# Real-Time Dynamic Motion Capture Using Multiple Kinects

Seongmin Baek(✉) and Myunggyu Kim

Electronics and Telecommunications Research Institute (ETRI),
218 Gajeong-ro, Yuseng-gu, Daejeon 34129, Korea
{baeksm, mgkim}@etri.re.kr

**Abstract.** The present paper proposes a method of capturing real-time motions without any inconvenient suit by using several inexpensive sensors vulnerable to joint occlusion and body rotation. Depth data and ICP algorithm are used for calibration. Then, the left and right sides of joints are determined, and the optimal joints are chosen based on the variation in rotation to restore postures. The similarity between the motions captured by the proposed multiple sensors and those captured by a commercial motion capture system is over 85 %.

**Keywords:** Multiple kinects · Motion capture · Dynamic motion · Joint selection

## 1 Introduction

3D motion capture has long been explored and characterized by high applicability in diverse fields for retrieving body motions. Optical systems are often used for capturing motions. Still, magnetic systems are also used to secure free movements. Yet, due to the need to wear inconvenient suits, motion capture systems are difficult to apply to ordinary users. By contrast, the marker-free motion capture can get the motions without the special suits, and thus are highly applicable to motion-based contents, e.g. dance and sports. The Kinect v2 released by Microsoft has been applied to many games as it is inexpensive and capable of extracting motions in real time. Yet, it has many limitations in extracting motions with a single sensor, resulting from the joint occlusion and other challenges.

To address the challenges resulting from the occlusion of body parts, more sensors are used to minimize the occluded parts. Lately, methods of using multiple Kinects have been suggested. Zhang and colleagues tracked postures with particle filtering and partition sampling [1]. Their method drew upon not skeleton data but template matching to estimate postures through optimization. Kitsikidis et al. used three Kinects to retrieve dance motions, and notably used HCRF to recognize motion patterns [2]. Kaenchan et al. analyzed walking motions based on the mean positions of joints tracked [3]. Moon et al. used the Kalman filtering to alter and mix accurate Kinect data [4]. Yet, they failed to capture 360-degree motion events because Kinects were placed in front and the motions were too simple. Jo et al. proposed a system using multi Kinects to track multiple users [5], but they focused on tracking the positions of


J.J. (Jong Hyuk) Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*, Lecture Notes in Electrical Engineering 421, DOI 10.1007/978-981-10-3023-9_5

multiple users instead of retrieving their motions. Ahmed used 4 Kinects to capture boxing and walking motions in 360° [6], tracked users' faces to determine a central Kinect with the joint inputs from the other Kinects being used to retrieve the joints that the central Kinect failed to track. Similarly, Baek et al. selected a central Kinect based on the movements of root joints and retrieved the postures by mixing the joints based on the weights of 5 segments tracked [7].

The present paper used 8 Kinects v2 to build a multiple Kinect system, and proposed a method of retrieving body motions from a series of noise joint data inputs from each sensor. The user motions were dynamic, e.g. Taekwondo, and could be captured in 360° in real time (30 fps). The proposed method captures dynamic motions with ease and fast without requiring any motion capture data or pre-trained probability model. The proposed multiple Kinect system was compared with a commercial motion capture system, Xsens to measure the accuracy of data recognized by the former.

## 2 Multi Kinects System and Data Transmission

As a single Kinect v2 can be connected to a single PC, $N$ Kinects need be connected to $N$ PCs. As in Fig. 1b, 2 Kinects were installed on all sides (front, rear, left and right), adding up to 8 Kinect systems. To incorporate the data inputs from each Kinect, the server-client model as in Fig. 1a was used, where $N$ PCs were connected to the server PC. Upon being connected with the clients, the server sends the background removal command to the clients, where the backgrounds and noises are removed from the depth data to transmit the data specific to the body (i.e. the depth and joint data whose color values are mapped).

