International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2017)
R. Lindeman, G. Bruder, and D. Iwai (Editors)

A Mutual Motion Capture System for Face-to-face Collaboration

Atsuyuki Nakamura[1], Kiyoshi Kiyokawa[1,2], Photchara Ratsamee[1,3], Tomohiro Mashita[1,3], Yuki Uranishi[1,3], and Haruo Takemura[1,3]

[1]Graduate School of Information Science and Technology, Osaka University, Japan
[2]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[3]Cybermedia Center, Osaka University, Japan

Abstract

In recent years, motion capture technology to measure the movement of the body has been used in many fields. Moreover, motion capture targeting multiple people is becoming necessary in multi-user virtual reality (VR) and augmented reality (AR) environments. It is desirable that motion capture requires no wearable devices to capture natural motion easily. Some systems require no wearable devices using an RGB-D camera fixed in the environment, but the user has to stay in front of the fixed the RGB-D camera. Therefore, in this research, proposed is a motion capture technique for a multi-user VR / AR environment using head mounted displays (HMDs), that does not limit the working range of the user nor require any wearable devices. In the proposed technique, an RGB-D camera is attached to each HMD and motion capture is carried out mutually. The motion capture accuracy is improved by modifying the depth image. A prototype system has been implemented to evaluate the effectiveness of the proposed method and motion capture accuracy has been compared with two conditions, with and without depth information correction while rotating the RGB-D camera. As a result, it was confirmed that the proposed method could decrease the number of frames with erroneous motion capture by 49% to 100% in comparison with the case without depth image conversion.

CCS Concepts

•***Human-centered computing*** → ***Mixed / augmented reality;*** *Virtual reality; Collaborative interaction;*

1. INTRODUCTION

Many systems have been developed that support human collaboration. In the field of computer-aided cooperative work (CSCW), such systems are classified into four types depending on whether the interaction is synchronous or asynchronous and whether participants are co-located or remotely distributed [Rod91]. In the present research, we will deal with synchronous co-located face-to-face collaboration powered by virtual reality (VR) or augmented reality (AR) that uses head mounted displays (HMDs).

In co-located collaboration, being able to see each other greatly helps collaboration. According to the study by Billinghurst et al. [BBGK03], sitting face-to-face with see-through HMDs makes the collaboration more natural and efficient than sitting side-by-side with a shared wall screen. Numerous AR systems support face-to-face collaboration [OSYT99,KNEO01]. For example, Studierstube [SSFG98] allows users to share information displayed in AR while being able to see each other. However, most of such systems require additional devices other than HMDs. In recent years, standalone HMDs such as Microsoft HoloLens and Google Glass have appeared and are gradually disseminating. Because such standalone HMDs are easier to be deployed without requiring additional hardware, their application scenarios are expanding including those support face-to-face collaboration. In collaboration using VR or AR, it is also desirable to have an easy-to-use motion capture system without requiring additional hardware in order to support natural body interaction in a similar manner to collaboration in the real environment.

Our goal is to develop an HMD-based easy-to-use motion capture system that requires no additional hardware other than the headset and has no limitation of working place. In this paper, we report on such a motion capture system suitable for face-to-face collaboration within VR or AR where each participant wears an HMD. We propose to attach a motion capture device on top of each HMD worn by users thereby capturing other users' motion mutually. By doing this, we minimize the device attached to the body, reduce the complication of attachment and detachment, and realize a motion capture system that does not limit the working volume.

2. RELATED WORK

In this section, we will introduce existing motion capture systems and discuss their characteristics. After that, we will introduce similar research that uses ego-centric video or first-person views.

Motion capture systems can be classified into three types according to their mechanical grounding configurations; the wearable-and-stationary type, the wearable type, and the stationary type.The wearable-and-stationary type motion capture systems can capture user motion by detecting markers or sensors attached to the user

DOI: 10.2312/egve.20171332

by cameras or sensors installed in the environment. For example, OptiTrack [OptiTrack] uses retroreflective markers attached to the body parts and a set of infrared cameras. User movements are measured by capturing the reflected infrared light and detecting marker positions. Magnetic motion capture systems such as LIBERTY of Polhemus [LIBERTY] and trakSTAR of Ascension [trakSTAR] use a transmitter applying a magnetic field and receivers which detect the magnetic field. Motion capture accuracy of the wearable-and-stationary type is generally very high, because on-body markers or sensors are detected by several cameras or sensors fixed in the environment. However, it is cumbersome to attach markers or sensors to the body parts and the working volume is limited because they need to be seen by the stationary devices.

The wearable type motion capture system measures user movements solely with sensors attached to the user. For example, those systems using inertial sensors [PERCEPTION NEURON, SH08, RLS09] can estimate user movement by integrating the values of the accelerometers attached to the body parts. Some motion capture systems use goniometers [TMSF86] to measure joint angles. Although the wearable type systems do not require a stationary device installed in the environment and the working volume is not limited, it is cumbersome and time-consuming to attach sensors to the body parts. In addition, wearable sensors often make the user uncomfortable and prevent natural body motion.

