IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 8, 2021, accepted February 5, 2021, date of publication February 19, 2021, date of current version March 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060385

Graph Matching for Marker Labeling and Missing Marker Reconstruction With Bone Constraint by LSTM in Optical Motion Capture

JIANFANG LI[1], DEGUI XIAO[1], KEQIN LI[2], (Fellow, IEEE), AND JIAZHI LI[1]
[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[2]Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Corresponding author: Jianfang Li (lijianfang@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China, under Grant 61272062, and in part by the Changsha Science and Technology Project through the Changsha Science and Technology Bureau under Grant kq2004012.

ABSTRACT Optical motion capture (MOCAP) is a commonly used technology to record the motion of non-rigid objects with high accuracy in 3D space. However, the MOCAP data has to be processed further before it can be used. The scattered reconstructed motion data must constitute a human configuration by labelling process according to the predefined template, and the missing markers have to be reconstructed to produce a stable motion trajectory. In this work, we propose a novel labelling method for motion sequences. First, a novel graph matching method is employed to determine the connection relationship of the scattered motion data for a single frame. Then, Kalman filtering is used for tracking in the motion sequence. As for the challenge coming from missing markers, we propose a new motion data preprocessing method considering the bone length constraint, which represents the information of variation in the relative position of adjacent markers. The processed motion data is input into a Long-Short Term Memory (LSTM) model to recover the missing markers and de-noise the motion data. The experiment conducted on our own dataset proves that our labelling method achieves a similar effect to Cortex, which is a commonly used commercial motion data analysis software. The experiment on CMU dataset demonstrates that our missing marker reconstruction method can achieve an art-of-state result. The labelling code will be pulished on https://github.com/Lijianfang6930/Graph-Matching-for-Marker-Labelling

INDEX TERMS Data preprocessing, graph matching, LSTM, MOCAP data.

I. INTRODUCTION

Motion capture is a technique which transforms human motion in the real world into a digital representation. Yet many scenarios have to take advantage of the digital representation of human motion, like movie special effects [1], animation production [2], kinematic analysis [3], etc. With the rapid development of computer vision technology [4]–[6], marker-based optical motion capture (MOCAP) is playing an increasingly important role, and several commercial devices show their reliability to capture human motion, like Motion Analysis and Vicon. While capturing the human motion, markers attached on the human body are recorded as 3D points in a real-world coordinate system for each timestamp and stored as digital representation [7]. So far, the motion data in each frame is scattered and our-of-order, each reconstrcuted marker must be assigned to the predefined location on the human body [8], meaning labelling marker trajectory. Only after accurately and efficiently labelling the marker trajectory, can the captured motion data be useful. Manual labelling is a time-consuming, susceptible and mistakable operation for users. Meanwhile, labelling would become much more difficult if some markers are failed to be reconstructed, the phenomenon is caused mainly by self-occlusion and failure marker detection [9], [10]. Missing markers would lead to discrete 3D motion trajectories, sometimes it will cause adjacent points to disappear for a while, that would break the related information of spatial domain and time domain [11].

Commercial processing software has provided a labelling solution to reduce manual labour, they build a template by manual initialization, then the subsequent motion trajectories learn the inner geometry structure of motion data from the predefined template. Nonetheless, the working mechanism of commercial software is unaccessible, and too many missing markers often give rise to failing tracking [12]–[14],

The associate editor coordinating the review of this manuscript and approving it for publication was Zhong Wu.

so missing marker reconstruction would bring about a better labelling result. Most tradition methods deal with the issue by the means of graph theory [15], [16]. Some methods employed interpolation algorithm to fill in the blanks of missing markers, however, these methods worked only in the condition of small-scale missing markers [17], and got poor performance once the markers continuously miss. Later a more logical thought that using the statistical model to correlate markers and dynamics across time was proposed, but the strategy was still challenging to infer long time missing markers. Under the modern view of data-driven model [18]–[21], Mall *et al.* [21] novelty proposed two neural-network-based architectures to model more complicated spatial and temporal correlation, their research achieved state-of-the-art result, however, they did not consider bone length constraint, which has been proven to be an important characteristic for the human skeleton in many types of research [22]. It is very important to keep constant bone length when MOCAP data is edited, otherwise, the created human model will be distorted.

In our previous researches, we have presented an open-source of multi-view calibration and 3D point reconstruction method for motion capture system [12]. Here we are proposing an automatical trajectory labelling method and missing marker reconstruction method. The proposed methodology produces a continuous stream of accurate labelled 3D motion data, and recovers the missing markers which disapears for an extended period time. We first propose a novel graph match algorithm to label each marker trajectory with the help of manual initialization. Instead of using original 3D coordinates as human pose representation [18], a data preprocessing method is proposed to include the information of variation in the relative position of adjacent markers. We train a Long-Short Term Memory(LSTM) model using the new representation of motion data with bone constraint. During the test, the pre-trained model maps the corrupted pose to the true one. Our main contributions are summarized as follows:

1. A novel trajectory labelling method is proposed. We propose a graph retrieval method to match the predefined template of the human body for a single frame, then Kalman filtering is employed as a tracking strategy for motion sequences. The method can obtain similar results compared to commercial motion analysis software.

2. We propose a data preprocessing method to generate a new representation of motion capture data with bone length constraint, which contains the information of variation in the relative position of adjacent markers. We set the processed motion data as the input of Long-Short Term Memory to model complicated spatial and temporal correlation for human motion. The experimental result shows the effectivity of our new motion data representation.

II. RELATED WORK

In the last two decades, Many types of research have made great progress for trajectory labelling and missing marker reconstruction, here we present a brief review of the related works.

A. MISSING MARKER RECONSTRUCTION

There are multiple kinds of motion capture installation to produce 3D human pose, unfortunately, almost all existing motion capture system must face the cruel truth that some missing markers are unavoidable. Recently many researches provide methods for missing marker reconstruction.

Preliminary occlusion filling related methods employed interpolation to smooth 3D human motion trajectory sequence, linear interpolation and splines reacted for short-time occlusions or occasional missing markers. But the interpolation was under-performing while the markers disappeared for a long time [23]. Then skeleton-based methods sprung up, these methods [16], [24], [25] employed human skeleton models to represent joint characteristics according to kinematics, and the human structures were repaired from the obtained motion data. Skeleton based methods run into the same difficulty that only worked well for a short segment of occlusions. Later, with the emergence of machine learning, many researchers combined machine learning approaches and dynamical systems. Some researchers directly reconstructed missing marker from linear models [26]–[28]. They usually combined PCA and Kalman smoothing together and operated models in a lower-dimensional space. Other researches [22], [29], [30], [31] introduced a chain of latent parameters to model human motion, thus a nonlinear map could be built from the latent space to the observed motion data, missing markers could be reconstructed by expectation maximum (EM) algorithm. These researches increased the accuracy of long-time missing marker reconstruction, but strong prior assumptions were imposed on these methods, leading to poor robust of irregular and complex motion. Further research is still required before the final goal of producing stable human motion. The limitation of previous works pushed researchers to look for new solutions, that is a neural network approach, which has a powerful ability to model non-linear mapping. Feng *et al.* [17] firstly introduced LSTM to handle missing marker reconstruction. Then, Holden [32] trained a deep neural network to transform corrupting human pose to the unbroken one. Mall *et al.* [21] improved the above two methods and proposed two much simpler neural network architectures with higher computation speed. Current deep-learning-based methods can achieve art-of-state accuracy for missing marker reconstruction, but they resolved the tasks of labelling and missing marker reconstruction independently, as a matter of fact, the two interact with each other. In conclusion, drawing bone length constraint into consideration will be beneficial to labelling while recovering missing markers.

B. LABELLING MAKER TRAJECTORY

A typical motion capture labelling process is composed of the initialization step and tracking step. There are lots of researches focused on labelling human motion trajectory.

The most common treatment is to estimate potential human skeleton structure according to the predefined human body. Ringer and Lasenby [33] took the association of segments of the human configures into consideration, this method reacted on tracking markers. Yu *et al.* [34] pre-encoded the rigid structure information into motion models, the approach could automatically label markers on multiple targets frame by frame. In many applications, most researchers [35]–[37] either built a standard human skeleton, which was easy to automatically label or utilized joints limits, which determined by human configuration. Meyer *et al.* [38] used T-pose-based initialization instead of manual initialization, they realized fully automatic labelling process, however, their method could only label the whole human skeleton structure, the method could not label specific part of the human configuration. Xia *et al.* [14] took advantage of Hungarian to optimize the soft graph matching problem, the method realized real-time marker labelling. The latest researches revealed that the data-driven methods also contributed to labelling. Kim [39] proposed a BRNN network with attention mechanism, which is also a deep-learning. Pavlo *et al.* [40] proposed deep-learning-based finger animation model, which was a two-stage pipeline based on deep neural network to reconstruct missing marker and repair joints deformation. Ghorbani *et al.* [8] designed a differentiable permutation learning model for automatic marker labelling, they utilized temporal consistency as a postprocessing method. The data-driven model could achieve impressive labelling results while training data and test data were from the same domain, but the adaptive capacity of the data-driven model needed further study.

