Estimating missing marker positions using low dimensional Kalman smoothing

M. Burke[a,b,*], J. Lasenby[a]

[a] University of Cambridge, Cambridge, UK, CB2 1PZ
[b] Mobile Intelligent Autonomous Systems, Council for Scientific and Industrial Research, Pretoria, South Africa

Abstract

Motion capture is frequently used for studies in biomechanics, and has proved particularly useful in understanding human motion. Unfortunately, motion capture approaches often fail when markers are occluded or missing and a mechanism by which the position of missing markers can be estimated is highly desirable. Of particular interest is the problem of estimating missing marker positions when no prior knowledge of marker placement is known. Existing approaches to marker completion in this scenario can be broadly divided into tracking approaches using dynamical modelling, and low rank matrix completion. This paper shows that these approaches can be combined to provide a marker completion algorithm that not only outperforms its respective components, but also solves the problem of incremental position error typically associated with tracking approaches.

Keywords: Motion capture, Missing Markers, Kalman filter, SVD

1. Introduction

Motion capture is a frequently used tool in biomechanics, typically relying on multiple cameras to track markers placed on joints or limbs of interest. Unfortunately, occlusions and marker detection failures often result in a number of missing markers. Missing markers are common in motion capture applications,

[*] Corresponding author

and typically result in a large amount of time being spent manually correcting marker trajectories before any in-depth analysis can occur.

The traditional approach to solving this problem uses manual marker correction (sometimes using spline or linear interpolation) or relies on skeleton fitting (Herda et al., 2000). The former is typically inaccurate and only suitable for short occlusion durations, while the latter usually requires specific marker placement and limits researchers to standard skeleton rigs. More recently, a number of studies (Xiao et al., 2011; Lai et al., 2011; Tan et al., 2013) have shown how missing marker positions can be estimated by using matrix factorisation techniques. While these can be effective, they are sometimes slow, and implementations often inaccessible to many biomechanics practitioners.

Tracking approaches that use dynamical motion models and temporal information to fill in missing trajectories have also been proposed previously, but these are often disregarded due to potential difficulties in designing dynamical models and concerns about efficacy. Wu and Boulanger (2011) use a Kalman filter together with a constant velocity motion model to estimate marker positions, but this approach is extremely susceptible to drift. Unfortunately, this latter behaviour has led a number of works (Feng et al., 2014; Federolf, 2013; Xiao et al., 2011; Baumann et al., 2011) to disregard Kalman filtering, under the misconception that this behaviour applies regardless of the dynamical model used.

Various approaches that attempt to learn a dynamical model while estimating marker positions have been proposed (Li et al., 2009), but are typically too slow to be of practical use.

Liu and McMillan (2006) use a family of low dimensional local linear models trained using multiple prior recordings. When a new sequence is provided, this approach selects an appropriate model for each frame using a random forest classifier, before predicting missing marker positions using the appropriate model. Unfortunately, this approach requires a priori training data, and is only applicable if the markers are placed in fixed, predefined positions that match those used in training.

Aristidou et al. (2008) utilise the fact that the distance between markers on a given limb should remain constant to estimate centres of rotation for limbs, and use centres of rotation tracked using a Kalman filter to infer marker position. Unfortunately, this approach requires that multiple markers are present for each limb, assumes limbs are rigid bodies, and is vulnerable to increasing error as the duration for which markers are missing increases. Tracking using an improved variable turn model and an unscented Kalman filter (Aristidou and Lasenby, 2013) lessens this effect, but still assumes the presence of rigid limbs with at least 3 markers placed on each limb.

Dorfmüller-Ulhaas (2003) track the rotation and translation of rigid bodies using an extended Kalman filter (a linear approximation to a non-linear motion model is used for state predictions and update), assuming a constant angular velocity rotation model and a constant acceleration translation model. Unfortunately, this approach requires specific marker placement for the rigid bodies to be detected, and is also liable to drift. This approach is similar to that described by Cerveri et al. (2003), who used an extended Kalman filter and second order motion model to track joints using 2D image measurements obtained from multiple cameras.

