# An Automatic Solution Framework for Robust and Computationally Efficient Joint Estimation in Optical Motion Capture

Jianwei Hang<sup>†</sup>, Joan Lasenby and Adrian Li<sup>\*</sup> Engineering Department University of Cambridge Cambridge, United Kingdom. Email: {jh845, jl221}@cam.ac.uk; adrian.li@cantab.net

Abstract—We propose an automatic and computationally efficient solution framework for addressing the joint estimation problem in marker-based optical motion capture. A fast joint estimator is presented which only requires an optimisation over 3 variables using marker-trajectory-bases (MTB). We also introduce the theory of *solvability propagation* to realise this automation. The framework acquires a 'hybrid' power making it able to deal with difficult cases where there are less than three markers on the body segments. It does this by combining the MTB-based and the *joint-marker-variance optimisation* methods. Computer simulations are used to examine the framework in terms of accuracy, speed and functionality. Results from these simulations show that the framework is robust and produces fast and accurate solutions.

Index Terms—Joint Estimation, Optical Motion Capture.

### I. INTRODUCTION

Marker-based optical motion capture (in short, Mo-Cap) is a technology that transforms human motion activities into 3D (x,y,z) position data. Mo-Cap is widely used for applications such as abnormality and asymmetry analyses in rehabilitation medicine [1], performance analysis in sports training [2], computer animations and 3D games in entertainment [3]. By using a multi-camera recording system, the 3D positions of the reflective markers which are placed on a bodysuit worn by a performer are recorded. The marker position data is then processed to reconstruct the body motion. The entire process of reproducing the body motion consists of three main parts: *Marker Identification, Topology Determination*, and *Joint Estimation*.

Marker identification is a matter of recognising markers and their placements on the body segments in each frame from the point positions obtained by the multi-camera recording system. Markers placed on the same limb will be gathered to form a marker group in this process. Many methods have been proposed for the marker identification problem over the past decade, e.g. *Spectral Clustering* [4], *Self-tuning Spectral Clustering* [5], *Model-based Dynamic Point Matching Algorithm* [6], etc. Topology determination determines how the body segments are connected by the joints to give a correctly rebuilt human skeleton. This is normally achieved by computing the *minimum spanning tree* [7] for a complete graph containing weighted linkages between all pairs of marker groups.

Joint estimation is a crucial process which determines the trajectories of joints between pairs of body segments, treating the human body as an articulated structure. Many research papers have focused on calculating the location of joints or in other words, the centres of rotation (CoR). For example, Kirk et al. in [8] define a joint cost which is the mean variance in distance between the joint and each marker. Under the practical assumption that a joint connecting two body segments should remain at a relatively constant distance to the markers on both segments, finding the joint locations essentially turns into a matter of minimising the joint cost. This approach is referred to in this paper as Joint-Marker-Variance Optimisation, or simply JMV Optimisation. Gamage and Lasenby in [9] introduce a closed-form least squares solution to the CoR or axis of rotation (AoR), using a cost function of the squared differences between unknown sphere radii and squared CoR-to-marker distances. Silaghi et al. in [10] employ a modified Levenberg-Marquardt method for least squares computations of all CoRs. The joint position is then estimated as the centre of mass of the CoRs weighted by the weights of the markers and the associated sphere radius. Cameron and Lasenby in [11] assume all markers are attached to rigid body segments and present a sequential algorithm for real-time joint localisation. Both Tan et al. [12] and Li [13] establish a local coordinate system using on-body markers to determine the joint positions. Tan et al. solve a series of overdetermined equations while Li proves the joint trajectory is in the space termed *eigentrajectory space* in [13]. However, these approaches are either computationally inefficient or need to satisfy hard conditions: the methods in [8] and [9] do not scale well and suffer from the optimisation time growing with the increase in the total number of markers and recorded frames; the methods in [10] and [11] require at least three markers on each body segment to yield reasonable results, whilst a further non-collinearity in three markers on a body segment should

<sup>&</sup>lt;sup>†</sup> Corresponding author. Email: jh845@cam.ac.uk.