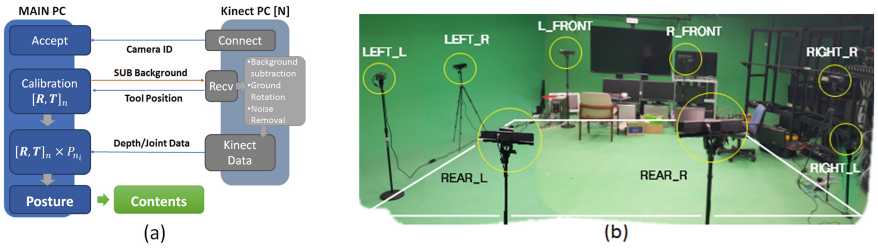


**Fig. 1.** System overview, (a) server-client model, (b) system configuration

The depth image that constitutes the background is saved when the initial Kinect depth data are acquired. Once the Kinect sensor senses the user, it compares the depth value of the $P_{ij}$ pixel with that of $B_{ij}$ pixel in the background saved. When the former is below the threshold, it is considered as the background, and excluded. As the depth data tend to show noises at the edges, the depth values of 8 pixels adjacent to the $P_{ij}$ pixel are compared to determine the similarity of depth values. When the depth similarity is below $k$, it is considered a noise and thus excluded. Finally, as the floor around the user still includes the noise owing to the depth data, the depth data below a certain

value of user's ankle joint data are considered noises and thus excluded. Here, as the Kinect sensor could be tilted, the normal value of the floor is determined based on the vector associated with the root and spine-mid joints while the user stands upright in the initial setting. The gradient is corrected by calculating the rotation matrix, where the normal vector for the floor is matched to the up vector (0, 1, 0).

## 3 Calibration

As the coordinate systems of the data inputs from each Kinect differ from one another, they need be unified into a single coordinate system. Here, the front Kinect is selected as the reference coordinate system. For calibration, a long thin stick (about 50 cm) with a light cubic object (for recognition) at its tip is used as a tool. Based on the resolution of the Kinect, the depth data of the stick are ignored, while the depth data of the cubic object at the tip are taken. As the user moves the tool in the capture space, the mean value of the depth data of the object at the tip is saved as the central point. The user can set the timing and number of data to be captured. Here, 300 data are collected at an interval of 50 ms (Fig. 2). Excluding the data occluded by the body, the rotation matrix (R) and the translation vector (t) are calculated by applying the ICP (iterative closest point) technique to the points from the Kinects corresponding to the input points from the reference Kinect. As data are collected at a certain interval of time, the input point matching the central point is easy to find, which is conducive to fast and accurate calculation. Figure 2 shows the data inputs from the multiple Kinects prior to the calibration and the rotation and translation of the points after the calibration.

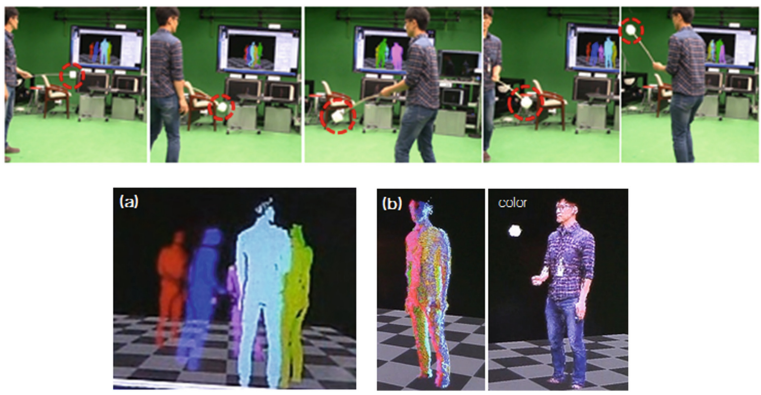


**Fig. 2.** Calibration process using the tool (Up), Calibration (a) before and (b) after (Down)

## 4 Joint Selection

The upgraded Kinect v2 enhances the accuracy of joint data over the existing version when the user faces forward. As in Fig. 3a, when the arms are lifted forward, the existing version tracks the elbow and wrist joints even in '*Not Tracked*' setting, calculates wrong positions, and is prone to errors. As in Fig. 3b, when the user turns right, it is impossible to track the positions because the right hip is blocked. Thus, the SDK 2.0 version continues to track wrong joint positions. Therefore, significant noises could occur due to the wrong joint positions when joint values are simply added up to determine mean positions or weighted. This challenge should be addressed to retrieve postures. In particular, the selection of root and hip joints when the user turns is most challenging.

