Top-down Pose Estimation Method based Human-Computer Interaction for Smart Space System with Digital Twin

Kwanho Kim, Junmyeong Kim and Kanghyun Jo
Dept. of Electrical, Electronic and Computer Engineering
School of Electrical Engineering
University of Ulsan, Ulsan, Korea
kjm7029@islab.ulsan.ac.kr, aarony12@naver.com, acejo@ulsan.ac.kr

***Abstract*—Human-Computer Interface (HCI), a technology for human interaction with computers, has been studied a lot for a long time. As technologies related to the metaverse have recently developed, digital twin technology is also used in various industries, and in the field of computer vision, various deep learning-based algorithms such as object classification, object detection, and pose estimation have been developed. In this paper, Using a deep learning-based top-down pose estimation algorithm, keypoints are extracted from three images and matched in a 3D virtual environment to create a digital twin. The coordinated digital twin delivers information to digital devices in the real environment through actions that cannot be simulated in the real environment, such as shooting lasers in a virtual space. Customized actions such as opening doors and turning off lights can be performed through IoT sensors and actuators in real environments. The experiment was performed using Unity, and the results showed 82.67% accuracy on average.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

With the advancement of communication technology and computing power, recent years have seen a surge in research and development of smart home [1] and smart space systems based on the Internet of Things (IoT). Numerous companies and researchers ayre actively engaged in studying these areas. When implementing a smart home, the Human-Computer Interaction (HCI) technology for the interaction between humans and computing devices is primarily implemented through specific(physical) devices such as smartphone, keyboard, and remote controller. However, these technologies suffer from a lack of convenience(intuitiveness) as people have to physically control the devices.

Because of the recent development of the technology about metaverse, there has been a significant increase in research focused on using AR glasses to recognize human hands and display interfaces in virtual spaces [2], thereby implementing HCI. This approach involves directly using the user's hands to interact with the computer without the need for controlling a separate device. However, it is not suitable as a method for HCI because it requires the user to wear AR glasses on their head specifically for the purpose of HCI. Above all, it has not yet been popularized among the general public due to issues such as high prices, display angles, resolution, battery life, and weight. This paper proposes a method to address these issues by utilizing multiple cameras installed in the physical space, without requiring any wearable devices for the user. It suggests an approach for interacting with computers using these cameras. In recent years, there has been significant development of high-performance object detection [3] and pose estimation [4] algorithms. By employing these methods, it is possible to extract the positions of human key-points from multiple camera images and utilize traditional techniques such as camera geometry to create a digital twin in a virtual space that mimics the pose of a person. Digital twins enable many actions that are difficult to implement in the physical world because they exist in a virtual space.

II. Background

A. Pose Estimation

Pose estimation is one of the computer vision tasks that refers to the task of extracting key point information, such as joints, of a person, animal, or object in an image or video. When implementing pose estimation using deep learning, there are two main approaches: the top-down method and the bottom-up method. Both methods consist of two steps.

The top-down method first utilizes an object detection model to estimate the area in the image or video where the object is present. It then performs pose estimation within that region. While this method provides high accuracy, it has the drawback of being slower when multiple people are present in the image, as pose estimation needs to be performed for each cropped area.

On the other hand, the bottom-up method extracts all key points present in the image or video and then groups them into individual poses by associating the corresponding joints. This approach performs pose estimation directly without using a separate object detection model, which makes it faster. However, its accuracy is relatively lower compared to the top-down method. This work uses the top-down approach to estimate the precise human pose because of higher accuracy.

Identify applicable funding agency here. If none, delete this.

