Human Motion Modeling from Complementary Skeleton Joints of Multiple Kinects

Ningping Sun[1,*] and Shunsuke Murakami[2]
[1]Dept. of Human-Oriented Information System Engineering, National Institute of Technology, Kumamoto College, 2659-2 Suya, Koshi, Kumamoto 861-1102, Japan
[2]Advanced Electronics and Information Systems Engineering Course, National Institute of Technology, Kumamoto College, 2659-2 Suya, Koshi, Kumamoto 861-1102, Japan

***Abstract*—Human motion modeling with the proposed algorithm of complementary skeleton joints can help us to solve the problem of visual blind spot when the motion are captured by Kinects. The shooting model can be remodeled and reconstructed with the recombination and complementation of the skeleton joints by a series of three-dimensional coordinate transformation according to the different Kinects and a virtual monocular camera. The reconstructed human skeleton model can be applied to further motion simulation from the virtual monocular camera.**

Keywords—motion capture; skeleton joint of Kinect; three-dimensional coordinate transformation; complementation

I. INTRODUCTION

It is common to model human motion using a motion capture system. The movement of a real person is digitally recorded, and these collected data are used to reproduce the movement of the characters in entertainment such as movies and games. Optical, mechanical, and magnetic sensors have been used to perform motion capture. However, in either method, it is necessary to shoot with special set at a fixed studio, the cost is very high and the place is limited.

Kinect, which is well-known as a simple motion capture originally developed as a device for home game machines. It is relatively inexpensive, easy to hand out, compact, and easy to carry. In addition, since Kinect is marker-less, it is possible to reduce the time and labor required to perform motion capture. We have been used Kinect V2 (Depth) as capture device for our motion modeling. However, the action captured using only one Kinect is sometimes unsatisfactory. This is because we cannot successfully track the bone being imaged when the bones enter the blind spot of the Kinect as the body parts overlap. Kinect can track sufficiently even if only one unit for planar motion tracking, but it is weak for stereoscopic motion tracking. Therefore, when we want to shoot not only directly in front of the target but also the full body, we use at least two or more Kinects to compensate for each other's blind spots and perform motion capture.

In this paper, we proposed a method of complementing the skeleton joints to solve the problem of visual blind point of Kinect. The model can be reconstructed by recombination and complementation of skeleton joints by a series of three-dimensional coordinate transformation according to the different Kinects and a virtual monocular camera. We shall explain our algorithm in details, and provide some verification experiment results also.

II. THE PROPOSED COMPLEMENTATION ALGORITHM

A. Motion Capturing with Kinect

Kinect can track motions of up to six models, in this study only one model is to be used. 25 skeleton joints are detected that are shown in FIGURE I, and each skeleton joint has information of type, position (three-dimensional coordinate), and tracking status (*Tracked, Inferred, NotTracked*), which is updated at 30 fps to track motion. Horizontal angle is 70 degrees, vertical angle is 60 degrees, and detectable distance is 0.5～4.5m [1].

First of all, we created a system for acquiring, three-dimensional computer graphics visualizing and preserving the data of the tracking motion, which can be operated remotely. The position and tracking status of each joint can be obtained as parameters of motion data. When the system starts, the detected model is rendered on the screen. While confirming the visualized motion in real time, we can acquire and check motion data, and save the data into a CSV file.

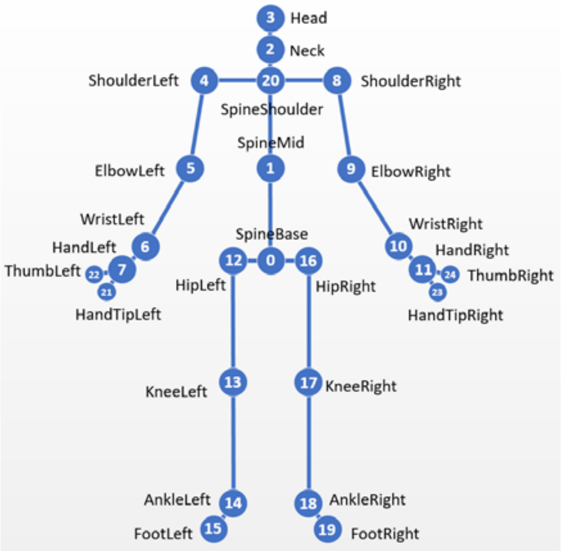


FIGURE I. SKELETON JOINTS OF KINECT

B. Three-dimensional Coordinate System Transformation

In our experiment we set two Kinects as same as shown in FIGURE II using Kinect B to compensate blind spot of Kinect A and vice versa.

