A Distributed Real-time 3D Pose Estimation Framework based on Asynchronous Multiviews

Taemin Hwang[1], **Jieun Kim**[1] **and Minjoon Kim**[1,*]
[1] Department of Data Convergence Platform Research Center
Korea Electronics Technology Institute
Seongnam, South Korea
[e-mail: taemin.hwang@keti.re.kr, jekim@keti.re.kr, mjoon@keti.re.kr]
*Corresponding author: Minjoon Kim

Received August 9, 2022; revised October 7, 2022; accepted October 27, 2022; published February 28, 2023

Abstract

3D human pose estimation is widely applied in various fields, including action recognition, sports analysis, and human-computer interaction. 3D human pose estimation has achieved significant progress with the introduction of convolutional neural network (CNN). Recently, several researches have proposed the use of multiview approaches to avoid occlusions in single-view approaches. However, as the number of cameras increases, a 3D pose estimation system relying on a CNN may lack in computational resources. In addition, when a single host system uses multiple cameras, the data transition speed becomes inadequate owing to bandwidth limitations. To address this problem, we propose a distributed real-time 3D pose estimation framework based on asynchronous multiple cameras. The proposed framework comprises a central server and multiple edge devices. Each multiple-edge device estimates a 2D human pose from its view and sends it to the central server. Subsequently, the central server synchronizes the received 2D human pose data based on the timestamps. Finally, the central server reconstructs a 3D human pose using geometrical triangulation. We demonstrate that the proposed framework increases the percentage of detected joints and successfully estimates 3D human poses in real-time.

Keywords: Computer vision, edge processing, multiple view geometry, object detection, pose estimation.

A preliminary version of this paper was presented at APIC-IST 2022, and was selected as an outstanding paper. This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project number: R2021040128)

http://doi.org/10.3837/tiis.2023.02.015 **ISSN : 1976-7277**

1. Introduction

Reconstructing 3D human poses accurately has been a long-standing problem in computer vision. The 3D human pose reconstruction is applied in various fields, such as motion capture, human-computer interaction, video surveillance, and sports broadcasting [1]. Traditionally, marker-based motion capture systems have been used to continuously track 3D human poses in fields, such as entertainment and gaming. However, marker-based systems have a limitation—their controllers must wear marker suits with attached sensors that include optical markers or mounted cameras for the system to capture their motions [2]. Further, the system is unable to capture the motions of people wearing casual clothing, and using marker suits involves additional costs.

Recently, new techniques in computer vision have enabled the use of markerless approaches in motion capture systems. The initial markerless approaches are machine learning-based strategies to convert a motion capture problem into a regression or pose classification problem. With the rapid development of convolutional neural network (CNN), several researches have proposed solutions to improve human pose estimation by utilizing a CNN. DeepPose was first proposed by Toshev et al. [3] to solve the 2D human pose estimation problem with a CNN model. Since then, several studies have proposed solutions to improve 3D pose estimation by introducing multistage convolution architectures [4]. Subsequently, various 3D pose estimation approaches from a single image have been proposed. Pavlakos et al. introduced deep CNNs based on the stacked hourglass architecture, instead of 2D pose regression, to infer 3D pose [5]. Further, Martinez et al. introduced a CNN model to lift a 2D ground-truth pose to a 3D space from a single image [6].

However, in these single-view scenarios, several occlusions issues arise. For example, self-occlusions occur when a person adopts complex poses, such as standing with one's back toward a single camera or when the locations of one's elbow and wrist are covered by the upper torso. To avoid these occlusions in a single view scenario and take advantage of the complementary information from the multiple views, multiview approaches have been proposed. Multiview approaches typically solve the single view occlusion issue using a two-stage concept. The first stage detects 2D human poses from multiple views, and the second stage aggregates the 2D human poses and lifts them to a 3D space using geometrical triangulation, reconstructing a 3D human pose [7, 8]. Although remarkable advances have been made in multiview 3D reconstruction, the following challenges still exist in practice. As the number of cameras increases, 3D pose estimation systems relying on CNN may lack adequate computational resources [9] and endure multimedia traffic overloads. When a single host system uses multiple cameras, the data transition speed becomes inadequate because of bandwidth limitations [10]. In addition, although only a few cameras are used to detect 2D joints, the resulting 3D human pose reconstruction might be inaccurate owing to the time difference between the capturing of motions by asynchronous cameras. Therefore, the projection rays from these cameras may not meet if the cameras capture a moving point at different times [10].

This paper aims to estimate 3D human pose in real time. The key idea is to use a distributed system based on edge processing, which comprises a single central server and multiple edge devices. The edge devices read RGB images from a calibrated camera and detect 2D human poses using a CNN model. The on-device AI boards that use the CNN model are considered as edge devices. Subsequently, each edge device transmits a 2D human pose with a timestamp, instead of the raw RGB image, to a central server. The data size of a 2D human pose and timestamp is much smaller than that of a raw RGB image, resulting in reduced data traffic;

furthermore, the computational load of the CNN is distributed among the edge devices. Finally, the central server receives multiple 2D human poses from the edge devices and reconstructs a 3D human pose by combining the 2D human poses using a geometrical triangulation technique, such as direct linear transform (DLT).

Additionally, the solution to the issue with asynchronous cameras is crucial to improve the accuracy of 3D pose estimation. Therefore, the central server creates a set of 2D human poses that were captured at a similar time by comparing the timestamps in the received data. This algorithm proposed for time synchronization improves the accuracy of the 3D human pose estimation.

In summary, the main contributions of this work are:

- We propose a distributed system which consists of multiple edge devices to estimate 3D human pose in real-time, while the earlier researches have focused on a single host system. The multiple edge devices can reduce the computational load of the central server and data size through the network.
- In addition, we propose an algorithm to increase the accuracy of 3D human pose estimation results by reducing data mismatch in the proposed distributed system.
- Also, we implement the prototype of the proposed system and demonstrate the proposed system successfully estimates 3D human pose in real-time in a practical environment.

2. Related Work

2.1 2D Human Pose Detection

Traditional human pose estimation was based on pictorial structure models, such as tree-structured graphical and hierarchical models [11-14]. These models were used to roughly encode the relationships among body parts. In addition, these approaches are based on manually created features and cannot achieve high performance. Lately, a deep neural network (DNN) has been applied in several areas of image processing. Compared with traditional methods, DNN-based methods are trained with a large set of images and are highly robust, resulting in a stable image-processing performance. Depending on the structure of a neural network, these methods are classified as single CNN [15, 16], multilevel CNN [7, 17], or recurrent neural network (RNN) methods [18, 19]. Each method has its advantages and helps achieve high image-processing performance. The single CNN method enables low network complexity, and the multilevel CNN method enhances the performance by cascading and synthesizing the network in various ways. The RNN method achieves better performance and overcomes the occlusion problem. However, these methods still endure the disadvantages of complex computations and high time consumption.