This paper addresses the drift problem in tracking approaches through the introduction of a fast and accurate marker completion algorithm that combines temporal smoothing and matrix factorisation, and which is accessible to the biomechanics community[1]. The primary contributions of this paper are as follows.

- We show that two previously separate families of approaches to marker completion are in fact complimentary and can be combined.

- We show how the most commonly listed flaw in tracking approaches (incremental error when markers are missing for extended periods of time) can be avoided.

[1] Matlab and Python implementations are available at `https://github.com/mgb45/MoGapFill`, together with a Python plugin for Vicon Nexus.

- We provide an approach to marker completion that is completely data driven, requiring no prior knowledge of marker placement or number and making no assumptions about the presence of rigid bodies.

2. Missing marker position estimation

We briefly describe the two primary families of approaches to missing marker completion, before introducing our approach, which combines elements of both.

2.1. Marker smoothing

Knowledge of the expected motion of markers in the form of a dynamical model can be used together with detected markers to estimate marker positions. Let $\mathbf{x}_t$ denote the position vector of markers at time t, and $\mathbf{z}_t$ the set of measured marker positions. Assuming N markers, the state vector is constructed as $\mathbf{x}_t = [x_1, y_1, z_1 \dots x_N, y_N, z_N]^T$. Our goal is to estimate $\mathbf{x}_t$ using measurements $\mathbf{z}_t$.

The Kalman filter (Kalman, 1960) is frequently used for problems like these, as it provides an optimal solution to tracking problems when states are governed by linear Gaussian motion and observation models. Let us assume that a state $\mathbf{x}_t$ evolves as

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_t, \tag{1}$$

with process noise $\mathbf{w}_t$ drawn from a zero-mean Gaussian distribution with process covariance $\mathbf{Q}$, and $\mathbf{F}_t$ a linear transition matrix that describes how states are likely to evolve over consecutive time steps. Further, let us assume that we can obtain measurements $\mathbf{z}_t$, which are related to the state at time t by the equation

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t, \tag{2}$$

with measurement or observation noise $\mathbf{v}_t$ drawn from a zero-mean Gaussian distribution with measurement covariance $\mathbf{R}$, and $\mathbf{H}_t$ a linear measurement matrix that maps measurements to states. The Kalman filter provides an optimal estimate of the state $\mathbf{x}_t$ given a history of measurements up to time t for models of this form. In post-processing applications, the Kalman filter can be extended to the Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965), which provides an optimal estimate of the state $\mathbf{x}_t$ given all measurements in the sequence (see Appendix A for the smoothing recursions).

Unfortunately, it can be hard to design the dynamical model $\mathbf{F}_t$, particularly without prior knowledge of marker placement, expected motion and the relationship between markers. A naive approach would be to model each marker's motion independently, but this failure to account for marker correlations limits the achievable accuracy and results in large errors if markers are missing for prolonged periods of time.

2.2. Low rank matrix completion

Motion capture sequences are typically recorded at an extremely high framerate, and there is often little change in motion over consecutive frames. As a result, the sequences can usually be described in a low dimensional space. This property has led to a number of approaches that try and find missing marker positions by using a low dimensional representation of motion capture sequences to reconstruct the original data.

We briefly illustrate these techniques using a representative approach termed mSVD (Srebro et al., 2003), but there are multiple decompositions that could be used. Let $\mathbf{X}$ denote a $T \times d$ training set, formed by stacking all marker position vectors in a motion capture sequence horizontally. Here, T is the length of the motion capture sequence, while d denotes the dimensionality of the state vector $\mathbf{x}_t$, typically $3N$, where N is the number of markers.