The stationary type motion capture system measures user movements using one or more image sensors that are fixed in the environment. For example, Tanaka et al.'s system [TNT08] uses eight RGB cameras around the user. Their system reconstructs a user volume by space carving and acquires the skeleton data by thinning the volume data. Microsoft's Kinect [SSK*13] can estimate human pose from a single depth image. Kinect uses machine learning with a depth image. Real-time human pose estimation using a single RGB camera has been a challenge, but it is becoming practical. For example, OpenPose [CSWS16] can estimate 2D pose of multi-person from a single image. Although users do not have to wear any devices, the working place is limited to the vicinity of the camera, which is typically fixed in the environment. Some systems [SK13, TSG14, SYJW16] cover a wide area by multiple depth sensors. To cover a wider area, these systems need more cameras and preparation time. In our system, we use an RGB-D camera as a motion capture device of the stationary type to HMD because it doesn't require any other device to capture human motion.

Some studies capture user motion using egocentric videos or first-person views. Ardeshir et al. [AB16] and Fan et al. [FLX*17] match an egocentric video and the viewer in a third-person video. Rogez et al. [RSK*14] and Sridhar et al. [SMOT15] use a body attached RGB-D camera for hand pose estimation. Bambach et al. [BCY15] developed a system to recognize hand activities from multiple first-person video streams. There are motion capture systems [CHC*15, RRC*16] that use one or more fisheye camera(s) attached to the user. Ess et al. [ELSV08] developed a multi-person tracking system to detect pedestrians, visualize their odometry, and estimate the depth with a pair of synchronized first-person views. The multi-person tracking system of Gammerter et al. [GEJ*08] uses the first-person view under ego-motion. A system of Yonetani et al. [YKS16] can recognize micro-actions and reactions from a paired egocentric video streams. Our system is a motion capture system using a pair of depth image sensors attached on top of each user mutually facing the collaboration partner.

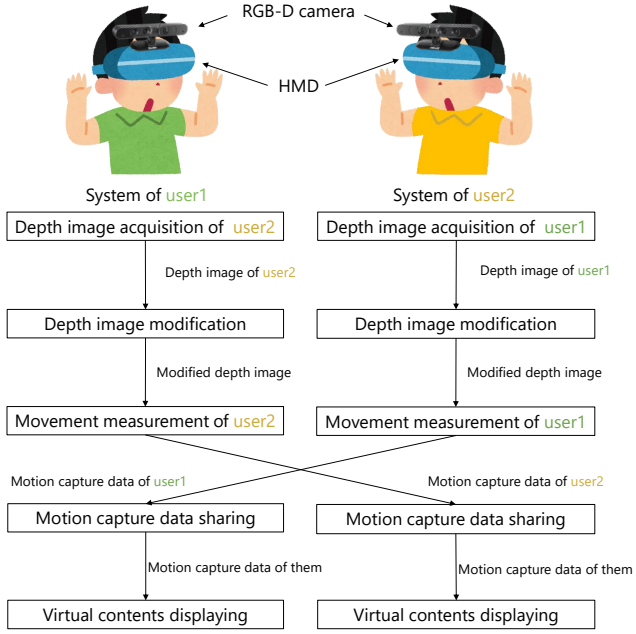


Figure 1: *Overview of the proposed system.*

3. A MUTUAL MOTION CAPTURE SYSTEM FOR FACE-TO-FACE COLLABORATION

In this section, we will introduce an overview of our proposed system and its process flow in detail.

3.1. Overview

The proposed motion capture system targets face-to-face collaboration within VR or AR using HMDs. Because the participants are facing each other, we can assume that a head mounted camera is also facing the other participants most of the time. In our system, motion capture of a user is mutually performed by an RGB-D camera mounted on the facing user's HMD. However, a raw depth image is not suitable for motion capture when the RGB-D camera is translated or rotated, so we modify the depth image as described below. After motion data is captured, it is shared by all users, and their movements will also be measured, as shown in Fig. 1.

3.2. Depth Image Correction

In our system, user motion is mutually captured by an RGB-D camera attached on a facing user's HMD. However, it is expected that the movement of the RGB-D camera will degrade the stability and the quality of the motion capture. This is because a standard motion capture algorithm using a depth image [SSK*13] assumes that the RGB-D camera is fixed horizontally in the environment orienting

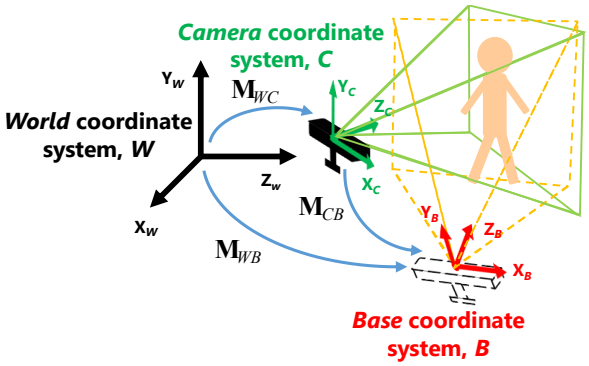


Figure 2: *Coordinate systems in the proposed system.*

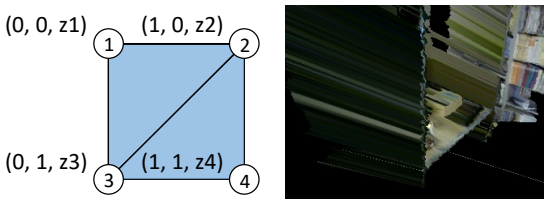


Figure 3: *Creating a triangle mesh: (a) how to create a mesh, and (b) false surfaces generated by connecting all the points without thresholding.*

toward the target user. In order to address this problem, we propose two methods to modify the depth image.