2.2 3D Human Pose Detection on Single Camera

With the rapid development of DNNs, several methods to estimate 3D human poses using a single camera are being studied. The simplest way to estimate 3D human poses is to design an end-to-end network to predict the 3D coordinates of the joints in each pose. There are two methods to directly map input images to 3D body joint positions: detection-based [20, 21] and regression-based methods [22, 23]. Detection-based methods predict a likelihood heatmap for each joint and determine the human pose by extracting the maximum likelihood. In contrast, regression-based methods directly predict the locations of the joints relative to the position of the root joint. Detection-based and regression-based methods endure calculation complexity

and low detection performance, respectively. Another approach to predict a 3D human pose is to lift a 2D pose to a 3D space. Several studies have attempted to improve the 3D pose estimation performance using the results of 2D pose estimation [6, 24]. Many approaches, such as depth value, geometric relationships, and additional networks, have been used to bridge the gap between 2D and 3D poses. However, owing to the lack of resources, generating a realistic 3D pose with monocular images is limited.

2.3 Multiview 3D Human Pose Detection

Multiview images can significantly reduce the ambiguity in image matching. However, the methods used to fuse information from multiple perspectives play an important role. Firstly, Pavlakos et al. combined each 2D joint heatmap obtained from multiple views using a 3D pictorial structures model [25]. The 3D pose was estimated using the average of the projected 3D location in each joint. Additional image optimization techniques, such as iterative refinement framework and feedback to the CNN model, were proposed to obtain precise joint locations [26, 27]. Secondly, methods using multiview consistency were proposed to reduce the need for annotated datasets by forcing systems to predict the same pose from all views only during training [28]. These methods were extended to realize an encoder-decoder network using unlabeled images by employing a semi-supervised CNN.

Furthermore, the triangulation method can be used for 3D human pose detection. In this method, feature maps are aggregated and processed by a 3D convolutional neural network without 3D projection [1, 29]. Further, 2D joint positions and confidence levels of multiple views are calculated using an algebraic triangulation module, yielding a 3D pose. However, because the 2D joint predictions are determined independently, noise inequality might exist in the results.

3. System Architecture

The proposed system comprises a single central server and multiple edge devices, as shown in **Fig. 1**. The proposed system shown in **Fig 1** is basically same with a well-known 3D pose estimation system with multiviews [8, 29]. The major difference from the earlier system is that the proposed system consists with multiple edge device. To reduce the computational load of the central server and data size through the network, the multiple edge device detects 2D poses simultaneously and send it to the central server. A video in standard HD resolution (1280 x 720) with 60 FPS has 3.5 Mbps bitrate. On the other hand, the bitrate of 2D pose detection results is around 64 Kbps when considering 25 keypoints of xy 2D coordinates (16 Bytes) with 20 FPS.

In this system, the edge devices are connected to an RGB camera from which the images are read. In addition, the central server and all the edge devices are connected to the same Ethernet network via an IP switch for data communication. All edge devices use the same coordinated universal time (UTC) using the network time protocol (NTP), an application protocol for clock synchronization of the host systems in Ethernet networks. Additionally, we consider the on-device AI boards, such as the Google Coral Dev board or Nvidia Jetson TX2, as edge devices to detect 2D human poses using the CNN model. The edge devices read an image, detect 2D human poses in real-time, and transmit the detected 2D poses and the timestamp based on UTC to a central server via Ethernet. Subsequently, the central server uses the received 2D human poses from multiple edge devices to reconstruct a 3D human pose using a geometrical triangulation technique, such as the DLT.

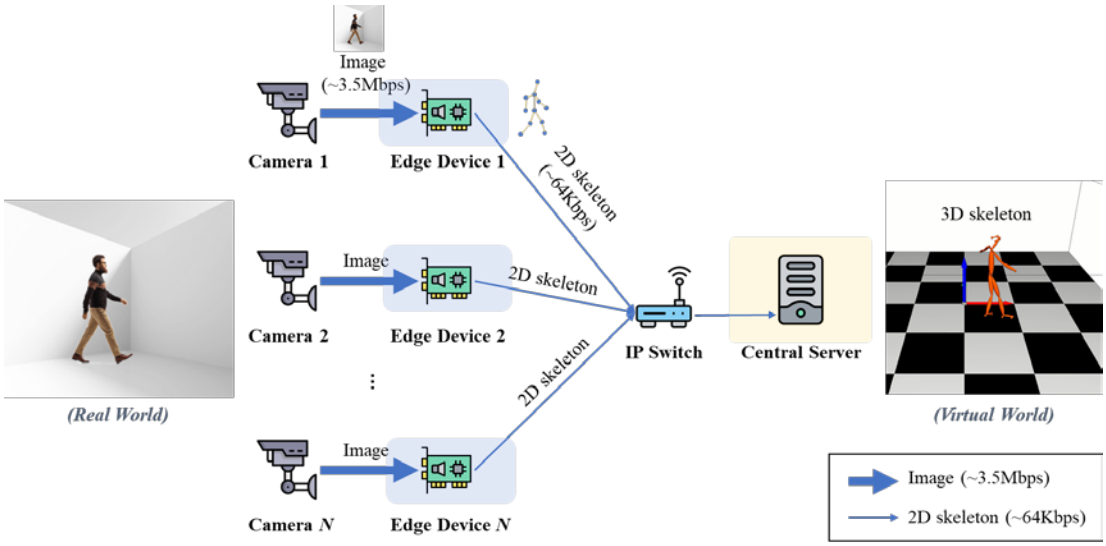


Fig. 1. System architecture for the proposed 3D human pose estimation framework.

Fig. 2 shows a block diagram of the proposed system. Each edge device independently detects the 2D human pose from an image [30] and sends it to the central server. $\mathbb{x}_n$ denotes the 2D coordinates $\{u_n, v_n\}$ with the confidence score $\{c_n\}$ from n^{th} camera. The i^{th} edge device transmits 2D coordinates $\mathbb{x}_i$ and the timestamp t_i to the central server. The central server makes a group $\{\mathbb{x}'_1, \ldots, \mathbb{x}'_m\}$of 2D coordinates that are detected at approximately the same time by comparing the timestamps. Finally, the central server reconstructs a 3D human pose using the derived 2D human pose group [29, 31]. $\mathbb{X}$ denotes the 3D world coordinates $\{x_w, y_w, z_w\}$ with confidence score $\{c_w\}$.

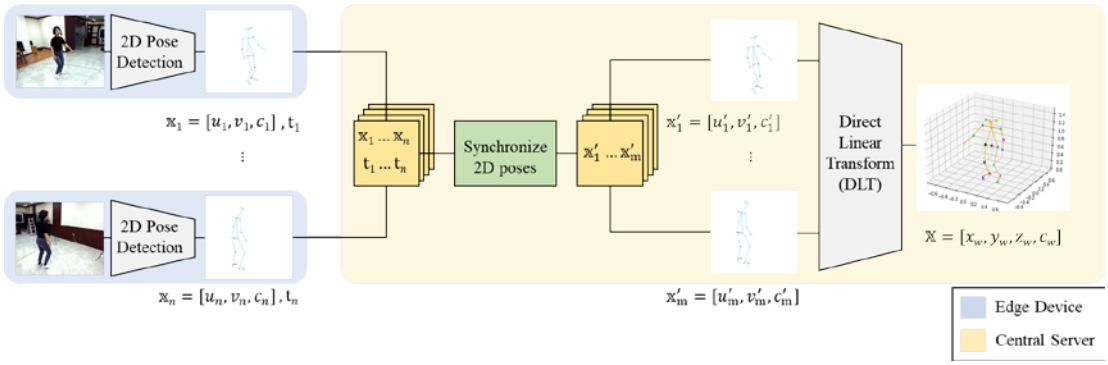


Fig. 2. Block diagram of the proposed 3D human pose estimation framework.