As a kind of data-driven model, the deep-learning-based methods can achieve a high degree of automation and accuracy in the process of labelling, the deep-learning-based methods are the development tendency. But the performance of current deep-learning-based methods for labelling extremely dependent on the training data, that limits their application in real world. By contrast, graph matching methods need the process of manual initialization. However, graph matching methods are not restricted by data domain, and can realize real-time and online labelling. In general, graph matching methods are more suitable in practical application under current technical framework.

III. METHOD OVERVIEW

Our purpose is to build a skeletal configuration which is consisted of the captured markers attached on the human body and obtain continuous motion trajectory of the configuration for motion sequences. In this section, we first propose a labelling method to confirm the associated markers and generate configuration automatically according to a predefined template. In order to recover the failure reconstruction markers and denoise the motion data, inspired by the work of [18], we propose a data pre-processing method to introduce the bone constraint into the input of LSTM model, so that the input can represent the variation in the relative position of adjacent markers.

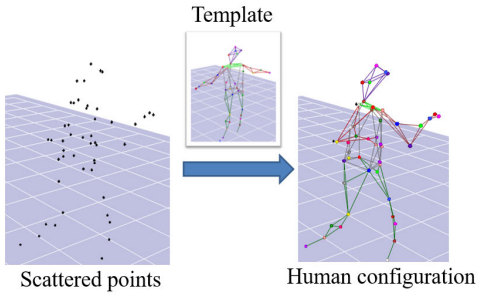


FIGURE 1. The process of marker labelling.

A. TRAJECTORY LABELLING BASED ON GRAPH MATCHING

After obtaining motion data, the markers attained on the captured object are connected by the manual operation to general a graph template. During motion capture, the scattered markers need to link to each other, and form a human configuration according to the predefined template, as shown in Fig. 1. Given a predefined template gragh and a set of scattered points, the labelling problem can be solved as a problem whether there is a subgraph, which is composed of scattered point sets, similar to the template graph. Our trajectory labelling algorithm is based on graph matching, as show in Algorithm 1, and the tracking strategy is based on Kalman filtering. The first stage is data preparation, the real-time edges and points are grouped according to the range of the predefined template. Then template edges and points are further traversed to find the matching real-time edges and points. Finally, consecutive frames of MOCAP data are labelled by Kalman filtering based tracking strategy. The whole process is shown in Fig. 2.

1) PREPROCESSING SCATTERED POINTS

The predefined template, which is the connection graph of markers on the captured object, is saved as a template in advance. The template provides the information of the total number of reconstructed points, the name and the serial number of each point, the total number of template graph edges, the minimum length of graph edges, the maximum length of graph edges, scalable range of each template graph edge, and the serial number of endpoints connected by each graph edge. In the following description, the points in the template are referred to as template points, and the edges in the template are referred to as template edges. The template determines the degree of variability in the human skeletal structure, larger scalable range of template edges indicates richer changes in human skeletal structure. During motion capture, the reconstructed markers in MOCAP data are called real-time points, the connectted lines between real-time points are called real-time edges. The real-time points are numbered according to the sequence of marker reconstruction, the real-time edges

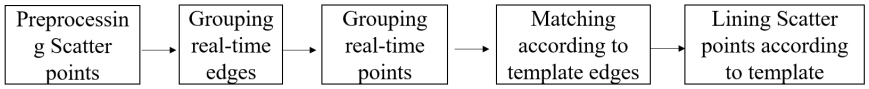


FIGURE 2. The process of marker labelling.

Algorithm 1 The Marker Labeling Process Based on Gragh Matching

Input: Template structure, real-time points.

Output: Subgraph similar to the template structure.

// Preparation work before matching template //

while i <= m **do**

  Filter alternative real-time edges for each template edge according to the length.

end while

for i = 1:n **do**

  Filter alternative real-time points for the template point.

end for

while i <= m **do**

  Update alternative real-time edges for each template edge.

end while

// Match the template edge according to the serial number from small to large //

Select the template point with the smallest number as the starting point, and calculate the real-time point number matching to the first template point.

Calculate the serial number of real-time points matching to the subsequent template points in turn according to the specific order of the template edges.

TABLE 1. The list of real-time edges with 46 real-time points.

Distance	Marker1	Marker2	Real-time edge
111.46	P_1^r	P_2^r	E_1^r
108.12	P_1^r	P_3^r	E_2^r
377.65	P_1^r	P_4^r	E_3^r
...	...	...	...
108.79	P_{44}^r	P_{45}^r	E_{1033}^r
39.50	P_{44}^r	P_{46}^r	E_{1034}^r
97.83	P_{45}^r	P_{46}^r	E_{1035}^r

include all line connecting each pair of real-time points, the total number of real-time edges is $\mathbf{C_n^2}$. In the following description, we illustrate the labelling method with an example. The total number of real-time points is 46, the total number of real-time edges is $\mathbf{C_{46}^2} = 1035$, as shown in Table 1, where Marker1 and Marker2 are the two endpoints of the real-time edge.

First, all real-time edges are sorted from small to large according to the length. In this example, there are 5 template points, these template edges are sorted in a particular order,

TABLE 2. The particular order of the template edges, Marker1 and Marker2 are the two endpoints of the template edge.

MinLength	MaxLength	Marker1	Marker2	serial number
140.69	143.93	P_2^t	P_1^t	E_1^t
115.64	119.33	P_3^t	P_1^t	E_2^t
146.05	150.67	P_4^t	P_2^t	E_3^t
165.74	170.59	P_4^t	P_3^t	E_4^t
106.66	110.20	P_5^t	P_1^t	E_5^t
143.65	147.26	P_5^t	P_2^t	E_6^t
110.91	114.19	P_5^t	P_3^t	E_7^t
153.37	158.13	P_5^t	P_4^t	E_8^t

```
[Markers]
Comment= Marker#, Name, Color, PhysicalColor, Size, Optional
NumberOf=5
1, Head_LF, 0, 0, 0.000000, 0
2, Head_LB, 1, 0, 0.000000, 0
3, Head_RF, 2, 0, 0.000000, 0
4, Head_RB, 3, 0, 0.000000, 0
5, Head_Top, 4, 0, 0.000000, 0
```

FIGURE 3. Storage form of template points.

```
[Linkages]
Comment= Name, LinkColor, LinkType, Marker1# Marker2#, MinLength, MaxLength, ExtraStretch
NumberOf=8
Link, 0, 0, 2 1, 140.696518, 143.928329, 0.150000, 0
Link, 0, 0, 3 1, 115.642090, 119.331139, 0.150000, 0
Link, 0, 0, 4 2, 146.049789, 150.671249, 0.150000, 0
Link, 0, 0, 4 3, 165.742493, 170.585114, 0.150000, 0
Link, 0, 0, 5 1, 106.658913, 110.201050, 0.150000, 0
Link, 0, 0, 5 2, 143.603760, 147.256042, 0.150000, 0
Link, 0, 0, 5 3, 110.913986, 114.186768, 0.150000, 0
Link, 0, 0, 5 4, 153.373093, 158.127029, 0.150000, 0
```

FIGURE 4. Storage form of template edges.

and their connection relationship is shown in Table 2. The serial number of Marker1 is larger than that of Marker2, and Marker1 is numbered from small to large. The arrangement can improve computational efficiency and avoid repetitive searching. Fig. 3 and Fig. 4 show the storage form of template points and template edges, respectively. The set of template points, which is sorted, is denoted as $\mathbf{P^t} = \{P_1^t, P_2^t, \cdots, P_i^t, \cdots, P_n^t\}$, n is total number of template points. The set of corresponding template edges is denoted as $\mathbf{E^t} = \{E_1^t, E_2^t, \cdots, E_u^t, \cdots, E_m^t\}$, m is total number of template edges. The set of sorted real-time points is denoted as $\mathbf{P^r} = \{P_1^r, P_2^r, \cdots, P_j^r, \cdots, P_p^r\}$, p is total number of real-time points. The set of real-time edges is denoted as $\mathbf{E^r} = \{E_1^r, E_2^r, \cdots, E_v^r, \cdots, E_q^r\}$, q is total number of real-time edges.

2) GROUPING REAL-TIME EDGES

The real-time edges whose length is within a template edge length range are grouped together as the alternative

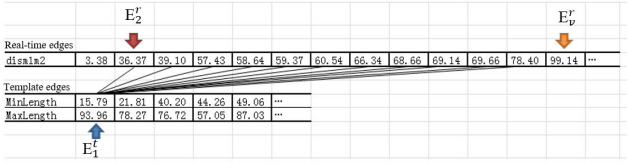


FIGURE 5. The result representation of all alternative matching real-time edges for E_1^t.

TABLE 3. The template edges sharing the same template point P_1^t.

MinLength	MaxLength	Marker1	Marker2	Template edge
106.65	110.20	P_5^t	P_1^t	E_5^t
115.64	119.33	P_3^t	P_1^t	E_2^t
140.69	143.92	P_2^t	P_1^t	E_1^t

TABLE 4. Partial alternative matching real-time edges in $\mathbf{G_1^p}$.