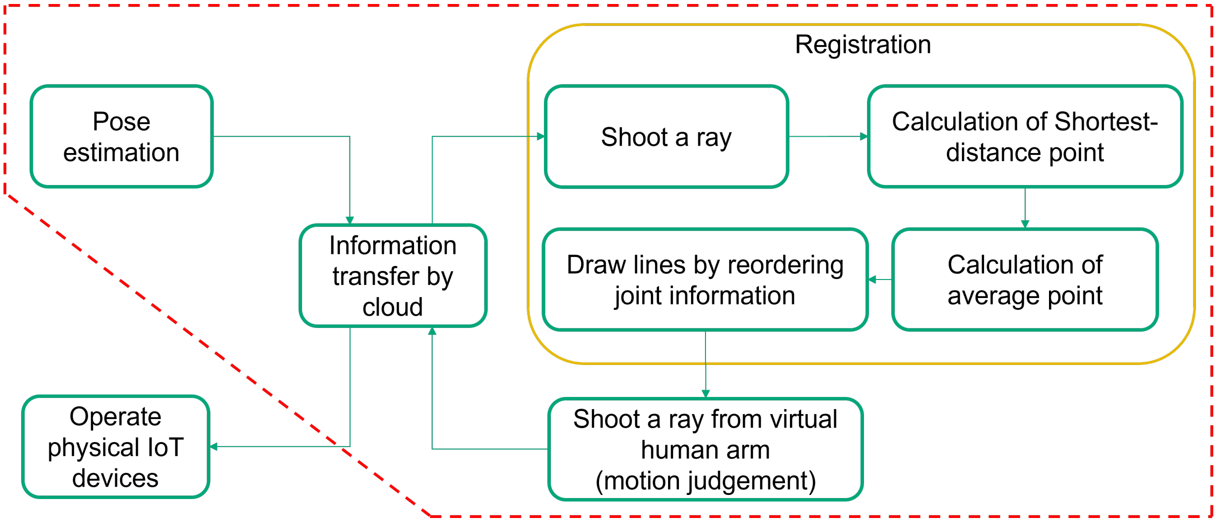
Registration
Pose estimation
Shoot a ray
Calculation of Shortest-distance point
Information transfer by cloud
Draw lines by reordering joint information
Calculation of average point
Operate physical IoT devices
Shoot a ray from virtual human arm (motion judgement)

Fig. 1. Overall processes of HCI system. Green boxes and lines indicate specific processes of proposed system. Yellow box represents the registration process which is one of the main contributions. The red dashed line means the processes implemented in this paper.

B. Object Tracking

Object tracking refers to the task of continuously tracking the motion of a specific object in a video or sequence of images. There are various methods to implement object tracking, but recently, with the emergence of exceptional object detection models, detection-based tracking approaches have been predominantly studied. These methods combine object detection and tracking by detecting the target object in each frame and performing tracking based on the detected bounding boxes. This paper employs the widely used Simple Online and Real-time Tracking (SORT) [5] and YOLO as the object detection models for the top-down approach. By utilizing these detection-based tracking methods, we achieved accurate and reliable object tracking throughout the video or image sequence

C. Camera Geometry

This work utilizes the relationship between image coordinates and world coordinates, as well as camera parameters, to reconstruct a three-dimensional object based on a camera. The image coordinate system refers to the coordinates of an image when an object is captured by the camera. The image coordinate system is represented on a two-dimensional plane. On the other hand, the world coordinate system is an absolute coordinate system that exists in the real world regardless of the camera. The origin of the image coordinate system is determined by factors such as the focal length of the camera, sensor size, and resolution. Figure 1 illustrates the relationship between the image coordinate system and the world coordinate system.

III. Proposed Methods

Before the processes depicted in Figure 1, this paper first establishes a virtual space that is similar to real space. The corresponding space replicates not only the actual shape of stationary objects such as tables, refrigerators, and desktops but also the parameters of the camera for implementing HCI. The task of constructing a virtual world that replicates the physical environment can be achieved through Simultaneously Localization And Mapping (SLAM) technology, which utilizes multiple devices. [6] Moreover, it is also possible to construct a virtual world by directly measuring the size of indoor spaces and objects and using 3D modeling programs.