A low dimensional representation of this matrix can be obtained by performing singular value decomposition (SVD), a factorisation of a matrix into the form

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, * denotes a conjugate transpose, and $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of $\mathbf{X}$, all positive and listed in decreasing order (Stewart, 1993). The magnitude of the singular values can be viewed as a measure of a mode's (columns of $\mathbf{U}$) contribution to the matrix

$\mathbf{X}$. A low rank approximation of the matrix $\mathbf{X}$ can be obtained by discarding the modes and basis functions (rows of $\mathbf{V}$) of $\mathbf{X}$, which correspond to singular values of smaller magnitude.

mSVD is an iterative approach that decomposes motion capture data, discards a portion of the basis functions, reconstructs the original data, replaces missing values in the sequence with reconstructed ones, and repeats until convergence. Essentially, this approach uses the low dimensional representation of sequences to find correlations between markers, and uses markers that are present to provide information about those that are missing. Unfortunately, this can be slow and memory intensive, as it relies on multiple decompositions.

3. Low dimensional Kalman smoothing

In previous sections, we introduced smoothing and low rank matrix completion approaches to missing marker problems. The former requires the design of a complex motion model typically utilising knowledge of marker placement, while the latter can be slow and memory intensive due to its iterative nature. Our approach combines the two by projecting markers into a lower dimensional space learned from the sequence, performing Kalman smoothing in this space using a random walk motion model and then returning to the original space, using correlated markers to reduce the average error in each marker position estimate.

In our formulation we let $\mathbf{X}$ denote an $M \times d$ training set, formed by taking M position vectors in a motion capture sequence for which all markers were present and stacking them horizontally. Let $\bar{\mathbf{m}}$ be a vector containing the means of columns of the motion capture matrix $\mathbf{X}$. If $\hat{\mathbf{V}}$ is a linear, $d \times \hat{d}$ projection learned by applying the SVD to mean-shifted data and retaining $\hat{d}$ basis functions using an appropriate energy measure (for example, use a scree plot to choose $\hat{d}$ such that $\gamma = 99\%$ of the energy in the original data is retained upon reconstruction), a full dimensional state sample, $\mathbf{x}_t$, can be projected into a low dimensional subspace to give $\tilde{\mathbf{x}}_t$ via

$$\tilde{\mathbf{x}}_t = \hat{\mathbf{V}}^{\mathrm{T}} \left(\mathbf{x}_t - \bar{\mathbf{m}} \right). \tag{4}$$

Assuming a simple random walk motion model is used in this low dimensional space, the only change required to apply Kalman smoothing is the replacement of the measurement matrix $\mathbf{H}_t$ in Section 2.1 with

$$\hat{\mathbf{H}}_t = \mathbf{H}_t \hat{\mathbf{V}}. \tag{5}$$

This approach has the benefit of performing both temporal smoothing and taking marker correlations into account, while requiring only a single SVD computation, thereby providing a large speed improvement over mSVD. Algorithm 1 describes our approach in more detail, providing the full Kalman smoothing recursions.

Algorithm 1 Low dimensional Kalman smoothing

```
Decompose $\mathbf{X} - \bar{\mathbf{m}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$
$\hat{d}$ = find(cumsum(diag($\mathbf{S}$))) $\geq \gamma$)
$\hat{\mathbf{V}}$ = first $d$ columns of $\mathbf{V}$
$\hat{\mathbf{m}}_0$ = randn(d,1)
$\hat{\mathbf{P}}_0$ = randn(d,d)
for $t = 1$ to $T$ do
    $\hat{\mathbf{m}}_t = \hat{\mathbf{m}}_{t-1}$
    $\hat{\mathbf{P}}_t = \hat{\mathbf{P}}_{t-1} + \mathbf{Q}$
    $\mathbf{K}_t = \hat{\mathbf{P}}_t \hat{\mathbf{H}}_t^{\mathrm{T}} (\hat{\mathbf{H}}_t \hat{\mathbf{P}}_t \hat{\mathbf{H}}_t^{\mathrm{T}} + \mathbf{R})^{-1}$
    $\mathbf{m}_t = \hat{\mathbf{m}}_t + \mathbf{K}_t(\mathbf{z}_t - (\hat{\mathbf{H}}_t \hat{\mathbf{m}}_t + \mathbf{H}_t \bar{\mathbf{m}}))$
    $\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \hat{\mathbf{H}}_t)\hat{\mathbf{P}}_t$
end for
$\tilde{\mathbf{m}}_T = \mathbf{m}_T$
$\tilde{\mathbf{P}}_T = \mathbf{P}_T$
for $t = T - 1$ to $1$ do
    $\tilde{\mathbf{m}}_t = \mathbf{m}_t + \mathbf{P}_t \hat{\mathbf{P}}_t^{-1}(\tilde{\mathbf{m}}_{t+1} - \hat{\mathbf{m}}_{t+1})$
    $\tilde{\mathbf{P}}_t = \mathbf{P}_t + \mathbf{P}_t \hat{\mathbf{P}}_t^{-1}(\tilde{\mathbf{P}}_{t+1} - \hat{\mathbf{P}}_{t+1})(\hat{\mathbf{P}}_t^{-1})^{\mathrm{T}} \mathbf{P}_t^{\mathrm{T}}$
    $\mathbf{y}_t = \hat{\mathbf{V}} \tilde{\mathbf{m}}_t + \bar{\mathbf{m}}$
end for
return $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_T]^{\mathrm{T}}$
```