Distance	Marker1	Marker2	Real-time edge
142.18	P_{15}^r	P_{43}^r	E_{567}^r
143.20	P_3^r	P_{32}^r	E_{122}^r
143.76	P_{42}^r	P_{46}^r	E_{1029}^r

matching edges of the template edge, denoted as $\mathbf{G^e} = \{\mathbf{G_1^e}, \mathbf{G_2^e}, \cdots, \mathbf{G_u^e}, \cdots, \mathbf{G_m^e}\}$. A real-time edge can correspond to multiple template edges at the same time. Here we explain our grouping algorithm in detail though the above example.

The length of E_1^r is 3.38, it is not within the range of the E_1^t, thus E_1^r is not the alternative matching edge of E_1^t, so we need to try the next template edge E_2^r, whose length is 36.37, it is within the length range of the first template edge, so we chose E_2^r as an alternative matching edge for E_1^t, and added E_2^r into $\mathbf{G_1^e}$. Then we keep on traversing the rest of real-time edges until all alternative matching real-time edges of E_1^t are found, the result is shown in Fig. 5. Repeating the above operation, and we can finally get all the alternative matching edges for $\mathbf{G^e}$.

3) GROUPING REAL-TIME POINTS

In this stage, real-time points are grouped as alternative matching points of template points, denoted as $\mathbf{G^p} = \{\mathbf{G_1^p}, \mathbf{G_2^p}, \cdots, \mathbf{G_i^p}, \cdots, \mathbf{G_n^p}\}$. Once a real-time point P_j^r is determined matching to template point P_i^t successfully, then the corresponding alternative matching real-time edges $\mathbf{G_u^e}$ of E_u^t whose endpoints include P_i^t must share more than one real-time point, these shared real-time points are chosen as the alternative matching points of template point P_i^t, and added into $\mathbf{G_i^p}$. For example, template edges with the endpoint of template point P_1^t are E_1^t, E_2^t and E_5^t, as shown in Table 3. The alternative matching real-time edges of these three template edges are shown in Table 4, Table 5 and Table 6, respectively.

Here for the sake of simplicity, we only present limited alternative matching real-time edges for E_1^t, E_2^t and E_5^t. The real-time edges matching to these three template edges must share at least one common point, and these shared points are chosen as the alternative matching points of P_1^t, and are added into $\mathbf{G_1^p}$. From Fig 6, we can see that only real-time point P_3^r is shared by $\mathbf{G_1^e}$, $\mathbf{G_2^e}$, $\mathbf{G_5^e}$ (In practice, P_{19}^r and P_{43}^r are the alternative real-time points of P_1^t too), meaning that only real-time point P_3^r is included in $\mathbf{G_1^p}$. As shown in Table 4, only E_{122}^r contains P_3^r, so P_{32}^r, as another endpoint of E_{122}^r, is the alternative real-time point of P_2^t, which is another endpoint of E_1^t. By repeating the above operation, $\mathbf{G^p}$ can be obtained, the results are shown in Table 8.

TABLE 5. Partial alternative matching real-time edges in $\mathbf{G_2^p}$.

Distance	Marker1	Marker2	Real-time edge
115.97	P_{43}^r	P_{44}^r	E_{1030}^r
116.12	P_{18}^r	P_{20}^r	E_{631}^r
118.60	P_2^r	P_3^r	E_{46}^r

TABLE 6. Partial alternative matching real-time edges in $\mathbf{G_5^p}$.

Distance	Marker1	Marker2	Real-time edge
108.89	P_5^r	P_7^r	E_{176}^r
109.76	P_1^r	P_3^r	E_2^r

TABLE 7. The definition of E_1^t.

MinLength	MaxLength	Marker1	Marker2	Template edge
140.69	143.92	P_1^t	P_2^t	E_1^t

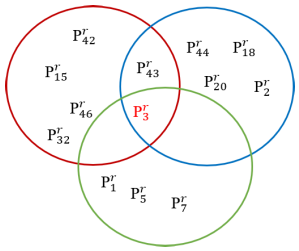


FIGURE 6. Points in the circle are the endpoints of alternative real-time edges of E_1^t, E_2^t and E_5^t, only P_3^r is shared by all the alternative real-time edges of E_1^t, E_2^t and E_5^t.

4) REFINING $\mathbf{G^e}$

After determining the alternative real-time points $\mathbf{G^p}$ for each template point, $\mathbf{G^e}$ can be refined according to $\mathbf{G^p}$. In the above example, the information of E_1^t is shown in Table 7. The two endpoints of E_1^t are P_1^t and P_2^t. The alternative matching points of all template points are shown in Table 8. We can see that the alternative matching real-time points of P_1^t are P_3^r, P_{19}^r, and P_{43}^r, the alternative matching real-time points of P_2^t are P_{15}^r, P_{16}^r, and P_{32}^r. The alternative matching real-time edges of E_1^t must include the alternative matching real-time points of P_1^t and P_2^t at the same time. According to initial $\mathbf{G_1^e}$, which is shown in Table 9, we can see that only E_{112}^r satisfies

TABLE 8. Real-time points in $\mathbf{G^p}$.

Template points	Alternative matching real-time points
P_1^t	P_3^r, P_{19}^r, P_{43}^r
P_2^t	P_{15}^r, P_{16}^r, P_{32}^r
P_3^t	P_2^r
P_4^t	P_{16}^r, P_{18}^r, P_{36}^r, P_{39}^r
P_5^t	P_1^r, P_{17}^r, P_{44}^r

TABLE 9. The alternative matching real-time edges in initial $\mathbf{G_1^e}$. Marker1 and Marker2 are endpoints of the real-time edge. Distance is the length of the real-time edge.

Distance	Marker1	Marker2	Real-time edges
134.36	P_9^r	P_{38}^r	E_{361}^r
134.56	P_{15}^r	P_{17}^r	E_{541}^r
138.48	P_{42}^r	P_{43}^r	E_{1026}^r
138.86	P_{36}^r	P_{46}^r	E_{990}^r
139.51	P_{36}^r	P_{44}^r	E_{988}^r
139.93	P_{10}^r	P_{35}^r	E_{394}^r
143.70	P_3^r	P_{32}^r	E_{112}^r
145.36	P_{16}^r	P_{17}^r	E_{571}^r
146.04	P_1^r	P_{32}^r	E_{31}^r
146.15	P_{43}^r	P_{46}^r	E_{1032}^r
147.88	P_{32}^r	P_{39}^r	E_{937}^r
151.25	P_{20}^r	P_{36}^r	E_{700}^r
153.17	P_{21}^r	P_{45}^r	E_{734}^r
155.19	P_{19}^r	P_{21}^r	E_{659}^r
156.96	P_1^r	P_{39}^r	E_{38}^r
158.00	P_{15}^r	P_{16}^r	E_{540}^r
159.67	P_{27}^r	P_{29}^r	E_{847}^r

the above requirement, E_{112}^r is remain in $\mathbf{G_1^e}$, other edges are deleted. The finall refined $\mathbf{G^e}$ is shown in Table 10.

5) MATCHING TEMPLATE EDGES ONE BY ONE

If the labelling objects are rigid, then the inner structures of the objects is fixed, the above grouping algorithm are sufficient to confirm the matching relationship between points and edges. However, for non-rigid targets, their structures are often flexible and variable, therefore multiple real-time edges frequently match to the same template edge. In order to solve this problem, the variable range of template edges should be appropriate, and all the alternative matching edges have to be traversed. Only all real-time edges are matched to the correct template edges, is the frame of motion data labelled successfully.

The basic idea of traversal is to search all matching cases of template points and alternative points according to ordered template edges. If there are multiple alternative real-time points in $\mathbf{G_i^p}$ for template point P_i^t, then a real-time point is randomly selected from $\mathbf{G_i^p}$ as the matching point of P_i^t, the rest alternative real-time points together with the previously matched points are stored in the stacks. If the subsequent matching process fails, a real-time point is extracted

TABLE 10. The alternative matching real-time edges in $\mathbf{G^e}$. Marker1 and Marker2 are endpoints of the real-time edge. Distance is the length of the real-time edge.

Template edge	Distance	Marker1	Marker2	Real-time edges
E_1^t	143.70	P_3^r	P_{361}^r	E_{112}^r
E_2^t	118.60	P_2^r	P_3^r	E_{46}^r
E_3^t	158.00	P_{15}^r	P_{16}^r	E_{540}^r
	162.84	P_{16}^r	P_{18}^r	E_{572}^r
	147.88	P_{32}^r	P_{39}^r	E_{937}^r
E_4^t	169.10	P_2^r	P_{39}^r	E_{82}^r
E_5^t	109.76	P_1^r	P_3^r	E_2^r
	113.70	P_{17}^r	P_{19}^r	E_{602}^r
	115.97	P_{43}^r	P_{44}^r	E_{1030}^r
	112.83	P_{44}^r	P_{45}^r	E_{1033}^r
E_6^t	148.03	P_1^r	P_{32}^r	E_{31}^r
	134.56	P_{15}^r	P_{17}^r	E_{541}^r
	145.36	P_{16}^r	P_{17}^r	E_{571}^r
E_7^t	112.19	P_1^r	P_2^r	E_1^r
E_8^t	156.96	P_1^r	P_{39}^r	E_{38}^r
	145.56	P_{16}^r	P_{17}^r	E_{571}^r
	161.80	P_{17}^r	P_{18}^r	E_{601}^r
	139.51	P_{36}^r	P_{44}^r	E_{988}^r

from the top of the stack, and a new round of the matching process is started.

