for the linear and circular measurements.

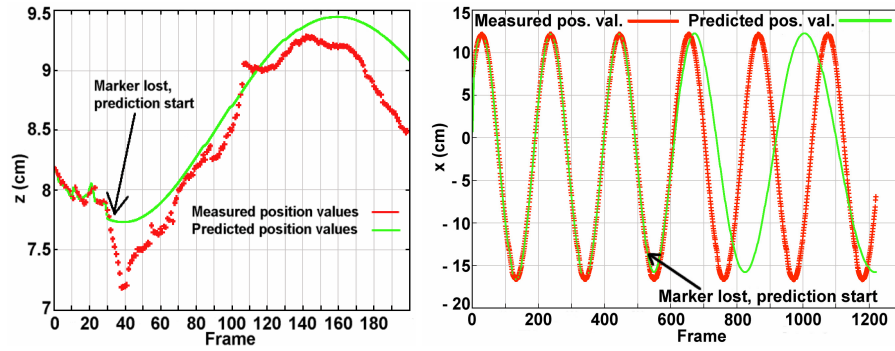
### 4.2 Measurement Results

To examine the performance of the proposed algorithm for a typical case, we show the results for the (almost) circular movement of a single marker. Data is predicted as soon as the marker disappears. The prediction is based on the 30 latest frames stored in the system's backlog. To determine how many frames the prediction algorithm is useful for, the prediction continues even after the marker reappears. While most of the graphs show a reduced data set, i.e. only frames 500 – 700 (Figs. 5 – 8). Fig. 9 shows the complete data set. The marker disappeared at frame 531 (or 31) and re-appeared in the next frame. The prediction algorithm was then triggered. Before the missing data frame, predicted data is a copy of measured position values. The three-dimensional plot in Fig. 5 shows that when the marker disappears, the prediction estimates a fairly circular movement. This indicates that the  $\alpha$ -coefficient used to combine linear and circular movement is close to, but not exactly equal to one due to an imperfect setup. While an absolutely circular movement of the marker corresponds to a constant frequency, a linear movement corresponds to a frequency equal to zero. Thus, an elliptic movement is a superposition of a constant frequency and a constant component. Due to the recursive character of the prediction, the constant value is superimposed again





**Figs. 5-7 (left to right measured (red) and predicted (green) data, arrow is where marker disappeared).** Fig. 5. Circular movement; shown in three dimensions. 200 frames at 120 FPS are shown. Fig. 6. x-axis of the circular movement. The arrow indicates frame 31, where the marker disappeared. Fig. 7. y-axis of the circular movement.



**Figs. 8 (left) and 9 (right): Measured (red) and predicted (green) data.** Fig. 8. z-axis of the circular movement. Fig. 9. x-axis of the circular movement over the complete experiment. The marker disappeared at frame number 531.

and again. This explains the observed growing cycle duration (Figs. 6 – 8), which becomes even clearer in the complete data set (Fig. 9). In the horizontal plane (Figs. 6 – 7), the prediction algorithm works well for 150 frames corresponding to 125 ms of missing data. Along the vertical axis (Fig. 8), it becomes evident that the  $\alpha$ -coefficient correctly mediates between the linear and circular estimation of the movement.

For the results shown here, without the constraint matrix, the algorithm continued to work also after the missing marker reappeared.

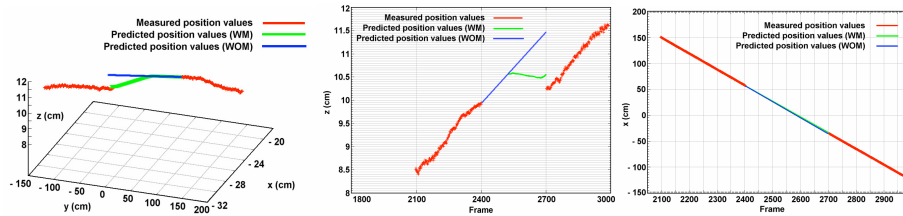
## 5 Measurements with Constraint Matrix (WM)

We developed a filter that stores min/max values of distances sampled between each (n) marker and all other (n-1) markers. This gives an  $n \times n$  matrix, where each cell contains a min-max tuple of two floats, initially set to  $\min = \inf.$  and  $\max = 0$ . For each frame and each marker visible, the matrix is updated. This matrix then serves as a filter in the case of a missing marker. After a first estimation of the missing marker's position, this estimate can be adjusted using the matrix. For each missing marker in a

frame, any other (n-1) marker is used as a reference marker if the difference (here: 10 cm) between the min and max distance is below a predefined threshold. If it is above, the markers are considered to be independent; hence the constraint is marked to be invalid. For each reference marker, the estimated position of the missing marker is successively adjusted towards each reference marker. If a reference marker itself is a missing marker, the adjustment is halved.