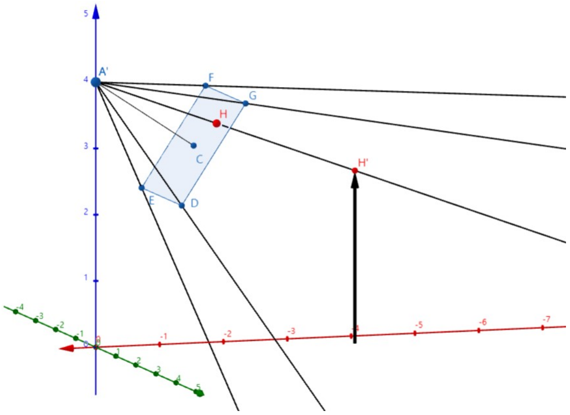

Fig. 2. The concept of Camrea geometry

A. Key-points Extraction

To perform 3D alignment, it is necessary to have videos captured from two or more cameras. In order to determine which points in each video correspond to the same location, classical key-points or feature extraction methods such as [7] can be used. However, these methods are primarily designed for use with videos captured from multiple adjacent cameras, which makes them unsuitable for the HCI system used in this paper. In this paper, a pose estimation algorithm is used to detect the human body's posture and the positions of its

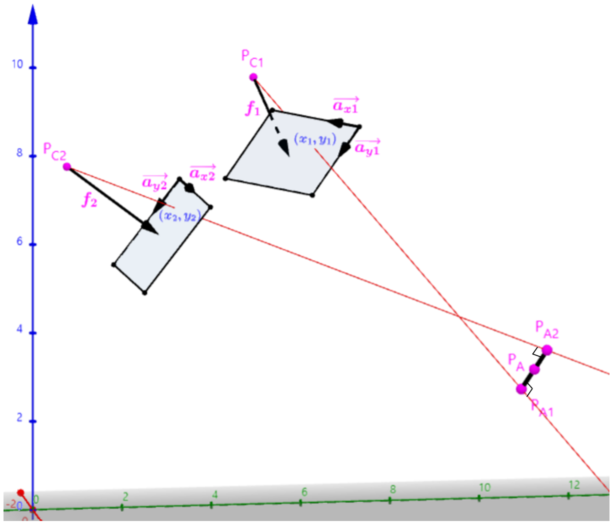
10
8
6
4
2
P_{C1}
$\overrightarrow{a_{x1}}$
f_1
(x_1, y_1)
$\overrightarrow{a_{y1}}$
P_{C2}
$\overrightarrow{a_{y2}}$
$\overrightarrow{a_{x2}}$
f_2
(x_2, y_2)
P_{A2}
P_A
P_{A1}
-20
0
2
4
6
8
10
12

Fig. 3. Simple motion capture coordinate system. Blue variables (x_1, y_1, x_2, y_2) are on the image coordinate while purple vectors (P_A,P_{A1},P_{A2},P_{C1}, P_{C2},$\overrightarrow{a_x1}$, $\overrightarrow{a_y1}$, $\overrightarrow{a_x2}$, $\overrightarrow{a_y2}$,f_1,f_2) are on the world coordinate. XYZ-axis of the world coordinate system are represented by red, green, blue axes. Red rays are emitted from the cameras P_C1,P_C2 in the direction of points (x_1,y_1), (x_2,y_2) in each image.

joints. By detecting a total of 33 joint positions such as the head, shoulders, and waist through pose estimation, it becomes possible to determine which points correspond to the same location in any video captured from any camera, using the joint information. This joint information represents the positional information in the 2D image coordinate system. The transformation to the 3D coordinate system is implemented in the virtual world.

B. Information Transfer

There are various methods for transmitting joint information from each image to a virtual world. If pose estimation and the implementation of the virtual world are performed on a single computing device, information transfer is straightforward, but each process requires a significant amount of computation. When using two computing devices, communication between the two computers is necessary. As mentioned again in the experimental section, this paper implemented information exchange between the two computers using cloud services.

C. Registration

Once the extracted joint information from each camera's captured images is received in the virtual world, the registration process converts the 2D joint position information into 3D coordinates. The position of each joint is represented as a single point within the image coordinates. That point is then projected onto a point in 3D space. By projecting all the points corresponding to the extracted joints into 3D space and reassembling them according to the joints, a digital twin is created in the same position and pose as the real counterpart.

Figure 2 demonstrates the method of projecting the points extracted from two images onto 3D space. The blue variables represent pixel-level coordinates of the points within the images, so they are first converted into meters, the unit of the world coordinate system, using Equation (1).