The traversal process starts with E_1^t, the real-time points matching to Marker1 and Marker2 of the current template edge are calculated in turn, the flow chart is shown as Fig. 7. First, the matching points of Marker2 is determined by initialization. One of the alternative real-time points of Marker2(P_1^t) is chosen as the matching point, and the other alternative real-time points are stored in the stack. From Table 8 we can see that the alternative real-time points of Marker2 are P_3^r, P_{19}^r and P_{43}^r. If P_{43}^r is chosen as the matching real-time point of Marker2, it is not the endpoints of any of the alternative real-time edge of E_1^t in $\mathbf{G_1^e}$, thus P_{43}^r cannot be the matching real-time point of Marker2, neither is P_{19}^r. In this condition, the algorithm returns back to choose P_3^r as the matching real-time point of Marker2, P_3^r is the endpoint of the alternative real-time edge of E_1^t in $\mathbf{G_1^e}$, and it is also within $\mathbf{G_2^p}$, which is alternative real-time points of Marker1. At this point, the matching process of $\mathbf{E_1^t}$ is ended. Then the algorithm continues to find the matching real-time edges of the rest template edges until all template edges are matched successfully, then the predefined template is considered to be labelled successfully.

6) TRACKING STRATEGY

After labelling the markers of the current frame, the matching relationship of each real-time point and template point in each view is recorded. Tracking strategy is used to label the continuous trajectory in the motion sequence. The most commonly used Kalman filtering is employed for tracking in our method. When matching the feature points of the first

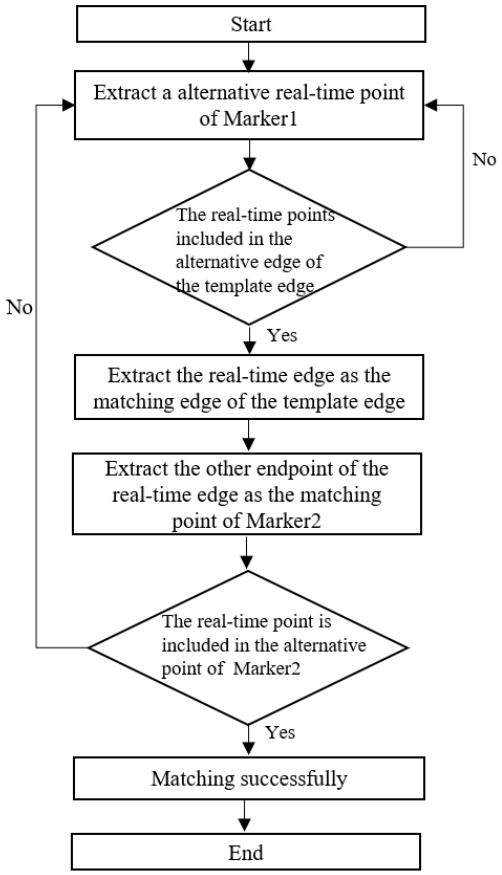


FIGURE 7. The flow chart of searching the alternative real-time edges of one template edge.

frame to the second frame, the velocity and acceleration of the filter are 0, so the predicted position of the feature points of the second frame will coincide with that of the first frame. To address this problem, The method proposed in [41] is utilized to determine the trajectory of each marker in the first three frames. From the fourth frame on, the markers are tracked by Kalman filter, and all the markers are matched to the template points according to the previous frame. Once the tracking object is missing, then we employ the above graph matching method to reconstruct the predefined template.

B. MISSING MARKER RECONSTRUCTION WITH BONE CONSTRAINT BY LSTM

Recent researches have proven that deep neural network can achieve art-of-state to denoise motion data and cover missing markers, they are able to model more suitable spatial and temporal correlations of human motion. In these methods, missing marker reconstruction is resolved as a separate task. However, in practical application, the motion of markers is enslaved to the bone constraint. Inspired by the thought, we propose a data pre-processing operation to introduce bone constraint into original motion data. The processed motion data represents the variation in the relative position of adjacent markers. In the training phase, the LSTM model is trained with the input of the procedural motion sequence, the trained model has the characteristic that human bones remain unchanged with time.

1) STANDARDIZING DATA

The motion capture data of human body records the spatial position and motion trajectory information of markers. Generally, motion capture data is directly represented as the form of three-dimensional coordinates of the markers in a world coordinate system. The motion data is stored with translation and rotation information of each marker, the rotation information is represented by Euler angles. The representation of MOCAP data in the world coordinate system can be decoded from AMC and ASF files, such representation of MOCAP data by coordinate is complex and changeable in the world coordinate system even for similar movement. Feng *et al.* proposed a method to standardize MOCAP data [42], [43], their method translated the coordinates of the markers into local coordinates relative to the root by translation and rotation. This kind of standardization reflects the similarity of motion data better, but only benefits to coordinate recover, and still does not take the bone constraint into account. Different from the researches of Feng *et al.*, in this paper, the 3D coordinates of each marker in the world coordinate system are transformed into the 3D local coordinates, which indicates the coordinate difference between adjacent markers.

The relative distance between two markers on the same bone represents the length of the bone. However, different people are likely to have different bone lengths, leading to a large difference in the standardized coordinates even generated by the same motion path and mode. Therefore, we should first standardize the bone length of the motion sequence. Human configuration can be divided into multiple bone chain. Supposing that there is a bone chain consisted of multiple joints, denoted as $\mathbf{C^i} = \{\mathbf{P_1^i}, \mathbf{P_2^i}, \cdots, \mathbf{P_n^i}\}$, i is the serial number of bone chain, n is the total number of joints in the bone chain. The standardization process of bone length is shown in Algorithm 2,

$$y_k^s = \Delta y_{k-1} + \frac{y_k^i - y_{k-1}^i}{\left\| y_k^i - y_{k-1}^i \right\|_2} L_{k,k-1}, \tag{1}$$

$$\Delta y_k = y_k^s - y_k^i, \tag{2}$$

Y_k^i is the initial coordinates of point P_k^i, which is the child node of P_{k-1}^i. $L_{k,k-1}$ is the bone length between the joints of $\mathbf{P_k^i}$ and $\mathbf{P_{k-1}^i}$. Y_k^s is the standardized coordinates of point P_k^i. The standardized MOCAP data is denoted as $\mathbf{F} = \{\mathbf{F_1}, \mathbf{F_2}, \cdots, \mathbf{F_N}\}$, N is total number of frames. The jth frame standardized MOCAP data is denoted as $\mathbf{F_j} = [x_{j1}, y_{j1}, z_{j1}, x_{j2}, y_{j2}, z_{j2}, \cdots, x_{jn}, y_{jn}, z_{jn}]$. x_{jm}, y_{jm} and z_{jm} is the X-axis, Y-axis and Z-axis coordinates of mth joints of jth frame, respectively.

At this point, $\mathbf{F}$ still represents the MOCAP data in the world coordinate system, our purpose is to transform the coordinates into local coordinate system. An elementary transformation matrix is defined according to the hierarchy of joints. $\mathbf{F1} = \mathbf{E} \times \mathbf{F}$ denotes the difference in coordinates

Algorithm 2 The Standardization Process of Bone Length

Input: $\mathbf{Y^i}$ /☆ Y^i is the set of the initial coordinates of points in $\mathbf{C^i}$ ☆/

Output: $\mathbf{Y^s}$ /☆ Y^s is the set of the standardized coordinates of points in $\mathbf{C^i}$ ☆/

1: $\Delta y_1 = 0$, $y_1^s = y_1^i$ /☆ y_1^s is the standardized coordinate of $\mathbf{P_1^i}$, y_1^i is the initial coordinate of $\mathbf{P_1^i}$, Δy_1 will be explained in the following part ☆/

2: **for** $k = 2 : n$ **do**

3: Calculate y_k^s according to Eq. (1)

4: Calculate Δy^k according to Eq. (2)

5: **end for**

between the markers connected by the same bone throughout the motion sequence. For example, if points P_1^i, P_2^i and P_3^i are located on the same bone, then $\mathbf{F1_j}$ can be denoted as $\mathbf{F1_j} = [x_{j1}, y_{j1}, z_{j1}, x_{j2}-x_{j1}, y_{j2}-y_{j1}, z_{j2}-z_{j1}, x_{j3}-x_{j2}, y_{j3}-y_{j2}, z_{j3}-z_{j2}]^T$. The transformed coordinates represent the difference of adjacent markers located on the same bone, and it reflects the relative positions of each marker. To take advantage of the bone constraint, the motion data need to be processed by Eq. (3) further,

$$\mathbf{F2} = sign(\mathbf{F1} \odot \mathbf{F1}), \tag{3}$$

$\odot$ represents Hadamard product, Eq. (3) can calculate the square of every element of **F1**. At the same time, symbolic information is retained to eliminate the ambiguity of relative position brought by square. The symbol for the missing markers in **F1** can inherit from the corresponding point of the adjacent frame. Given that the bone length does not change, the sum of the absolute values of the three coordinates of the same point in **F2** is a constant, as shown in Eq. (4),

$$\mathbf{T_a} \times \mathbf{F2_b} = \mathbf{T_a} \times \mathbf{F2_c} = \mathbf{L}, \tag{4}$$