<sup>\*</sup> Previously with University of Cambridge but now with Google.

be ensured when performing the methods in [12] and [13].

In this paper, we are concerned with the joint estimation problem and develop a computationally efficient and robust solution framework which is capable of automatically finding the motion trajectories of the joints of an articulated body structure. In this framework, a fast joint localisation method is proposed which only needs to optimise three variables via the use of marker trajectory bases (MTB), regardless of the total number of markers and recorded frames. This method is combined with the inefficient but condition-free JMV Optimisation approach to form a hybrid system which is robust to tough situations by smartly toggling to the appropriate joint estimation approach. Along with the solvability propagation theorem, the framework is able to automatically find all joint trajectories of an articulated body structure without intervention or tuning of parameters. Simulated results show that our solution framework is able to survive with reasonable results in tough cases where there are less than three markers on all body segments, and even performs slightly better in terms of speed than the commonly considered efficient approach using sequential algorithms as described in [11] in cases where there are at least three markers on each body segment. Note that since we are solely addressing the joint estimation problem, the process of marker identification is assumed to have already been done prior to this framework so that markers have been recognised and knowledge of marker placement has been gained in advance.

The rest of the paper is organised as follows: the MTB-based joint estimation method and the solvability propagation theorem are described in Section II; the structure of the framework is explained in Section III; simulated results of our framework are compared with that of some other existing joint estimation approaches in Section IV in terms of speed, accuracy and functionality; the conclusions are given in Section V.

#### II. THEORY AND METHODOLOGY

In this paper, markers belonging to the same group are assumed to move together as a rigid body. The proposed MTB-based joint estimation approach is governed by the *joint solvability* while the automation of the framework is handled by the *solvability propagation*.

## A. Joint Solvability

If a joint is connected to a rigid body with four non-coplanar markers, then the trajectory of this joint over all frames will lie in the space spanned by the trajectories of the markers. Such a joint is termed as a 'solvable' joint and can be expressed by a linear combination of the marker trajectories.

*Proof.* Suppose the 3D positions of the four non-coplanar markers on a rigid body at an arbitrary frame number f are denoted as  $r_1^{(f)}, r_2^{(f)}, r_3^{(f)}$  and  $r_4^{(f)}$ . A joint connected with that body segment has a position labelled as  $r_c^{(f)}$ . The trajectories of the markers and joint over all frames are termed  $t_1, t_2, t_3, t_4$  and  $t_c$ , respectively. In a reference frame R, three basis vectors can be obtained from the marker positions, making

 $r_4^{(R)}$  the origin:  $a_1^{(R)} = r_1^{(R)} - r_4^{(R)}$ ;  $a_2^{(R)} = r_2^{(R)} - r_4^{(R)}$ ; and  $a_3^{(R)} = r_3^{(R)} - r_4^{(R)}$ .

Since the four markers are non-coplanar, the basis vectors will span the  $\mathbb{R}^3$  space, though they may not be orthonormal to each other. As in [12] and [13], the origin-to-joint vector can therefore be expressed as a linear combination of the basis vectors:

$$r_c^{(R)} - r_4^{(R)} = \lambda_1 a_1^{(R)} + \lambda_2 a_2^{(R)} + \lambda_3 a_3^{(R)}$$
(1)

from which the joint position can be deduced:

$$r_c^{(R)} = r_4^{(R)} + \lambda_1 a_1^{(R)} + \lambda_2 a_2^{(R)} + \lambda_3 a_3^{(R)}$$
(2)

Suppose the rigid body is brought from the reference frame to an arbitrary frame f by a rotation  $R^{(f)}$ , the resulting rotated basis vectors will be:

$$a_i^{(f)} = R^{(f)} a_i^{(R)} \qquad i = 1, 2, 3$$
 (3)

and the rotated origin-to-joint vector in frame f becomes:

$$r_c^{(f)} - r_4^{(f)} = R^{(f)} \left[ r_c^{(R)} - r_4^{(R)} \right]$$
(4)