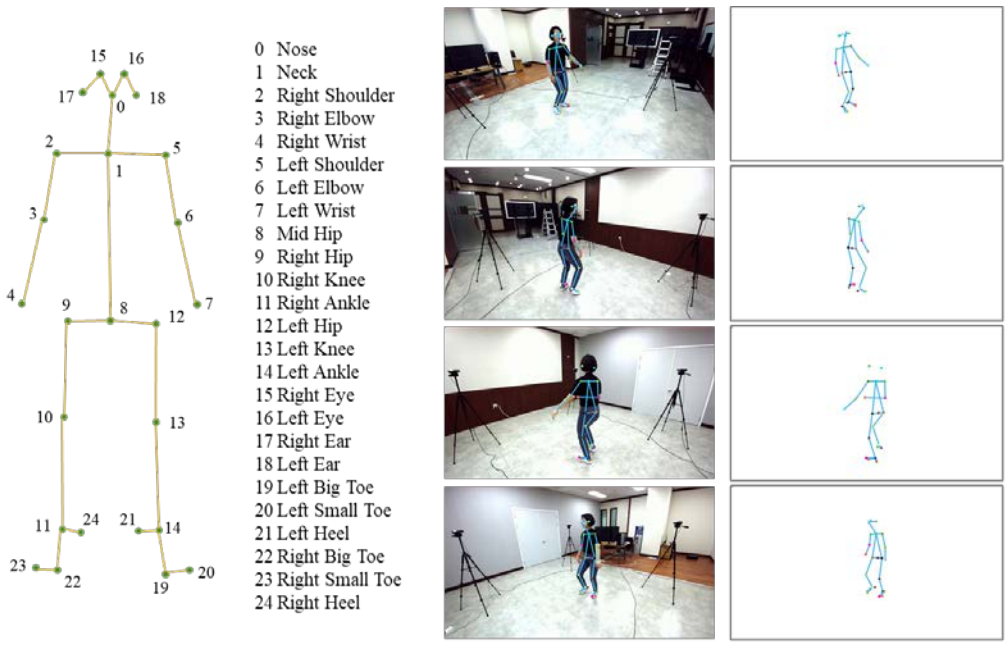
4. Proposed Framework

4.1 2D Human Pose Detection

First, the 2D human poses need to be detected before a 3D human pose is reconstructed. 2D human pose detection involves predicting the locations of individual joints of a person from an image. With the recent developments in CNN architectures, research in 2D human pose estimation has achieved significant progress in terms of performance and accuracy. We adopt

pose proposal networks to infer the 2D human poses in the edge devices, such as on-device boards [30]. Pose proposal networks detect an unknown number of 2D poses in real time.

The results of the 2D human pose detection that comprises 25 keypoints, including the nose, neck, elbows, wrists, and knees, are shown in **Fig. 3** (a). The results of the simultaneously detected 2D human poses from four edge devices are shown in **Fig. 3** (b). To detect moving objects in any viewpoint, and also avoid the occlusion problem, multiple viewpoints should be placed variously. For instance, **Fig 3** (b) shows the four corner viewpoints. As mentioned earlier, each edge device detects the 2D human pose and transmits the detected pose, instead of the raw RGB image, to a central server. For a better understanding, the result of each 2D human pose detection is overlaid on each RGB image in **Fig. 3** (b). During the 2D human pose detection, n edge devices simultaneously detect a 2D human pose. Subsequently, the i^{th} edge device detects the 2D human pose $\mathbb{x}_i$ that comprises xy 2D coordinates of the 25 keypoints, and transmits the result along with the UTC timestamp t_i to the central server. The performance of each edge device, while detecting a 2D human pose, continuously varies depending on the remaining computational resources of the device.



(a) human pose keypoint format (b) 2D human pose detection results from 4 viewpoints

Fig. 3. (a) 2D human pose keypoint format and (b) 2D human pose detection results from four viewpoints.

4.2 Time Synchronization

After 2D human pose detection, the 2D human poses need to be synchronized based on the timestamps. In the distributed system, the received 2D human poses have a time difference because of the continuous variation in the performance of the edge device. If the time difference between the detected 2D human poses is significant, the estimated 3D human pose could be noisy and inaccurate. In this study, we propose a time-synchronization algorithm to improve the accuracy of 3D human reconstruction. We consider a data frame that includes the time and 2D poses for time synchronization. **Fig. 4** shows the proposed data frame for time

synchronization. The central server divides the time of the 2D poses in the data frame into discrete intervals, Δt, called slots. The central server uses the data frame to verify whether each 2D pose was successfully received within a time slot.

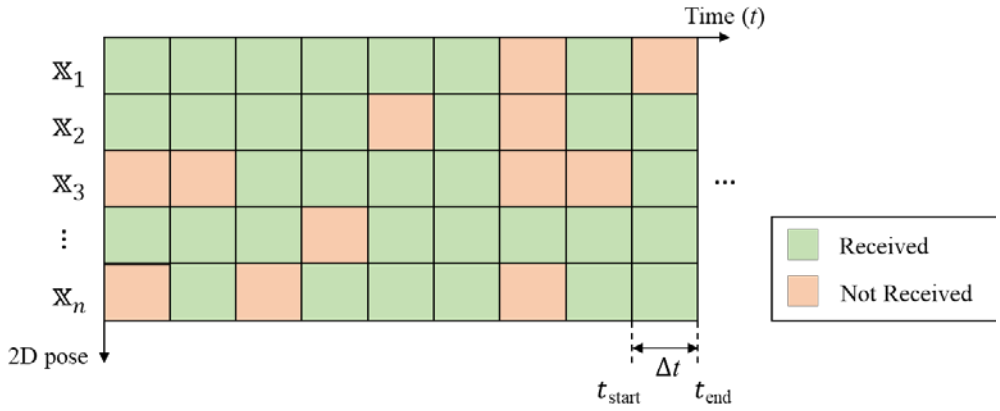


Fig. 4. Proposed time-synchronization data frame.

The time-synchronization algorithm is shown in **Algorithm 1**, where $\mathbb{x}=\{\mathbb{x}_1,\ldots,\mathbb{x}_n\}$ denotes the set of the received 2D human poses, $\boldsymbol{T}=\{t_1,\ldots,t_n\}$ denotes the set of timestamps from n cameras, and $\boldsymbol{S}=\{\mathbb{x}'_1, \ldots, \mathbb{x}'_m\}$ denotes the set of time-synchronized 2D human poses across m multiple views. The algorithm takes input values $\mathbb{x}$ and $\boldsymbol{T}$ and produces an output $\boldsymbol{S}$ for each set of inputs. The central server executes this algorithm in every time slot.