3.2.1. Coordinate transformation

We transform a depth image for more robust motion capture. As illustrated in Fig. 2, three coordinate systems are involved in this transformation; *Camera* coordinate system, *World* coordinate system, and *Base* coordinate system, denoted as C, W, and B, respectively. *Base* coordinate system, B, is dynamically positioned near *Camera* coordinate system that is assumed to be more appropriate for motion capture. A transformation matrix from *World* coordinate system to *Camera* coordinate system, denoted as $\mathbf{M}_{WC}$, represents the RGB-D camera pose in *World* coordinate system. $\mathbf{M}_{WC}$ can be estimated by a variety of approaches, e.g., visual SLAM [DRMS07, KSC13, ESC14] with RGB-D images, an inertial sensor in HMD, or a localization system [MEE14] combined them. In our prototype system, we use an inertial sensor embedded in the HMD. $\mathbf{M}_{WC}$ has Rotation components $\mathbf{R}$ and translation components $\mathbf{t}$. They can be expressed as Eq. 1 and Eq. 2, respectively,

$$\mathbf{R} = \mathbf{R}_x(\varphi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi) \quad (1)$$

$$\mathbf{t} = (tx, ty, tz) \quad (2)$$

where $\mathbf{R}_x, \mathbf{R}_y$, and $\mathbf{R}_z$ are rotation matrices around X, Y and Z axes, respectively. Similarly to $\mathbf{M}_{WC}$, the transformation matrix from *World* coordinate system to *Base* coordinate system, denoted as $\mathbf{M}_{WB}$, has rotation components $\mathbf{R}'$ and translation components $\mathbf{t}'$.

Similarly to $\mathbf{M}_{WC}$, the transformation matrix from *World* coordinate system to *Base* coordinate system, denoted as $\mathbf{M}_{WB}$, has rotation components $\mathbf{R}'$ and translation components $\mathbf{t}'$. It is assumed that the motion capture algorithm with an RGB-D camera [SSK*13] works correctly if the RGB-D camera is placed horizontally and oriented straight toward the target user at an appropriate distance. Therefore, by using a horizontal position tx' and tz' and an azimuthal angle θ', $\mathbf{R}'$ and $\mathbf{t}'$ can be further expressed as in Eq. 3 and Eq. 4, respectively. tx', tz' and θ' can be automatically determined by some geometric constraints, for example, the distance to the user is within a reasonable range (e.g. 2 to 3m), the target user is on its Z axis, and $\mathbf{t}'$ is close to $\mathbf{t}$ as possible.

$$\mathbf{R}' = \mathbf{R}_y(\theta') \quad (3)$$

$$\mathbf{t}' = (tx', ty, tz') \quad (4)$$

Then we modify the depth image by a transformation matrix from *Camera* coordinate system to *Base* coordinate system, $\mathbf{M}_{CB}$, which is simply calculated as in Eq. 5. By doing this, we obtain a new depth image that are taken from a virtual RGB-D camera placed at the origin of *Base* coordinate system.

$$\mathbf{M}_{CB} = {\mathbf{M}_{WC}}^{-1}\mathbf{M}_{WB} \quad (5)$$

Now we can obtain motion capture data by using the new depth image, but their positions are in *Base* coordinate system. We then transform found joint positions in *Base* coordinate system, $\mathbf{P}_B$, into those in *World* coordinate system, $\mathbf{P}_W$, by Eq. 6 below.

$$\mathbf{P}_W = {\mathbf{M}_{WB}}^{-1}\mathbf{P}_B \quad (6)$$

3.2.2. Hole filling

A depth image contains a discrete 2.5-dimensional information of a 3D space. Once the entire point cloud is transformed into another coordinate system and rendered from its origin as the new viewpoint, there will be holes and cracks in the areas that were not observable from the original camera position. To prevent motion capture degradation due to this lack of depths, we implement a hole filling method. Although there are many hole filling methods [SF11, CS12, PKT*14], we use a very simple and fast solution for real-time processing. In our system, we create a triangle mesh connecting adjacent points in the point cloud as shown in Fig. 3 (a) to fill the holes and cracks before coordinate transformation. If we create a mesh for all the points, false surfaces that do not exist in reality will be generated and they often conceal other surfaces as shown in Fig. 3 (b). To prevent this, a distance threshold th is introduced. For example in Fig. 3 (a), we do not create a surface among points where the absolute depth value of $(z1 - z4)$ or $(z2 - z3)$ is bigger than th.