Therefore, using (2) (3) (4), the joint position in frame f can be obtained:

$$r_{c}^{(f)} = r_{4}^{(f)} + R^{(f)} \left[ r_{c}^{(R)} - r_{4}^{(R)} \right]$$
  
=  $r_{4}^{(f)} + R^{(f)} \left[ \lambda_{1} a_{1}^{(R)} + \lambda_{2} a_{2}^{(R)} + \lambda_{3} a_{3}^{(R)} \right]$   
=  $\lambda_{1} r_{1}^{(f)} + \lambda_{2} r_{2}^{(f)} + \lambda_{3} r_{3}^{(f)}$   
+  $(1 - \lambda_{1} - \lambda_{2} - \lambda_{3}) r_{4}^{(f)}$  (5)

The trajectory of the joint can be obtained via stacking the joint's 3D positions calculated in (5) over all frames:

$$t_{c} = \lambda_{1}t_{1} + \lambda_{2}t_{2} + \lambda_{3}t_{3} + (1 - \lambda_{1} - \lambda_{2} - \lambda_{3})t_{4}$$
(6)

where

$$t_i = \left[r_i^{(1)}, r_i^{(2)}, \cdots, r_i^{(f)}\right]^{\mathrm{T}}$$
  $i = c, 1, 2, 3, 4$ 

As shown in (6), the trajectory of the joint over all frames is essentially a linear combination of the trajectories of the four non-coplanar markers.  $\Box$ 

To obtain the trajectory of the joint, we minimise the cost function defined in [8]:

$$\hat{t}_c = \arg\min_t Q \tag{7}$$

with

$$Q = \frac{1}{N_a + N_b} \sum_{m \in N_a \cup N_b} \sigma(r_c, r_m) + \alpha \bar{d}(r_c, r_m)$$
(8)

where  $\bar{d}(r_c, r_m)$  is the average distance between a marker and a joint:  $\bar{d}(r_c, r_m) = \frac{1}{|F|} \sum_{f=1}^{F} ||r_c^{(f)} - r_m^{(f)}||$ , with F being the total number of frames;  $\sigma(r_c, r_m)$  is the corresponding variance in distance:  $\sigma(r_c, r_m) = \frac{1}{|F|} \sum_{f=1}^{F} (||r_c^{(f)} - r_m^{(f)}|| - \bar{d})^2$ ;  $\alpha$  is a penalty factor weighting the average distance  $\bar{d}$ ;  $N_a$  and  $N_b$ are the number of markers on the two rigid body segments connected by the target joint.



Fig. 1. Block diagram of the proposed solution framework

Note that only three unknowns,  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , are optimised during the process of minimising the cost function.

The joint solvability is also applicable to the case where a rigid body segment is formed by three non-collinear markers, with the aid of an auxiliary marker  $r_{+}^{(f)}$  obtained from the cross-product of the inter-marker vectors:

$$r_{+}^{(f)} = r_{3}^{(f)} + \left(r_{1}^{(f)} - r_{3}^{(f)}\right) \times \left(r_{2}^{(f)} - r_{3}^{(f)}\right)$$
(9)

Since the three actual markers are assumed to remain at a relatively constant distance to each other,  $r_{+}^{(f)}$  can be treated as a virtual marker on the rigid body. In this way, a set of four non-coplanar markers  $\{r_1^{(f)}, r_2^{(f)}, r_3^{(f)}, r_{+}^{(f)}\}$  is obtained for the MTB-based joint estimation method described above.

# B. Solvability Propagation

If a rigid body segment has three non-collinear markers and is connected with at least one already-solved joint, then all other joints connected with that body segment will be 'solvable' joints as long as the already-solved joint and the other three markers are non-coplanar.

*Proof.* The already-solved joint connected to the rigid body segment can be treated as an extra marker on that body, which

brings back the scenario in which the body segment has 'four' markers. If the 'four' markers are non-coplanar, all other joints connected with the rigid body can be solved with the solvability theory described in the previous section.  $\Box$ 

The solvability propagation may also be extended to the case where there are only two markers on a rigid body segment connected with an already-solved joint that is non-collinear with the existing two markers, since an auxiliary marker can be generated (see (9)) based on the marker set consisting of the already-solved joint and the existing two markers.