L is the bone length matrix, which is consisted of a bone length of the bone chain. $\mathbf{T_a}$ is the transformation matrix, which is used to compute the sum of the absolute values of the three coordinates of the same point in **F2**. The bone length invariance of MOCAP data can be kept by maintaining this constraint during missing marker reconstruction. $\mathbf{F2_b}$ and $\mathbf{F2_c}$ are any two frames of MOCAP data in **F2**. **F2** is the MOCAP data after standardization with bone constraint, the standardized processing reduces the variability of similar motion data. Meanwhile, the whole process is reversible, as shown in Eq. (5),

$$\mathbf{F} = \mathbf{E}^{-1}\{sign(\mathbf{F1}) \odot \sqrt{sign(\mathbf{F1}) * \mathbf{F2}}\}, \tag{5}$$

To show that the above process does not destroy the correlation of motion sequence, the singular value spectrum diagram of the original motion sequence **X** and the transformed motion sequence **F2** are shown in Fig. 8. The curve of **F2** and **X** approach 0 at the same speed, or even **F2** approaches 0 first, indicating that the transformed motion sequence **F2** has a lower rank and its data correlation is not reduced or even higher than the original data.

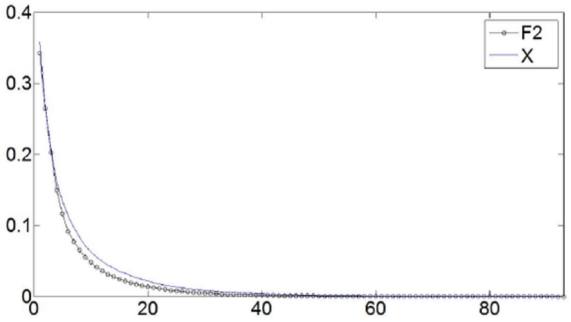


FIGURE 8. The singular value spectrum diagram of original motion sequence **X** and the transformed motion sequence **F2**.

2) TRAINING LSTM MODEL WITH TRANSFORMED MOTION SEQUENCE

Missing marker reconstruction can be regarded as mapping motion sequence with missing points to a complete motion sequence, Long-Short Term Memory (LSTM) has shown the powerful effect for sequence-to-sequence mapping [18]. Our network is enlightened by [18]. Motion data with missing marker is derived from complete original complete motion data, details will be explained in Section 4. Supposing that x is the true pose, and $\bar{x}$ is the corrupted pose. $\bar{x}$ can be mapped to x by Eq. (6).

$$x \approx C^{-1}(\bar{x}), \tag{6}$$

C is the mappint from $\bar{x}$ to x, we use LSTM model as function approximation. While training the LSTM model, Gaussian additive noise is also added into the corrupted pose as Eq. (7),

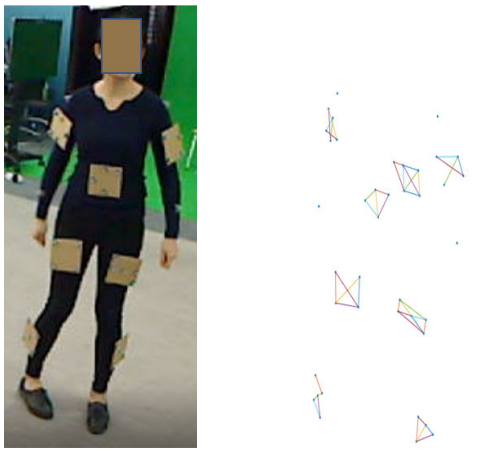
$$\bar{x}_t = M_t(\bar{x} + \mathcal{N}(0, \sigma(X) * \alpha)), \tag{7}$$

$\sigma(X)$ is the standard deviation of training dataset, α is a coefficient of proportionality, it is set as the value of 0.3 [18]. The input of LSTM model is the standardized motion data with bone constraint, which represents the relative position change of the adjacent marker points, the output of LSTM is transformed to the original coordinate representation of markers by Eq. (5). Though this way, the trained LSTM model can not only mine the correlation between successive frames but also maintain the spatial correlation between markers in every frame.

IV. EXPERIMENT AND EVALUATION

In this section, we present the experiment results of labelling marker trajectory and missing marker reconstruction. The evaluation of labelling marker trajectory is based on our motion dataset, while evaluation of missing marker reconstruction is based on CMU Mocap Dataset.

Our motion dataset was captured by MotionAnalysis, which included 16 infrared fish-eye cameras. Our motion dataset contains 21 motion sequences, including 4 motion sequence with one subject and 17 motion sequence with 2 subjects. Motions sequence with one subject has 36, 40, 50, 60 marker points, motion sequence with two subjects has 70, 80, 90, 100, 110, 120, 130, 140, 150, 160,170, 180



(a) (b)

FIGURE 9. Real motion capture scene and its corresponding MOCAP data with 40 markers. (a) presents the marker location on human body in real capture scene. (b) shows the reconstructed template in the corresponding motion sequence.

FIGURE 10. Marker position on the subject, the markers in the red circle is used in our test.

190, 200 marker points, all these motion data are presented as 3D coordinates in a predefined word coordinate system, and sampled at the rate of 60Hz. Fig. 9 shows a frame of real motion capture scene and its corresponding MOCAP data. The activities in each motion sequence include multiple common actions, like jumping, walking, running, Tai Chi, and so on. There are also some interactive activities between the motion sequences with two people.

To evaluate the effect of our labelling method, we define a template with 36 points, and then count ratio of the frames with the successfully reconstructed template in the motion sequence, we denote the ratio as RF. Fig. 10 shows the predefined template. As for missing marker reconstruction error calculation, we use Root Mean Squared Error (RMSE) over the missing markers.

CMU Mocap Dataset is a big database for motion capture. In Fig. 10, we show the marker location during motion

TABLE 11. The Key setting of LSTM model.

Network types	Width	Depth	batchsize	optimizer
LSTM	1024	2	32	Adam optimizer

capture, we can see that there are 41 markers on the subjects. In the experiment, each MOCAP sequence is turned into the hips-centre coordinate scheme. 3D coordinates of the joints are generated by translating joint angles from the BVH file, the source code of 3D coordinate extraction is provided by the paper [18]. The validation dataset consists of the 2 motion sequences from pantomime, sports, jumping and general motion. The test dataset includes motion sequences of basketball, boxing and jump. The training dataset includes all the 25 motion sequence folders except for the validation dataset and the test dataset.

Our labelling method is implemented by MATLAB2017, the motion data is exported from Cortex software, which is the professional tools for building skeleton, monitoring motion, modelling dynamics and camera calibration. The methods for standardizing motion capture data are implemented by Python 3.6, the source code provided in [18] is employed as the implementation code of the LSTM model, whose setting is shown in Table 11. We conduct our experiment on RTX2070 GPU, 16GB RAM and Intel i7-8700 CPU 3.19 GHz. The operating system is Ubuntu 16.04.

A. LABELLING EVALUATION

In this part, we will evaluate the proposed marking method from three aspects based on our MOCAP data. First, we test different values of the scalable range of template edges. Then we calculate and analyse the RF value for successfully identifying the predefined template in the motion sequence with different markers. At last, we present some successful labelling examples.

1) THE SCALABLE RANGE OF TEMPLATE EDGES

Theoretically, our labelling method can achieve good results for rigid bodies, since rigid bodies have constant intrinsic structure. When the motion capture objects are non-rigid body, the internal structures dynamically change, thus appropriately increasing the value of the scalable range of template edges, which is denoted as the threshold in the following introduction, can improve the success rate of labelling. However, the larger threshold will lead to more real-time points and real-time edges per retrieval, thus the labelling process will cost more time. In this experiment, we estimate the value of the threshold through experience, we test different values of the threshold from 2 to 40. We randomly pick 100 frames from each motion sequence of our dataset, and obtain the average value of RF for very threshold, the experimental result is shown in Fig. We can see that the average RF value becomes larger with the increase of the threshold, while after the value of the threshold reaching to 20, the increase in average RF value starts to be very slow.

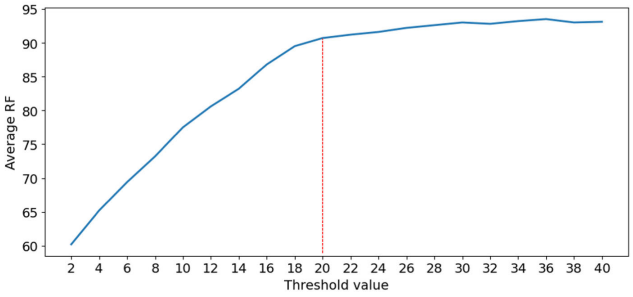


FIGURE 11. The average RF values corresponding to different threshold values.