In **Algorithm 1**, the procedure is divided into three cases: $t_i \leq t_{start}$, $t_i > t_{end}$, and $t_{start} < t_i \leq t_{end}$, where t_i is the timestamp of the $\mathbb{x}_i$

- If the input timestamp is earlier than t_{start} ($t_i \leq t_{start}$), the central server considers the input 2D pose from i^{th} edge device as outdated and clears the data.
- If the timestamp is later than t_{end} ($t_i > t_{end}$), the central server considers the input 2D pose from i^{th} edge device as an earlier pose and pushes the data into the buffer $\boldsymbol{Q}$.
- If neither of the first two cases are true, then the central server considers the input 2D pose as appropriate ($t_{start} < t_i \leq t_{end}$) and appends the 2D pose $\mathbb{x}_i$ to $\boldsymbol{S}$, the set of 2D poses to be used for 3D human pose reconstruction.

Additionally, in anticipation of a poor case where the minimum number of 2D poses for 3D human pose reconstruction is not received, n_{min}, we have programmed the algorithm to check whether the number of the time-synchronized 2D human poses in set $\boldsymbol{S}$ is greater than n_{min}. If not, the central server uses the data stored earlier in the buffer $\boldsymbol{Q}$ to meet the triangulation requirement of the minimum number of poses.

Algorithm 1: Time-Synchronization Algorithm

Input: Dataset $\mathbb{x}$ and $\boldsymbol{T}$
Output: Dataset $\boldsymbol{S} = \{\mathbb{x}'_1, \ldots, \mathbb{x}'_m\}$
(1) Initialize queue $\boldsymbol{Q}$ and $\boldsymbol{S}$
(2) Insert $\forall\{\mathbb{x}_i, t_i\} \in \mathbb{x}$, $\boldsymbol{T}$ into $\boldsymbol{Q}$
(3) **while** $t_{start} < t \leq t_{end}$:
(4) **for** each {$\mathbb{x}_i$, t_i} in $\boldsymbol{Q}$:
(5) pop {$\mathbb{x}_i$, t_i} from $\boldsymbol{Q}$
(6) **if** $t_i \leq t_{start}$ **then** (*In case 2D pose $\mathbb{x}_i$ is outdated, clear the data*)
(7) continue
(8) **else if** $t_i > t_{end}$ **then** (*In case 2D pose $\mathbb{x}_i$ is early, reserve the data*)
(9) push{$\mathbb{x}_i$, t_i} into $\boldsymbol{Q}$
(10) **else** (*Otherwise, 2D pose $\mathbb{x}_i$ is appropriate, use the data*)
(11) push $\mathbb{x}_i$ into $\boldsymbol{S}$
(12) **end**
(13) **end**
(14) **for** each {$\mathbb{x}_i$, t_i} in $\boldsymbol{Q}$:
(15) pop {$\mathbb{x}_i$, t_i} from $\boldsymbol{Q}$
(16) **if** n($\boldsymbol{S}$) < $\boldsymbol{n_{min}}$ **then** (*If the number of 2D poses is insufficient, use the early data*)
(17) push $\mathbb{x}_i$ into $\boldsymbol{S}$
(18) push{$\mathbb{x}_i$, t_i} into $\boldsymbol{Q}$
(19) **else**
(20) break
(21) **end**
(22) **end**

4.3 3D Pose Reconstruction

The final stage in the development of the proposed framework is 3D pose reconstruction. A 3D pose is reconstructed using a geometrical triangulation technique with the set of 2D human poses, $S = \{\mathbb{x}'_i\}_{i=1}^{m}$, which is the output of the previous time-synchronization algorithm.

As the previous 3D pose estimation researches [29, 31], we reconstruct a 3D human pose $\mathbb{X}$ = {x_w, y_w, z_w} using the DLT method with the following procedures. Assuming this as a pinhole camera model, we can derive $\mathbb{x}'_i = kP_i\mathbb{X}$ where $\mathbb{x}'_i$ is the set of 2D keypoints of the human pose, k is an unknown scale factor, and P_i is a projection matrix of the i^{th} camera. Therefore, the cross product of the two vectors is zero ($, \mathbb{x}'_i \times P_i\mathbb{X} = 0$) because $\mathbb{x}'_i$ and $P_i\mathbb{X}$ have the same direction, except for the scale factor k. On expanding on the equations, we derive (**1**).

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} p_{i1}^{\mathrm{T}}\mathbb{X} \\ p_{i2}^{\mathrm{T}}\mathbb{X} \\ p_{i3}^{\mathrm{T}}\mathbb{X} \end{bmatrix} = \begin{bmatrix} v_i p_{i3}^{\mathrm{T}}\mathbb{X} - p_{i2}^{\mathrm{T}}\mathbb{X} \\ p_{i1}^{\mathrm{T}}\mathbb{X} - u_i p_{i3}^{\mathrm{T}}\mathbb{X} \\ u_i p_{i2}^{\mathrm{T}}\mathbb{X} - v_i p_{i1}^{\mathrm{T}}\mathbb{X} \end{bmatrix} = 0, \quad (\mathbf{1})$$

where p_{ij}^{T} denotes the j^{th} row of the projection matrix of the i^{th} camera. Further, using (**1**), we obtain linear equations (**2**) and (**3**).

$$\begin{bmatrix} v_i p_{i3}^{\mathrm{T}} - p_{i2}^{\mathrm{T}} \\ p_{i1}^{\mathrm{T}} - u_i p_{i3}^{\mathrm{T}} \end{bmatrix} \times \mathbb{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2}$$

$$A_i \mathbb{X} = 0 \tag{3}$$

By associating $A_i \mathbb{X} = 0$ with m views, we represent it as a homogeneous linear system. Therefore, the approximation of $\mathbb{X}$ is the last column of V corresponding to the smallest singular value of $U \sum V^{\mathrm{T}}$,which is the result computed using singular value decomposition (SVD) [29, 31]. The resulting 3D human pose estimation using the DLT method is shown in **Fig. 5**.

