Multi-Kinects fusion for full-body tracking in virtual reality-aided assembly simulation

Yu Wang1, Fuxiang Chang1, Yuanjie Wu2, Ziran Hu1, Lihui Li1, Pengyu Li3, Pu Lang4 and Shouwen Yao1

Abstract
Skeleton tracking based on multiple Kinects data fusion has been proved to have better accuracy and robustness than single Kinect. However, previous works did not consider the inconsistency of tracking accuracy in the tracking field of Kinect and the self-occlusion of human body in assembly operation, which are of vital importance to the fusion performance of the multiple Kinects data in assembly task simulation. In this work, we developed a multi-Kinect fusion algorithm to achieve robust full-body tracking in virtual reality (VR)-aided assembly simulation. Two reliability functions are first applied to evaluate the tracking confidences reflecting the impacts of the position-related error and the self-occlusion error on the tracked skeletons. Then, the tracking skeletons from multiple Kinects are fused based on weighted arithmetic average and generalized covariance intersection. To evaluate the tracking confidence, the ellipsoidal surface fitting was used to model the tracking accuracy distribution of Kinect, and the relations between the user-Kinect crossing angles and the influences of the self-occlusion on the tracking of different parts of body were studied. On the basis, the two reliability functions were developed. We implemented a prototype system leveraging six Kinects and applied the distributed computing in the system to improve the computing efficiency. Experiment results showed that the proposed algorithm has superior fusion performance compared to the peer works.

Keywords
Full-body motion capture, marker-less motion tracking, multiple Kinects fusion, distributed sensor network

Date received: 24 August 2021; accepted: 17 March 2022

Handling Editor: Peio Lopez Iturri

Introduction
Design for assembly/manufacture (DFA/DFM) has gained many attentions because of their advantages in improving the design efficiency and reducing the costs.1 The core of the DFA/DFM theories is evaluating the assembly in the early design phase. Current computer-aided design and manufacturing (CAD/CAM) software enables accurate evaluations of assembly design in virtual environments. However, the assessment of human factors in CAD/CAM software is still problematic, especially for the evaluation of worker’s perception of assembly complexity.2 Virtual reality (VR) has been proved to be a promising tool for evaluating human factors in assembly since it offers chances to simulate the human factors through natural interaction and stereo display, which is much more intuitive and usable than traditional CAD/CAM software.3,4
Full-body motion capture (MoCap) is one of the key technologies in VR-aided assembly simulation. In contrast to the MoCap with wearable sensors or markers, marker-less MoCap does not require users to wear additional devices, causing less interference with human movements during the simulation, which is very desirable for VR-based industrial applications. Microsoft Kinect is a kind of marker-less full-body MoCap sensor with considerable tracking performance and affordable price, thus has been widely used for full-body MoCap and industrial applications. However, the tracking field of single Kinect is relatively narrow, and the tracking performance is easily affected by the inconsistent tracking accuracy and self-occlusion of human body. Full-body MoCap based on single Kinect becomes more challenging in the assembly simulation of large gearbox since the assembly task requires the user to usually move around in a large area, where the skeleton tracking accuracy may be affected by the inconsistent tracking performance in the tracking field of Kinect. Moreover, the assembly operations contain many postures with self-occlusion, such as crouching, bending forward, and bimanual operations, which seriously affect the tracking performance of the Kinect.

To solve this problem, researchers suggested to process the depth data using the machine learning to optimize the tracking results under the self-occlusion conditions. However, the performance of this approach relies on the quality of the training database, and large errors could occur if the tracking posture is out of the database or extreme occlusion persists for a long time. Another solution is to fuse the tracking data of multiple sensors with different views, in which the parts of human body occluded in one sensor’s view can be supplemented by other effective sensors.

The core of this solution is adequately evaluating the fusion weighting of the data from different Kinects according to their tracking states. However, most of the previous works were designed for the evaluation of general body posture, which did not consider the impacts of the tracking accuracy inconsistency in the tracking field of Kinect, and were not optimized for the self-occlusion of body parts in assembly simulation.

In this article, we improved previous works by considering the tracking accuracy inconsistency and the self-occlusion of body parts during assembly simulation in the data fusion, and developed a multi-Kinects fusion algorithm with finer evaluation granularity for full-body motion tracking in VR-aided assembly simulation. The structure of this article is listed as follows.

- We modeled the tracking error distribution in Kinect’s tracking field leveraging the ellipsoidal surface fitting, and studied the relations between the user-Kinect crossing angles and the self-occlusion. Based on the findings, we propose two reliability functions with finer evaluation granularity on the position-related errors and self-occlusion errors.
- We achieved a distributed computing solution of the multi-Kinects tracking system. The tracking confidences were calculated on the client devices and transmitted to the main workstation through the local area network, which reduces the computing load of the main workstation.
- We implemented a marker-less full-body tracking system consisted of six Kinects for VR-aided assembly simulation.

Related work

Kinect accuracy

Kinect V2 is an RGB-D sensor which consists of an RGB camera and a time-of-flight (ToF) depth camera. The tracking range of Kinect V2 is $70^\circ \times 60^\circ$ (horizontal $\times$ vertical) and the optimal tracking range is 1.2–3.5 m from the sensor. Kinect V2 tracks human body as a 25-joints skeleton model. Related research shows that the body-tracking performance of Kinect is mainly influenced by three factors: system error, target position, and self-occlusion. The system error of Kinect is related to the environmental lighting, target color, distortion of lens, etc. The position of the target could also influence the tracking accuracy due to the inconsistent accuracy distribution in the tracking field of Kinect.