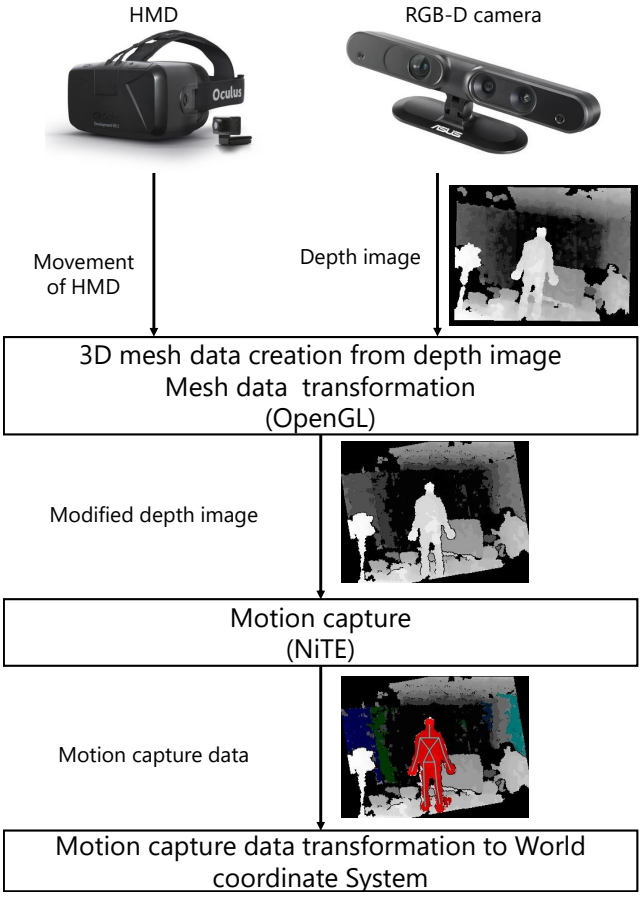


Figure 4: *Module diagram of the proposed system.*

4. Prototype System

In this section, we present a prototype system implemented to evaluate the effectiveness of the proposed methods and describe the preliminary experiment.

4.1. Implementation

Figure 4 shows a module diagram of the prototype system. The prototype consists of a desktop computer (CPU: Core i7-6700k, memory: 8GB×2 and GPU: NVIDIA GeForce GTX 1080 8GB), an RGB-D camera (ASUS Xtion PRO LIVE, resolution: 320×240, frame rate: 60 Hz, weight: 210g), and an HMD (Oculus Rift DK2, resolution: 2160×1200, weight: 440g). OpenNI 1.5.4.0 for Windows, NiTE 1.5.2.21 for Windows, PrimeSense Sensor 5.1.2.1 for Windows are used in the program. OpenGL 1.1 and glut 3.7 are also used to generate point cloud data from depth images and to transform them.

4.2. Preliminary experiment

As a preliminary experiment, we examine if coordinate transformation and hole filling work properly. We also measure the processing time, because our system targets face-to-face real-time collaboration within VR or AR. For simplicity, we fix *Base* coordinate system at *World* coordinate system, namely, $\theta = 0$, $tx' = 0$ and $tz' = 0$. In this case, depth image modification generates a depth image canceled RGB-D camera's transformation and rotation. The threshold th for mesh generation is set to 10 cm. During the experiment, we only rotate the HMD together with the RGB-D camera on a tripod around three axes as shown in Fig. 8. So, we can roughly assume that $tx \simeq tx' = 0$ and $tz \simeq tz' = 0$. We measure φ, θ, and ψ by an inertial sensor embedded in the HMD.

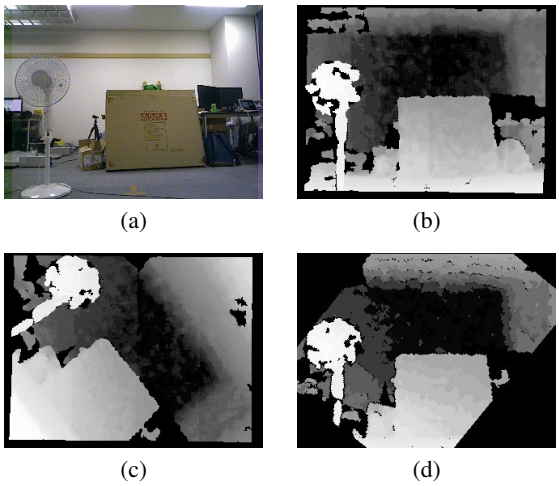


Figure 5: *Example coordinate transformation: (a) an RGB image of an environment, (b) a depth image of the same environment, (c) a depth image after camera rotation around the roll axis, and (d) a modified depth image after coordinate transformation.*

4.2.1. Coordinate transformation

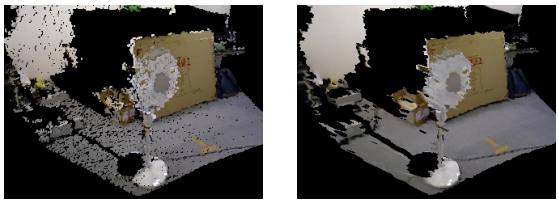
Figure 5 shows an example of coordinate transformation. Figure 5 (a) and (b) are an RGB image and a depth image of an environment, respectively. Figure 5 (c) is a depth image taken after rotating the RGB-D camera around the roll axis. Figure 5 (d) is a modified depth image generated by rendering a transformed point cloud. From these figures, we can confirm that the coordinate transformation works properly so that the system can virtually cancel the RGB-D camera's movement and synthesize a depth image rendered from the origin of *Base* coordinate system.

4.2.2. Hole filling

Figure 6 shows an example of hole filling. Figure 6 (a) is a colored point cloud generated from the depth image in Fig. 5 (b). Figure 6 (b) is a triangle mesh with thresholding generated from the same depth image. From these figures, we can confirm that hole filling works properly to some extent. Note that they are rendered from an oblique angle to emphasize the difference.