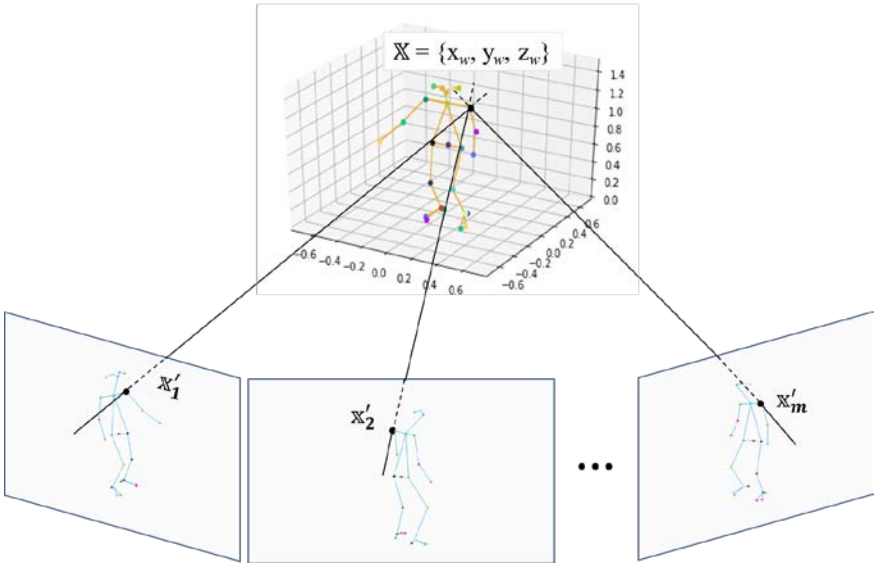


Fig. 5. 3D human pose estimation from multiple 2D human poses using the DLT method.

4.4 Weighted Moving Average

Once the 3D human pose is estimated, a weighted moving average filter can be adapted to smooth the 3D human pose and reduce noise. 3D human poses are sequentially stored into a frame buffer, and a weighted moving average filter is subsequently applied. Further, to add weight to the 3D human pose with a higher confidence score, the confidence score c_w is used as the weight of the moving average filter.

5. Experiment Results

5.1 Performance Metrics

To evaluate the performance of the proposed time-synchronization algorithm, we used the percentage of detected joints (PDJ) as an evaluation metric [3]. The PDJ is a simple and widely accepted evaluation metric for the accuracy of human pose estimation. The metric shows a

ratio of the correctly predicted keypoint to the ground-truth keypoint. As the accuracy of the prediction increases, the PDJ approaches 100%. A joint is considered as detected when the distance between the predicted and the ground-truth keypoints is within a certain fraction of the torso diameter. The PDJ metric is calculated as follows:

$$PDJ^{\alpha}\ (\%) = \frac{\sum_{i=1}^{k} B(d_i < \alpha D)}{k} \times 100, \tag{4}$$

where d_i is the Euclidean distance between the i^{th} predicted and ground-truth keypoints, D is the Euclidean distance of the 3D bounding box of the human body from the edge devices, and α is a certain distance threshold to verify the accuracy of the estimation. In addition, k is the number of keypoints on the human body, and B is a boolean function that returns one if the condition is true and zero if it is false.

5.2 Simulation Results

We evaluated the performanceof the proposed framework, in terms of the PDJ, for 3D human pose estimation. The following dataset was used to compare the detected keypoints with the ground-truth keypoints [32]. It was a dataset consisting of 2D human poses from 23 calibrated cameras of a person moving in an indoor environment. The dataset had 18,400 frames, with each calibrated camera capturing 800 frames. We assumed that the dataset was time-synchronized because it was extracted from videos recorded at the same time. In this simulation environment, multiple asynchronous threads read the 2D human pose data captured by different cameras and transmitted the data at time intervals with a random time difference σ, *rand*(-σ, σ). Subsequently, the main thread collected the 2D human poses and estimated 3D human poses. The values of the simulation parameters were considered as follows. The total number of the RGB camera and edge devices was 8 (n = 8), the minimum number of 2D keypoints for DLT was 3 (n_{min} = 3), and the time interval of the time-synchronization data frame was 20 ms (Δt = 20 ms). In addition, the edge devices were assumed to transmit 2D keypoints at intervals of 20 ms and with a random σ.

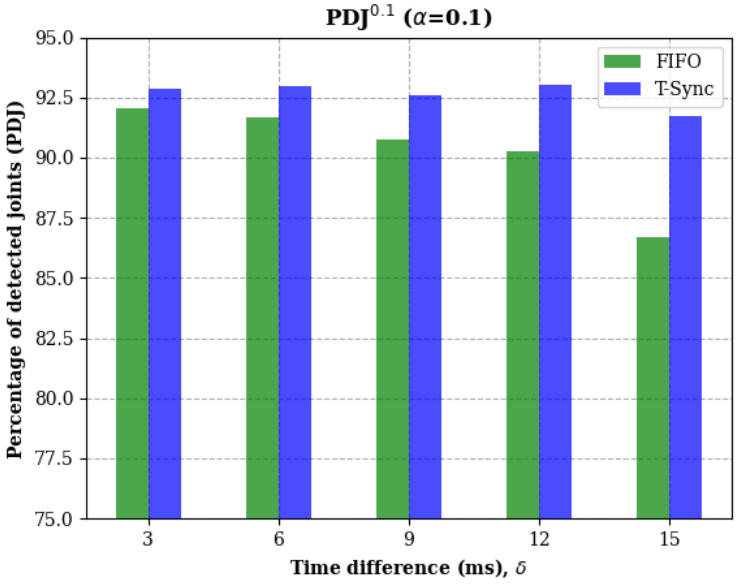


Fig. 6. PDJ with a distance threshold of 0.1.

Fig. 6 shows the PDJ with a distance threshold of 0.1 based on σ. As the value of σ increased, the PDJ decreased because of the reduction in accuracy of 3D human pose estimation, and the mismatched 2D human poses increased in the time-synchronization algorithm. To evaluate the performance improvement of using the time-synchronization algorithm, we compared the proposed algorithm with the typical first-in, first-out (FIFO) algorithm that handles data in a sequence without any time synchronization. When σ was 15 ms, the $PDJ^{0.1}$ derived using the time-synchronization was approximately 5% higher than that derived using the FIFO algorithm. The proposed time-synchronization algorithm (T-Sync) exhibited a higher PDJ than the FIFO algorithm because the proposed algorithm selects only the 2D human poses in the same time domain to increase the estimation accuracy.

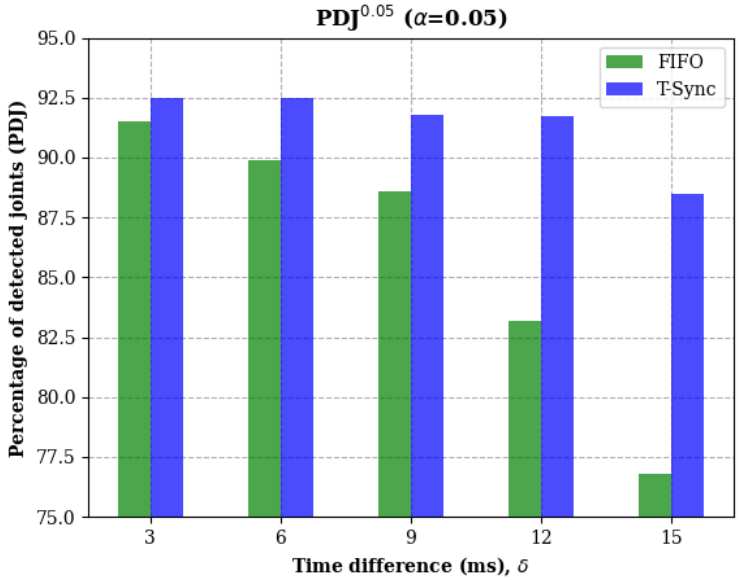


Fig. 7. PDJ with a distance threshold of 0.05.