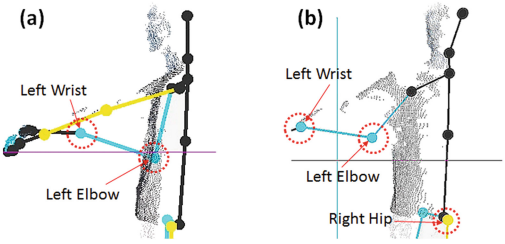


**Fig. 3.** Example of tracking error: (a) Arms are lifted forward, (b) User turns right

The present paper builds a model based on the user's initial posture and proposes a method of choosing the optimal joints for each part. The initial posture model is generated with the user standing with both arms spread wide while facing the front Kinect. At the same time, the length of each joint is measured. The initial model becomes the reference model, which is used to determine the left and right sides of the joints in the postures following. The lengths of joints may vary with the noises arising in the process of retrieving the posture. Thus, the reference model is used to correct the variation of joint lengths. As the Kinect does not tell left from right, the left joint seen from the front may be tracked as the right joint by another Kinect. Therefore, the calculation varies with whether it is necessary to distinguish left from right in retrieving a posture.

To retrieve the joints, the top nodes (root and hip joints) are first located. As aforementioned, it is not easy to find the accurate position of the hip joint because of lots of noises arising when the user rotates. In generating the initial model, not only the distance between the root and hips but also that between the hips should be measured. The closest values to the triangle (*LHip-Root-RHip*) measured in the initial model are found for the root and hip joint inputs from each Kinect. Based on the ratios of joint lengths, the joints whose values are below the given values are chosen. It is most likely that the data from the Kinect sensors facing the user and those placed in the rear are selected. Usually, up to two candidates are chosen, weighted based on similarities and mixed.

As the torso joints have no left and right sides clearly separated, mean values are used to calculate the joint positions, which are in turn adjusted based on the normal vectors associated with parent joints and the initial joint lengths.

As for the arms and legs, the K-means algorithm is used and the joint data are divided into two parts ($S_A$ and $S_B$), each of which is to become the left or right side. The arms start from the top nodes, or the shoulder joints. The minimal-difference pattern in the square values of the distance between the previous posture's shoulder joint positions($Lj_{fr-1}$, $Rj_{fr-1}$) and the two-part data's mean joint positions $(P_A, P_B)$ is used to determine the left and right arms. Likewise, the elbows/knees and the wrists/ankles are calculated based on the differences in distances. Still, the parent joints serve as the references for comparing the joints.

As aforementioned, given the mixed values found by weights are significantly affected by noises, the selection is made based on the variation of joint angles. As for the reference for joint selection, the joints $(J_s)$, which correspond to the minimal sum of the vector rotation direction $(D_s)$ and rotation angle $(A_s)$ calculated from the joint positions ($Pj_{\text{fr-2}}$, $Pj_{\text{fr-1}}$) and current joint position ($Pj_{\text{fr}}$), are selected.

Kinect finds the body parts based on learning data but sometimes fail to yield the joint values especially when feet go higher than the lower back as in kick motions. When no joint data are gained, the joint vector generated in previous postures (the vector between parent and children joints) is used to retrieve the posture.

## 5 Results

The present paper proposes a multiple Kinect system that captures motions in real time by minimizing the joint occlusion. As in Fig. 4, the user's posture can be retrieved although the hand is blocked or when the user rotates in 360°. Also, the proposed system can capture dynamic motions, e.g. Taekwondo.

Here, the proposed system is compared with the commercial motion capture system, Xsens to determine its accuracy. Xsens' data are saved as 120 and 240 frames. To match the initial setting, a T-pose is taken first. Motions are converted by matching the joint scales between Xsens' data and the multiple Kinect's data. To synchronize with

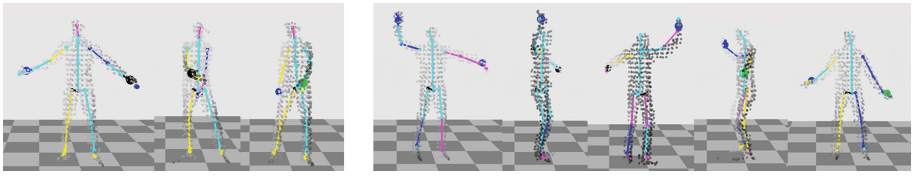

**Fig. 4.** Motion capture result: arm joint occluded by body (left), 360-degree left turn (right)

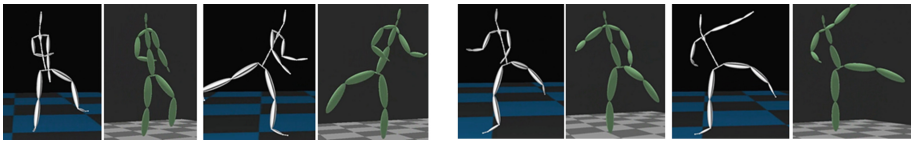

**Fig. 5.** Comparison of the motion: Xsens data (left-white), multi Kinects data (right-green)

the 30-fps Kinect data, the multiple Kinect system saves the data together with the timing of captures (milliseconds). The Xsens' data saved at a time closest to the recorded time are compared. The angular variation of joints from *N* data is calculated to compare the accuracy based on the difference in variation.