When shooting motion with two Kinects, each Kinect tracks the motion of the model from different places and viewing angles. In other words, the coordinate group of the motion data obtained by each Kinect is represented on the coordinate system A and B unique to each Kinect. It is necessary to unify the respective coordinate systems before combining and completing the coordinate of skeleton joint. According to the coordinate system of a virtual monocular camera, a series of affine transformations are applied to the coordinates group of each motion data [2] [3].

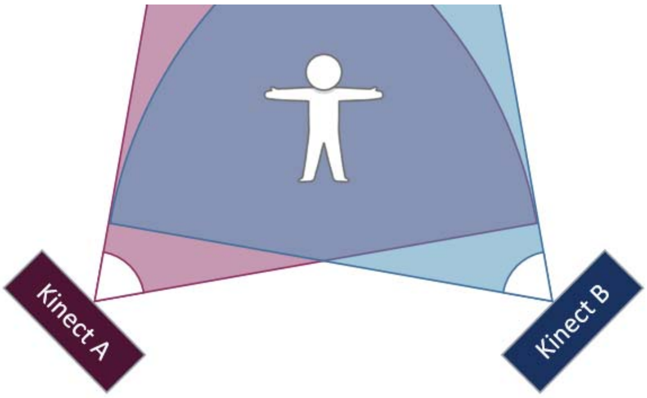


FIGURE II. A MOTION CAPTURE SYSTEM WITH KINECT A AND B

Let D_A be the 3D coordinate group of the motion data of Kinect A. When D_A is divided into frames, the first frame D_{A1} before affine transformation is shown in FIGURE III, where blue represents the coordinates of the left foot (FootLeft), the green is the right foot (FootRight), and the orange is the head (Head).

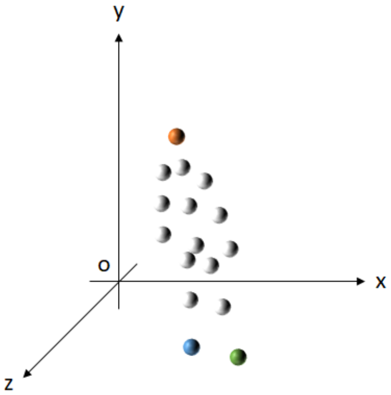


FIGURE III. D_{A1} BEFORE AFFINE TRANSFORMATION

ALGORITHM

a) Translate D_A so that the coordinates of FootLeft of D_{A1} overlap the origin of the virtual monocular camera.

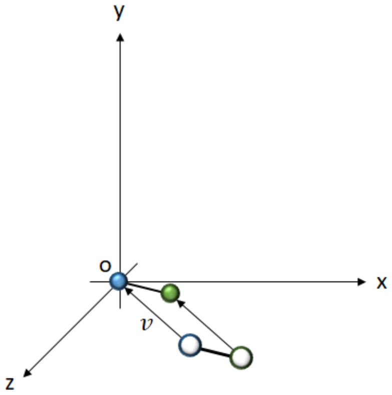


FIGURE IV. FTRANSLATE D_A

b) Find out the angle α around the *y* axis so that the *z* component of the position vector of FootRight becomes *0* and rotate D_A around the *y* axis.