Fig. 7 shows the PDJ with a distance threshold of 0.05. The value of the $PDJ^{0.05}$ was lesser than $PDJ^{0.1}$ because the criterion to evaluate the accuracy of the estimation became much stringent as the distance threshold decreased. In the proposed framework, when σ was 6 ms and 15 ms, the $PDJ^{0.05}$ was approximately 2.5% and 11.7% higher, respectively, compared with that of the FIFO algorithm. The numerical results of the simulation are summarized in **Table 1**.

Table 1. The PDJ with distance thresholds 0.1 and 0.05

Time difference σ (ms)	$PDJ^{0.1}$ (%)		$PDJ^{0.05}$ (%)	
	FIFO	T-Sync	FIFO	T-Sync
3	92.0	92.8	91.5	92.4
6	91.7	92.9	89.9	92.4
9	90.7	92.6	88.5	91.7
12	90.2	93.0	83.2	91.7
15	86.7	91.7	76.8	88.5

The performance improvement using the proposed T-Sync algorithm was validated by the 3D pose estimation results of FIFO and T-Sync, as shown in **Fig. 8**. The first row shows the result where σ is 0 ms, with no time difference between the edge devices. When we increased σ to 15 ms, FIFO exhibited an inaccurate estimation of the 3D pose because the 2D human poses were mismatched. As shown in red circles in **Fig. 8** (a), some of the joint locations were poorly reconstructed compared with the reconstructed joint locations (**Table 1**) when σ was 0 ms. In contrast, even after we increased σ to 15 ms, the proposed T-Sync algorithm exhibited a more accurate estimation of the 3D pose as shown in **Fig. 8** (b).

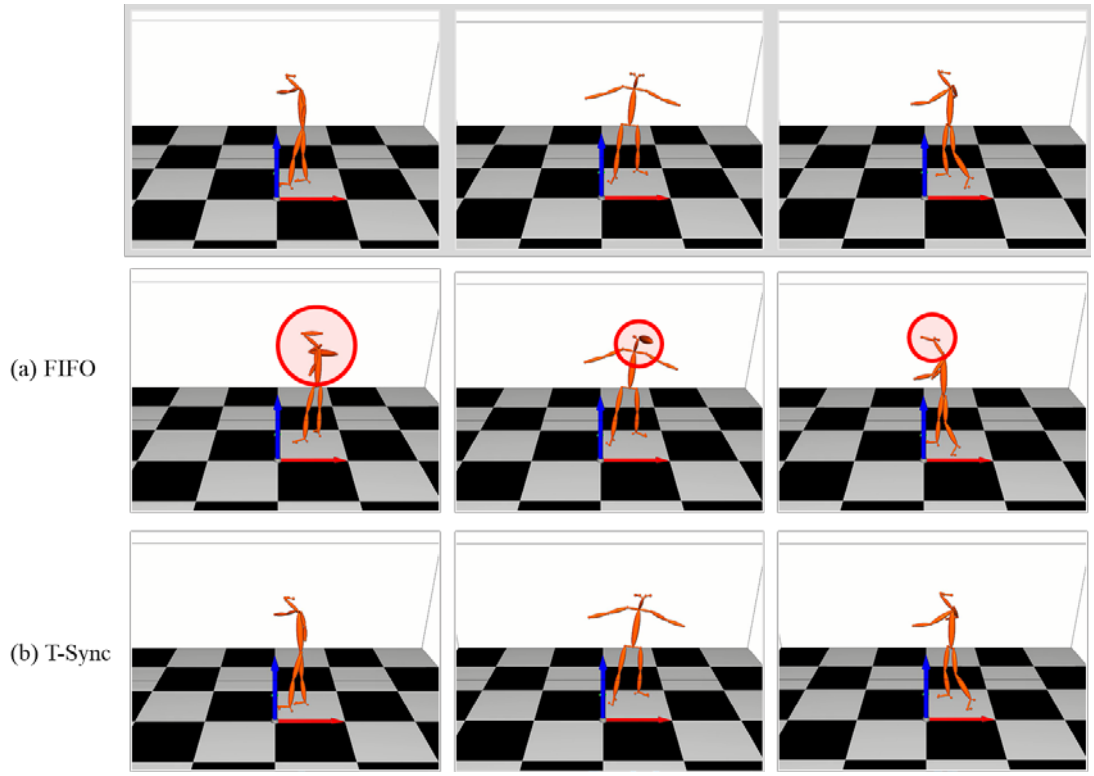


Fig. 8. 3D pose estimation results of FIFO and T-Sync with σ as 15 ms.

5.3 Prototyping Results

In addition to deriving the simulation result, we implemented a prototype to demonstrate that the proposed framework could successfully estimate 3D human pose in real time. We implemented and installed the prototype that comprises a single central server and four edge devices in an empty space, as shown in **Fig. 9**. We used Sterolabs Zed 2i RGB camera and Nvidia Jetson TX2 as the edge device to read an RGB image and detect a 2D human pose, respectively. In addition, we used an ASUS laptop with Intel i9 and RTX3080 GPU as the central server. Further, all the devices were connected to an iptime A3008 IP switch for data communication. **Table 2** summarizes the hardware specifications of the implemented prototype.

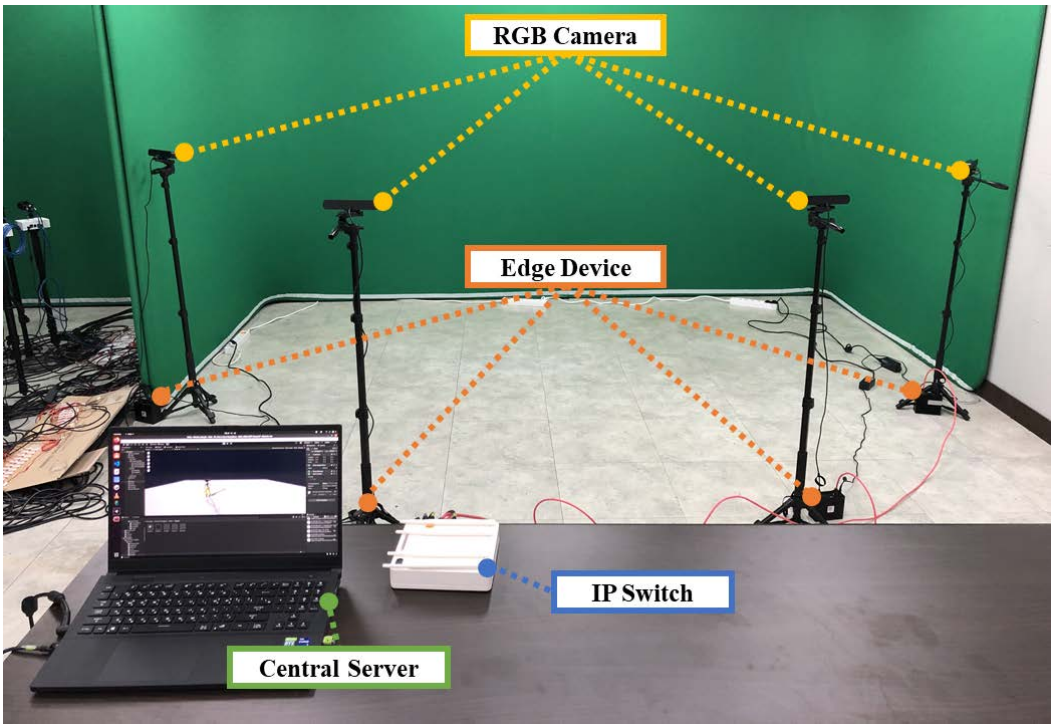


Fig. 9. Prototype that comprises a single central server and 4 edge devices.