#### **III. FRAMEWORK EXPLAINED**

In this paper, a group with N markers (we assume no more than four markers are placed on the same limb so  $N \in [1, 4]$ ) is called an N-marker group. Remember that we assume the marker identification process has been done in advance, which means markers are recognised and those belonging to the same limb are grouped together. We also define a new type of marker group named a *special group*. A four or threemarker group becomes a special group when all markers within that group are collinear. The co-planarity checking process governs the determination of a special group and picks only two characteristic markers from a special group, because all other markers can be expressed through linear scaling of the vector formed by the chosen markers.

The overall picture of our framework is illustrated in Fig. 1. The input marker groups are first sorted by the number of markers in each group, resulting in four collections of groups having 4, 3, 2 and 1 marker(s), respectively. The rest of the framework consists of three main subsystems, in order:

- 1) PRE-JOINT SOLVER
- 2) SOLVED-JOINT-DEPENDENT SOLVER
- 3) REDUCED JMV JOINT SOLVER

## A. PRE-JOINT SOLVER

From the theories described in the previous section, joints connected with a rigid body segment having four non-coplanar or three non-collinear markers can be directly solved. Therefore, groups with four or three markers are processed first by the governing subsystem, i.e. the PRE-JOINT SOLVER.

The PRE-JOINT SOLVER takes the three or four-marker groups as inputs. These groups then go through JOINT SOLVER #1. If all markers in a group are found to be collinear by the co-planarity checking process, the corresponding group will be marked as a special group. Otherwise, the trajectories of the joint(s) connected with this group can be obtained based on the joint solvability theory. In this way, all special groups are identified and put into a special 'container', and the joints solved by this subsystem will enable possible solvability propagation in later stages.

One essential variable across the entire framework is the system state which plays a very important role in tracking the implementation and avoiding repeated joint calculations. It consists of two sub-states:

- Joint State: a record of whether a joint is solved.
- Group State: a record of the presence of marker groups connected with unsolved joints.

The system state is checked and updated in every implemented step. Updating the system state consists of two parallel parts:

- The joint state will be updated once a new joint is found.
- The groups whose connected joints are all solved will be removed from the corresponding group collections.

#### B. SOLVED-JOINT-DEPENDENT SOLVER

After implementing the first subsystem, i.e. the PRE JOINT SOLVER, there will be three types of marker groups left: the special four or three-marker groups, the two-marker groups, and the one-marker groups (if they exist). These groups share a common feature that they do not have at least three noncollinear markers, which means the joint solvability theory cannot be directly applied to solve joints connected with these groups. However, if there are already-solved joints connected with these groups (Note: at least two solved joints for the one-marker groups in this case), solvability propagation may be applied so that the conditions for joint solvability may again be satisfied and further joints may be solved. As shown in Fig. 1, the system state is taken as a feedback in a loop that implements this subsystem. In every iteration, the system state is checked. If the previous iteration leads to an update to the system state, i.e.  $state_i \neq state_{i-1}$ , four steps will be implemented by JOINT SOLVER #2, in order:

- Find already-solved joints connected with the 4-marker special groups. Solve all possible joints connected with these groups.
- Find already-solved joints connected with the 3-marker special groups. Solve all possible joints connected with these groups.
- Find already-solved joints connected with the 2-marker groups. Solve all associated joints.
- 4) Find already-solved joints connected with the 1-marker special groups. If there are at least two such joints for a group, solve all associated joints.

If there are no satisfactory solved joints found connected, or the co-planarity checking process cannot be passed, the corresponding groups will be put into the *waiting list* for further possible processing. The subsystem will keep being implemented until the system state no longer updates, which means no more joints can be solved. The rest of the unsolved joints connected with the wait-listed marker groups therefore need to be dealt with by the condition-free *JMV Optimisation* approach.

# C. REDUCED JMV JOINT SOLVER

After the previous two subsystems, all joints that can be solved by the joint solvability and solvability propagation theories have been solved. If there are still unsolved joints, the condition-free *JMV Optimisation* approach serving as a complementary method could then be employed.