In terms of data uncertainty, the software development kit (SDK) of Kinect V2 provides a confidence parameter that describes the tracking states of each skeleton joint. The parameter consists of three confidence levels: tracked, inferred, and not-tracked. But this parameter is found unreliable in the self-occlusion conditions.
direction in which the user is facing Kinect. They found that the Kinect has a better tracking performance when users are facing or facing away from the Kinect. Furthermore, Wu found that Kinect is unable to identify the front of human body. Thus, the skeleton joints tracking data of left and right sides should be swapped when users turn around.\(^\text{12}\)

**Multi-Kinects data fusion**

To solve the tracking problems mentioned above, many researchers proposed their multi-Kinects data fusion algorithm. A straightforward solution is to calculate the weighted average of the multiple Kinects data. Azis et al.\(^\text{25}\) proposed a weighted averaging method which fuses the multiple Kinects data based on the distance between the center joints and the tracked joints of the stably tracking Kinects. Yeung et al.\(^\text{19}\) proposed a data fusion algorithm which optimizes the bone length differences between the fused skeleton and original skeletons. This method can correct the inconsistency between multiple Kinects, but cannot eliminate the poor tracking data. Multi-sensor fusion algorithm based on possibility theory—such as the Kalman filter (KF),\(^\text{26}\) particle filter,\(^\text{27}\) or the combination of both—\(^\text{28}\)—was also widely used to process multi-Kinect data fusion. These algorithms works well when the skeleton tracking follows the pre-determined distribution model; however, the above works set the measuring error of the sensor model as constant since most of them tracked body movements at a static position. In addition, the probability theory model cannot properly handle the tracking singularity, such as incorrect recognition of facing direction and body parts, which are somehow difficult to be represented by general probability models.

To eliminate the tracking singularity, heuristic methods were applied, which adjust the fusion weightings of Kinects based on empirical rules. Kim et al.\(^\text{24}\) proposed to assign higher weights to skeletons captured by Kinect in which the user faces forward to obtain accurate tracking data, and a front vector was defined to determine the fusion weights. Otto et al.\(^\text{29}\) proposed a set of quality heuristics to determine the fusion weightings of Kinects’ data based on the facing direction and tracking position of the tracked skeleton. Wu et al.\(^\text{12,23}\) suggested an adaptive weighting calculation method which determines the fusion weighting according to the angle between the observing direction of the Kinect and user’s facing direction. Above works lined out the patterns of the tracking singularity and proposed proper solutions to avoid them; however, the weighting function of their works was basically based on the thumb-rules, and was not optimized for assembly operation simulation in workstation.

**Fusion rules**

Generalized covariance integration (GCI) and weighted arithmetic average (WAA) are two most popular fusion rules for multiple sensor data fusion.\(^\text{30}\) The methods based on GCI minimizes the sum of the Kullback–Leibler divergences between the fused results and the source densities,\(^\text{31}\) while the methods based on WAA minimizes the sum of the Cauchy–Schwarz divergences between the fused density and the local densities.\(^\text{32}\) Related research works found that the GCI methods can achieve higher estimation accuracy, but is easily affected by the miss-detection and is computation consuming,\(^\text{33}\) while the WAA methods have lower computational costs and are more robust but less accurate. Both the GCI methods and the WAA methods have been widely applied in multi-target tracking,\(^\text{34}\) and hand tracking based on multi-sensor data fusion\(^\text{35}\) and have shown outstanding performance.

**Modeling tracking reliability of Kinect**

This section introduced the modeling of the tracking performance of the Kinect under different conditions of target position and self-occlusion, which are the kernel of the reliability functions. The presented models include a position-related errors model, which depicts the error distribution in the tracking field of Kinect, and a user-Kinect crossing-angle model, which describes the tracking errors under the self-occlusion in different crossing angles. The details of the modeling are presented below.

**Position-related error**

The position-related error model is based on Yang et al.’s\(^\text{13}\) work, which described the tracking accuracy distribution of Kinect. We used two ellipsoidal models to fit the boundaries of high accuracy (depth error <2 mm) and 4 mm (depth error <4 mm) as stated in Yang’s model (see Figure 6 of Yang et al.\(^\text{13}\)). We fitted the boundaries in the camera coordinates system, which is a left-handed system with origin located at the center of Kinect front surface, \(x\)-axis facing to left and \(z\)-axis facing away from the camera. The general equation of the ellipsoidal model can be expressed as equation (1)

\[
x^2 + ay^2 + bz^2 + cx + dy + ez + f = 0
\]  

where \(a, b, c, d, e, f\) are the fitting parameters of the ellipsoidal model. By inputting samples of the surface
points on the boundary, the parameters of equation (1) can be obtained by solving a least square problem. We sampled 10 points for each boundary as listed in Table 1. The fitting results of the high and low accuracy boundaries are presented as equations (2) and (3). The residual errors are 0.0288 and 0.0075 m, respectively. Figure 1 shows the fitting surfaces of the ellipsoidal models in the camera coordinate system

\[ x^2 + 2.312y^2 + 0.095z^2 - 0.165z - 0.36 = 0 \]  
\[ x^2 + 1.163y^2 + 0.111z^2 + 0.34z - 2.546 = 0 \]

There are three constrains related with the tracking range of Kinect according to Yang's model, which the ellipsoidal models should work with:

- **Constrain 1**: \( z \geq 0.5 \) (m).
- **Constrain 2**: \( \arctan(x/z) \in (-35, 35) \) (degree).
- **Constrain 3**: \( \arctan(y/z) \in (-30, 30) \) (degree).

Constrain 1 corresponds to the minimum limitation of the depth measurement along the z-axis. Constrains 2 and 3 refer to the horizontal and vertical field of view, respectively. We defined the space bounded by the three constrains as \( S \). It is reasonable to regard the tracking data outside \( S \) as unreliable.

### Crossing-angle model

The user-Kinect crossing angle is defined as the angle between user's front vector and the opposite direction of Kinect's observing vector. We use a clock-wise rule to determine the sign of the crossing angle. When the Kinect is in front of the user, the crossing angle is zero degree. If the Kinect is on the left side of the user, the crossing angle is negative. Otherwise, the crossing angle is positive.

To test the tracking performance of Kinect under different crossing angles, we designed a test scene as Figure 2 shows. Six Kinect V2 were evenly placed on a quarter-circle with a radius of 2.5 m, facing to the center of the circle. The Kinect at zero degree direction was labeled as Kinect1, and the Kinect at 90 degree direction was labeled as Kinect6. The crossing angles of the Kinects were indicated in the figure. A high-precision marker-based tracking system Optitrack \(^{36}\) was applied to obtain the ground truth of the skeleton joints position. A least-square method \(^{35}\) was used to register the Kinect system and the Optitrack system, and the averaged calibration error is 0.052 m. Please seek the details of the device setup and data transmission in the following section.