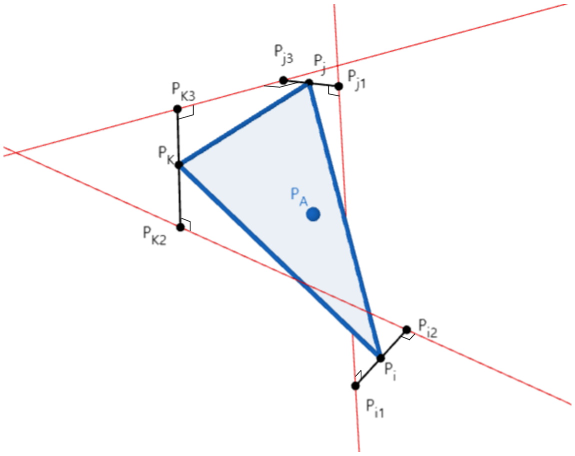

Fig. 4. Point determination with three cameras. Three rays are emitted by cameras. P_A represent the determined point by weighted mean. Subscripts (i, j, k) indicate the points about closest distance between two rays.

$$x_{im}, y_{im} = \frac{pixel}{resolution} * SensorSize \tag{1}$$

The coordinates of Equation (1) are in the same units as the world coordinate system, but they are still in the image coordinate system. To draw a red ray, Equ ation (2) is used to transform the coordinates (x_{im}, y_{im}) to be based on the world coordinate system

$$P_{im} = P_C + f + (x_{im} - \frac{w}{2})\overrightarrow{a_x} + (y_{im} - \frac{h}{2})\overrightarrow{a_y} \tag{2}$$

In Figure 2, the purple vectors are all based on the world coordinate system, so Equation (2) holds true. P_{im} represents a 3-dimensional vector obtained by transforming the coordinates of Equation 1 to be based on the world coordinate system. P_C represents the position of the camera, f represents the focal length vector, and w and h represent the width and height of the sensor, respectively. $\overrightarrow{a_x}$ and $\overrightarrow{a_y}$ represent a directional vector corresponding to the XY axes of the image coordinate system based on the world coordinate system. The focal length vector contains information about the direction the camera is facing. By using the coordinates of a point in the image based on the world coordinate system, a red ray can be projected in Figure 2.

By projecting rays from two or more cameras, the two points with the minimum distance between the rays can be calculated. Assuming the two points are P_{A1} and P_{A2}, Equations (3) and (4) can be used to represent the equation of the line, and as shown in Figure 2, the rays and the line $(P_{A1} - P_{A2})$ are perpendicular. Therefore, a simple quadratic equation can be formulated with variables such as shown in Equations (5) and (6). By solving the equations, the two points can be calculated, and the midpoint of the two points is designated as P_A.

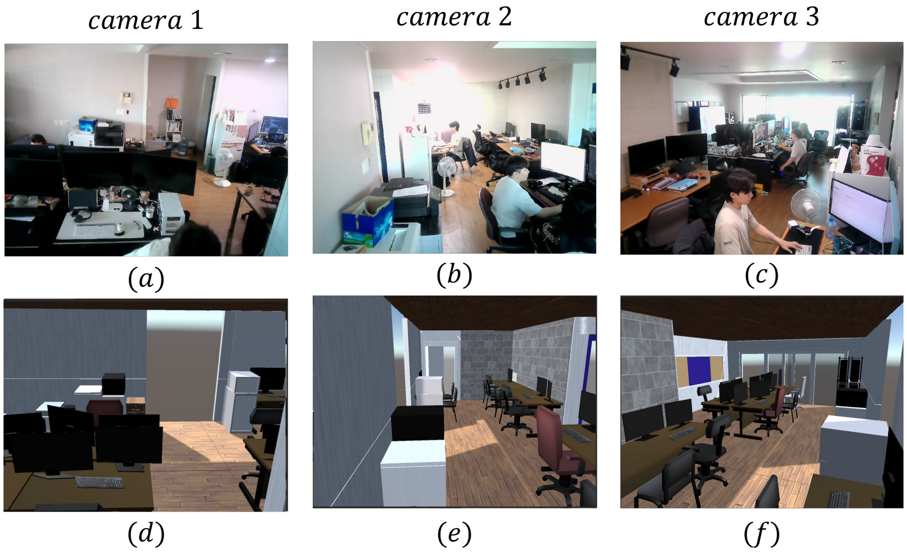

Fig. 5. Experiment environment with three cameras. (a), (b), (c) represent captured images from the cameras of real environment while (d), (e), (f) for virtual environment. (a) and (d) were captured from camera 1, (b) and (e) from camera 2, and (c) and (f) from camera 3.

