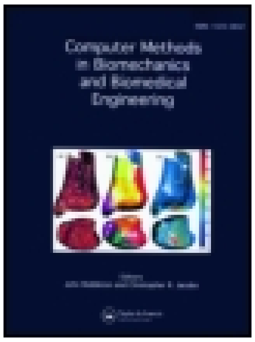

Computer Methods in Biomechanics and Biomedical Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gcmb20

Automated gap-filling for marker-based biomechanical motion capture data

Jonathan Camargo , Aditya Ramanathan , Noel Csomay-Shanklin & Aaron Young

To cite this article: Jonathan Camargo , Aditya Ramanathan , Noel Csomay-Shanklin & Aaron Young (2020): Automated gap-filling for marker-based biomechanical motion capture data, Computer Methods in Biomechanics and Biomedical Engineering, DOI: 10.1080/10255842.2020.1789971

To link to this article: https://doi.org/10.1080/10255842.2020.1789971

Published online: 11 Jul 2020.

COMPUTER METHODS IN BIOMECHANICS AND BIOMEDICAL ENGINEERING
https://doi.org/10.1080/10255842.2020.1789971

Automated gap-filling for marker-based biomechanical motion capture data

Jonathan Camargo[a], Aditya Ramanathan[a], Noel Csomay-Shanklin[b] and Aaron Young[a]

[a]George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA; [b]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

ABSTRACT

Marker-based motion capture presents the problem of gaps, which are traditionally processed using motion capture software, requiring intensive manual input. We propose and study an automated method of gap-filling that uses inverse kinematics (IK) to close the loop of an iterative process to minimize error, while nearly eliminating user input. Comparing our method to manual gap-filling, we observe a 21% reduction in the worst-case gap-filling error ($p < 0.05$), and an 80% reduction in completion time ($p < 0.01$). Our contribution encompasses the release of an open-source repository of the method and interaction with OpenSim.

ARTICLE HISTORY
Received 27 March 2020
Accepted 28 June 2020

KEYWORDS
Motion capture; inverse kinematics; gap-filling; biomechanics

1. Introduction

Motion capture is often used in the analysis of human motion, as well as many other fields, including performance capture for animation and video games, and sports biomechanics (Moeslund et al. 2006; Starck and Hilton 2007; Federolf 2013). Marker-based motion capture, such as that offered by Vicon Motion Systems Limited (XXXX), Qualisys (Qualisys Motion Capture Systems XXXX), OptiTrack (Nagymáté and Kiss XXXX; OptiTrack inc), and other leading motion capture companies, tracks the position of small spherical retro-reflective markers, whereas markerless options, such as Microsoft Kinect (Zhang 2012), use computer vision to extract information about the motion of subjects. Occlusion is a problem that can occur in all of these scenarios due to limbs, clothes, or simply the environment obstructing the camera's view. In marker-based motion capture, this leads to gaps in the marker trajectories that must be filled to obtain the position information and to perform subsequent analysis (Liu and McMillan 2006; Begon et al. 2008; Federolf 2013; Feng et al. 2014).

Traditional methods of gap-filling marker-based motion capture data are time-intensive manual methods where the user must visually inspect each gap and decide how it should be filled with interpolation (Vicon Motion Systems Limited, Fill gaps in trial data, XXXX). The user must subjectively choose the type of interpolation to be used, rather than taking a data-driven approach. These interpolation methods are designed around the assumption that the data present before and after the gap were reconstructed and labeled accurately (Howarth and Callaghan 2010). Some approaches have taken advantage of probabilistic models (Tits et al. 2018), Kalman filters (Aristidou et al., 2008), or information from additional sensors (Bobilev et al. 2012) to help in reconstruction of missing marker data. Newer methods are resorting to the use of machine learning or principal component analysis (Liu et al. 2006; Federolf 2013; Gloersen and Federolf 2016), such as the work by Liu and McMillan (Liu and McMillan 2006) which can recover missing data even if only half the markers are available by first training a classifier to determine principal components, and then using this classifier to recover missing data based on the principal components. One pitfall of such methods is that they require clean and filled data to implement the training stage, which may be difficult to obtain. There are also many matrix based gap-filling methods (Feng et al. 2014; Peng et al. 2015), such as one by Tan et al. (2015), which uses skeleton constrained singular value thresholds. This has the benefit of enforcing kinematic constraints and using existing data for training, so no additional data are needed. Due to the importance of kinematic constraints, other methods have also included skeletal properties (Herda et al. 2001; Begon et al. 2008).

For all of these methods, however, the existing data must be valid, which may not be the case. Marker data may be incorrectly labeled if there are too few cameras to capture the subject markers

CONTACT Jonathan Camargo jon-cama@gatech.edu

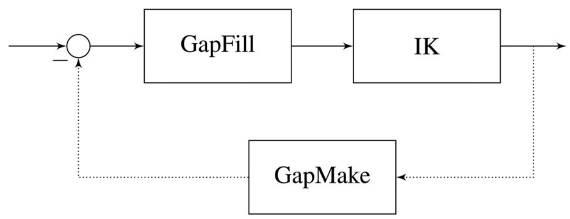

Figure 1. Visualization of data flow.

accurately (Herda et al. 2001; Chiari et al. 2005; Maycock et al., XXXX; Meyer et al. 2014), and using this data to reconstruct missing trajectories will result in more invalid data. Marker data may also be invalid if other reflective objects in the environment are incorrectly registered as markers by the motion capture system. Being able to effectively gap-fill data with errors such as this would require a method to determine where the existing marker data is incorrect so that it can be removed. Then, any traditional gap-filling method could be applied to the removed regions to replace the invalid data. We consider an inverse kinematics (IK) validation that could act as an intermediate step to find regions where certain markers violate kinematic constraints. In these regions, the inverse kinematics marker error will be especially high compared to the surrounding regions, indicating that they were likely captured incorrectly. IK only requires information about marker placement and rigid body relations, which means that it can be processed with no data from other sensors and without any manually filled training data. In regions of invalid kinematics, motion capture data can be adjusted to repair the errors in labels or to correct inaccurately filled marker data (Figure 1).