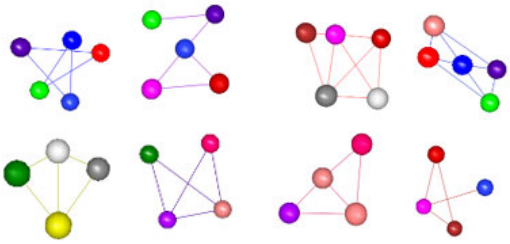


FIGURE 12. The structure of the predefined template.

Increasing the threshold further would bring more computation time but limited improvement, so we set the value of the of the scalable range of template edges as 20. Our further test results show that our method can successfully generate predefined templates in most common human poses.

2) LABELLING RESULT ANALYSIS

This part introduces the overall performance of our method across motion sequence with a different number of markers by using a predefined template, as shown in Fig. 12. We conduct two experiments, first we test our labelling method on the motion sequences whose all markers coincide exactly with the template points, then we proceed our experiment on the motion sequences with a different number of makers. We divide the valid motion sequence into three parts. In the first part, the activity of the subjects in the motion sequence is changing from a T-pose to a normal walking status. In the second part, the subjects walk at normal speed. In the third part, the subjects do movements with a larger range of motion, like jumping, ticking, and so on. The difficulty of labelling increases with the increase in the size of the movement.

For the first experiment, we find that the RF value can reach 100% for the first part motion sequence, 98.5% for the second part motion sequence, and 93.8% for the third part motion sequence. In the absence of interference of other points, our method can construct the template completely for T-pose. RF decreased by 1.5% as the size of movement becomes bigger, the result is almost perfect for normal human motion. As for the hardest part of motion sequence, the third part, the value of RF can still reach 93.8%, that is an ideal result in practice, and it achieves the effect similar to that of Cortex through our observation.

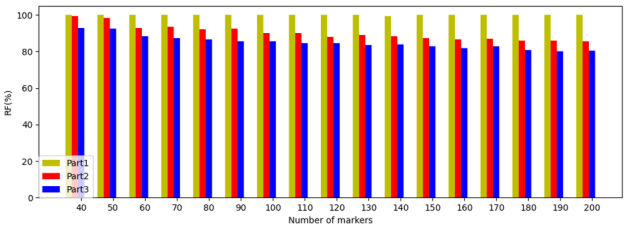


FIGURE 13. The experimental result with the introduction of interferential markers.

TABLE 12. The value of *RF of our method and Cortex on part* 2 *motion sequences.*

Method	40marker	80marker	120marker	160marker	200marker	Average
Ours	99.5%	92.0%	88.0%	86.5%	85.5%	90.3%
Cortex	99.0%	93.0%	90.5%	86.5%	87.0%	91.5%

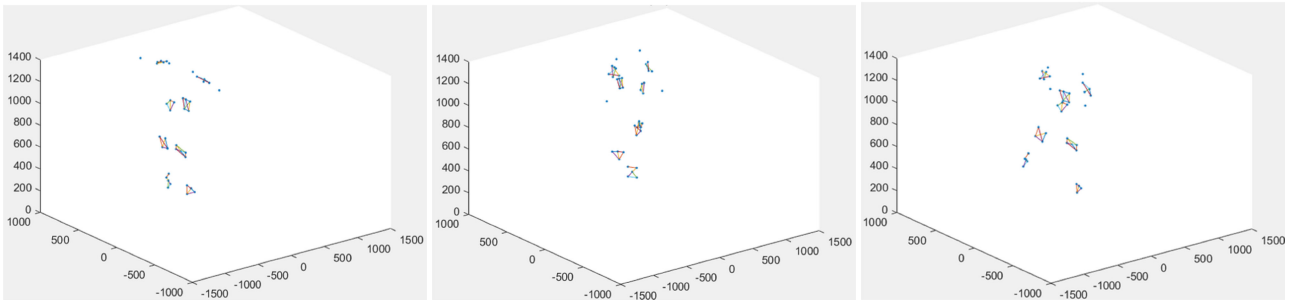
The second experiment introduces the disturbance of other irrelevant markers, the result is shown in Fig. 13. We can see that the disturbance markers do not influence the effect of T-pose labelling, the value of RF reaches 100% on all parts of the motion sequence. As long as the size of movement becomes bigger, the performance of our method decreases, the value of RF drops 14.6% at most on part 2 motion sequence, and 19.1% at most on part 3 motion sequence. Meanwhile, the increase of the number of markers also gets the value of RF down, the value of RF drops 12.6% on part 2 motion sequence, and 12.1% on part 3 motion sequence. Further observation found that some failure labelling was due to the failure of marker reconstruction, the actual performance of our method should be better. The experimental results can still prove the effectivity of our method.

We compare our method to Cortex and test the performance of our method and Cortex on 5 part 2 motion sequences with 40, 80, 120, 160, and 200 markers, respectively. The results are shown in Table 12, we can see that our method can achieve similar result comparing to Cortex software, the average value of RF is only 1.2% difference.

We present the labelling samples of 10th, 310th, and 610th frame on motion sequence with 40, 80, 120, and 160 markers, motion sequence with 40 markers have only one subject, while the rest motion sequences have two subjects. Although the reconstructed markers in some motion sequences with two subjects are quite dense, our method can still recognize the template successfully in both subjects, as shown in Fig. 14, Fig. 15, Fig. 16 and Fig. 17. We also present a sample of walking motion sequence for the full-body template in Fig. 18, the result proves the applicability of our method further.

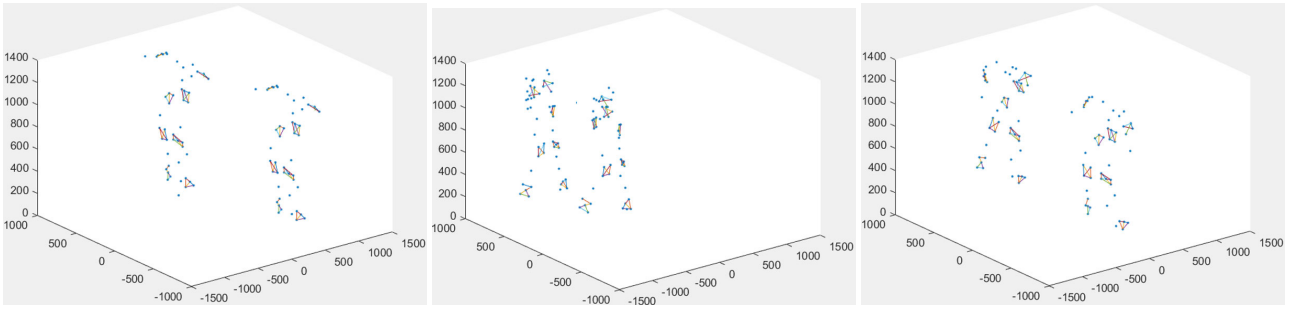
B. MISSING MARKER RECONSTRUCTION EVALUATION

To prove the effectivity of our data preparation method on LSTM-based missing marker reconstruction, we chose 12 markers which constitute the bones of the arm and the calf from CMU dataset, other markers are not included in the



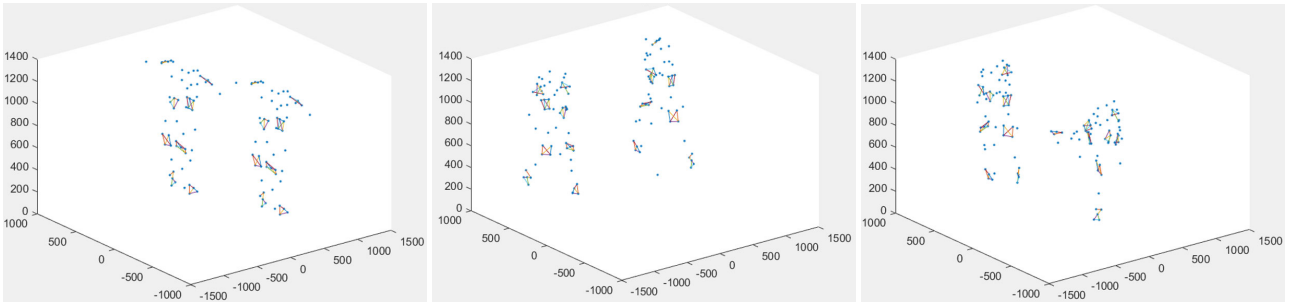
(a) The 10th frame (b) The 310th frame (c) The 610th frame

FIGURE 14. The visualization of labelling results of motion sequence with 40 markers. (a) is the result of 10th frame;(b) is the result of 310th frame;(c) is the result of 610th frame.



(a) The 10th frame (b) The 310th frame (c) The 610th frame

FIGURE 15. The visualization of labelling results of motion sequence with 100 markers. (a) is the result of 10th frame;(b) is the result of 310th frame;(c) is the result of 610th frame.