However, it is possible that the MTB-based approach can still be used to reduce computational cost in certain circumstances. Consider an example case below for a better understanding.

*Example* Suppose there is a chain of 5 rigid body segments (not limited to a human body in this example – so there can be an arbitrary number of segments), as shown in Fig. 2.



Fig. 2. A chain of 5 rigid body segments

where a circled number is called a node which denotes an N-marker group on the corresponding rigid body segment; an edge linking two adjacent nodes represents the joint connecting two related marker groups.

After going through the first two subsystems of the framework, only the joint linking the first two nodes (from left to right), i.e. (4) and (1), will be solved, if (4) is not a special group. To estimate other joints, one method would be to use the *JMV Optimisation* approach. However, in this example, the *JMV Optimisation* approach need only be applied once and the rest of the joints can then be solved by the primary approach. We outline this in what follows.

Suppose we estimate the joint linking the second and third nodes, i.e. (1) and (2), using the *JMV Optimisation*, the resulting

chain is illustrated in Fig. 3, with the solved joints denoted by thickened edges.



Fig. 3. A chain of 5 rigid body segments with 2 joints solved

It can be seen in Fig. 3 that there is a solved joint connected with the third node which has two markers. As long as the solved joint and these two markers are non-collinear (which is a practical assumption), the solvability propagation can be applied so that all the remaining joints can be solved.

The above example shows that in certain situations, the MTB-based approach can be re-applied after implementing the JMV optimisation for a few joints, which can potentially maintain a low computational cost.

In this subsystem, as shown in Fig. 1, the wait-listed three or four-marker special groups and the two or one-marker groups, are taken as the inputs. The aim is therefore to obtain all remaining trajectories using the *JMV Optimisation* as little as possible, thus reducing the computational cost. To achieve this, four steps are implemented by JOINT SOLVER #3, in order:

- Estimate joints connected with the wait-listed 4-marker special groups one by one using JMV optimisation. Apply the MTBbased approach where possible.
- Estimate joints connected with the wait-listed 3-marker special groups one by one using JMV optimisation. Apply the MTBbased approach where possible.
- 3) Estimate joints connected with the wait-listed 2-marker groups one by one using JMV optimisation. Apply the MTB-based approach where possible.
- Estimate joints connected with the wait-listed 1-marker groups one by one using JMV optimisation. Apply the MTB-based approach where possible.

In each of the above steps, as long as the conditions for solvability propagation are met after a joint is estimated by *JMV Optimisation*, the subsystem SOLVED-JOINT-DEPENDENT SOLVER will be called to perform an automatic joint estimation process until no further joints can be solved using the MTB-based approach, in which case the JMV optimisation is toggled back for the next tough joint. The toggling between the *JMV Optimisation* and the MTB-based approach will continue until all joints are solved, which reveals the *hybrid* power of this framework.

#### **IV. SIMULATED RESULTS**

The framework is validated by simulated results using MATLAB in three aspects: Accuracy, Speed and Functionality. Synthetic chains of rigid body segments connected by joints (see Fig. 2 as an example) are generated to mimic articulated structures. A plain text notation  $\langle N_1 - N_2 - \cdots - N_L \rangle$  denotes a chain of L rigid body segments with  $N_i$  ( $i \in [1, L]$ ) markers on the  $i^{th}$  segment. Each of the rigid body segments rotates relative to its neighbours with randomly generated angular velocities. For simplicity, it is assumed that we do not have three collinear markers on any rigid body segment used in the simulations, which can be easily fulfilled in real world tasks.