The contributions of this work are (1) the integration of inverse kinematics to monitor and correct the gap-filling process, (2) evaluating the performance of the method on real datasets in comparison to traditional manual tools, (3) releasing a public repository that contains the tools we developed to process motion capture data. With a simple to use API developed in MATLAB, our automated gap-filling method iterates through the process, removing the need for user intervention, reducing the time needed to fill gaps, and improving the accuracy of the gap-filled results. Unlike other methods of gap-filling, our method yields accurate results even when there are incorrectly labeled regions in the input data by automating the process with OpenSim (Delp et al. 2007; Seth et al. 2018), an open-source musculoskeletal modeling and simulation software, to perform IK and establish such regions. The steps of IK-based marker validation and subsequent gap creation are not present in existing gap-filling methods, and its effectiveness is investigated. The core impact of this work is in the improvement of existing gap-filling methods that rely on intense manual work that, if reduced, would allow researchers in different fields that dependent on motion capture (e.g., gait analysis) to conduct analyses much more quickly and with greater accuracy.

2. Methods

2.1. overview

Analysis of improperly labeled data to calculate joint angles through inverse kinematics (IK) yields large deviations between the kinematic fit and the original marker data wherever the marker data is incongruent with a possible posture. We suggest that large deviations from the output of IK indicate that the data were reconstructed inaccurately, possibly due to improper labeling, invalid gap-filling, or interference from other reflective objects in the motion capture space, and we investigate this claim below. Because of this, trajectories with sufficiently large IK deviations should be removed from those regions. The new marker data with artificial gaps created can be gap-filled again. This process can be iterated until the IK errors converge below some user-set thresholds, or until a maximum iteration limit is reached. This process starts from the motion capture data in a C3D format (C3D. The 3D Biomechanics Data Standard 2018), a standard format supported by major manufacturers of 3 D motion capture systems. Gap-filling is realized by combining interpolation-based methods with information about rigid body segments from an OpenSim model to intelligently select the closest available marker on the same segment for interpolation methods that require information from such additional donor markers. The interpolation methods are implemented in MATLAB, based on openly available descriptions of methods used by Vicon Nexus (Vicon Motion Systems Limited, What Gap Filling Algorithms are Used Nexus 2, XXXX). After each gap-filling process, the gap creation process uses the information output from IK.

Algorithm 1 Iterative Gap-Filling Algorithm

procedure IterativeGapFill
  change ← true
  while change **do**
    markers ← *GapFill*(*markers*)
    markers, change ← *GapMake*(*markers*)
  return *markers*

2.2. Gap-Filling

The gap-filling step uses MATLAB to mimic manual approaches to gap-filling offered by motion capture software. For example, Vicon Nexus (Vicon Motion Systems Limited, What Gap Filling Algorithms are Used Nexus 2, XXXX), supplies various gap-filling algorithms that may be useful in different situations, among which are: spline fill, pattern fill, and rigid body fill. Spline fill uses a cubic or a quintic spline to interpolate missing marker data. Pattern fill uses the trajectory of one donor marker to estimate the trajectory of the missing marker. Rigid body fill uses three or more donors to estimate the trajectory of the missing marker, under the assumption that the donors and the fill trajectory are all part of a rigid body segment.

The donors for rigid body fill are determined through rigid body segment information contained in the OpenSim model of the subject. For a given marker, all other markers in the same segment are considered as potential donors and will be used if they have data during the gap. The donor used for pattern fill is given by the marker on the same segment that has the lowest average distance to the missing marker at the frames before and after the gap.

Rigid body fill uses many donors in calculating the missing trajectory, so it is less sensitive to errors in any one of the donors, whereas pattern fill, using only one donor, will copy any errors in the donor to the missing trajectory. Spline fill, using only the endpoints of the missing trajectory, is highly subject to small errors in the data near the endpoints of the gap, and is best for filling short gaps (Liu and McMillan 2006; Howarth and Callaghan 2010). These methods are implemented in MATLAB to allow for complete automation.

All of these methods work best with shorter gaps, so gaps are filled from shortest to longest, which allows short gaps to be filled and used to fill longer gaps. Rigid body fill is used first wherever possible, and if there are insufficient donors, then rigid body fill will fail, and pattern fill will be tried next. If there are no valid donors for pattern fill, then the algorithm will move on to the next longest gap. Only when the algorithm has tried to fill every gap with rigid body and pattern fill will spline fill be used on the shortest gap, in order to minimize its usage. After this single usage of spline fill, gaps will be recalculated, and the algorithm will again start with rigid body fill on the new shortest gap.