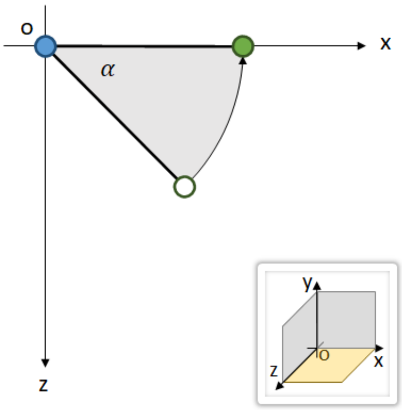


FIGURE V. ROTATE D_A AROUND THE *Y* AXIS

c) Find out the angle β about the *z* axis so that the *y* component of the FootRight's position vector becomes *0* and rotate D_A about the *z* axis.

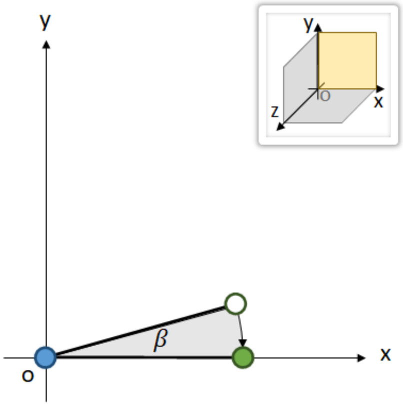


FIGURE VI. ROTATE D_A AROUND THE *Z* AXIS

d) Find out the angle γ about the *x* axis such that the *z* component of the position vector of the Head becomes *0* and rotate D_A around the *x* axis.

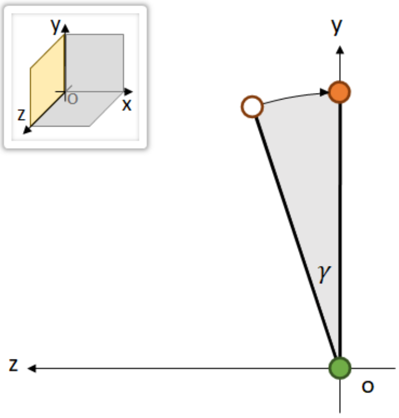


FIGURE VII. ROTATE D_A AROUND THE *X* AXIS

e) As the result of the above process, the FootLeft are on the origin, the FootRight are on the *x* axis, and the Head are in contact with the *xy* plane shown in FIGURE VIII.

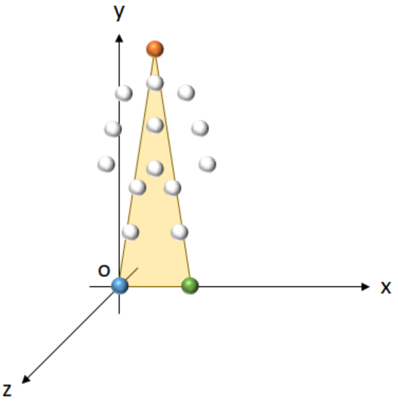


FIGURE VIII. THE FINAL RESULT OF CONVERSION

The coordinates group D_A' after the final affine transformation is obtained by (1),

$$D_A{}' = R(D_A - v) \tag{1}$$

where *v* is the position vector of the left foot of D_{A1}, and *R* is the rotation matrix as follows.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{2}$$

$R_{11} = cos\alpha cos\beta$
$R_{12} = -sin\beta$
$R_{13} = sin\alpha cos\beta$
$R_{21} = cos\alpha sin\beta cos\gamma + sin\alpha sin\gamma$
$R_{22} = cos\beta cos\gamma$
$R_{23} = sin\alpha sin\beta cos\gamma - cos\alpha sin\gamma$
$R_{31} = cos\alpha sin\beta sin\gamma - sin\alpha cos\gamma$
$R_{32} = -cos\beta sin\gamma$
$R_{33} = sin\alpha sin\beta sin\gamma + cos\alpha cos\gamma$

By performing affine transformation with the proposed algorithm on all of the motion data captured from A and B according to a virtual monocular camera, the complementation of the skeleton joints and motion remodeling become possible.