Table 2. Hardware specification of the implemented prototype

Name	Specification	Number of machines
RGB Camera	Stereolabs Zed 2i (720p/60FPS)	4
Edge device	Jetson TX2, NVIDIA Denver 2 CPU, Pascal GPU, 8GB RAM	4
IP Switch	IPtime A3008 2Gbps	1
Central Server	Intel i9 2.5GHz CPU, RTX3080 GPU, 32GB RAM	1

Similar to the simulation environment, we set Δt as 20 ms, and $n_{\min}$ as 3. When we executed the prototype, each edge device detected a 2D human pose with the images captured atapproximately 15–20 frames per second (FPS). The FPS of each edge device continuously varied depending on the remaining computational resources of the device. For the quantitative evaluation, we verified that the prototype captured a static 20 FPS when we set Δt as 20 ms. For the qualitative performance evaluation, we used a real-time 3D visualizer to analyze the estimated 3D human pose. The 3D visualizer showed the results of 2D human detection and 3D human reconstruction in real time. The prototyping results obtained by the real-time 3D visualizer are shown in **Fig. 10**. The first row (a) shows a person with different poses. The second row (b) shows the results of 2D pose detection from the edge devices, and the third row (c) shows the results of 3D pose estimation as 2D projections of the estimated 3D skeletons representing the joints and bone locations. The last row (d) shows a 3D character duplicating the different poses of the person based on the estimated 3D poses. To ensure that the 3D character had the same pose as the person, we mapped each result of the respective 3D pose estimation to a 3D character model. Using the real-time 3D visualizer, we verified that the proposed framework could estimate 3D human poses in real time across multiple asynchronous views.

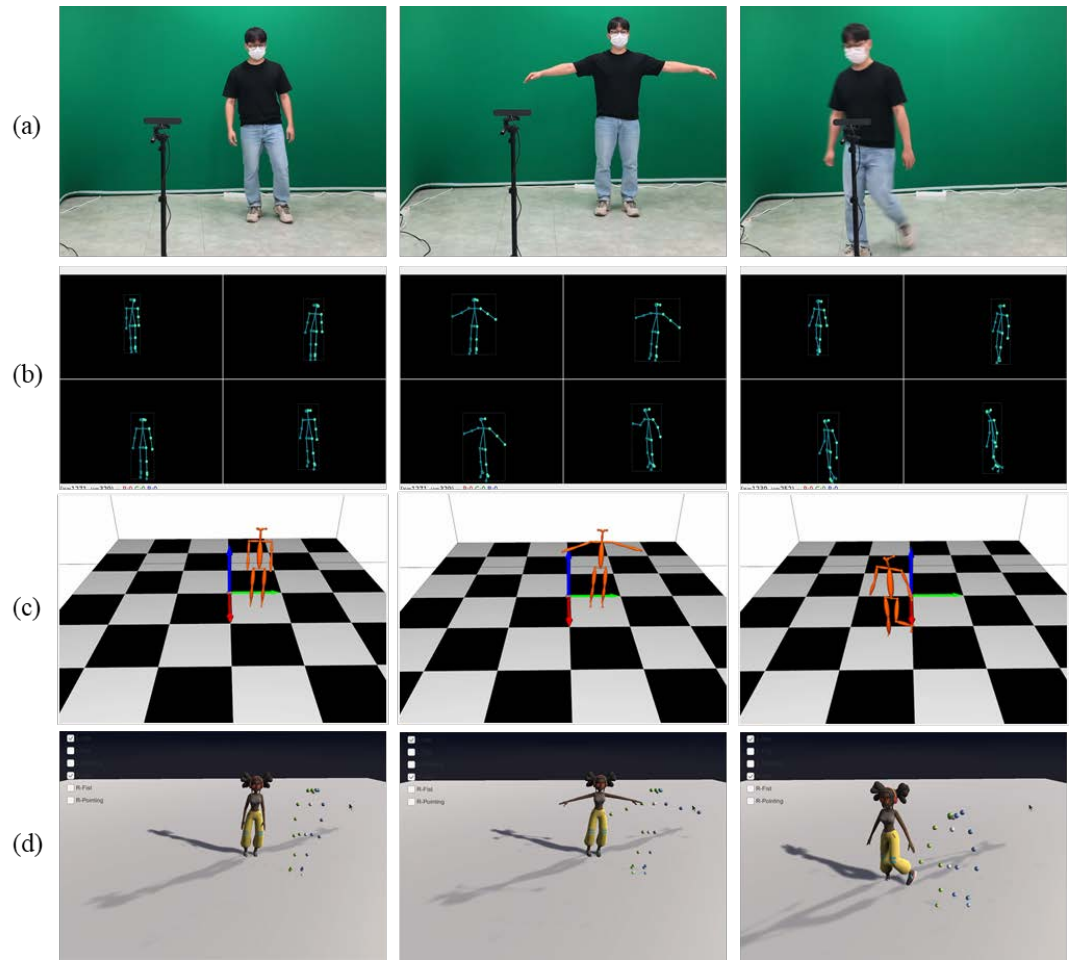


Fig. 10. Prototyping results obtained by the real-time 3D visualizer.

6. Conclusion

In this study, we propose a distributed framework for real-time 3D pose estimation based on asynchronous multiviews. In the earlier multi-view 3D human pose estimation system which has a single host server, the system may lack in computational resources, and its multimedia traffic becomes overloaded. To overcome this challenge, we propose a distributed framework that comprises a single central server and multiple edge devices. The edge devices detect 2D human poses and transmit them, instead of RGB images, with timestamps to a central server. In addition, to solve the issue of inaccuracy owing to asynchronous cameras in the distributed system, we propose a time-synchronization algorithm in which the central server creates a set of keypoints of the 2D poses detected at approximately the same time. The simulation results demonstrated that, when σ was 6 ms and 15 ms, the $PDJ^{0.05}$ of the proposed framework was approximately 2.5% and 11.7% higher, respectively, compared with that of the FIFO algorithm. In addition, we implemented and installed a prototype that comprises a single central server and four edge devices. The qualitative performance evaluation verified that the proposed framework could estimate a 3D human pose in real time using multiple asynchronous views.