The equations to calculate filled marker positions are reproduced below for the sake of completeness, and to provide more specific details on implementation, although they are based on the openly available methods used in Vicon Nexus (Vicon Motion Systems Limited, What Gap Filling Algorithms are Used Nexus 2, XXXX). For the following discussion, $s = [0, 1, \ldots, n] \subset \mathbf{Z}$ will represent the data sample number during a gap of length $n-2$, and $\tau = \{s_i/n : s_i \in s\}$ will be a parameterization of time over the duration of a gap where $\tau = 0$ is the last time that data exists before the gap, $\tau = 1$ is the first time when data is seen again after the gap. Let $\mathbf{p_{d,i}} \in \mathbf{R}^{m\times 3}$ represent the global x, y, and z coordinates of the m donor markers at $\tau = i$, and let $\mathbf{p_{f,i}} \in \mathbf{R}^3$ represent the coordinates of the marker to be filled at the same point in time.

1. Spline fill ($m = 0$)

Spline fill fits a cubic polynomial to the gap data with boundary conditions defined at the edges of the gap. This is done using MATLAB's *interp1()* function with the *'spline'* option.

2. Pattern fill (m = 1or2)

Pattern fill takes the offset for one donor marker between its actual data and a linear interpolation, and adds that offset to a linear interpolation of the desired marker. If there are two valid donor markers, the one with the closest average position to the desired marker at the frames before and after the gap will be selected. The coordinates of the missing trajectory can be calculated as

$$\widehat{o_i} = \mathbf{p_{d,i}} - (\mathbf{p_{d,0}} + \tau(\mathbf{p_{d,1}} - \mathbf{p_{d,0}})) \tag{1}$$

$$\mathbf{p_{f,i}} = (\mathbf{p_{f,0}} + \tau(\mathbf{p_{f,1}} - \mathbf{p_{f,0}})) + \widehat{o_i} \tag{2}$$

where $\widehat{o_i}$ is the offset between the actual data and the linearly interpolated data, for both the donor marker and the desired marker, at $\tau = i$. As stated above, τ is a parameterization of time, $\mathbf{p_{d,i}}$ is the position of the donor marker at $\tau = i$, and $\mathbf{p_{f,i}}$ is the estimated position of the missing marker at $\tau = i$.

3. Rigid body fill ($m \geq 3$)

The method for rigid body filling uses the Kabsch algorithm to determine the optimal rotation matrix between the donor marker data at $\tau = 0$ and at $\tau = i$, and another optimal rotation matrix between the donor marker data at $\tau = 1$ and at $\tau = i$. Let

$$\mathbf{O}(i) = \frac{1}{m} \sum_{p \in \mathbf{p_{d,i}}} p \tag{3}$$

represent an origin row vector on the rigid body. Then set $\mathbf{p_{d,i}} = \mathbf{p_{d,i}} - \mathbf{O}(i)$ in order to offset $\mathbf{p_{d,i}}$ to be with respect to $\mathbf{O}(i)$ by subtracting the origin from each of the rows.

Next, create covariance matrices C_0 and C_1 where $C_j(i) = \mathbf{p_{d,j}}^\top \mathbf{p_{d,i}}$ and perform a singular value decomposition such that $C_j = \mathbf{U_j \Sigma_j V_j^*}$. Let $\mathbf{L_j} = \mathbf{I} \in \mathbf{R}^{3\times3}$ with $L_j[3,3] = \text{sgn}(\det(\mathbf{U_j V_j^*}))$. The rotation matrices $R_j = \mathbf{V_j L_j U_j^*}$ can then be defined. Given R_0 and R_1, estimated trajectories $\mathbf{G}_0$ and $\mathbf{G}_1$ can be written as

$$\mathbf{G}_j(i) = R_j(i)(\mathbf{p_{f,j}} - O(j))^\top + O(i)^\top \tag{4}$$

with $j = 0$ estimating from $\tau = 0$ and $j = 1$ estimating from $\tau = 1$. The interpolated data can then be expressed as

$$\mathbf{p_{f,i}} = \tau \mathbf{G}_1(i) + (1 - \tau)\mathbf{G}_0(i) \tag{5}$$

For each time frame, every available donor will be used to interpolate the gap, which means that donors can vary between frames. In other implementations, fills may be rejected if there are fewer than three valid donors at any time frame. In the proposed implementation, only those frames with fewer than three donors will be rejected, and the rest of the gap will be filled. This ensures the maximal use of rigid body fill over other methods of gap-filling, because it is the least susceptible to errors from bad donor data due to its inclusion of several donors.

Algorithm 2. Gap-Filling Algorithm

```
procedure GapFill(markers)
  gapTable ← findGaps(markers)
  for gap in sort(gapTable) do
    if gapLength(gap) = 1 then
      SplineFill(gap)
  gapTable ← findGaps(markers)
  change ← true
  while change do
    while change do
      change ← false
      for gap in sort(gapTable) do
        gapChanged ← true
        if! RigidBodyFill(gap) then
          if! PatternFill(gap) then
            gapChanged ← false
        change ← change or gapChanged
      gapTable ← findGaps(markers)
  if ! SplineFill(smallestGap) then
    change ← false
  return markers
```