### 5.1 Linear Setup

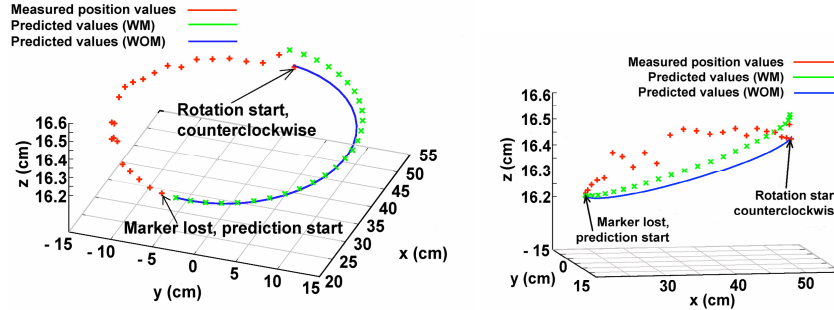
Figs. 10, 11, and 12 are generated from the data collected in the case of an almost linear movement. Fig. 10 shows the movement from the right to the left. The movement was generated using a car carrying three markers. There was a marker on each side and on the top. The latter disappeared at frame 2403 and reappeared at frame 2703, being lost for a total of 327 frames, which is an unusually long time. In Fig. 11, at frame 2524 the two prediction modes start to output significantly different values. While the prediction without the constraint matrix (WOM) gives a steep rising trajectory based upon the last 30 frames (see straight line in Figure 11), the prediction with the matrix (WM) is constrained by the other visible markers (see curved line in Figure 11). Once the missing marker reached the maximum allowed position by the constraints, it re-approached the other markers. This allows detecting and also abrupt changes in the movement and correcting the trajectory (see Figure 10). This behavior cannot be seen in Fig. 12 since the car moved in a more ideal straight line along the x-axis rather along the other axes.



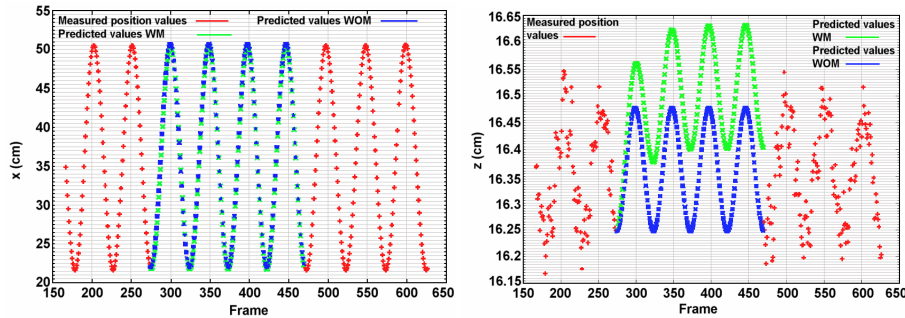
**Figs. 10-12 (left to right, measured (red) and predicted (green) data).** Fig. 10. Linear movement in all three dimensions: the direction of movement is from right to left. Fig. 11. Linear movement along the z-axis. Fig. 12. Linear movement along the x-axis.

### 5.2 Circular Setup

In this test, with its set-up shown in Figure 4, two markers are used. The first marker was placed in the center of rotation while the second was placed on the end of one of the spinning arms. At frame 274 the second marker was lost and the prediction started. The second marker reappeared at frame 470, making for a gap of 196 frames. In Figs. 13 and 14, one complete spin can be observed. In Figs. 15 and 16, the complete circular experiment is depicted. The predicted values with (WM) and without (WOM) the constraint matrices are shown. In Fig. 15, the behavior of both prediction modes seems accurate. Then, in Fig. 16, the behavior along the z-axis shows how the



**Figs. 13 (left) and 14 (right).** Measured (red), predicted (WM) (green), and predicted (WOM) (blue) data. Fig. 13. Circular movement. Fig. 14. Same as in Fig. 13, but from a different angle. Now it is more evident where the predictions (WM) and (WOM) start to differ.



**Figs. 15 (left) and 16 (right).** Measured (red), predicted (WM) (green), and predicted (WOM) (blue) complete circular trajectory data. Arrows indicate start of rotation (right) and marker disappearance (left). Fig. 15. x-axis, algorithms (WM)/(WOM) behave correctly. Fig. 16. z-axis, prediction with (WM) affected by central marker, vibrating along z-axis.

prediction mode with the constraint matrix (WM) is influenced by the position of the central marker, which vibrated along the z-axis.