References

[1] J. Dong, Q. Fang, W. Jiang, Y. Yang, Q. Huang, H. Bao, and X. Zhou, "Fast and Robust Multi-Person 3D Pose Estimation and Tracking from Multiple Views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6981-6992, Oct., 2022. Article (CrossRef Link)

[2] M. Sandau, H. Koblauch, T. B. Moeslund, H. Aanæs, T. Alkjær, and E. B. Simonsen, "Markerless motion capture can provide reliable 3D gait kinematics in the sagittal and frontal plane," *Elsevier Medical Engineering & Physics*, vol. 36, no. 9, pp. 1168-1175, Sep., 2014. Article (CrossRef Link)

[3] A. Toshev, and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp.1653-1660, 2014. Article (CrossRef Link)

[4] Y. Nie, J. Lee, S. Yoon, and D. S. Park, "A Multi-Stage Convolution Machine with Scaling and Dilation for Human Pose Estimation," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 6, pp. 3182-3198, Jun. 2019. Article (CrossRef Link)

[5] G. Pavlakos, X. Zhou, K. G. Derpanis and K. Daniilidis, "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, pp. 1-10, 2017. Article (CrossRef Link)

[6] J. Martinez, R. Hossain, J. Romero and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, pp. 2640-2649, 2017. Article (CrossRef Link)

[7] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D Pictorial Structures for Multiple Human Pose Estimation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 1669-1676, 2014. Article (CrossRef Link)

[8] A. Elmi, D. Mazzini, and P. Tortella, "Light3DPose: Real-time Multi-Person 3D Pose Estimation from Multiple Views," in *Proc. of IEEE International Conference on Pattern Recognition (ICPR)*, Milan, Italy, pp. 2755-2762, 2020. Article (CrossRef Link)

[9] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu, "Cross-View Tracking for Multi-Human 3D Pose Estimation at Over 100 FPS," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 3276-3285, 2020. Article (CrossRef Link)

[10] C. Albl, Z. Kukelova, A. Fitzgibbon, J. Heller, M. Smid, and T. Pajdla, "On the Two-View Geometry of Unsynchronized Cameras," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, pp. 5593-5602, 2017. Article (CrossRef Link)

[11] Y. Yang, and D. Ramanan, "Articulated Human Detection with Flexible Mixtures of Parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878-2890, 2013. Article (CrossRef Link)

[12] M. Kiefel, and P. V. Gehler, "Human Pose Estimation with Fields of Parts," in *Proc. of Springer European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, pp. 331-346, 2014. Article (CrossRef Link)

[13] L. Fu, J. Zhang, and K. Huang, "Beyond Tree Structure Models: A New Occlusion Aware Graphical Model for Human Pose Estimation," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1976-1984, 2015. Article (CrossRef Link)

[14] M. Dantone, J. Gall, C. Leistner, L. V. Gool, "Body Parts Dependent Joint Regressors for Human Pose Estimation in Still Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2131-2143, Nov., 2014. Article (CrossRef Link)

[15] J. Tompson, A. Jain, Y. Lecun, C. Bregler, "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation," in *Proc. of International Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, pp. 1799-1807, 2014. Article (CrossRef Link)

[16] X. Chu, W. Ouyang, H. Li, X. Wang, "Structured Feature Learning for Pose Estimation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, pp. 4715-4723, 2016. Article (CrossRef Link)

[17] K. Luu, T. D. Bui, C. Y. Suen, K. Ricanek, "Combining local appearance and holistic view: Dual-Source Deep Neural Networks for human pose estimation," in *Proc. of IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore, pp. 1976-1984, 2010. Article (CrossRef Link)
[18] G. Gkioxari, A. Toshev, and N. Jaitly, "Chained Predictions Using Convolutional Neural Networks," in *Proc. of Springer European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, pp. 728-743, 2018. Article (CrossRef Link)
[19] V. Belagiannis, and A. Zisserman, "Recurrent Human Pose Estimation," in *Proc. of IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, Washinton, DC, USA, pp. 468-475, 2017. Article (CrossRef Link)
[20] G. Pavlakos, N. Kolotouros, K. Daniilidis, "TexturePose: Supervising Human Mesh Estimation with Texture Consistency," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 803–812, 2019. Article (CrossRef Link)
[21] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 5137–5146, 2018. Article (CrossRef Link)
[22] S. Li, and A. B. Chan, "3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network," in *Proc. of Springer Asian Conference on Computer Vision (ACCV)*, Singapre, pp. 332–347, 2015. Article (CrossRef Link)
[23] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu, "HEMlets pose: Learning Part-Centric Heatmap Triplets for Accurate 3D Human Pose Estimation," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 2344–2353, 2019. Article (CrossRef Link)
[24] S. Park, M. Ji, and J. Chun, "2D Human Pose Estimation based on Object Detection using RGB-D information," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 800-816, 2018. Article (CrossRef Link)
[25] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 6988–6997, 2017. Article (CrossRef Link)
[26] D. Tome, M. Toso, L. Agapito, and C. Russell, "Rethinking Pose in 3D: Multi-stage Refinement and Recovery for Markerless Motion Capture," in *Proc. of IEEE International Conference on 3D Vision (3DV)*, Verona, Italy, pp. 474–483, 2018. Article (CrossRef Link)
[27] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross View Fusion for 3D Human Pose Estimation," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, pp. 4342–4351, 2019. Article (CrossRef Link)
[28] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning Monocular 3D Human Pose Estimation from Multi-view Images," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 8437–8446, 2018. Article (CrossRef Link)
[29] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang, "Lightweight Multi-View 3D Pose Estimation Through Camera-Disentangled Representation," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 6039-6048, 2020. Article (CrossRef Link)
[30] T. Sekii, "Pose Proposal Networks," in *Proc. of Springer European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 350-366, 2018. Article (CrossRef Link)
[31] R. Hartley, and A. Zisserman, "Two-View Geometry," in *Multiple View Geometry in Computer Vision*, Cambridge, England: Cambridge University Press, pp. 237-238, 2004. Article (CrossRef Link)
[32] EasyMoCap (2021) [Online]. Available: https://github.com/zju3dv/EasyMocap

Taemin Hwang received the B.S. and the M.S. degrees in electronic engineering from Sogang University, Seoul, Korea, in 2015 and 2017, respectively. From 2017 to 2021, he was a Senior Engineer at LG Electronics, Seoul, Korea, where he involved in the design and development of Adaptive AUTOSAR. Since 2021, he has been with Korea Electronics Technology Institute, Seongnam, Korea, where he is currently a senior researcher. His current research interests include computer vision, machine learning, and human pose estimation.

Jieun Kim received the B.S. degree in electrical engineering from Kyungpook National University, Daegu, South Korea, in 2012 and Ph.D degree in electronic engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2020. Since 2021, she has been with Korea Electronics Technology Institute (KETI) as a senior researcher. Her research interests include signal processing and hardware architecture of radar & lidar system and vision technoloy for 3D reconstruction.

Minjoon Kim received the B.S. and Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2012 and 2018, respectively. He is currently a senior researcher in the Korea Electronics Technology Institute, Korea. His research interests include digital signal processing algorithm and SoC/VLSI implementation.