2.4. Gap creation

After each gap-filling step, the gap creation step is necessary to validate the filled trajectories, and to remove those that are not viable. Marker data with initial errors will show impossible or unlikely trajectories in both the initial data and the filled data, all of which must first be deleted, before being filled again. Without the gap creation step, any unchecked initial errors would propagate to the filled trajectories and remain there, giving an inaccurate representation of the true motion of the markers. To determine which sections have invalid trajectories after the motion capture data has been filled entirely, the marker data is run through inverse kinematics (IK) in OpenSim, which is generally used to ascertain the joint angles that would yield that marker data using the method of least squares fit. For our gap-filling purposes, however, the joint angles are not of direct interest but are an intermediate output to obtain the positions of markers on the kinematic fit model. By using the position of markers given the joint angles and comparing with the marker positions in the original marker data, we determine a measure of the error for each marker at every frame by taking the Euclidian distance between them. The marker positions on the kinematic fit model are provided from OpenSim, but the calculation of errors for each marker and frame, as well as all subsequent steps, are done in MATLAB. These IK errors are compared against two user-set thresholds to determine regions of deletion. For each marker, every region with error exceeding the lower threshold will be a candidate for deletion, and for each candidate region, any region that has a peak error exceeding the higher threshold at any point will be deleted. The entire region above the lower threshold will be deleted to ensure that the fill for this gap uses data with low error. With these artificial gaps created, the gap-filling step can be run again to attempt to find a more accurate fill based on the IK error metric.

Algorithm 3 Gap Making Algorithm

```
procedure GapMake(markers)
  errTable ← IK(markers)
  badRegions ← errTable>lowThreshold
  change ← false
  for region in badRegions do
    if MaxError(region)>highThreshold then
      delete region from markers
      change ← true
  return markers, change
```

2.5. Validation

The motion capture data that was used for validation of our gap-filling method was recorded with cameras at a rate of 200 Hz. Subjects consented participation in protocols approved by the Institutional Review

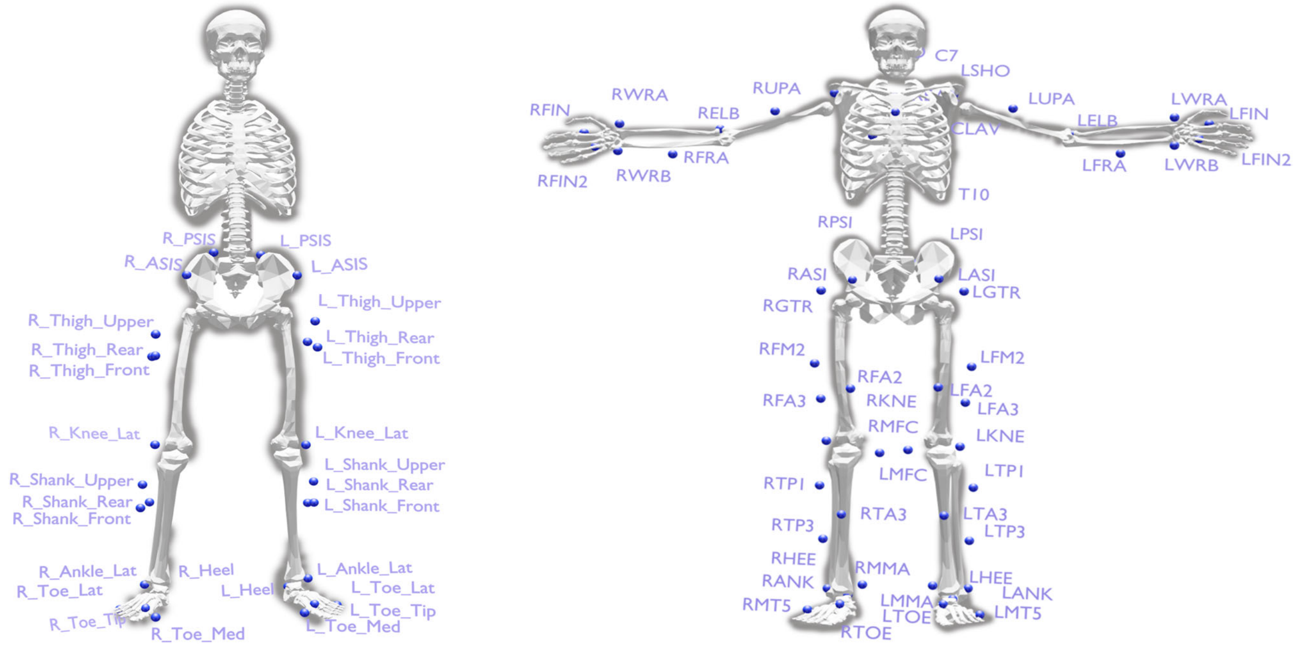

Figure 2. (a) Lower body marker setup. A 28-marker set was used for lower limb kinematics in ambulation. (b) A 52-Marker Full Body Setup for locomotion and throwing.

Board at Georgia Institute of Technology. Lower body locomotion on level ground, stairs and ramps was performed by 22 subjects instrumented with a 28-marker lower body setup. In addition, 10 subjects performed level ground walking and tossing actions while instrumented with a 52-marker full body setup based on the Cleveland Clinic marker set (Zeller et al. 2003); both marker sets are presented in Figure 2.

The motion capture data was used in four different aspects of validation. First, we evaluated the stability of reconstruction by comparing automatically filled data to manually filled data as the ground truth (2.4.1). For this, we inserted artificial gaps to sections of the marker set to determine the reconstruction error as the deviation from the ground truth.