Kalman smoothing recursions are applied in the low dimensional space, using a matrix $\mathbf{H}_t$ at each time step to map measurements $\mathbf{z}_t$ (detected markers at time t) to states. The measurement matrix $\mathbf{H}_t$ varies depending on the number of measurements obtained at each time step, and is constructed by removing the rows corresponding to missing markers from an $d \times d$ identity matrix. Noise covariance matrix $\mathbf{R}$ is diagonal, with elements selected empirically, while $\mathbf{Q}$ is obtained by determining the standard deviation of the rate of change of marker positions and projecting this into the low rank space. Note that in our application, the use of a random walk motion model means that $\mathbf{F}_t = \mathbf{I}$. The state is also tracked in a lower dimensional subspace, which results in the modified recursions of Algorithm 1.

4. Results

We tested our approach on a challenging marker tracking sequence of an actor performing a fighting sequence, which was obtained from Aristidou (2013). This sequence comprises 25000 frames of 47 markers captured at 480 fps, with existing missing marker positions manually corrected by assuming smooth motion. Figure 1 shows the gaps that were present in the sequence prior to manual correction.

Test data was generated by inserting gaps of various lengths at randomly selected locations for randomly selected markers, with gaps following the distribution shown in Figure 1. This process was repeated 100 times, to generate 100 different test sequences. The test data generation process selected ensures that test sequences contain the error types likely to be seen in real data, while providing a wide variety of test conditions.

Figure 2 shows a box plot of the root mean square error for marker positions when the proposed low dimensional smoothing was applied to the 100 test sequences using a random walk motion model. The results obtained by RTS smoothing using random walk and constant velocity motion models (Wu and Boulanger, 2011), cubic spline fitting, applying mSVD (Srebro et al., 2003) and by unscented Kalman filtering using a variable turn model (VTM) (Aristidou and Lasenby, 2013) are also provided for comparison. The latter assumes that limbs are rigid bodies and requires that at least 3 markers are present per limb.

The low dimensional smoothing took place in a 77 dimensional subspace, while the random walk smoother used a state space of dimension 141 and the