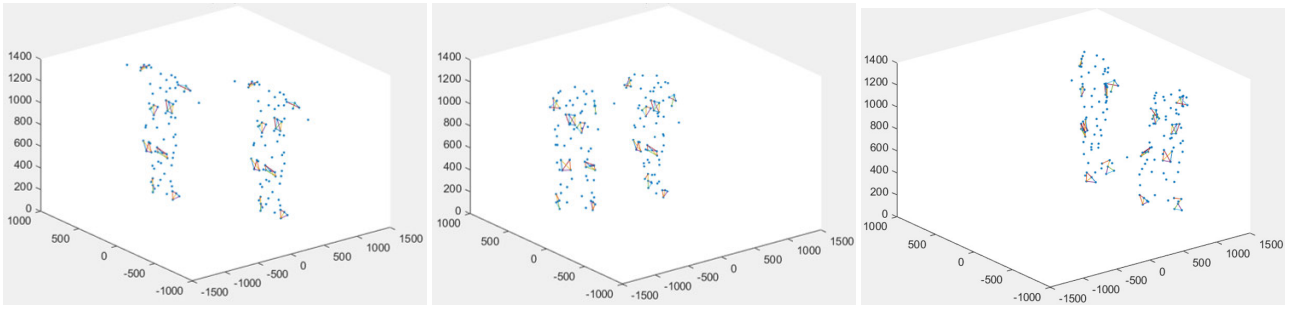
(a) The 10th frame (b) The 310th frame (c) The 610th frame

FIGURE 16. The visualization of labelling results of motion sequence with 120 markers. (a) is the result of 10th frame;(b) is the result of 310th frame;(c) is the result of 610th frame.

TABLE 13. Comparison results of *RMSE value under the condition of 2 missing markers and 4 missing markers with 0.1 duration gap. The unit of measurement is centimeter.*

Methods	Two missing markers			Four missing markers		
	Basketball	Boxing	Jump turn	Basketball	Boxing	Jump turn
Interpolation	0.62	0.99	1.81	0.74	1.31	2.50
Taras[18]	1.42	1.66	3.01	1.58	1.81	3.62
Ours	1.35	1.48	2.18	1.52	1.62	2.43

evaluation, as shown in Fig. 10. The length of the bone can be obtained from the CMU dataset. For the test, we delete two points(a wrist joint and an ankle joint) and four points(two wrist joint and two ankle joints) in the motion sequence, respectively, as shown in Fig. 19. The experimental setup follows an experiment in [18], which is the baseline of our experiment, the only difference is that we employ the 12 markers which are shown in Fig. 10(a) instead of all



(a) The 10th frame (b) The 310th frame (c) The 610th frame

FIGURE 17. The visualization of labelling results of motion sequence with 160 markers. (a) is the result of 10th frame;(b) is the result of 310th frame;(c) is the result of 610th frame.

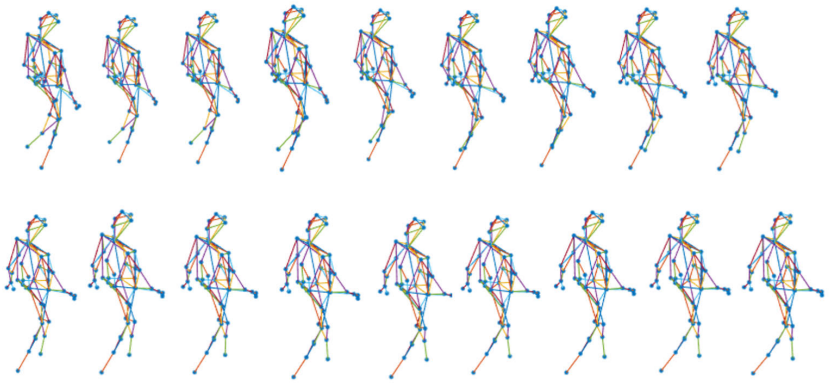
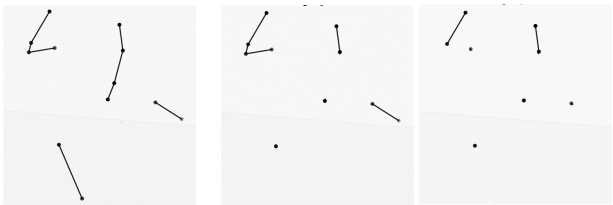


FIGURE 18. The fully template labelling result of a walking motion sequence.

41 markers, and the missing points are specified rather than random. We repeat the experiment of Taras method [18] with all 41 markers according to their guidance, Taras method inputs original motion data into the LSTM model without our motion data preprocessing method, and we only calculates the RMSE of the specific 12 markers in the test. Besides, a standard interpolation method is accomplished and tested on the 12 markers motion sequences. The evaluation is divided into two parts, the first experiment part sets the duration gap of missing markers as 0.1s. In the second part experiment, the length of the gap is set as 100 frames, 150 frames, 200 frames, 250 frames and 300 frames, respectively.

Table 13 shows the experimental results of the first part experiment. We can see that the increase in missing markers makes RMSE get higher, and declines the performance of our method. As the missing markers increase from 2 to 4, the RMSE of the interpolation method goes up by 0.38 on average, 0.31 for Taras method and for ours. Our method has the smallest RMSE change as the number of missing markers increases, the results also show that the stability of our method is the highest. The interpolation method obtains the least this is due to the short duration gap. The RMSE value of our method is 0.36 lower than Taras method for average with two missing markers, and 1.5 for average with four markers. Our method can achieve better results as the missing



(a) (b) (c)

FIGURE 19. The 12 markers used for markers and their connection. (a) is the unbroken human skeleton structure; (b) is the human skeleton with a missing wrist joint and an ankle joint; (c) is the human skeleton with two missing wrist joints and two ankle joints.

markers increase. The result indicates that the new representation of the motion sequence generated by our method makes an impact in our experiment.

In the second part experiment, we show the adaptive capacity of our method to the different length of gaps, the experimental results are shown in Fig. 20. The interpolation method fails to reconstruct the missing markers with a long gap. Both our method and Taras method have strong adaptability to the long gap. Moreover, our method shows lower RMSE than Taras method for each frame gap, and the longer the gap, the smaller the RMSE of our method. It proves that our method has stronger adaptability to long gaps in our experiment, even better even better than Taras method.

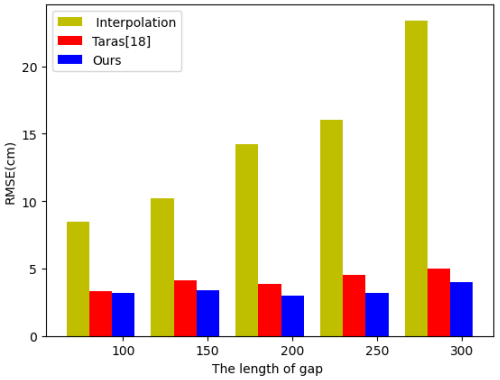


FIGURE 20. The comparison of the average value of *RMSE with different length of gap.*

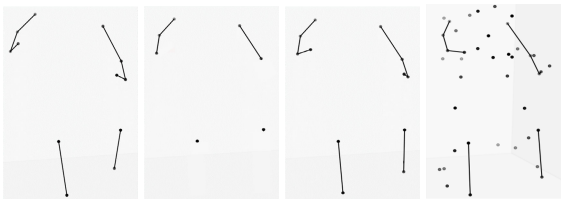


FIGURE 21. Reconstruction result of motion sequences with four missing markers by our method and Taras method.(a) is the groud-truth, (b) is the sample with four missing markers, (c) is the missing marker reconstruction sample by our method, (d) is the missing marker reconstruction sample by Taras method.

In Fig. 21, we show the visualization results of Taras method and our method on motion sequences from CMU dataset. The number of missing markers is four. We can see that the missing marker reconstruction result of our method more like the ground-truth rather than the result of Taras method.

V. CONCLUSION

This study presents a novel trajectory labelling method based on graph matching and a motion data representation for missing marker reconstruction. After defining a template manually whose scalable range of the template edges is set as 20mm empirically, the predefined graph template is traversed to match the real-time scattered points successfully in a single frame, and a Kalman filtering based tracking strategy is employed to track consecutive motion trajectory. Experiments show that our method can label the trajectory of most common human motion, and achieve similar labelling result comparing to Cortex software. To reconstruct the missing markers, we propose a new motion data presentation with information of variation in the relative position of adjacent markers. The experiment shows that the new presentation improves the accuracy and stability of the LSTM model, which is the art-of-state in missing marker reconstruction.

Since there are lots of traversal search and comparison operations in our labelling method, in the future, we intend to optimize the labelling method further. We also plan to improve the tracking strategy to combine the missing marker reconstruction method.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their suggestions to make our paper more perfect. They also thank Beijing Nokov Technology Company Ltd., for providing Cortex, Multiple Cameras Systems (MCSs), and 3D motion capture dataset.