It is evident that to compute the true reconstruction error of gap filling, a ground truth set of trajectories with no gaps would be necessary. However, when the algorithm is applied to a new unprocessed dataset as it is intended to be used, there is no ground truth of what the real marker positions are, and it is therefore impossible to calculate true reconstruction accuracy. Because of this, the next validation (2.4.2) is to show that IK error is correlated to the reconstruction error, supporting the use of IK error as the criterion for our algorithm.

Third, we compared manual gap-filling to our proposed method in terms of the processing time and error reflected in the inverse kinematics (2.4.3). Fourth, we evaluated the performance of the automated gap-filling method for completely processing the dataset (2.4.4).

1. Stability of reconstruction accuracy

The stability of reconstruction accuracy for the gap-filling method was studied by using a subset of four motion capture trials, from two subjects instrumented with the lower body setup (Figure 2(a)). $N = 5000$ artificial gaps were introduced in each trial that were automatically gap-filled using our iterative method, including feedback from IK, with up to 10 iterations. The deviation between manually filled and automatically filled data was used as the reconstruction error.

2. Validation of IK error

In a true experimental setting, neither true reconstruction error nor ground truth marker data would be available, but the IK error can always be determined, so a strong correlation between IK error and reconstruction error would imply that IK error can be used to determine where marker data is incorrect, instead of using reconstruction error.

To determine if there is a relation between reconstruction error and IK error, four motion capture trials, from two subjects instrumented with the lower body setup (Figure 2(a)) were first manually filled by individuals with experience in motion capture software. This process included a thorough inspection, including relabeling markers that were incorrectly labeled and deleting measured data that did not correspond to any actual markers. This manually filled data established an estimated ground truth of the marker positions over time. The raw data was filled automatically using only interpolation methods, with no iterative IK feedback and no deletion of marker data to create an example of reconstructed trajectories. The deviation between the automatically filled and manually filled data provides the reconstruction error of the automatically filled data. For every marker at every frame of the automatically filled, N = 79365 pairs of reconstruction error and IK error were computed, and the correlation between the IK error and the reconstruction error was determined.

3. Comparison with manual Gap-Filling

Our gap-filling method was compared against manual gap-filling with respect to the time taken to complete the process, in addition to the final IK error. Within a trial, for each marker, the maximum IK error over all frames was calculated, and these marker maximums were averaged over all markers and all trials as a metric of the typical worst-case reconstruction error. The mean of the IK error over all frames and markers was also calculated as a metric of the average reconstruction error. The gap-filling data was produced from N = 18 trials from 4 subjects. One subject was instrumented with the full body setup (Figure 2(b)), while the other three were instrumented with the lower body setup (Figure 2(a)). As reference for the processing time, the automated gap-filling was performed on a laptop computer with an Intel Core i7-9750H 2.60 GHz processor and 32GB RAM, and ran for 5 iterations. The manual gap-filling was executed by personnel with previous experience in the use of the motion capture software (Vicon Nexus).

4. Deployment of the algorithm to the complete dataset

Finally, we evaluated the performance of the method by iteratively filling gaps and calculating the maximum IK error after each iteration to determine if the highest IK error decreases through iterations as a result of using the IK error as feedback. We evaluated the use of this method to fill gaps for a total of N = 138 trials across 20 different subjects instrumented with the lower body marker setup (Figure 2(a)), and N = 42 trials across 10 different subjects instrumented with the full body marker setup (Figure 2(b)). The trials were filled with up to 4 iterations of gap-filling, and terminated early with a high IK error threshold of 60 mm.

3. Results

3.1. Reconstruction accuracy

The distribution of reconstruction errors across all four trials is presented in Figure 3, which shows that

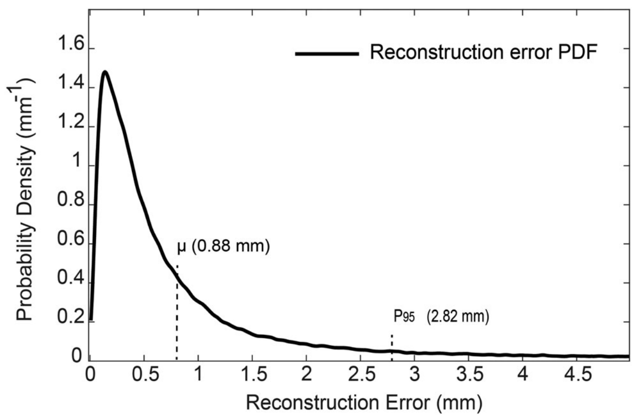

Figure 3. Distribution of reconstruction error. Probability density function of marker reconstruction from a filled with artificially-created gaps.

most frames are reconstructed with low error. The mean reconstruction error was 0.88 mm with a standard deviation of 4.0 mm. A sample region with high reconstruction error is presented in Figure 4, and gives an example of a possible inaccurate reconstruction that the algorithm may generate. We observe, however, that the reconstructed trajectory follows the same overall trend as the original data.

3.2. Validity of IK error

The reconstruction error and IK error plotted in Figure 5 have a Pearson's correlation coefficient of $\rho = 0.733$, which was highly significant with $p < 0.01$, indicating that reconstruction error and IK error are strongly correlated. Since reconstruction error cannot be obtained without having gap-less or perfectly filled data in the first place, the strong correlation between IK error and reconstruction error shows that IK error can be used instead, and regions with high IK error are also regions with high reconstruction error.