## 6 Human-Body Movement

The proposed method (WM) was also applied to a less predictable movement such as that of a walking person. Here, seven markers were attached to one side of a person's shoulder, elbow, hand, hip, knee, ankle, and foot. The missing marker was chosen to be the one on the ankle in the moment in which the foot approaches and points to the ground and the knee moves forward to generate the successive step. The marker is supposed to move downward until the foot has touched the ground and then stay still until the foot is lifted from the ground and moved forward. The prediction was accurate enough to fake a visually satisfactory human kinematic for the ankle.

## 7 Conclusions and Outlook

While we avoided making use of knowledge about the tracked object (R1), we have still not tested whether the results scale other experimental platforms (R2). The results showed that the suggested algorithm (WOM) delivers valuable data for up to 150 ms of occlusion. For longer periods, prediction may decrease in accuracy. This is due to the  $\alpha$ -coefficient that combines linear and circular movements (11). Hence, an initially small linear or circular component may increasingly cause predicted data differences. However, under normal conditions it is quite unusual that markers disappear for a longer period of time. Thus, the proposed algorithm copes with most of the missing marker problems. To further improve the prediction, a constraint matrix (WM) was introduced, storing all inter-marker distances that are above a certain threshold. Now, missing markers can be predicted more easily. If no reference marker is visible, the matrix can not correct the predicted trajectory for the missing markers and the only effect is to keep all missing markers in a geometrical context to each other.

## References

1. Menache, A.: Understanding Motion Capture for Computer Animation and Video Games. Morgan Kaufmann Publishers Inc., San Francisco (1999)
2. Hashiguchi, J., Nivomiya, H., Tanake, H., Nakamura, M., Nobuhara, K.: Biomechanical Analysis of a Golfswing using Motion Capture System. *Proceedings of the Annual Meeting of Japanese Society for Orthopaedic Biomechanics* 27, 325–330 (2006)
3. Broeren, J., Sunnerhagen, K.S., Rydmark, M.: A Kinematic Analysis of a Handheld Stylus in a Virtual Environment: A Study in Healthy Subjects. *Journal of NeuroEngineering and Rehabilitation* 4(13) (2007)
4. Aristidou, A., Cameron, J., Lasenby, J.: Real-Time Estimation of Missing Markers in Human Motion Capture (2008)
5. Wiley, D., Hahn, J.: Interpolation Synthesis of Articulated Figure Motion. *IEEE Computer Graphics and Applications* 17(6), 39–46 (1997)
6. Nebel, J.: Keyframe Interpolation with Self-collision Avoidance. In: *Eurographics*, pp. 77–86. Springer Computer Science, Heidelberg (1999)
7. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and Adverbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics and Applications* 18(5), 32–40 (1998)
8. Rhijn, A., Mulder, J.: Optical Tracking and Automatic Model Estimation of Composite Interaction Devices. In: *Proceedings of IEEE Virtual Reality Conference*, pp. 135–142 (2006)
9. Herda, L., Fua, P., Plänkers, R., Boulic, R., Thalmann, D.: Skeleton-based Motion Capture for Robust Reconstruction of Human Motion. In: *IEEE Proceedings of Computer Animation*, p. 77 (2000)
10. Siemens: Jack Human Modeling and Simulation, [http://www.plm.automation.siemens.com/en\\_us/products/tecnomatix/assembly\\_planning/jack/index.shtml](http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/jack/index.shtml) (accessed 4/2/2009)
11. Hornung, A., Sar-Dessai, S.: Self-calibrating Optical Motion Tracking for Articulated Bodies. In: *Proceedings of IEEE Virtual Reality Conference 2005*, pp. 75–82 (2005)

12. Catmull, E., Rom, R.: A Class of Local Interpolating Splines. In: Barnhill, R.E., Reisenfeld, R.F. (eds.) *Computer Aided Geometric Design*, pp. 317–326. Academic Press, London (1974)
13. Qualisys Inc., <http://www.qualisys.com/> (accessed 5/2/2009)
14. Hanson, A.: Visualizing Quaternions. In: *ACM Siggraph 2005 Course* (2005)
15. Furniss, M.: Motion Capture. Massachusetts Inst. of Technology Comm. Forum (2005)
16. Molet, T., Boulic, R., Thalmann, D.: A Real-Time Anatomical Converter for Human Motion Capture. In: Boulic, R., Hegron, G. (eds.) *Eurographics Workshop on Computer Animation and Simulation* 96, pp. 79–94 (1996)
17. Welch, G., Foxlin, F.: Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *IEEE Computer Graphics and Applications*, 24–38 (2002)
18. Aliaga, D.G., Xu, Y., Popescu, V.: Occlusion-Resistant Camera Design for Acquiring Active Environments. In: *IEEE Computer Graphics and Applications*, pp. 68–78 (2007)
19. Hornung, A., Sar-Dessai, S., Kobbelt, L.: Self-calibrating Optical Motion Tracking for Articulated Bodies. *Proceedings of IEEE Virtual Reality 2005*, 75–82 (2005)