The testing tasks consist of a static task, that is holding a T-pose statically, and a dynamic task, that is marching on the spot. Two testers (male: 175 mm,

| Sample | High      | Low      |
|--------|-----------|----------|
| 1      | (0.7, 0.1)| (1.2, 0.2)|
| 2      | (0.5, 0.2)| (0.7, 0.3)|
| 3      | (0.0, 0.3)| (0.03, 0.35)|
| 4      | (-0.5, 0.2)| (-0.7, 0.3)|
| 5      | (-0.7, 0.1)| (-1.2, 0.2)|
| 6      | (0.04, 0.1)| (0.11, 1.2)|
| 7      | (0.04, 0.2)| (0.07, 3.3)|
| 8      | (0.03, 0.3)| (0.03, 3.5)|
| 9      | (0, -0.4, 2)| (0, -0.3, 3)|
| 10     | (0, -0.4, 1)| (0, -1.1, 2)|

**Figure 1.** Fitting surfaces of accuracy boundaries: (a) high accuracy boundary and (b) low accuracy boundary. The dots represent the sample points.
medium shape; female: 160 mm, slim shape) were invited to perform the test. During the test, the tester was asked to stand in the center of the quarter-circle and perform the tasks facing to the Kinect1, and then perform the tasks again facing to the Kinect6. In each task, 2000 frames of body-tracking data were sampled, respectively.

To investigate the tracking performance of different body parts, we separated the skeleton joints into five groups, as listed in Table 2. The joints of hand-tip, thumb, and toe were excluded from the data analysis since the tracking noise of these joints is relatively large.

The relations between the tracking error and the crossing angle are shown in Figure 3. The error under a crossing angle is the sum of the averaged differences between the skeleton joints captured by the Kinect and the Optitrack system. As figure shows, the tracking error of torso is weakly related to the crossing angles, revealing that the Kinect has robust tracking performance of the torso under the self-occlusion conditions.

The tracking errors of arms are obviously related to the crossing angle, showing as the large errors when the arms are on the side away from the Kinect (36 degrees or more). This accords with the fact that the arms on the side away from Kinect are easily occluded by the torso and the other arm. The tracking error of legs is slightly related with the crossing angle, with moderate errors near the vertical crossing angles (72 degrees or more). Although the tracking of legs should be sensitive to the self-occlusion since they can be easily occluded by the other leg. The robust tracking on the legs probably due to that the legs movements are less intensive than arms, resulting in less tracking errors.

Multi-Kinects data fusion

In this section, we introduce the details of the proposed multi-Kinects data fusion algorithm. The reliability functions were built based on the tracking error models presented in the above section. We defined two confidences, $\omega_p$ and $\omega_o$, to evaluate the influences of position-related error and the self-occlusion error on the tracking performance, respectively.

Algorithm overview

The algorithm flowchart is shown in Figure 4. The raw skeleton data captured by the Kinects are first filtered using a double exponential filter. Then, the skeleton data are evaluated in two layers. In the first layer, called data-layer evaluation, the confidence $\omega_p$ of the skeleton joints in each Kinect is evaluated. This evaluation is achieved by non-governed distributed computing without communicating with the master system. In the second layer, called system-layer evaluation, the confidence $\omega_o$ of the body parts in each Kinect skeleton is calculated. This evaluation depends on the feedback of the predicted front vector of the fused skeleton from the master system. Finally, the skeleton data from each Kinect are first fused with the predicted estimation based on the GCI rules separately, then the posterior estimation of all Kinects is merged using the WAA method. A correlation confidence $\omega_c$ describing the

Table 2. Grouping of the skeleton joints for self-occlusion investigation.

| Group     | Skeleton joints* |
|-----------|------------------|
| Torso     | Head, Neck, Spine |
|           | Shoulder, SpineBase, SpineMid |
| Left arm  | ShoulderLeft, ElbowLeft, WristLeft, HandLeft, HandTipLeft, ThumbLeft |
| Left leg  | HipLeft, KneeLeft, AnkleLeft, FootLeft |
| Right arm | ShoulderLeft, ElbowRight, WristRight, HandRight, HandTipRight, ThumbRight |
| Right leg | HipRight, KneeRight, AnkleRight, FootRight |

*The name of joints refers to Kinect SDK C++.
correlations between the predicted state and the Kinect measurements is generated from the GCI process.

**Position-related reliability function**

In the data-layer evaluation, the confidence $\omega_p$ of each joint is calculated by the position-related reliability function, which describes the tracking uncertainty according to the position of the skeleton joints in the camera coordinate system. We defined a vector $\Omega_p$ which is a 25 dimensions vector ($\omega_{p,1}, ..., \omega_{p,25}$), corresponding to the confidence of 25 joints of the Kinect skeleton. To calculate the confidence $\omega_{p,j}$ of the $j$th tracked joint in $S$, we first transfer the Cartesian

---

**Figure 3.** The averaged errors of the skeleton joints in different groups. $X$-axis is the angles (degree) between the front vector of user and the observing direction of the Kinect. $Y$-axis is the averaged errors (m) between skeleton joints captured by Kinect and Optitrack: (a) left arm, (b) torso, (c) right arm, (d) left leg, and (e) right leg.