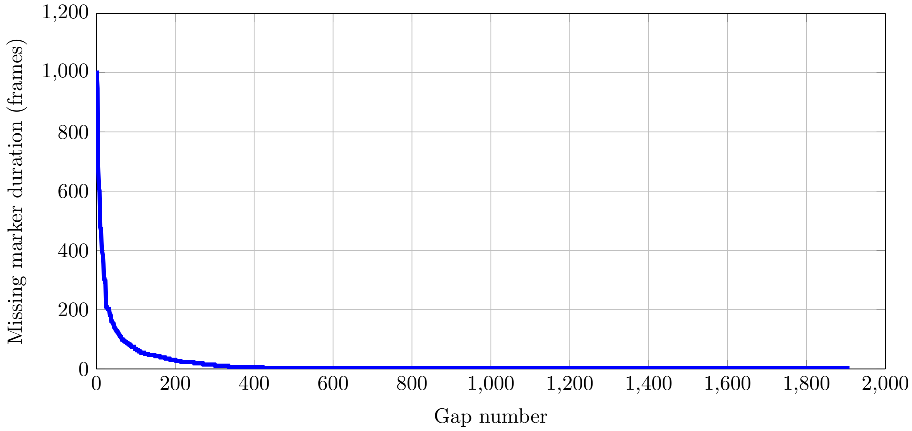

Figure 1: A plot of the gap durations extracted from a real motion capture sequence shows that markers are primarily missing for short periods of time, but that longer gaps are also present. The gaps shown here were inserted into the test sequence at random locations for randomly selected markers. This process was repeated to produce 100 different test sequences, each containing over 1900 gaps.

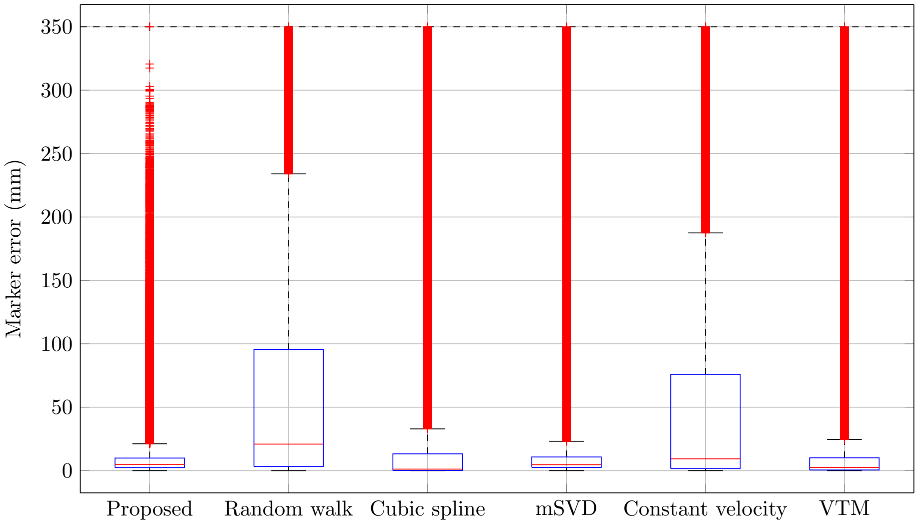

Figure 2: A boxplot shows that the distribution of the marker error at each time instant in the tracking sequences is significantly more compact for the low dimensional smoothing approach, with significantly smaller outlier errors. (Outlier errors larger than 350 mm have been suppressed to aid visualisation)

Table 1: Summary statistics (mm) for the tested approaches

Statistic	Proposed	RW	Cubic spline	mSVD	CV	VTM
Median	4.93	20.91	1.18	4.69	9.32	2.50
Mean	8.36	87.17	25.029	23.81	106.10	10.21
SD	11.42	158.72	83.50	81.58	272.59	23.75
Q3	9.91	95.60	13.24	10.77	75.93	10.13
Max	441.23	3671.01	10240.21	10240.02	6736.50	1664.70

constant velocity smoother required a state space of 282 dimensions.

A Kruskal-Wallis H test showed that there was a statistically significant difference in errors obtained using the different approaches, $\chi^2(2)$=2.5033E6, p=0, with mean ranks 1.0408E7, 1.3755E7, 7.6475E6, 1.0926E7, 1.2270E7 and 8.3967E6. The latter are ordered as in Figure 2. A post-hoc test shows that the mean ranks for each group are significantly different to those of others at a 95 % confidence level. As expected, tracking using the VTM model, which requires knowledge of the marker placement and at least 3 markers placed on each limb, performs extremely well, but surprisingly a cubic spline fit exhibits the lowest mean rank of all the approaches tested. This can be attributed to the gap distribution used for testing, which contains significantly more small gaps. The cubic spline performs well over short durations and as a result exhibits good overall results. However, for longer gaps, the cubic spline performs extremely poorly. Table 1 provides summary results that emphasise this.