$$P_{A1} = P_{C1} + t_1(P_{im1} - P_{C1}) \tag{3}$$

$$P_{A2} = P_{C2} + t_1(P_{im2} - P_{C2}) \tag{4}$$

$$(P_{A1} - P_{A2})(P_{im1} - P_{C1}) = 0 \tag{5}$$

$$(P_{A1} - P_{A2})(P_{im2} - P_{C2}) = 0 \tag{6}$$

If there are only two cameras, the average point obtained above can be transformed into a 3-dimensional position. However, when there are three or more cameras, more than three rays are projected from each camera. In this case, weights are assigned to calculate the 3-dimensional position. Figure 3 illustrates the method for calculating the 3-dimensional coordinates when there are three cameras. In this paper, the distance between two lines is used as a weight for the determined point, and the final 3-dimensional position is determined through weighted averaging. For example, when there are three cameras, the point P_A is calculated using Equation (7).

$$P_A = \frac{|P_{i1} - P_{i2}|P_i + |P_{j1} - P_{j2}|P_j + |P_{k2} - P_{k3}|P_k}{|P_{i1} - P_{i2}| + |P_{j1} - P_{j3}| + |P_{k2} - P_{k3}|} \tag{7}$$

By converting all extracted joints from the 2-dimensional image to the 3-dimensional world coordinate system and reassembling them based on the joint information, a skeletal structure can be created by drawing lines in the virtual world. This skeletal structure represents a digital twin with a pose similar to that of a real person.

D. Interaction

In this paper, to enable a specific object to respond when a person points at it, the coordinates and angles of the skeleton are used to shoot virtual lasers. Depending on which object the laser hits, various actions can be performed, such as turning on/off a monitor, turning on/off a light, or opening/closing a door. Additionally, by estimating the person's pose, customized IoT services can be provided.

IV. Experiments

A. Environment

The experiment was conducted in an environment similar to the one depicted in Figure 5. A virtual environment resembling a real laboratory was created using Unity, and three cameras with different perspectives were placed. Figure 6 illustrates the structure of the laboratory, and the camera parameters are provided in Table 1.

TABLE I
Camera parameters

Set	Focal length (mm)	Sensor size x (mm)	Sensor size y (mm)
1	9.73981	13.3	13
2	10.25152	13.3	10
3	12.04143	13.3	10

B. Pose Estimation

In this paper, [8] was used for the pose estimation. [8] is a top-down pose estimation algorithm based on YOLOv7 [9]. Figure 7 shows the extracted 17 keypoints from each camera using The x and y coordinates, along with the confidence of the extracted keypoints, are transmitted to the cloud through Firebase [10].

C. Reconstruction

To represent a digital twin in a 3D virtual environment, x and y coordinates along with confidence values for each joint are obtained from Firebase. After performing 3D registration based on the x and y coordinates, a Kalman filter [11] and a

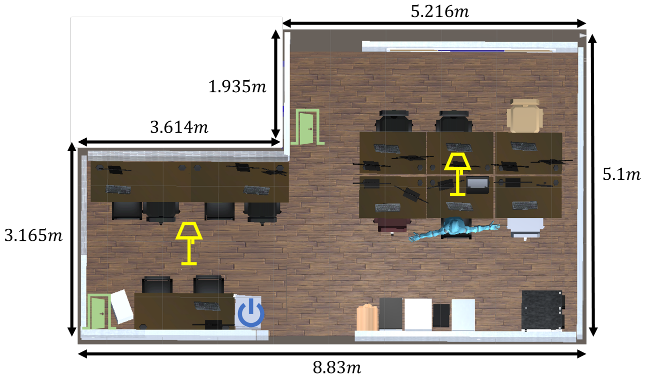

Fig. 6. Top view of laboratory. The IoT device used in the virtual environment is set up as a refrigerator door, an entrance door, lights.