**Figure 4.** The flowchart of the proposed fusion algorithm. $\Omega_p$, $\Omega_o$, and $\Omega_C$ are the position-related confidence vector, the self-occlusion confidence vector, and the interrelation confidence vector, respectively.
coordinate \((x_j, y_j, z_j)\) of the tracked joint to polar coordinates \((\rho_j, \alpha_j, \beta_j)\), expressed as equations (4)–(6). Here, we set the XoZ plane as the equatorial plane

\[
x_j = \rho_j \cos \beta_j \cos \alpha_j
\]

\[
y_j = \rho_j \sin \beta_j
\]

\[
z_j = \rho_j \cos \beta_j \sin \alpha_j
\]

By substituting equations (4)–(6) into equations (2) and (3), the ellipsoidal models in polar coordinates system are obtained. Next, the polar coordinates of the \(j\)th tracked joint can be simply obtained using the inverse transform equation of equations (4)–(6), which is expressed as

\[
\rho_j = \sqrt{x_j^2 + y_j^2 + z_j^2}
\]

\[
\alpha_j = \arccos \frac{x_j}{\sqrt{x_j^2 + y_j^2}}
\]

\[
\beta_j = \arcsin \frac{y_j}{\rho_j}
\]

Substituting equations (8) and (9) into the polar ellipsoidal models, we obtained two quadratic equations about \(\rho\). Then, the intersection points between the connecting line from origin to the \(j\)th tracked joint and the two ellipsoidal surfaces can be obtained by solving the quadratic equations (discarding the negative solutions). The distance between the origin and the intersection points are noted as, \(\rho_h\) for high accuracy boundary, and \(\rho_l\) for low accuracy boundary. Finally, the position-related reliability function can be expressed as that is

\[
\omega_p = \begin{cases} 
1 & \rho_j \leq \rho_h \\
1 - 0.5 \frac{\rho_h - \rho_l}{\rho_h - \rho_l} & \rho_h \leq \rho_j \leq \rho_l \\
0.5 e^{(\rho_h - \rho_j)} & \rho_l \leq \rho_j
\end{cases}
\]

In the case of the tracked joint outside the low accuracy boundary, we used an exponential function to quickly dampen the unreliable tracking data. Moreover, \(\omega_p\) was set to zero for the joints outside \(\mathbb{S}\). By sequentially calculating \(\omega_p\) of the 25 joints, the confidence vector \(\Omega_p\) is obtained.

**Prediction**

Before performing the system-layer evaluation, the algorithm predicts the skeleton state based on the posteriori state of the previous frame \(k-1\). Assume the posteriori skeleton state of the frame \(k-1\) is \(X_{k-1}\), which is a set of the tracking states of the 25 skeleton joints \(\{x_{1,k-1}, \ldots, x_{25,k-1}\}\). The tracking state of the \(j\)th joint is \(x_{j,k-1}\), which is a six dimensions vector \((x, y, z, \dot{x}, \dot{y}, \dot{z})\) consisting of the position \((x, y, z)\) and the velocity \((\dot{x}, \dot{y}, \dot{z})\) in the three axes. The covariance matrix is \(P_{j,k-1}\). We used a Gaussian dynamic model to predict the state. The prediction function is

\[
x_{j,k-1} = Fx_{j,k-1}
\]

\[
P_{j,k-1} = Q + FP_{j,k-1}F^T
\]

where \(x_{j,k-1}\) is the predicted state of the \(j\)th joint, \(P_{j,k-1}\) is the predicted covariance matrix, \(F\) is the transmission matrix, and \(Q\) is the process noise matrix. \(F\) and \(Q\) are expressed as

\[
F = \begin{bmatrix} I_3 & \Delta I_3 \\ 0 & I_3 \end{bmatrix}, Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta^2}{3} I_3 & \frac{\Delta^2}{3} I_3 \\ \frac{\Delta^2}{3} I_3 & \Delta^2 I_3 \end{bmatrix}
\]

where \(\Delta\) is the step length, and \(\sigma_a\) is the process noise. After applying the Gaussian dynamic function to predict the 25 joints, the predicted skeleton is obtained.

**Self-occlusion reliability function**

In the system-layer evaluation, a self-occlusion reliability function is defined to calculate the confidence \(\omega_o\) of the body parts in each Kinect skeleton, which reflects the tracking uncertainty of the body parts caused by the self-occlusion. A confidence vector \(\Omega_o\) is defined with 25 dimensions, of which the elements are determined by the grouping in Table 2. The procedure of the \(\omega_o\) calculation is presented below.

The function is a equation of the user-Kinect crossing angle defined in the previous section. The front vector is first calculated based on the prediction of the previous frame. We define the front vector as the crossing product between the vector pointing from left shoulder to right shoulder and the up vector of the world coordinates system. Then, the crossing angle is calculated using the inverse cosine function. To calculate the self-occlusion confidence, we extrapolated the error curves in Figure 3 to the domain of 360 degrees based on the fact that the human body is symmetry about the coronal plane. Then, the reliability function is defined by transferring the error curves to a quasi-probability function, which is expressed as

\[
\omega_o = \frac{\omega_u - \omega_l}{\omega_u - \omega_l}
\]

where \(\omega_u\) and \(\omega_l\) represent the upper limit and the lower limit of the extrapolated curves of Figure 3, \(\omega(\gamma)\) is the tracking error under the crossing angle \(\gamma\). Because the torso tracking is unaffected by the self-occlusion, the \(\omega_o\) of joints in the torso group is always set to 1. For other joints, we calculated the curve of \(\omega_o\) against the crossing angle for left arm, right arm, left leg, and right leg, as Figure 5 shows.
Next, according to the type of the $j$th joint, the self-occlusion confidence $\omega_{o,j}$ is sampled from the corresponding curves in Figure 5. By sequentially calculating the $\omega_o$ of the 25 joints, the confidence vector $\Omega_o$ is obtained.

**Data fusion**

First, the predicted skeleton is merged with the captured skeleton of each Kinect sequentially based on KF. For the predicted state $x_{j,k}^{(i)}$ of the $j$th joint and the measurement $z_{j,k} = (z_x, z_y, z_z)$ of the $j$th joint in the skeleton captured by the $i$th Kinect, the KF propagation is given as

$$x_{j,k}^{(i)} = x_{j,k|k-1}^{(i)} + K^{(i)}(j_x - Hx_{j,k|k-1}^{(i)})$$

$$P_{j,k}^{(i)} = [I - K^{(i)}H]P_{j,k|k-1}^{(i)}$$

$$K^{(i)} = P_{j,k|k-1}^{(i)}H(HP_{j,k|k-1}^{(i)}H^T + R)^{-1}$$

where $x_{j,k}^{(i)}$ is the posterior of joint $j$ of the skeleton captured by Kinect $i$, $P_{j,k}^{(i)}$ is the posteriori covariance matrix, $K^{(i)}$ is the Kalman gain, $H = [I, \theta]$ is the transform matrix from the state space to the measurement space, $R = \sigma_k I_3$ is the measurement noise matrix, and $\sigma_k$ is the measurement noise.