C. *Complementing Skeleton Joint Algorithm*

As mentioned above, each joint has a tracking status, such as *Tracked*, *Inferred* and *NotTracked*, and Kinect defines and provides the joint's state during its motion capture processing. The joint whose tracking state is *Inferred* indicates that the coordinates are acquired, but the reliability is low. If part of the body is hidden and cannot be captured, the associated joints become either *Inferred* or *NotTracked.* In the case of *Inferred* Kinect makes some dummy motions for those joints.

In order to remedy these modeling drawbacks and make a reliable motion, we developed a motion interpolation method with complementary skeleton joints shown in TABLE II. Let the coordinates of the reference motion data be d_A obtained from Kinect A, and let d_B be the coordinates of the same motion corresponding to d_A that is obtained from Kinect B. TABLE II shows how to determine which group of coordinates to be used for reconstruction of the motion.

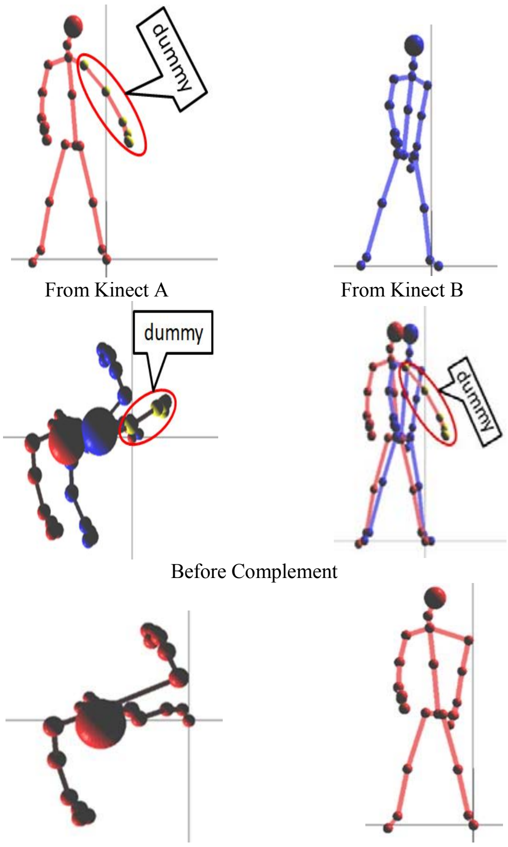
TABLE I. SKELETON JOINT COMPLEMENTATION METHOD

Tracking status		***reconstructed coordinates***
d_A	d_B	
Tracked	*Tracked*	d_A
	Inferred	
	NotTracked	
Inferred	*Tracked*	d_B
NotTracked		

III. THE RESULTS OF VERIFICATION EXPERIMENTS

As illustrated in FIGURE II, we installed two Kinects and conducted some verification experiments on the proposed method. The experimental results are represented and rendered by our system described in section I.

In the Example 1 of FIGURE IX, the red model captured by Kinect A is supplemented with the coordinates of the blue model that is captured by Kinect B. The yellow skeleton joint indicates that the tracking state was *Inferred*. We can see there are *Inferred* joints in red, not in blue. The coordinates of red *Inferred* are supplemented with the blue *Tracked* coordinates based on the method of TABLE II. From FIGURE IX we can see that the hidden red motion is reproduced well.