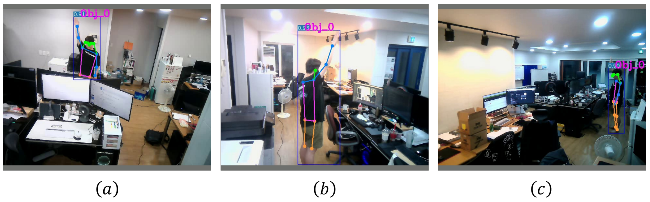
(a) (b) (c)

Fig. 7. Pose estimation from (a) camera 1, (b) camera 2, (c) camera 3.

low-pass filter were utilized to compensate for errors caused by the performance of pose estimation. Figure 8 showcases a digital twin generated at the same viewpoint as Figure 7, demonstrating the effectiveness of the aforementioned techniques.

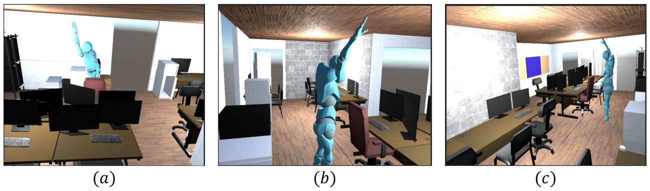
(a) (b) (c)

Fig. 8. (a) Reconstructed skeleton and (b) character by registering each joints to virtual environment at the same time as in Fig. 7.

D. Experiment result of interaction

The devices controlled through HCI in the experiment include a refrigerator, an entrance door, and others. When the laser emitted by the digital twin hits the refrigerator, it opens, and when it hits again, it closes. The entrance door is designed in a similar method, where it opens upon the laser impact and closes when hit again. Additionally, lasers hitting the ceiling of each room turn on the lights, and hitting them again turns them off. The lasers should only be emitted when the user desires, and should not be emitted at other times. In this paper, the laser is emitted when the angle of the user's arm exceeded 170 degrees. Table 2 presents the accuracy for each interaction measured when using all three cameras and two cameras. A total of 15 runs are executed, and a successful operation within 3 seconds is considered a success. When using two cameras instead of three, the average accuracy of interaction decreased by 21.33 %. Based on this result, it was confirmed that significant results can be obtained when performing 3D reconstruction and implementing HCI using three cameras.

TABLE II
ACCURACY OF INTERACTIONS (%)

	Door1	Door2	Light1	Light2	Refrigerator
Two camera	46.67	60	66.67	46.67	86.67
Three camera	80	66.67	86.67	80	100

V. CONCLUSION

This paper suggests how to implement HCI using human keypoints information from multiple cameras. Using Unity 3D program, a realistic environment was created, and experiments were conducted on five interactive scenarios that can be applied in daily life. The experimental results were compared between using two cameras and using three cameras. When using three cameras, an average accuracy of 82.67 % was achieved, demonstrating the validity of the proposed method.

ACKNOWLEDGMENT

This result is supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

REFERENCES

[1] L. Y. Rock, F. P. Tajudeen, and Y. W. Chung, "Usage and impact of the internet-of-things-based smart home technology: a quality-of-life perspective," *Universal Access in the Information Society*, pp. 1–20, 2022.

[2] M. Kim, S. H. Choi, K.-B. Park, and J. Y. Lee, "User interactions for augmented reality smart glasses: A comparative evaluation of visual contexts and interaction gestures," *Applied Sciences*, vol. 9, no. 15, p. 3171, 2019.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[6] T. Laidlow, J. Czarnowski, and S. Leutenegger, "Deepfusion: Real-time dense 3d reconstruction for monocular slam using single-view depth and gradient predictions," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4068–4074.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] M. R. Munawar. yolov7-pose-estimation. [Online]. Available: https://github.com/RizwanMunawar/yolov7-pose-estimation

[9] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[10] L. Moroney and L. Moroney, "The firebase realtime database," *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*, pp. 51–71, 2017.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.