A correlation confidence $\omega_c$ is generated from the Kalman filtering which is expressed as

$$\omega_c^{(i)} = \frac{\alpha^{(i)}}{\kappa + \sum_{i=1}^{n} \alpha^{(i)}}$$

where $\kappa$ is the clutter density, and $\alpha^{(i)}$ represents the abstract strength of the correlation between the measurements of the $i$th Kinect and the predicted state. The higher $\omega_c^{(i)}$ indicates the measurement has a stronger correlation with the predicted states, that is, the tracking results are considered more reliable. In contrast, if the $\alpha^{(i)}$ is so weak even lower than the threshold $\kappa$, the $\omega_c^{(i)}$ will be very small and the measurement is likely to be an outlier. By sequentially propagating the 25 joints, the posteriori skeleton $X_{k}^{(i)}$ of the $i$th Kinect and the corresponding correlation confidence vector $\Omega_c^{(i)}$ are obtained which contains the correlation confidence of the 25 joints of the tracked skeleton generated by the $i$th Kinect.

Next, the confidence vectors $\Omega_p^{(i)}$, $\Omega_o^{(i)}$, and $\Omega_c^{(i)}$, $i = 1 \ldots n$ are transformed to the diagonal matrices $\Omega_p^{(i)}$, $\Omega_o^{(i)}$, and $\Omega_c^{(i)}$, $i = 1 \ldots n$, respectively. Then, the fused skeleton $X_k$ in the frame $k$ can be calculated by

$$X_k = \sum_{i=1}^{n} \Omega_p^{(i)} X_k^{(i)}$$

$$P_k = \sum_{i=1}^{n} \Omega_p^{(i)} \left( P_k^{(i)} + \left( X_k - X_k^{(i)} \right) \left( X_k - X_k^{(i)} \right)^T \right)$$

$$\Omega^{(i)} = \Omega_p^{(i)} \Omega_o^{(i)} \Omega_c^{(i)}$$

where $P$ is a set of the covariance matrices of the 25 skeleton joints. For clarification, we give the calculation of the $j$th joint in the fused skeleton, which is expressed as

$$x_{j,k} = \frac{\sum_{i=1}^{n} \omega_{o,j}^{(i)} x_{j,k}^{(i)} + \sum_{i=1}^{n} \omega_{c,j}^{(i)} (x_{j,k} - x_{j,k}^{(i)})}{\sum_{i=1}^{n} \omega_{o,j}^{(i)} + \sum_{i=1}^{n} \omega_{c,j}^{(i)}}$$

$$P_{j,k} = \frac{\sum_{i=1}^{n} \omega_{o,j}^{(i)} P_{j,k}^{(i)} + \sum_{i=1}^{n} \omega_{c,j}^{(i)} (x_{j,k} - x_{j,k}^{(i)}) (x_{j,k} - x_{j,k}^{(i)})^T}{\sum_{i=1}^{n} \omega_{o,j}^{(i)} + \sum_{i=1}^{n} \omega_{c,j}^{(i)}}$$

where $x_{j,k}$ and $P_{j,k}$ are the mean and covariance matrix of the tracking state of the $j$th joints in the fused skeleton.

Since the propagation of our fusion algorithm is processed per joints for all tracked skeletons, the principle computational complexity of our algorithm is $O(mn)$, where $m$ is the amount of the tracking sensors included.
in the system and \( n \) is the joints number of the fused skeleton.

**System implementation**

We implemented a prototype of the marker-less full-body tracking system using six Kinect V2. The system setup is shown in Figure 6. Because the six Kinects cannot be connected to a single computer, we used six Intel NUC computers (Intel Core i5-8259U at 2.3 GHz, 8 GB RAM, and Iris PlusGraphics 655) as the client devices for driving the Kinect and the client program. A graphic workstation (Intel Xeon Gold 6128 CPU, 128 GB RAM, NVIDIA Quadro RTX 6000) was applied as the server to drive the main program. The Kinect are connected to the NUC through USB3.0 port, and the MoCap data are transmitted to the server through wireless UDP LAN using Open Sound Control (OSC) messages. The LAN was established using a gigabit wireless adaptor (TP-Link AC1900). The adaptor was connected to the server through a network cable.

The six Kinects were evenly placed on a circle with a radius of 2.5 m, facing the center of the circle. With the setup, the tracking space of the system is shown in Figure 7. The available tracking space is a hexagon with a long axis of 5 m (shown as the blue area). The fine tracking space, where the user can be tracked by all Kinects in a distance of 0.5–3.5 m, is a hexagon area with a diameter of 2.5–3.5 m. The registrations between the Kinects are achieved using a Least-square method. The maximum calibration error of the skeleton joints is 0.085 m, and the averaged calibration errors of the six Kinects are less than 0.053 m.

The algorithm was developed in C#. We developed two programs to implement the proposed algorithm. The first program is the client program, running on the client computer. The client program accounts for collecting tracking data from Kinect SDK, the double exponential filtering, calculation of the position-related confidence and self-occlusion confidence, and transmitting the skeleton data to the server. The second program runs on the server and performs the functions of receiving client data, calculation of the correlation confidence, data fusion, and rendering the fused skeleton.