Figure 5 compares the data of two motions synchronized. 6 dynamic motions are measured in terms of the similarities of postures. The similarities are found as in Table 1. The lower extremity is less accurate than the upper one, because noises arise in the sensors attached to the feet in Xsens, and because errors occur in the lower extremity as the sensors are placed a bit high to increase the recognition of kicks in the multiple Kinect system. In particular, some motions such as the jump kick are not recognized by the multiple Kinect system, resulting in significant noises and errors (Fig. 6). Future research will draw upon the depth data to develop the technology for correcting the joint positions and for removing noises associated with legs and thus to increase the overall accuracy.

**Table 1.** Similarities of postures: six dynamic motions (M1–M6)

| | Right-Arm | Left-Arm | Right-Leg | Left-Leg | Average |
|---|---|---|---|---|---|
| M1. 2,572 frames (240 Hz) | 90.2 % | 91.7 % | 81.2 % | 83.1 % | 86.6 % |
| M2. 2,072 frames (240 Hz) | 90.3 % | 92.5 % | 79.2 % | 82.0 % | 86.0 % |
| M3. 4,371 frames (120 Hz) | 91.0 % | 91.6 % | 78.2 % | 78.1 % | 84.7 % |
| M4. 7,518 frames (120 Hz) | 91.1 % | 91.9 % | 79.2 % | 79.2 % | 85.4 % |
| M5. 10,060 frames (120 Hz) | 89.0 % | 90.0 % | 79.0 % | 78.9 % | 84.2 % |
| M6. 12,986 frames (120 Hz) | 90.0 % | 90.9 % | 79.6 % | 80.0 % | 85.1 % |
| Total 39,579 frames | 90.3 % | 91.6 % | 79.4 % | 80.2 % | 85.3 % |

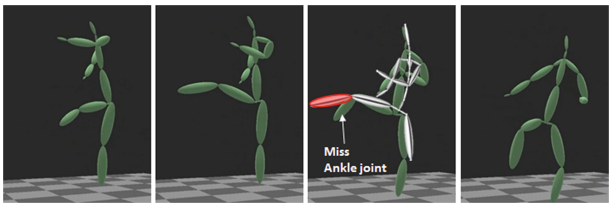


**Fig. 6.** Tracking failure of the ankle joint in the jump kick

**Acknowledgments.** This research project was supported by the Sports Promotion Fund of Seoul Olympic Sports Promotion Foundation from Ministry of Culture, Sports and Tourism.

## References

1. Zhang, L., Sturm, J., Cremers, D., Lee, D.: Real-time human motion tracking using multiple depth cameras. In: IEEE/RSJ IROS 2012, pp. 2389–2395 (2012)
2. Kitsikidis, A., Dimitropoulos, K., Douka, S., Grammalidis, N.: Dance analysis using multiple kinect sensors. In: VISAPP 2014, Lisbon, Portugal, pp. 5–8, January 2014

3. Kaenchan, S., Mongkolnam, P., Watanapa, B., Sathienpong, S.: Automatic multiple kinect cameras setting for simple walking posture analysis. In: ICSEC 2013, pp. 245–249 (2013)
4. Moon, S., Park, Y., Ko, D.W., Suh, I.H.: Multiple kinect sensor fusion for human skeleton tracking using Kalman filtering. Int. J. Adv. Robot Syst. **2016**, 1–10 (2016)
5. Ahmed, N.: Unified skeletal animation reconstruction with multiple kinects. In: Eurographics 2014, pp. 5–8 (2014)
6. Baek, S., Kim, M.: Dance experience system using multiple kinects. Int. J. Future Comput. Commun. **4**(1), 45–49 (2015)
7. Jo, H., Yu, H., Kim, K., Jung, H.S.: Motion tracking system for multi-user with multiple kinects. IJUNESST **8**(7), 99–108 (2015)