4.2.3. Processing time

In measurement of the processing time, we compared three conditions; a naive motion capture condition without the proposed meth-



(a) (b)

Figure 6: *Example of hole filling: (a) a point cloud and (b) a triangle mesh generated from the point cloud with thresholding.*

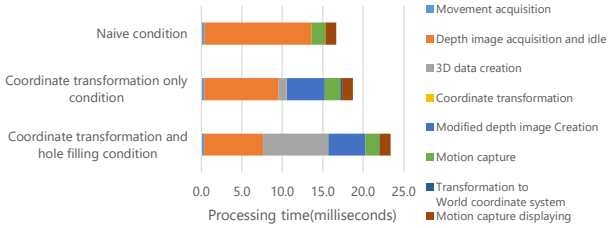


Figure 7: *Processing time.*

ods, a condition with coordinate transformation without hole filling, and a condition with coordinate transformation and hole filling. We measured the processing time of motion capture targeting a person standing still as illustrated in Fig. 9 in the three conditions. Figure 7 shows the break-down processing time in each condition. In the figure, '3D data creation' is the time for creating a point cloud or a triangle mesh from it, and 'Coordinate transformation' is the time for transforming the acquired motion capture data from *Base* coordinate system to *World* coordinate system. Frame rates of the prototype system were approximately 60 frames/second, 52 frames/second, and 42 frames/second, for the naive condition, the coordinate transformation only condition, and the coordinate transformation and hole filling condition, respectively. These results show that the proposed system runs in real-time with a little slowdown.

Figure 8: *Rotation axes.*

Figure 9: *Subject of main experiment.*

Table 1: *Motion sequence.*

Rotation	Time
Hold	1 second
Rotation counterclockwise by 10 degrees	1 second
Hold	1 second
Rotation clockwise by 20 degrees	2 seconds
Hold	1 second
Rotation counterclockwise by 10 degrees	1 second
Hold	3 seconds

5. MAIN EXPERIMENT

5.1. Procedure

In the main experiment, we examine if the proposed methods improve the accuracy and stability of motion capture. We use the same hardware and software configurations as in the preliminary experiment, targeting a standing person and recording the captured images while rotating the RGB-D camera around one of pitch, yaw, and roll axes manually in a sequence shown in Table 1. The recorded images are then fed to the prototype system with the same three conditions as in the preliminary experiment. To compare the motion capture accuracy in each condition, we define the error as the distance between the center of a visual marker attached to the body part of the target subject and the reprojected screen coordinates of the corresponding joint as in Eq. 7,

$$Error = ||\mathbf{x} - \mathbf{x}^*|| \tag{7}$$

where $\mathbf{x}$ is a position from the motion capture data in the depth image and $\mathbf{x}^*$ is the center of the corresponding marker attached to the target.

5.2. Results

Figure 10 shows the motion capture errors for different parts of the body in each condition. We can observe that motion capture with coordinate transformation is more robust and errors in those conditions are generally smaller than that in the naive condition without transformation. The differences among the three conditions are particularly clear in the case of rotation around the roll axis.

Figure 11 shows typical examples of motion capture at a moment during rotation around the roll axis. As Figure 11 (a) shows, erroneous motion capture is often caused by wrong human region extraction which is attributed to rotated depth images. As Figure 11 (b) shows, coordinate transformation suppresses such errors, however, human region is not always fully extracted due to holes and cracks yielding the recognized skeleton in a wrong size or pose. And as Figure 11 (c) shows, hole filling makes human region extraction more robust and further decreases the error.

We conducted t-tests between errors for different conditions for each body part in the case of camera rotation around the roll axis. Comparing the errors for each body part in the naive condition and the coordinate transformation only condition, p values for all body parts are less than 0.05 (p = 3.1E-21 for head, p = 2.4E-09 for left

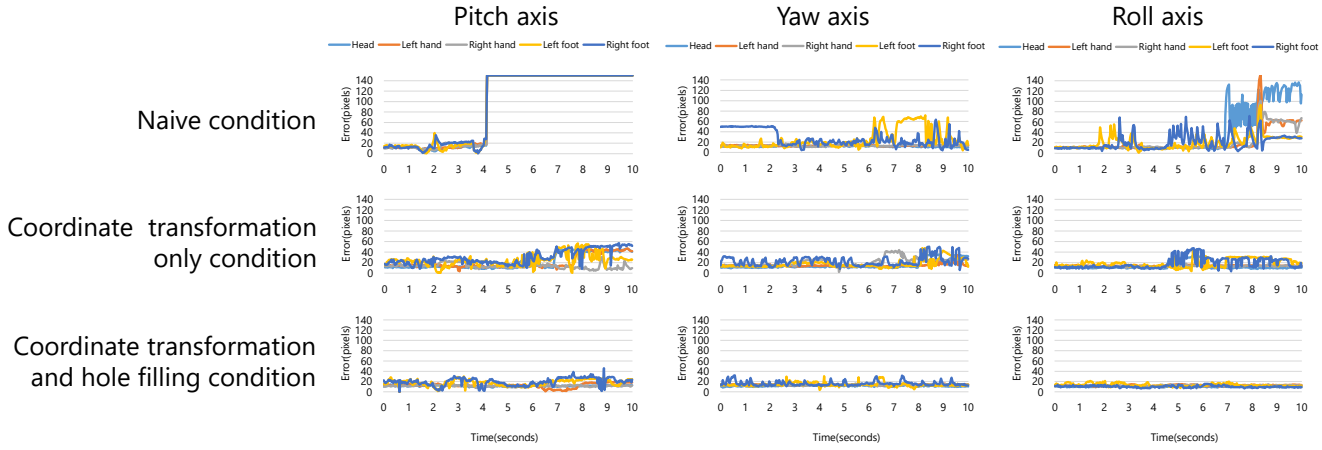
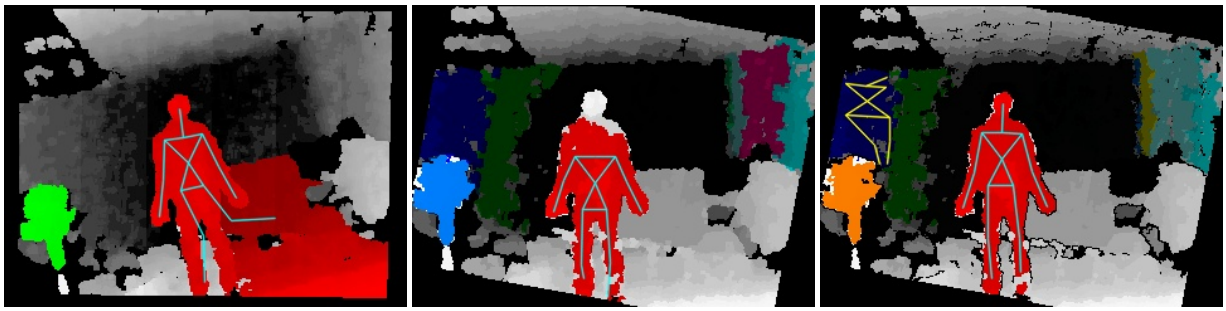