**Experiment**

In this section, we illustrate the fusion performance of the proposed algorithm. To compare the performance, we implemented another two fusion algorithms...
referring to the peer works based on our system setup. The first algorithm (notated as Simple Average, SA) simply averages the posteriori skeletons according to the joint tracking state provided by the Kinect SDK, where the state of tracked is assigned with a confidence value of 1, the inferred is assigned with 0.5, and the not-tracked is assigned with 0. The second algorithm is a six-Kinects version of the improved adaptive weight calculation (IAWC) algorithm proposed in Wu et al. Experiment design

The experiment task is divided into two groups. The first group consists of 10 common actions selected from typical workout movements, as listed in Table 3. The common actions are classified into two types. The actions of locomotion require testers to move around the center of the tracking space for five rounds with the specifying movements. The actions of limbs move require testers to complete the movements at three positions on a circle with a radius of 2 m, centered on the tracking space center. The actions of limb move are repeated for four rounds in each position, and each round contains five repetitions of the movements. The second group of the experiment task consists of three assembly tasks (ATask) of a transmission case. The three tasks are as follows:

1. **ATask1**: transport and assembly the transmission gear, oil distribution sleeve, bearing, and bearing cover of the transmission shaft. The four parts are distributed around the station, requiring the tester to move around to get the parts. The assembly operations contain movements of bending, reaching, bimanual manipulation, locomotion, and so on.
2. **ATask2**: fetch and assembly five lifting lugs on the transmission case. This task is similar with ATask1, the difference is that the volume of the case is larger than the shaft; thus, the tester need to perform the task at the peripheral area of the tracking space, where the body tracking is more challenging than ATask1.
3. **ATask3**: transport and assembly the idle gear, bearing, and bearing pedestal. The idle gear is located inside the transmission case, requiring the tester to kneel and reach into the front opening of the case to assemble the gear. In this task, the tester needs to complete the task with strong self-occlusion at legs.

We invited six college students to perform the experiment. For the test of common action group, the tester directly completed the actions in the tracking space of the multi-Kinects system. For the test of assembly task group, we developed a virtual experiment scene in Unity3D game engine (2019.4.7f1) as shown in Figure 8. During the task, the tester observes the virtual environment through the head-mounted display (HMD) of Oculus Rift S, and interacts with the virtual object through the Leap Motion controller. The HMD is only used for visualization and does not provide the head-tracking data. The ground truth of the experiment is obtained using the Optitrack system, and the tester needed to wear a black bodysuit attached with the Optitrack markers during the test.

Results and discussion

The skeleton data captured by the six Kinects, the fusion results of the proposed fusion algorithm, and the ground truth captured by the Optitrack system were recorded online during the test. The frame sequence length is about 600–700 frames for a common action trial, and is about 1600–1900 frames for an assembly trial. The fusion results of the SA and IAWC algorithms were calculated offline based on the records.

To represent the fusion performance of each algorithm, we defined a skeleton fusion error $e_{sf}$, which is the mean of the averaged joint errors for the 25 skeleton joints. To calculate the error, the averaged errors of the skeleton joints between the fusion results and the ground truth were used.

Table 3. Grouping of the skeleton joints for self-occlusion investigation.

| Type            | Action name       | Description                               |
|-----------------|-------------------|-------------------------------------------|
| Locomotion      | Side jumping      | Jump left and right                       |
|                 | Walking           | Walk at a normal pace                     |
|                 | Bend walking      | Walk with upper body bent                 |
|                 | Jogging           | Jog in the tracking space                 |
|                 | Frog-jump         | Move with frog-jumping                    |
| Limbs move      | Boxing            | Punch with left and right fists alternately|
|                 | Waving            | Wave two hands at the sides of body       |
|                 | Clapping          | Clap hands in front of chest              |
|                 | Pick and throw    | Pick object from ground and throw it       |
|                 | Squat             | Squat over and over again                 |

International Journal of Distributed Sensor Networks
truth are first calculated in the frame sequences of all tasks. Then, the $e_{sf}$ is obtained by calculating the mean of the averaged joint errors for the 25 skeleton joints. The $e_{sf}$ for each fusion algorithm are presented below.

Figure 9 shows the $e_{sf}$ of the proposed algorithm, the SA algorithm, and the IAWC algorithm for the common actions. Table 4 shows the statistical results of the $e_{sf}$. Generally, the proposed algorithm has a superior performance in fusion accuracy for the common actions compared to peer works. The improvements are obvious for the actions of walking, pick and throw, and squat (improved by more than 20%) compared to the SA. This shows the superiority of the position-related confidence and the self-occlusion confidence over the SDK confidence. We noted that the proposed algorithm also outperforms the IAWC in these actions. This could attribute to that the finer evaluation granularity in the proposed algorithm than in IAWC, and the weightings of IAWC were modeled based on the experience, which may not be reasonable under some crossing angles. While the proposed algorithm has similar performance with the IAWC for the actions of waving and clapping. Moreover, we found that the proposed algorithm has a relatively larger error for the frog-jump action (more than 0.1 m), which could be interpreted by that the motion in frog-jump action is the most intensive in all actions, which is challenging for Kinect tracking.

**Table 4.** Comparisons of the skeleton fusion errors (mm) between the proposed algorithm and the peer works for the common actions.

| Action            | Ours  | SA    | IAWC  |
|-------------------|-------|-------|-------|
| Bend walking      | 0.082 | 0.102 | 0.082 |
| Side jumping      | 0.065 | 0.081 | 0.075 |
| Frog-jump         | **0.103** | 0.121 | 0.118 |
| Jogging           | 0.079 | 0.095 | 0.0905|
| Walking           | 0.065 | 0.083 | 0.079 |
| Boxing            | 0.064 | 0.078 | 0.075 |
| Clapping          | 0.058 | 0.072 | 0.063 |
| Waving            | 0.070 | 0.08  | 0.0715|
| Pick and throw    | 0.066 | 0.088 | 0.076 |
| Squat             | 0.068 | 0.088 | 0.077 |

SA: simple average; IAWC: improved adaptive weight calculation. Noticeable results, such as the largest error of our algorithm, significant improvement (greater than 20%) and similar performance (improvement less than 10%), are in bold.

Figure 8. Virtual experiment scene for the test of the assembly task group.