It is clear that the proposed low dimensional Kalman smoothing approach significantly reduces the magnitude of outlier errors and results in lower error variability. The more compact marker error distribution is obtained because the filtering action in the low dimensional subspace serves to constrain the error in individual markers, and automatically allows for knowledge of the behaviour of other markers to be leveraged to provide insight into the expected position of missing markers. Smoothing using the constant velocity motion model performs well when markers are missing for only short periods of time, but fails dismally when markers are missing for extended periods of time.

This behaviour can be observed when the marker position error is viewed as a function of the duration of time the marker was missing for (Figure 3). As expected, random walk and constant velocity smoothing are highly susceptible to drift, undergoing incremental position error, and a cubic spline fit is unable to handle large gaps, but the matrix factorisation approaches are more robust to the missing marker duration, as they take the positions of other markers into account when estimating position, thereby limiting the drift in error. The VTM filter performs well as it also takes the position of other markers into account, but requires prior knowledge of marker placement. mSVD and the cubic spline perform similarly because mSVD applies a cubic spline initialisation stage.

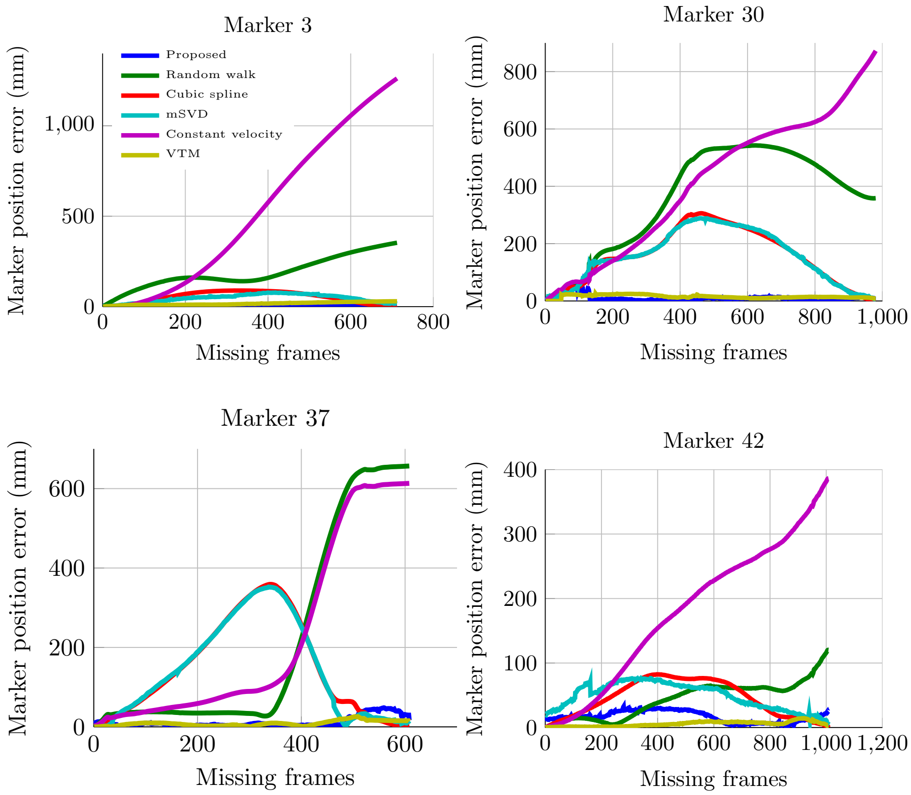

Figure 3: Marker position error as a function of missing marker duration shows that low dimensional smoothing and the VTM filter are less susceptible to drift than the other approaches compared against.