Figure 10: *Error of motion capture.*



(a) (b) (c)

Figure 11: *Example of motion capture in the case of rotation around the roll axis: (a) naive motion capture without transformation, (b) coordinate transformation without hole filling, and (c) coordinate transformation with hole filling.*

hand, p = 2.6E-12 for right hand, p = 0.011 for left leg, and p = 0.0022 for right leg), therefore, it was proven that the coordinate transformation decreases the errors of motion capture. And comparing the errors in the coordinate transformation only condition and the coordinate transformation and hole filling condition, p values for all body parts are less than 0.01 (p = 2.4E-15 for head, p = 6.5E-116 for left hand, p = 1.1E-48 for right hand, p = 2.7E-29 for left foot, and p = 6.6E-42 for right foot), therefore, it was proven that the hole filling decreases further the error of motion capture. In these results, both the coordinate transformation and the hole filling are effective for motion capture in our system.

The motion capture algorithm used in the prototype [SSK*13] heavily relies on the learning dataset, and those body postures that are not included in it are much more difficult to capture. In the case of rotation around the yaw axis, motion capture errors are much smaller even without coordinate transformation. This is because horizontal movement occurs in natural motion and it is already learned in the algorithm. On the other hand, rotation around the pitch axis severely impacts the accuracy and it became impossible to capture motion at all beyond a certain amount of rotation. This is because vertical movement of the entire body rarely occurs in natural motion and it is not learned in the algorithm.

Figure 12 shows the total number of frames that had an error larger than 17 pixels (approximately equivalent to the width of a forearm) for each part of the body in each condition. Comparing to the condition without coordinate transformation, that with coordinate transformation without hole filling decreased the number of such erroneous frames by 17%, 57%, 25%, 4%, 0% for head, left hand, right hand, left foot, and right foot, respectively. This result again confirms that coordinate transformation is effective in improving robustness of motion capture against camera movement. Comparing to the condition with coordinate transformation without hole filling, the condition with hole filling further decreased the number of erroneous frames by 100%, 79%, 100%, 49%, 49% for head, left hand, right hand, left foot, and right foot, respectively. This result confirms that hole filling is also effective for robust motion capture.

6. DISCUSSION

The results indicate that depth image modification canceling camera rotation improves the motion capture accuracy while the RGB-D camera is moving. In this experiment, the depth image is modified only by rotation. We need to investigate the effectiveness of depth image modification for camera translation and more natural

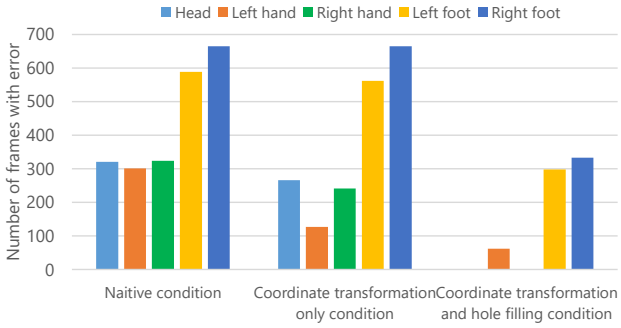


Figure 12: *Number of frames with erroneous motion capture.*

camera motion both in rotation and translation. In actual face-to-face collaboration, the head-attached RGB-D cameras will move very quickly and motion blur will frequently occur. To address the out-of-frame problem due to head rotation, stitching depth images to create a wider view will be effective [TCK*15] Our coordinate transformation and hole filling will be effective to depth images with motion blur to some extent. However, we will need to introduce some compensation method to cope better with motion blur and intra-frame time differences due to the rolling shutter [TMN*16]. Our proposed system is designed for AR and VR environments. To display virtual contents on the HMDs, we need to acquire users' position in the world coordinates. Nowadays, many self-localization techniques are available. For example, visual SLAM with or without depth sensors can be used with our system [DRMS07, IKH*11, KSC13, ESC14, NFS15]. Moreover, visual-inertial SLAM with an RGB camera and an inertial sensor on the headset can also be effective. The users of our system are expected to face each other. Therefore, the motion of a user who is not seen by any other user cannot be captured (for example, when all users look down to see the shared content on the table). In such a case, the position (at least the head) of the lost user can be calculated by one's own sensors using SLAM methods. For a collaboration with three or more users, sharing and completing motion data will also help accurate motion capture. If more users use our system, a wider area can be covered and more users can be tracked. With multiple users, a user will be more often seen by more than one user. In this case, the user will be detected from different views. To recognize a single user from such multiple views our system will need to integrate similar 3D skeletons into one in real-time.