REFERENCES

[1] J. Dong, Q. Shuai, Y. Zhang, X. Liu, X. Zhou, and H. Bao, "Motion capture from Internet videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 210–227.
[2] D. Xiao, Q. Chen, and S. Li, "A multi-scale cascaded hierarchical model for image labeling," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 09, Nov. 2016, Art. no. 1660005.
[3] Z. Li, K. Nai, G. Li, and S. Jiang, "Learning a dynamic feature fusion tracker for object tracking," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 7, 2020, doi: 10.1109/TITS.2020.3027521.
[4] G. Li, M. Peng, K. Nai, Z. Li, and K. Li, "Reliable correlation tracking via dual-memory selection model," *Inf. Sci.*, vol. 518, pp. 238–255, May 2020.
[5] Motion Analysis Corporation. *Cortex*. Accessed: Sep. 20, 2020. [Online]. Available: http://www.tp-ontrol.hu/index.php/TPToolbox
[6] Vicon Corporation. *Vicon*. Accessed: Sep. 20, 2020. [Online]. Available: https://www.vicon.com
[7] Q. Cui, B. Chen, and H. Sun, "Nonlocal low-rank regularization for human motion recovery based on similarity analysis," *Inf. Sci.*, vol. 493, pp. 57–74, Aug. 2019.
[8] S. Ghorbani, A. Etemad, and N. F. Troje, "Auto-labelling of markers in optical motion capture by permutation learning," in *Proc. Comput. Graph. Int. Conf.* Cham, Switzerland: Springer, 2019, pp. 167–178.
[9] S.-J. Li, H.-S. Zhu, L.-P. Zheng, and L. Li, "A perceptual-based noise agnostic 3D skeleton motion data refinement network," *IEEE Access*, vol. 8, pp. 52927–52940, 2020.
[10] Z. Li, S. Gao, and K. Nai, "Robust object tracking based on adaptive templates matching via the fusion of multiple features," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 1–20, Apr. 2017.
[11] S.-K. Oh and H. Park, "Analysis of IDCT and motion-compensation mismatches between spatial-domain and transform-domain motion-compensated coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 835–843, Jul. 2005.
[12] D. Xiao, J. Li, and K. Li, "Robust precise dynamic point reconstruction from multi-view," *IEEE Access*, vol. 7, pp. 22408–22420, 2019.
[13] K. Nai, Z. Li, G. Li, and S. Wang, "Robust object tracking via local sparse appearance model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4958–4970, Oct. 2018.
[14] S. Xia, L. Su, X. Fei, and H. Wang, "Toward accurate real-time marker labeling for live optical motion capture," *Vis. Comput.*, vol. 33, nos. 6–8, pp. 993–1003, Jun. 2017.
[15] D. Xiao, Q. Yang, B. Yang, and W. Wei, "Monocular scene flow estimation via variational method," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10575–10597, Apr. 2017.
[16] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann, "Skeleton based motion capture for robust reconstruction of human motion," in *Proc. Comput. Animation*, 2000, pp. 77–83.
[17] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J. J. Zhang, and R. Song, "Exploiting temporal stability and low-rank structure for motion capture data refinement," *Inf. Sci.*, vol. 277, pp. 777–793, Sep. 2014.
[18] T. Kucherenko, J. Beskow, and H. Kjellström, "A neural network approach to missing marker reconstruction in human motion capture," 2018, *arXiv:1803.02665*. [Online]. Available: http://arxiv.org/abs/1803.02665
[19] D. Xiao, X. Yang, J. Li, and M. Islam, "Attention deep neural network for lane marking detection," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105584.
[20] H. Wang, S. Wang, J. Lv, C. Hu, and Z. Li, "Non-local attention association scheme for online multi-object tracking," *Image Vis. Comput.*, vol. 102, Oct. 2020, Art. no. 103983.
[21] U. Mall, G. R. Lal, S. Chaudhuri, and P. Chaudhuri, "A deep recurrent framework for cleaning motion capture data," 2017, *arXiv:1712.03380*. [Online]. Available: http://arxiv.org/abs/1712.03380
[22] L. Li, J. McCann, N. Pollard, and C. Faloutsos, "BoLeRo: A principled technique for including bone length constraints in motion capture occlusion filling," in *Proc. SCA*, 2010, pp. 179–188.

[23] J. Lv, Z. Li, K. Nai, Y. Chen, and J. Yuan, "Person re-identification with expanded neighborhoods distance re-ranking," *Image Vis. Comput.*, vol. 95, Mar. 2020, Art. no. 103875.

[24] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth, "Skeletal parameter estimation from optical motion capture data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 782–788.

[25] V. B. Zordan and N. C. Van Der Horst, "Mapping optical motion capture data to skeletal motion using a physical model," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2003, pp. 245–250.

[26] S. I. Park and J. K. Hodgins, "Capturing and animating skin deformation in human motion," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 881–889, Jul. 2006.

[27] M. Burke and J. Lasenby, "Estimating missing marker positions using low dimensional Kalman smoothing," *J. Biomechanics*, vol. 49, no. 9, pp. 1854–1858, Jun. 2016.

[28] Ø. Gløersen and P. Federolf, "Predicting missing marker trajectories in human motion data using marker intercorrelations," *PLoS ONE*, vol. 11, no. 3, Mar. 2016, Art. no. e0152616.

[29] A. Aristidou, J. Cameron, and J. Lasenby, "Predicting missing markers to drive real-time centre of rotation estimation," in *Proc. Int. Conf. Articulated Motion Deformable Objects*. Cham, Switzerland: Springer, 2008, pp. 238–247.

[30] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, "DynaMMO: Mining and summarization of coevolving sequences with missing values," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 507–516.

[31] Z. Wang, S. Liu, R. Qian, T. Jiang, X. Yang, and J. J. Zhang, "Human motion data refinement unitizing structural sparsity and spatial-temporal information," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 975–982.

[32] D. Holden, "Robust solving of optical motion capture data by denoising," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Aug. 2018.

[33] M. Ringer and J. Lasenby, "A procedure for automatically estimating model parameters in optical motion capture," *Image Vis. Comput.*, vol. 22, no. 10, pp. 843–850, Sep. 2004.

[34] Q. Yu, Q. Li, and Z. Deng, "Online motion capture marker labeling for multiple interacting articulated targets," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 477–483, Sep. 2007.

[35] Y. Xiang, S. Rahmatalla, J. S. Arora, and K. Abdel-Malek, "Enhanced optimisation-based inverse kinematics methodology considering joint discomfort," *Int. J. Hum. Factors Model. Simul.*, vol. 2, nos. 1–2, pp. 111–126, 2011.

[36] L. Herda, R. Urtasun, P. Fua, and A. Hanson, "Automatic determination of shoulder joint limits using quaternion field boundaries," *Int. J. Robot. Res.*, vol. 22, no. 6, pp. 419–436, Jun. 2003.

[37] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools Appl.*, vol. 79, pp. 18201–18219, Mar. 2020.

[38] J. Meyer, M. Kuderer, J. Muller, and W. Burgard, "Online marker labeling for automatic skeleton tracking in optical motion capture," in *Proc. Workshop Comput. Techn. Natural Motion Anal. Reconstruct., IEEE Int. Conf. Robot. Automat. (ICRA)*, Karlsruhe, Germany, 2013.

[39] S. U. Kim, H. Jang, and J. Kim, "Human motion denoising using attention based bidirectional recurrent neural network," in *Proc. SIGGRAPH Asia Posters*, 2019, pp. 1–2.

[40] D. Pavllo, T. Porssut, B. Herbelin, and R. Boulic, "Real-time finger tracking using active motion capture: A neural network approach robust to occlusions," in *Proc. 11th Annu. Int. Conf. Motion, Interact., Games*, Nov. 2018, pp. 1–10.

[41] I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, pp. 56–73, Jan. 1987.

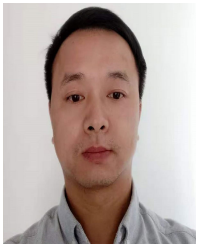
[42] J. Xiao, Y. Feng, M. Ji, X. Yang, J. J. Zhang, and Y. Zhuang, "Sparse motion bases selection for human motion denoising," *Signal Process.*, vol. 110, pp. 108–122, May 2015.

[43] Z. Wang, Y. Feng, S. Liu, J. Xiao, X. Yang, and J. J. Zhang, "A 3D human motion refinement method based on sparse motion bases selection," in *Proc. 29th Int. Conf. Comput. Animation Social Agents*, May 2016, pp. 53–60.

JIANFANG LI is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include stereoscopic evaluation algorithms and motion capture.

DEGUI XIAO received the B.E. degree in industrial automation from the Wuhan University of Textile, Wuhan, China, in 1994, and the Ph.D. degree in computer science and technology from the Huazhong University of Science and Technology, Wuwhan, in 2003. He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His current research interests include image and video processing, computer vision, and edge computing.

KEQIN LI (Fellow, IEEE) is currently a SUNY Distinguished Professor of computer science with the State University of New York. He is also a Distinguished Professor with Hunan University, China. He has published over 760 journal articles, book chapters, and refereed conference papers. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber- physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, and intelligent and soft computing. He has received several best paper awards. He has served on the editorial boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.



JIAZHI LI is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include computer vision and face recognition.