Figure 9. The skeleton fusion errors of the SA, IAWC, and proposed fusion algorithms for the common actions: (a) locomotion and (b) limb move.
To qualitatively compare the tracking performance, we took several snapshots in the frame sequence of the frog-jump action, as shown in Figure 10. The sequence depicts a process of landing and getting up during a frog jump. As can be seen from the figure, during the landing process, all the fusion algorithms have poor fusion performance in the leg part, which is manifested as the abnormal shape of the leg in the second and third frames. The possible reason could be that the movements of leg were quick during the landing, leading to lower tracking accuracy of all Kinects. With the tester getting up (the fourth and fifth frames), the skeleton of the proposed algorithm returns to normal quickly, while the legs of the IAWC and SA remain in the abnormal shape. We also noticed that in the fusion of the arm part, the proposed algorithm and the IAWC show better consistencies with the ground truth than the SA algorithm, showing as the incorrect shoulder position of the SA skeleton in the second frame. However, the arm fusion of the IAWC algorithm is less stable than the proposed algorithm, indicated by the abnormal arm shape of the IAWC skeleton in the forth and the fifth frames.

Figure 11 shows the $e_{sf}$ in the assembly tasks. Generally speaking, the proposed algorithm has superior fusion performance for the assembly tasks. Compared to the IAWC, the $e_{sf}$ in the proposed algorithm were reduced by 16.2% for ATask1, 13.9% for ATask2, and 15.2% for ATask3. Because the tracking of the torso is less affected by the self-occlusion, the differences between the proposed algorithm and the peer works are inconspicuous. Thus, we analyzed the fusion performance by comparing the critical body parts in each task. In ATask1, the tracking

![Image](image_url)

**Figure 10.** The appearance of the skeletons produced by the Optitrack, proposed algorithm, IAWC, and SA for the frog-jump action.

![Image](image_url)

**Figure 11.** The skeleton fusion errors of the SA, IAWC, and the proposed fusion algorithms for the assembly tasks.
of the arm joints is easily affected by the self-occlusion since the task requires bimanual operations. Similarly, the leg joints are critical in ATask3 because the tester is squatting during the task, leading to the self-occlusion of the leg joints. Hence, we calculated the fusion errors of the right arm in the ATask1 as well as the errors of the right leg in ATask3 to further analyze the fusion performance. Figure 12 shows the fusion errors of the critical body parts. As shown in figure, the proposed algorithm outperforms the peer works, indicated by the obviously lower fusion errors in right wrist for ATask1, and in right knee for ATask3. Results indicates that the proposed algorithm efficiently improve the fusion performance at critical body parts in assembly simulation compared to the peer works.

Conclusion

This article proposed a multi-Kinects data fusion algorithm for improving the MoCap accuracy in VR-aided assembly simulation. To better evaluate the tracking performance of the Kinect, we proposed two tracking confidences, $v_p$ and $v_o$, to consider the position-related tracking accuracy of Kinect and the influences of the self-occlusion. The fusion performance of the proposed algorithm was tested using ten common actions and three assembly tasks of a transmission case.

The results of the common actions test illustrate the proposed confidence can efficiently improve the MoCap accuracy under the conditions of user moving and self-occlusion. In the qualitatively analysis of the frog-jump actions, we found that the legs tracking of Kinect is poor during the landing process, indicating that the evaluation on legs tracking data should be further optimized for more intensive leg movements. The testing results of the assembly task show that the proposed algorithm has advantages in dealing with the fusion error caused by the user’s moving and the self-occlusion of body parts.

Moreover, the proposed algorithm is beneficial to distributed computing. The efficiency of the algorithm can be further improved by realizing the data transmission environment with high synchronicity. In this environment, the calculation of system-layer evaluation and Kalman filtering can be moved to the client device because these steps require only the predicted state of the last frame.

In general, the multi-Kinects data fusion algorithm proposed by this article is able to improve MoCap performance in the assembly simulation in VR. We summarize the main novelties of this article are listed as follows:

- The algorithm considers the tracking accuracy inconsistency of the Kinect, which improves the fusion performance of the multiple Kinect data in the assembly simulation, where the user could frequently move in the tracking space.
- The reliability function of self-occlusion factor is designed at the level of body parts, instead of the whole skeleton. This significantly improves the fusion accuracy in the self-occlusion conditions.
- We designed the algorithm in a distributed computing manner, by which the computing efficiency can be further improved by implementing data transmission environments with high synchronicity.

In future, we consider to improve the fusion performance of the proposed algorithm by optimizing the evaluation reliability under stronger self-occlusion conditions using the methods like neutral networks and machine learning. The distributed computing of the proposed algorithm will also be completed for better efficiency.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (grant no. 51975051).