3.3. Comparison with manual Gap-Filling

The 18 filled trials had an average of 9105 frames and 792 gaps, with an average gap length of 5 frames, and a maximum gap length of 326 frames. Comparing the results of our method with data filled manually in Vicon Nexus, we can see in Figure 6 that our method finishes in much less time (80% decrease from 0.276 sec/frame to 0.056 sec/frame). For the IK error, the mean of marker maximums was also lower with our automated method (21% decrease from 69.4 mm to 54.6 mm). The higher IK errors for the manually filled data suggest that it was not reconstructed as close to the true marker positions as the automatically filled data, especially since this set of manually filled

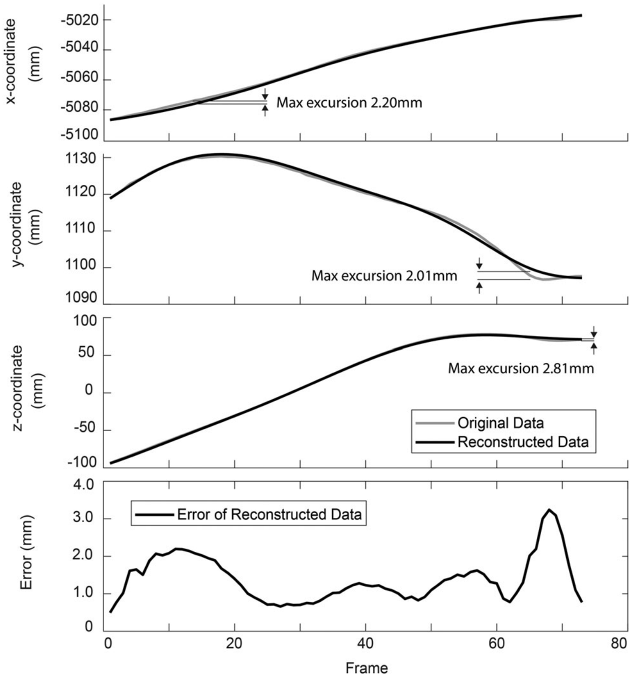

Figure 4. Original and reconstructed marker trajectory of 59-frame artificially-created gap. This is the longest gap created and shows a trajectory reconstructed worse than average.

data was not as thoroughly vetted as the data from the previous sections. The average of all IK errors was nearly the same between both (17.5 mm for manual, 17.4 mm for auto).

A Welch's t-test on the results shows that the decrease in processing time is highly significant ($p < 0.01$), and that the decrease in the mean of maximum IK errors was also significant ($p < 0.05$), but the difference in the mean of all IK errors was not significant. In comparing the variances, we can see that the variance in time taken is significantly less for the iterative gap-filling method. (F-test of variances $p < 0.01$), but there is no statistically significant difference in the variance of the mean of the maximum of IK errors or the mean of all IK errors.

3.4. Error over iterations

The maximum IK error over four iterations is presented for the automated gap-filling of the entire dataset. The trials had an average of 22370 frames and 3044 gaps, with an average gap length of 6 frames and a maximum gap length of 3405 frames. The data for all trials, as well as the mean across the different trials is presented in Figure 7. The total IK error decreases significantly with respect to the first iteration (47% drop from 129.0 mm to 67.6 mm). Some trials show an increase in the error between iterations, yet the general trend is that the error will reduce in later iterations. The trials that do diverge in error corresponded to trials with especially inaccurate initial marker labels, or had many regions with no marker labels at all.

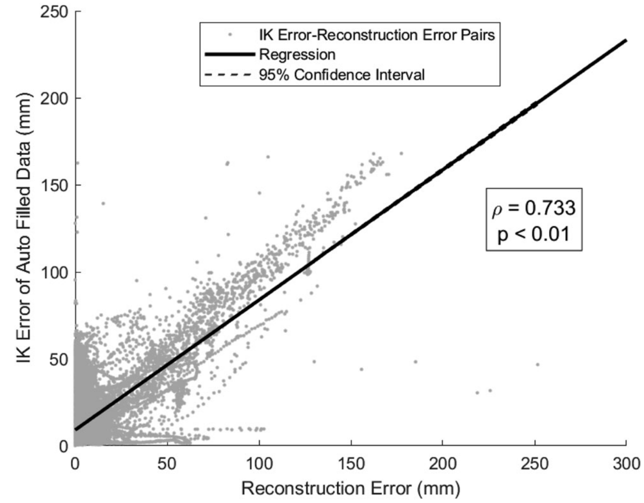

Figure 5. IK Error of automatically filled data vs. reconstruction error of automatically filled data. Reconstruction error was calculated as the deviation between automatically filled data and manually filled ground truth data.

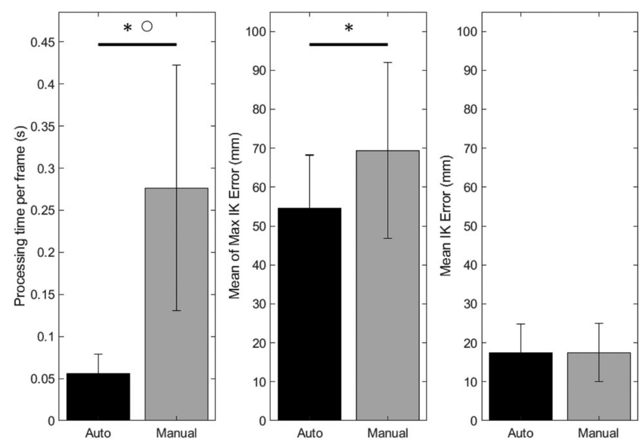

Figure 6. Comparison of automated gap-filling and manual gap-filling for processing time (left) mean of marker maximum IK error (center), and mean of all IK errors (right). * represents a significant difference of means, ° represents a significant difference of variance ($p < 0.05$).