From Kinect A

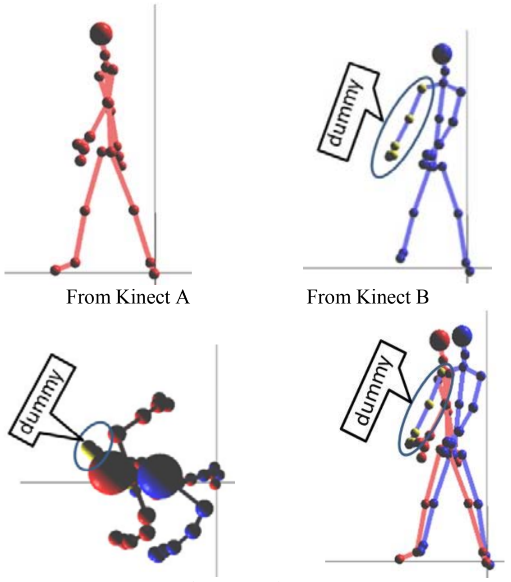
From Kinect B

Before Complement

After Complemented

FIGURE IX. THE RESULTS OF COMPLEMETATION CONVERSION OF EXAMPLE 1

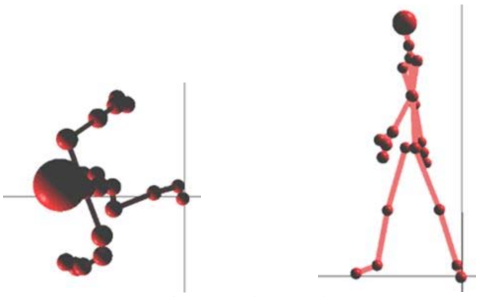
Looking at Example 2, there are no *Inferred* coordinates in red but it is in blue. However, since the red model is supplemented with the coordinates of the blue model, no complement is needed in this case, so the result is a red model as it is, which is shown in FIGURE X.



From Kinect A

From Kinect B

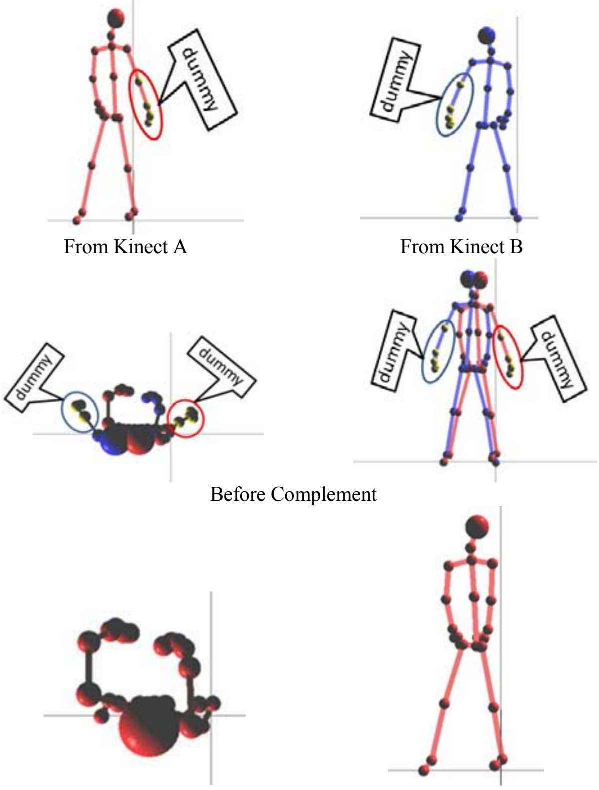
Before Complement



After Complemented

FIGURE X. THE RESULTS OF COMPLEMETATION CONVERSION OF EXAMPLE 2

In Example 3, both red and blue have *Inferred* coordinates. Since joints, which are *Inferred* in red, are *Tracked* in blue, supplementing red with blue makes hidden motion reproduce as FIGURE XI.



From Kinect A

From Kinect B

Before Complement

After Complemented

FIGURE XI. THE RESULTS OF COMPLEMETATION CONVERSION OF EXAMPLE 3

Using the proposed algorithm, we could match the position coordinates of individual motion that are captured by different Kinects to a virtual coordinate system, which is for our computer graphics modeling and rendering. Furthermore, we can use the motion interpolation with complementary skeleton joints instead of those incomplete ones to create a better representation of the motion.

REFERENCES

[1] K. Nakamura, T. Sugiura, and T. Takada, “KINECT for Windows SDK Programming,” Shuwa System Co., LTD, Tokyo, May 2015, pp. 103–117.

[2] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughess, “ Computer Graphics: Principles and Practice (3rd Edition),” Addison – Wesley, 2015, pp.263-286, pp.299-320.

[3] G. Sellers, R. S. Wright, Jr., N.Haemel, “ OpenGL SuperBible (6th Edition),” Addison – Wesley, 2014, pp.49-90.