Table 2 shows the processing times for a single test sequence. While not as

Table 2: Comparison of processing times

Algorithm	Duration (25000 frames)
Cubic spline	0.6 s
Proposed	238.0 s
Random walk	376.0 s
mSVD	402.5 s
Constant velocity	626.2 s
VTM	1382.8 s

fast as a simple cubic spline fit, the proposed approach is still suitable for practical use. In terms of computational complexity, Kalman smoothing is dominated by the number of markers visible at each time step (worst case $O(d^3)$), so the processing rate is unlikely to fall below 100 frames per second in most biomechanics applications. It should be noted that while the cubic spline provides low median error and is extremely fast, the large number of outliers (marker position estimates with unusually large errors) produced by this approach make it infeasible for practical use.

5. Conclusion

Missing markers are a frequently encountered problem in studies relying on motion capture. This paper has described a simple and effective approach to missing marker position estimation that combines temporal smoothing and low rank matrix completion to remedy drift related problems typically associated with tracking approaches to marker completion. While we have provided results for the pairing of singular value decomposition and a random walk Kalman smoother, the unified marker estimation approach proposed here can also be applied in alternative linear subspaces, or with modified motion models.

It should be noted that low dimensional smoothing requires a representative set of frames in which all markers were present to learn an appropriate subspace in which to filter, so is unable to track the position of markers that fall off early on in a motion capture sequence. This can be remedied by ensuring that all markers are present for an initial training phase, but is not ideal.The proposed approach is independent of marker placement and motions and does not require a specific skeleton rig. An implementation of our approach is available at `https://github.com/mgb45/MoGapFill` and we hope that this will prove useful within the greater biomechanics community.

Over 75% of the errors obtained using the proposed approach were below 1 cm, a suitable level for many biomechanics applications, but an error of this magnitude can affect gait kinematic estimates significantly (Szczerbik and Kalinowska, 2011). Errors are dependent on the nature of the test sequence and the lengths of gaps in a given test sequence influences the choice of marker completion algorithm. As a result, biomechanics researchers still need to be cognisant of the accuracy required for their studies, and the type of gaps present in their dataset when selecting marker completion algorithms.

Acknowledgements

This work was supported by funding from the Council for Scientific and Industrial Research (CSIR), South Africa and the Cambridge Trusts under a CSIR-Cambridge scholarship. Thanks to Andreas Aristidou from the University of Cyprus, who provided the test data.

References

Aristidou, A., 2013. Marker prediction and skeletal reconstruction in motion capture technology. Tech. rep., University of Cyprus.

Aristidou, A., Cameron, J., Lasenby, J., May 2008. Real-time estimation of missing markers in human motion capture. In: Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on. pp. 1343–1346.

Aristidou, A., Lasenby, J., 2013. Real-time marker prediction and cor estimation in optical motion capture. The Visual Computer 29 (1), 7–26.

Baumann, J., Krüger, B., Zinke, A., Weber, A., 2011. Data-driven completion of motion capture data. In: VRIPHYS. pp. 111–118.

Cerveri, P., Pedotti, A., Ferrigno, G., 2003. Robust recovery of human motion from video using kalman filters and virtual humans. Human Movement Science 22 (3), 377 – 404.

Dorfmüller-Ulhaas, K., 2003. Robust optical user motion tracking using a Kalman filter. In: 10th ACM Symposium on Virtual Reality Software and Technology.

Federolf, P. A., 10 2013. A novel approach to solve the missing marker problem in marker-based motion analysis that exploits the segment coordination patterns in multi-limb motion data. PLoS ONE 8 (10), e78689.

Feng, Y., Xiao, J., Zhuang, Y., Yang, X., Zhang, J. J., Song, R., 2014. Exploiting temporal stability and low-rank structure for motion capture data refinement. Information Sciences 277 (0), 777 – 793.

Herda, L., Fua, P., Plankers, R., Boulic, R., Thalmann, D., 2000. Skeleton-based motion capture for robust reconstruction of human motion. In: Computer Animation 2000. Proceedings. pp. 77–83.

Kalman, R. E., 1960. A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82 (Series D), 35–45.

Lai, R. Y., Yuen, P. C., Lee, K., 2011. Motion capture data completion and denoising by singular value thresholding. Proceedings of Eurographics, Eurographics Association, 45–48.