TABLE I Root Squared Error per joint per frame in different rigid body chains solved by different joint estimation methods

	RSE per Joint per Frame in			
	Chain#1	Chain#2	Chain#3	
JMV Opt.	1.1E-02	3.4E-03	4.4E-03	
Seq. Algorithm	5.7E-17	3.3E-17	3.1E-17	
Proposed Framework	3.9E-07	2.9E-07	1.8E-07	

#### A. Accuracy

The accuracy of the framework is checked with the *root-squared-error* (RSE) per joint per frame, given by

$$\xi = \frac{\sqrt{\sum_{i=1}^{F} \sum_{j=1}^{M} r_{i,j}^2}}{M \times F}$$
(10)

with

$$r_{i,j}^2 = [x_o(i,j) - \hat{x}(i,j)]^2 + [y_o(i,j) - \hat{y}(i,j)]^2 + [z_o(i,j) - \hat{z}(i,j)]^2$$

where F is the total number of frames; M is the number of joints;  $(\hat{x}(i,j), \hat{y}(i,j), \hat{z}(i,j))$  and  $(x_o(i,j), y_o(i,j), z_o(i,j))$  are the estimated and ground truth 3D positions of the  $j^{th}$  joint at the  $i^{th}$  frame, respectively.

Three chains with an increasing number of rigid body segments are used for the validation. The distance between successive joints is fixed at 1.

chain#1: 
$$\langle 4 - 3 - 3 \rangle$$
  
chain#2:  $\langle 4 - 3 - 3 - 4 - 3 \rangle$   
chain#3:  $\langle 4 - 3 - 3 - 4 - 3 - 3 - 3 - 3 \rangle$ 

The number of frames F is fixed at 100 and 50 runs of simulations are performed for each chain to obtain the average values of  $\xi$ . The results are summarised and compared with those of the approaches using JMV Optimisation [8] and Sequential Estimation [11] in TABLE I.

From TABLE I, the joint estimation approach using the Sequential Algorithm is shown to be error free with  $\xi$  of the order of  $10^{-17}$ . The proposed framework also achieves a reasonable error level of  $10^{-7}$ , regardless of the increase in the number of body segments. The *JMV Optimisation* method, however, is much less competitive with a error level of  $10^{-2}$  to  $10^{-3}$ . Therefore, the accuracy of our framework is high when dealing with rigid bodies with at least three non-collinear markers, which is practically feasible.

# B. Speed

The speed is examined by the time consumed per joint using an Intel<sup>®</sup>Core<sup>TM</sup>i7-2620M processor. Due to the limited computer power, only *chain*#1  $\langle 4-3-3 \rangle$  is used. The number of frames *F* is varied from 100 to 300 with an increment of 50. To obtain the average values, 50 runs of simulations are performed per change of frame number. Comparisons of results are given in TABLE II.

#### TABLE II

TIME CONSUMED PER JOINT FOR SOLVING RIGID BODY CHAIN#1 USING DIFFERENT JOINT ESTIMATION METHODS (NUMBER OF FRAMES VARIES)

	Time Consumed per Joint Using			
	JMV Opt.	Seq. Algorithm	<b>Proposed Framework</b>	
F = 100	6.772 sec	0.056 sec	0.0211 sec	
F = 150	7.186 sec	0.084 sec	0.0226 sec	
F = 200	7.879 sec	0.112 sec	0.0300 sec	
F = 250	14.872 sec	0.138 sec	0.0304 sec	
F = 300	18.143 sec	0.166 sec	0.0313 sec	

TABLE III Validation of Solvability Propagation of the Proposed Framework using rigid body Chain#4

	JMV Opt.	Proposed Framework
RSE per joint per frame	1.6E-02	1.5E-06
Time consumed per joint	5.63 sec	0.018 sec

TABLE IV Validation of Hybrid scheme of the Proposed Framework using rigid body Chain#5

	JMV Opt.	<b>Proposed Framework</b>
RSE per joint per frame	4.5E-02	2.8E-02
Time consumed per joint	5.35 sec	0.65 sec

As shown in TABLE II, our proposed framework has the edge over the Sequential Algorithm approach in terms of speed while beating the *JMV Optimisation* by a huge margin. It is also worth noting that using our framework, the time consumed per joint barely increases with the number of frames, which shows great efficiency.