The data used for this section had a total of 4,026,000 frames, and based on the results from section III.C, it would take approximately 51.5 hours of continuous computer processing to finish, or just over 2 days for one computer. On the other hand, processing this data manually would have taken 309 total man-hours. A small team of 5 people working 8 hours per day would need almost 8 days to finish processing the data, further illustrating the benefits of our automated gap-filling method.

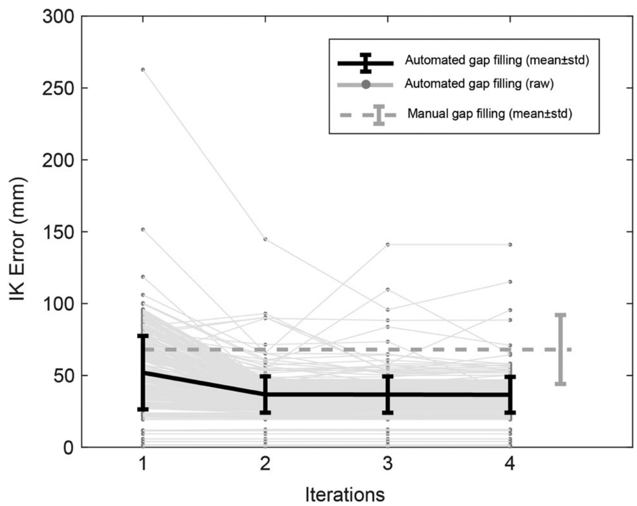

Figure 7. Mean of marker maximum IK error over iterations. The method reduces the IK error for all trials evaluated, and also yields an error lower than the average of manually filled gaps.

4. Discussion

The results show that IK error is strongly correlated with reconstruction error, indicating that IK error can be used as a readily available metric to determine where reconstructions are inaccurate and must be corrected. Taking advantage of this correlation, the proposed method of gap-filling is able to reconstruct missing marker data to within a low average error, indicating that our method of gap-filling is viable in terms of reconstruction accuracy.

The feedback given from inverse kinematics (IK) allows the gap-creation step to target regions with high IK error so that they will be reduced between iterations of gap-filling, causing the resulting fills to be more accurate in later iterations. The error presented in the first iteration is from using only the existing interpolation-based methods of gap-filling. The errors in the later iterations include the gap creation step using the feedback from IK and are much lower than the error in the first iteration, showing that the iterative process is necessary for the error to converge to a minimum. Although some trials show an increase in the error between iterations, these same trials also show a decrease in error afterwards, suggesting that if more iterations were performed, the error would continue to decrease. Based on these results from the error comparison over iterations, we can see that the iterative process is necessary for our method to decrease the IK error.

Compared with manual gap-filling, our iterative gap-filling method yields lower worst-case IK errors due to the feedback given from IK, and can do so in a significantly shorter period of time, indicating that our method is useful to reduce the processing time. It is important to note that this processing time is computational only. Thus, our method removes the burden of gap-filling manually by the user, virtually eliminating the number of man-hours that must be spent for the gap-filling process.

Although our algorithm takes advantage of existing gap-filling methods, the high errors in the first iteration of the process indicate that these methods are not sufficient for producing quality data if the original data is inaccurate. Our original contribution is found in the next steps of the algorithm, where IK error, as a metric of reconstruction accuracy, informs a gap creation step, after which gaps can be filled again. Our evaluations of the method reveal that it can serve as a more accurate and efficient replacement for manual gap-filling and can be easily implemented in existing processing pipelines, especially if MATLAB and OpenSim are already in use.

One barrier to fully eliminating user effort in this process is that the method requires an initial labeling of the data; this could be expanded by using a labeling approach such as the methods suggested by Maycock et al. (XXXX), or Meyer et al. (2014). Such labeling methods could be used in conjunction with our iterative gap-filling method, and would also reduce the impact of incorrect labels, leaving improper fills and interference from other reflective objects as the primary sources of invalid marker data. Such work in labeling could be directly integrated with the code behind our algorithm due to its open-source nature. Additionally, biomechanics research groups have the ability to alter various aspects of the algorithm to allow it to coordinate with other processing steps.

5. Conclusion

The proposed method of iterative gap-filling allows for completely automated and accurate reconstruction of missing marker data without the requirement of entirely accurate initial data due to the feedback given by IK. It is capable of doing so in significantly lower processing time, especially eliminating manual user input and the use of graphical user interfaces. Thus, our method has the potential to replace the manual gap-filling and allow research requiring motion capture to be conducted much more quickly without sacrificing quality, by using feedback from IK to determine and reduce the errors.