ORCID iD
Yu Wang https://orcid.org/0000-0002-6318-0446

References
1. Boothroyd G and Alting L. Design for assembly and dis-assembly. CIRP Ann 1992; 41(2): 625–636.
2. Mattsson S, Tarrar M and Fast-Berglund Å. Perceived production complexity—understanding more than parts of a system. Int J Prod Res 2016; 54(20): 6008–6016.
3. Peruzzini M, Pellicciari M and Gadaleta M. A comparative study on computer-integrated set-ups to design human-centred manufacturing systems. Robot Comput Integ Manuf 2019; 55: 265–278.
4. De Casenave L and Lugo JE. Effects of immersion on virtual reality prototype design reviews of mechanical assemblies. In: Proceedings of the ASME 2018 international design engineering technical conferences and computers and information in engineering conference (vol. 51845), Quebec City, QC, Canada, 26–29 August 2018. New York: The American Society of Mechanical Engineers.
5. Putheveetil SC, Daphalapurkar CP, Zhu W, et al. Comparison of marker-based and marker-less systems for low-cost human motion capture. In: Proceedings of the ASME 2013 international design engineering technical conferences and computers and information in engineering conference (vol. 55867), Portland, OR, 4–7 August 2013. New York: The American Society of Mechanical Engineers.
6. Berg LP and Vance JM. Industry use of virtual reality in product design and manufacturing: a survey. Virt Real 2017; 21(1): 1–17.
7. Lawson G, Salanitri D and Waterfield B. Future directions for the development of virtual reality within an automotive manufacturer. Appl Ergon 2016; 53: 323–330.
8. Microsoft. Kinect for Windows, https://developer.microsoft.com/zh-cn/windows/kinect/ (accessed 19 August 2021).
9. Friörksson FA, Kristjánsson HS, Sigurðsson DA, et al. Become your avatar: fast skeletal reconstruction from sparse data for fully-tracked VR. In: Proceedings of the 26th international conference on artificial reality and tele-existence and the 21st eurographics symposium on virtual environments: posters and demos (vol. 10), Little Rock, AR, 7–9 December 2016. Geneva: The Eurographics Association.
10. Halim I, Radin Umar RZ, Ahmad N, et al. Usability study of integrated RULA-KinectTM system for work posture assessment. Int J Integ Erg 2018; 10(8): 175–184.
11. Plantard P, Shum HPH, Le Piérres AS, et al. Validation of an ergonomic assessment method using Kinect data in real workplace conditions. Appl Ergon 2017; 65: 562–569.
12. Wu Y, Wang Y, Jung S, et al. Towards an articulated avatar in VR: improving body and hand tracking using only depth cameras. Entertain Comput 2019; 31: 100303.
13. Yang L, Zhang L, Dong H, et al. Evaluating and improving the depth accuracy of Kinect for Windows v2. IEEE Sens J 2015; 15(8): 4275–4285.
14. Plantard P, Shum HPH and Multon F. Filtered pose graph for efficient kinetic pose reconstruction. Multim Tool Appl 2017; 76(3): 4291–4312.
15. Xu Y and Qiú TT. Human activity recognition and embedded application based on convolutional neural network. J Artif Intell Technol 2021; 1(1): 51–60.
16. Xing Y and Zhu J. Deep learning-based action recognition with 3D skeleton: a survey. CAAI Trans Intell Technol 2021; 6: 80–92.
17. Kaenchan S, Mongkolnam P, Watana B, et al. Automatic multiple kinect cameras setting for simple walking posture analysis. In: Proceedings of the 2013 international computer science and engineering conference (ICSECE). Nakhonpathom, Thailand, 4–6 September 2013. New York: IEEE.
18. Asteriadis S, Chatzitofiáis A, Zarparas D, et al. Estimating human motion from multiple Kinect sensors. In: Proceedings of the 6th international conference on computer vision/ computer graphics collaboration techniques and applications, Berlin, 6–7 June 2013. New York: ACM.
19. Yeung KY, Kwok TH and Wang CCL. Improved skeleton tracking by duplex kinects: a practical approach for real-time applications. J Comput Inform Sci Eng 2013; 13(4): 041007.
20. Livingston MA, Sebastian J, Ai Z, et al. Performance measurements for the Microsoft Kinect skeleton. In: Proceedings of the 2012 IEEE virtual reality workshops (VRW), Costa Mesa, CA, 4–8 March 2012. New York: IEEE.
21. Wasenmüller O and Stricker D. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In: Proceedings of the 13th Asian conference on computer vision, Taipei, 20–24 November 2016. Cham: Springer.
22. Microsoft. TrackingState Enumeration, https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn758898(v=ieb.10) (2014, accessed 19 August 2021).
23. Wu Y, Gao L, Hoermann S, et al. Towards robust 3D skeleton tracking using data fusion from multiple depth sensors. In: Proceedings of the 2018 10th international conference on virtual worlds and games for serious applications (VS-Games), Würzburg, 5–7 September 2018. New York: IEEE.
24. Kim J, Lee I, Kim J, et al. Implementation of an omnidirectional human motion capture system using multiple Kinect sensors. IEICE Trans Fund Electron Commun Comput Sci 2015; 98(9): 2004–2008.
25. Azis NA, Jeong YS, Choi HJ, et al. Weighted averaging fusion for multi-view skeletal data and its application in action recognition. IET Comput Vis 2016; 10(2): 134–142.
26. Moon S, Park Y, Ko DW, et al. Multiple kinect sensor fusion for human skeleton tracking using Kalman filtering. *Int J Adv Robot Syst* 2016; 13(2): 65.
27. Kim Y. Dance motion capture and composition using multiple RGB and depth sensors. *Int J Distrib Sens Netw* 2017; 13(2): 7696083.
28. Morato C, Kaipa KN, Zhao B, et al. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *J Comput Inform Sci Eng* 2014; 14(1): 011006.
29. Otto MM, Agethen P, Geiselhart F, et al. Towards ubiquitous tracking: presenting a scalable, markerless tracking approach using multiple depth cameras. In: *Proceedings of the EuroVR 2015*, https://www.uni-ulm.de/fileadmin/web-site_uni_ulm/iui.inst.100/institut/Papers/Prof_Rukzio/2015/OttoEtAl_Towards_ubiquitous_trac king.pdf
30. Yi W, Li G and Battistelli G. Distributed multi-sensor fusion of PHD filters with different sensor fields of view. *IEEE Trans Signal Process* 2020; 68: 5204–5218.
31. Battistelli G and Chisci L. Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica* 2014; 50(3): 707–718.
32. Hoang HG, Vo BN, Vo BT, et al. The Cauchy–Schwarz divergence for Poisson point processes. *IEEE Trans Inform Theory* 2015; 61(8): 4475–4485.
33. Üney M, Houssineau J, Delande E, et al. Fusion of finite-set distributions: pointwise consistency and global cardinality. *IEEE Trans Aerosp Electron Syst* 2019; 55(6): 2759–2773.
34. Li G, Battistelli G, Yi W, et al. Distributed multi-sensor multi-view fusion based on generalized covariance intersection. *Signal Process* 2020; 166: 107246.
35. Wang Y, Wu Y, Jung S, et al. Enlarging the usable hand tracking area by using multiple leap motion controllers in VR. *IEEE Sens J* 2021; 21(16): 17947–17961.
36. OptiTrack. Flex 13, https://www.optitrack.com/cameras/flex-13/ (accessed 19 August 2021).
37. Microsoft. JointType Enumeration, https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn758663(v=ieb.10) (2014, accessed 19 August 2021).
38. Homepage of Rug.Osc, https://bitbucket.org/rugcode/rug.osc/src/master/ (2021, accessed 19 August 2021).
39. Microsoft. Kinect for Windows SDK, https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn799271(v=ieb.10) (2014, accessed 19 August 2021).