# C. Functionality

The added functionalities of our framework are the capabilities of performing *solvability propagation* and the *hybrid* scheme to address tough circumstances where there are insufficient markers on the body segments (i.e. less than three markers). Two chains of rigid body segments are generated to validate the added functionalities:

chain#4:	$\langle 3 -$	2 -	2 -	2 -	$2\rangle$	
chain#5:	$\langle 3 -$	1 -	2 -	2 -	2 -	$2\rangle$

By fixing F at 100 and averaging over 50 runs of simulations, the accuracy and speed for each chain are compared with that of the *JMV Optimisation*. Note the Sequential Algorithm is not considered here since it requires at least three markers on each body segment. Results for chain#4 and chain#5 are given in TABLE III and TABLE IV, respectively.

In TABLE III, our framework shows the ability to propagate the *joint solvability* with much faster speed than the *JMV Optimisation* while maintaining a very low level of error. In TABLE IV, our framework reveals comparable and slightly better accuracy than the *JMV Optimisation* when dealing with very tough cases where only one marker is present on a body segment (chain#5). The *hybrid* scheme also greatly reduces the time consumed per joint calculation.

## V. CONCLUSION

This paper presents an automatic solution framework to address the joint estimation problem in optical motion capture. A fast joint estimator which only needs to optimise three variables is proposed (Section II-A). A hybrid scheme is formed by automatically toggling between the proposed MTB-based and the existing condition-free JMV Optimisation approaches for joint estimation so that the system is able to deal with tough circumstances where there are less than three markers on the body segments. The automation of the framework is realised by the solvability propagation (Section II-B) and the hybrid scheme. Simulated results verify the feasibility of our framework. In normal cases where there are at least three noncollinearly placed markers on each body limb, our framework shows good accuracy and is even slightly faster than the commonly considered efficient Sequential Algorithm, let alone the JMV Optimisation. The performance is not degraded when solvability propagation is performed. Results (TABLE IV) also show that the *hybrid* scheme is comparable in terms of accuracy with the JMV Optimisation on tough cases while taking much less time.

#### REFERENCES

- J. Broeren et al., "A kinematic analysis of a haptic handheld stylus in a virtual environment: a study in healthy subjects," Journal of NeuroEngineering and Rehabilitation, vol. 4, p. 13, 2007.
- [2] J. Hashiguichi et al., "Biomechanical analysis of a golf swing using motion capture system," in Proceedings of Annual Meeting of Japanese Society for Orthopaedic Biomechanics, vol. 27, 2006, pp. 325–330.
- [3] A. Menache, Understanding Motion Capture for Computer Animation and Video Games. San Francisco: Morgan Kaufmann, 2000.
- [4] A. Y. Ng et al., "On spectral clustering: Analysis and an algorithm," in Advances in Neural Information Processing Systems. MIT Press, 2001, pp. 849–856.
- [5] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in Advances in Neural Information Processing Systems 17. MIT Press, 2004, pp. 1601–1608.
- [6] B. Li et al., "Articulated motion reconstruction from feature points," Pattern Recognition, vol. 41, pp. 418–431, 2008.
- [7] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proceedings of the American Mathematical Society*, no. 7, 1956, pp. 48–50.
- [8] A. Kirk et al., "Skeletal parameter estimation from optical motion capture data," in CVPR 2005. IEEE Computer Society Conference on, vol. 2, 2005, pp. 1185 vol. 2–.
- [9] S. Gamage and J. Lasenby, "New least squares solutions for estimating the average centre of rotation and the axis of rotation," *Journal of Biomechanics*, vol. 35, no. 1, pp. 87 – 93, 2002.
- [10] M. Silaghi et al., "Local and global skeleton fitting techniques for optical motion capture," in *Proceedings of the International Workshop* on Modelling and Motion Capture Techniques for Virtual Environment, 1998, pp. 26–40.
- [11] J. Cameron and J. Lasenby, "A real-time sequential algorithm for human joint localization," in ACM SIGGRAPH 2005 Posters, ser. SIGGRAPH '05, 2005.
- [12] G. Tan et al., "Human skeleton reconstruction for optical motion capture," *Journal of Computational Information Systems*, vol. 9, no. 20, pp. 8073– 8080, 2013.
- [13] A. Li, "Automatic estimation of skeleton parameters from optical motion capture data," Master's thesis, University of Cambridge, 2014.