Li, L., McCann, J., Pollard, N. S., Faloutsos, C., 2009. Dynammo: Mining and summarization of coevolving sequences with missing values. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 507–516.

Liu, G., McMillan, L., 2006. Estimation of missing markers in human motion capture. The Visual Computer 22 (9-11), 721–728.

Rauch, H. E., Striebel, C., Tung, F., 1965. Maximum likelihood estimates of linear dynamic systems. AIAA journal 3 (8), 1445–1450.

Srebro, N., Jaakkola, T., et al., 2003. Weighted low-rank approximations. In: ICML. Vol. 3. pp. 720–727.

Stewart, G. W., 1993. On the early history of the singular value decomposition. SIAM Review 35 (4), 551–566.

Szczerbik, E., Kalinowska, M., 2011. The influence of knee marker placement error on evaluation of gait kinematic parameters. Acta Bioeng Biomech Wroc Univ Technol 13, 43–46.

Tan, C.-H., Hou, J., Chau, L.-P., June 2013. Human motion capture data recovery using trajectory-based matrix completion. Electronics Letters 49 (12), 752–754.

Wu, Q., Boulanger, P., May 2011. Real-time estimation of missing markers for reconstruction of human motion. In: Virtual Reality (SVR), 2011 XIII Symposium on. pp. 161–168.

Xiao, J., Feng, Y., Hu, W., 2011. Predicting missing markers in human motion capture using l1-sparse representation. Computer Animation and Virtual Worlds 22 (2-3), 221–228.

Appendix A. RTS smoothing

This section provides the Rauch-Tung-Striebel smoothing (Rauch et al., 1965) recursions used in this work. Here, a forward prediction of a state mean and uncertainty at time t is made using

$$\hat{\mathbf{m}}_t = \mathbf{F}_t \hat{\mathbf{m}}_{t-1}, \tag{A.1}$$

$$\hat{\mathbf{P}}_t = \mathbf{F}_t \hat{\mathbf{P}}_{t-1} \mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_t, \tag{A.2}$$

under the assumption of a linear motion model $\mathbf{F}_t$ and Gaussian process noise with covariance $\mathbf{Q}_t$. When a measurement $\mathbf{z}_t$ is made the state mean and covariance is updated using

$$\begin{aligned}\mathbf{m}_t &= \hat{\mathbf{m}}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{m}}_t), &\text{(A.3)}\\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{P}}_t), &\text{(A.4)}\end{aligned}$$

with $\mathbf{H}_t$ denoting the mapping between measurements and state, and

$$\mathbf{K}_t = \hat{\mathbf{P}}_t\mathbf{H}_t^{\mathrm{T}}(\mathbf{H}_t\hat{\mathbf{P}}_t\mathbf{H}_t^{\mathrm{T}} + \mathbf{R})^{-1} \tag{A.5}$$

the optimal Kalman gain. $\mathbf{R}$ denotes the uncertainty in measurements. The forward pass Kalman filter estimate $\mathbf{m}_t$ only considers the effect of measurements obtained prior to time t, so the RTS smoother applies a second backward pass through the data to take future measurements into account. The backward recursions

$$\begin{aligned}\mathbf{C}_t &= \mathbf{P}_t\mathbf{F}_t^T\hat{\mathbf{P}}_t^{-1}, &\text{(A.6)}\\ \tilde{\mathbf{m}}_t &= \mathbf{m}_t + \mathbf{C}_t(\tilde{\mathbf{m}}_{t+1} - \hat{\mathbf{m}}_{t+1}), &\text{(A.7)}\\ \tilde{\mathbf{P}}_t &= \mathbf{P}_t + \mathbf{C}_t(\tilde{\mathbf{P}}_{t+1} - \hat{\mathbf{P}}_{t+1})\mathbf{C}_t^{\mathrm{T}}, &\text{(A.8)}\end{aligned}$$

are used to accomplish this.

Note that in our application, the use of a random walk motion model means that $\mathbf{F}_t = \mathbf{I}$. The state is also tracked in a lower dimensional subspace, which results in the modified recursions of Algorithm 1.