7. CONCLUSION AND FUTURE WORK

In this research, we proposed a motion capture system for HMD-based face-to-face collaboration within VR or AR, which needs for minimum devices and no limitation of working place. In this system, we generate a novel depth image from a virtual viewpoint by coordinate transformation and hole filling that is considered to yield more robust motion capture. Through experiments, it was confirmed that the proposed methods run in real-time and greatly improve the robustness of motion capture.

In the future, we will develop a coordinate transformation method that automatically optimizes θ', tx', tz' in *Base* coordinate system at run-time. To do this, we have to implement a method to get user positions in the world coordinate system. After that, our system can communicate and share the users' motion data. In our prototype system, we used ASUS Xtion PRO LIVE because it weighs only 210g and can be used as a motion capture device with OpenNI. However, its measurable range is short (up to 3.5m) and multiple of such light-coding depth sensors will interfere with each other. Therefore, finding an appropriate depth sensor so that multiple of them can be used at the same time will also be an issue to address. Moreover, we will also develop a complete face-to-face collaboration system and verify the motion capture system through experiments in which, using the system, users do face-to-face collaboration task within VR or AR environment.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Number JP16H02858.

References

[AB16] ARDESHIR S., BORJI A.: Ego2top: Matching viewers in egocentric and top-view videos. In *Proc. ECCV 2016* (2016), Springer, pp. 253–268. doi:10.1007/978-3-319-46454-1_16. 2

[BBGK03] BILLINGHURST M., BELCHER D., GUPTA A., KIYOKAWA K.: Communication behaviors in colocated collaborative ar interfaces. *IJHCI 16*, 3 (2003), 395–423. doi:10.1207/S15327590IJHC1603_2. 1

[BCY15] BAMBACH S., CRANDALL D. J., YU C.: Viewpoint integration for hand-based recognition of social interactions from a first-person view. In *Proc. ICMI 2015* (2015), ACM, pp. 351–354. doi:10.1145/2818346.2820771. 2

[CHC*15] CHAN L., HSIEH C.-H., CHEN Y.-L., YANG S., HUANG D.-Y., LIANG R.-H., CHEN B.-Y.: Cyclops: Wearable and single-piece full-body gesture input devices. In *Proc. CHI 2015* (2015), ACM, pp. 3001–3009. doi:10.1145/2702123.2702464. 2

[CS12] CAMPLANI M., SALGADO L.: Efficient spatio-temporal hole filling strategy for kinect depth maps. *3DIP and Applications 8290* (2012). doi:10.1117/12.911909. 3

[CSWS16] CAO Z., SIMON T., WEI S.-E., SHEIKH Y.: Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050* (2016). 2

[DRMS07] DAVISON A. J., REID I. D., MOLTON N. D., STASSE O.: Monoslam: Real-time single camera slam. *IEEE TPAMI 29*, 6 (2007), 1052–1067. doi:10.1109/TPAMI.2007.1049. 3, 7

[ELSV08] ESS A., LEIBE B., SCHINDLER K., VAN GOOL L.: A mobile vision system for robust multi-person tracking. In *Proc. CVPR 2008* (2008), IEEE, pp. 1–8. doi:10.1109/CVPR.2008.4587581. 2

[ESC14] ENGEL J., SCHÖPS T., CREMERS D.: Lsd-slam: Large-scale direct monocular slam. In *Proc. ECCV 2014* (2014), Springer, pp. 834–849. 3, 7

[FLX*17] FAN C., LEE J., XU M., SINGH K. K., LEE Y. J., CRANDALL D. J., RYOO M. S.: Identifying first-person camera wearers in third-person videos. *arXiv preprint arXiv:1704.06340* (2017). 2

[GEJ*08] GAMMETER S., ESS A., JÄGGLI T., SCHINDLER K., LEIBE B., VAN GOOL L.: Articulated multi-body tracking under egomotion. *ECCV* (2008), 816–830. doi:10.1007/978-3-540-88688-4_60. 2

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: Kinectfusion: real-time 3d reconstruction and

interaction using a moving depth camera. In *Proc. UIST 2011* (2011), ACM, pp. 559–568. doi:10.1145/2047196.2047270. 7

[KNEO01] KIYOKAWA K., NIIMI M., EBINA T., OHNO H.: Mr^2(mr square): a mixed-reality meeting room. In *Proc. ISAR 2001* (2001), IEEE, pp. 169–170. doi:10.1109/ISAR.2001.970526. 1

[KSC13] KERL C., STURM J., CREMERS D.: Dense visual slam for rgb-d cameras. In *Proc. IROS 2013* (2013), IEEE, pp. 2100–2106. doi:10.1109/IROS.2013.6696650. 3, 7

[LIBERTY] : *LIBERTY*. Accessed:2017-2-13. http://polhemus.com/motion-tracking/all-trackers/liberty. 2

[MEE14] MOSTOFI N., ELHABIBY M., EL-SHEIMY N.: Indoor localization and mapping using camera and inertial measurement unit (imu). In *Proc. PLANS 2014* (2014), IEEE, pp. 1329–1335. doi:10.1109/PLANS.2014.6851507. 3

[NFS15] NEWCOMBE R. A., FOX D., SEITZ S. M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. CVPR 2015* (2015), pp. 343–352. doi:10.1109/CVPR.2015.7298631. 7