Further optimizations can be made to increase the efficiency of the algorithm, namely to only perform IK on the sections that were deleted and refilled since those are the only regions that should show changes in the IK errors. If additional data are available, the gap-filling step of the iterative process could be replaced with any other algorithm that completely fills any missing trajectories, such as machine learning methods or probabilistic methods, and the iterative process with IK feedback can still be applied to validate and correct the resulting gap-filling. The algorithm could also be improved by automatically detecting when there is no improvement between iterations and terminating early instead of continuing until a maximum iteration limit is reached. As our primary contribution is to show the benefits of an IK-based validation, such improvements are beyond the scope of this work. However, the method is released as an open-source repository in the supplemental material and at https://github.com/JonathanCamargo/MoCapTools, so other biomechanics research groups have the opportunity to expand on this work, should the need arise.

Acknowledgments

The authors would like to thank Dean Molinaro for sharing the full body motion capture data that was analyzed in this paper, and Bharat Kanwar and Will Flanagan for their discussions during the conception of the gap-filling method. This work was supported, in part, by a Fulbright fellowship awarded to Jonathan Camargo-Leyva.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Jonathan Camargo http://orcid.org/0000-0001-5219-2110

References

Aristidou A, Cameron J, Lasenby J. 2008. Real-time estimation of missing markers in human motion capture. 2nd International Conference on Bioinformatics and Biomedical Engineering.

Begon M, Wieber P-B, Yeadon MR. 2008. Kinematics estimation of straddled movements on high bar from a limited number of skin markers using a chain model. J Biomech. 41(3):581–586.

Bobilev AN, Bolotin YV, Voronov AV, Kruchinin PA. 2012. About two the least-squares method modifications for lost data recovery in videoanalysis system based on accelerometer data. Russ J Biomech. 16(1):80–91.

The 3D Biomechanics Data Standard (C3D) 2018. https://www.c3d.org/index.html. Recent Innovations in Mechatronics (RIiM), Vol. 5. No. 1.

Chiari L, Croce UD, Leardini A, Cappozzo A. 2005. Human movement analysis using stereophotogrammetry: Part 2: Instrumental errors. Gait Post. 21(2):197–211.

Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng. 54(11): 1940–1950.

Federolf P. 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 JJ, Song R. 2014. Exploiting temporal stability and low-rank structure for motion capture data refinement. Inf Sci. 277:777–793.

Gloersen O, Federolf P. 2016. Predicting missing marker trajectories in human motion data using marker intercorrelations. PLoS One. 11(3):e0152616.

Herda L, Fua P, Plänkers R, Boulic R, Thalmann D. 2001. Using skeleton-based tracking to increase the reliability of optical motion capture. Hum Mov Sci. 20(3):313–341.

Howarth SJ, Callaghan JP. 2010. Quantitative assessment of the accuracy for three interpolation techniques in kinematic analysis of human movement. Comput Methods Biomech Biomed Eng. 13(6):847–855.

Kabsch W. A solution for the best rotation to relate two sets of vectors. Acta Crystallographica.

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

Liu G, Zhang J, Wang W, McMillan L. 2006. Human Motion Estimation from a Reduced Marker Set. Symposium on Interactive 3D graphics and games, pp. 35–42, https://dl.acm.org/citation.cfm?id=1111418.

Maycock J, Rohlig T, Schroder M, Botsch M, Ritter H. XXXX. Fully automatic optical motion tracking using an inverse kinematics approach. 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids).

Meyer J, Kuderer M, Mueller J, Burgard W. XXXX. Online marker labeling for fully automatic skeleton tracking in optical motion capture. 2014 IEEE International Conference on Robotics and Automation (ICRA).

Moeslund TB, Hilton A, Krüger V. 2006. A survey of advances in vision-based human motion capture and analysis. Comput Vision Image Understand. 104(2/3): 90–126.

Nagymáté G, Kiss RM. XXXX. Application of OptiTrack motion capture systems in human movement analysis: a systematic literature review.

OptiTrack Inc. Motion capture systems. XXXX. https://optitrack.com/. https://www.vicon.com/faqs/software/what-gap-filling-algorithms-are-used-nexus-2.

Peng S, He G, Liu X, Wang H. 2015. Hierarchical block-based incomplete human mocap data recovery using adaptive nonnegative matrix factorization. Comput Graph. 49:10–23.

Qualisys Motion Capture Systems. XXXX. https://www.qualisys.com/.

Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, Ong CF, DeMers MS, Rajagopal A, Millard M, et al. 2018. OpenSim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. PLoS Comput Biol. 14(7):e1006223.

Starck J, Hilton A. 2007. Surface capture for performance-based animation. IEEE Comput Grap Appl. 27(3):21–31.

Tan C, Hou J, Chau L. 2015. Motion capture data recovery using skeleton constrained singular value thresholding. Vis Comput. 31(11):1521–1532.

Tits M, Tilmanne J, Dutoit T. 2018. Robust and automatic motion-capture data recovery using soft skeleton constraints and model averaging. PLoS One. 13(7):e0199744.

Vicon Motion Systems Limited. XXXX. Fill gaps in trial data. https://docs.vicon.com/display/Nexus25/Fill+gaps+in+trial+data.

Vicon Motion Systems Limited. XXXX. https://www.vicon.com/.

Vicon Motion Systems Limited. XXXX. What Gap Filling Algorithms are Used Nexus 2, https://www.vicon.com/.

Zeller BL, McCrory JL, Kibler WB, Uhl TL. 2003. Differences in kinematics and electromyographic activity between men and women during the single-legged squat. Am J Sports Med. 31(3):449–456.

Zhang Z. 2012. Microsoft kinect sensor and its effect. IEEE Multimedia Mag. 19(2):4–10.