[OptiTrack] : *OptiTrack*. Accessed:2017-2-13. https://www.optitrack.co.jp/. 2

[OSYT99] OHSHIMA T., SATOH K., YAMAMOTO H., TAMURA H.: Rv-border guards: A multi-player mixed reality entertainment. *Trans. VRSJ 4*, 4 (1999), 699–705. doi:10.18974/tvrsj.4.4_699. 1

[PERCEPTION NEURON] : *PERCEPTION NEURON*. Accessed:2017-1-25. https://neuronmocap.com/ja/products/perception_neuron. 2

[PKT*14] PARK J., KIM H., TAI Y.-W., BROWN M. S., KWEON I. S.: High-quality depth map upsampling and completion for rgb-d cameras. *IEEE TIP 23*, 12 (2014), 5559–5572. doi:10.1109/TIP.2014.2361034. 3

[RLS09] ROETENBERG D., LUINGE H., SLYCKE P.: Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV* (2009). 2

[Rod91] RODDEN T.: A survey of cscw systems. *IwC 3*, 3 (1991), 319–353. doi:10.1016/0953-5438(91)90020-3. 1

[RRC*16] RHODIN H., RICHARDT C., CASAS D., INSAFUTDINOV E., SHAFIEI M., SEIDEL H.-P., SCHIELE B., THEOBALT C.: Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM TOG 35*, 6 (2016), 162. doi:10.1145/2980179.2980235. 2

[RSK*14] ROGEZ G., SUPANCIC III J. S., KHADEMI M., MONTIEL J. M. M., RAMANAN D.: 3d hand pose detection in egocentric rgb-d images. *arXiv preprint arXiv:1412.0065* (2014). doi:10.1007/978-3-319-16178-5_25. 2

[SF11] SILBERMAN N., FERGUS R.: Indoor scene segmentation using a structured light sensor. In *Proc. ICCV Workshops 2011* (2011), IEEE, pp. 601–608. doi:10.1109/ICCVW.2011.6130298. 3

[SH08] SLYPER R., HODGINS J. K.: Action capture with accelerometers. In *Proc. SIGGRAPH 2008/Eurographics Symposium on Computer Animation* (2008), Eurographics Association, pp. 193–199. doi:10.2312/SCA/SCA08/193-199. 2

[SK13] SCHÖNAUER C., KAUFMANN H.: Wide area motion tracking using consumer hardware. *IJVR 12*, 1 (2013). 2

[SMOT15] SRIDHAR S., MUELLER F., OULASVIRTA A., THEOBALT C.: Fast and robust hand tracking using detection-guided optimization. In *Proc. CVPR 2015* (2015), pp. 3213–3221. doi:10.1109/CVPR.2015.7298941. 2

[SSFG98] SZALAVÁRI Z., SCHMALSTIEG D., FUHRMANN A., GERVAUTZ M.: "studierstube": An environment for collaboration in augmented reality. *Virtual Reality 3*, 1 (1998), 37–48. doi:10.1007/BF01409796. 1

[SSK*13] SHOTTON J., SHARP T., KIPMAN A., FITZGIBBON A., FINOCCHIO M., BLAKE A., COOK M., MOORE R.: Real-time human pose recognition in parts from single depth images. *Communications of the ACM 56*, 1 (2013), 116–124. doi:10.1145/2398356.2398381. 2, 3, 6

[SYJW16] SONG W., YUN S., JUNG S.-W., WON C. S.: Rotated top-bottom dual-kinect for improved field of view. *MTAP 75*, 14 (2016), 8569–8593. doi:10.1007/s11042-015-2772-5. 2

[TCK*15] TAYLOR C. J., COWLEY A., KETTLER R., NINOMIYA K., GUPTA M., NIU B.: Mapping with depth panoramas. In *Proc. IROS 2015* (2015), IEEE, pp. 6265–6272. doi:10.1109/IROS.2015.7354271. 7

[TMN*16] TOURANI S., MITTAL S., NAGARIYA A., CHARI V., KRISHNA M.: Rolling shutter and motion blur removal for depth cameras. In *Proc. ICRA 2016* (2016), IEEE, pp. 5098–5105. doi:10.1109/ICRA.2016.7487715. 7

[TMSF86] TAGUCHI H., MASUDA K., SHIMIZU T., FUJII K.: Handwriting analysis technique using an electro-goniometer. *Biomechanisms 8* (1986), 119–130. doi:10.3951/biomechanisms.8.119. 2

[TNT08] TANAKA H., NAKAZAWA A., TAKEMURA H.: Example based approach for human pose estimation using volume data and graph matching. *The IEICE transactions. D 91*, 6 (2008), 1580–1591. doi:10.11371/wiieej.06-04.0_57. 2

[trakSTAR] : *trakSTAR*. Accessed:2017-1-25. https://www.ascension-tech.com/products/. 2

[TSG14] TAN Y. F., SAIN M., GOOK L. B.: User detection in real-time panoramic view through image synchronization using multiple camera in cloud. In *Proc. ICACT 2014* (2014), IEEE, pp. 1118–1123. doi:10.1109/ICACT.2014.6779133. 2

[YKS16] YONETANI R., KITANI K. M., SATO Y.: Recognizing micro-actions and reactions from paired egocentric videos. In *Proc. CVPR 2016* (2016), pp. 2629–2638. doi:10.1109/CVPR.2016.288. 2