2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA) | 979-8-3503-2845-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/IMETA59369.2023.10294844

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

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

***Abstract*—Human-Computer Interaction (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. 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 IoT 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 experiments are performed using Unity, and the results showed 82.67% accuracy on average.**

***Index Terms*—Human-Computer Interaction, Computer vision, Digital twin**

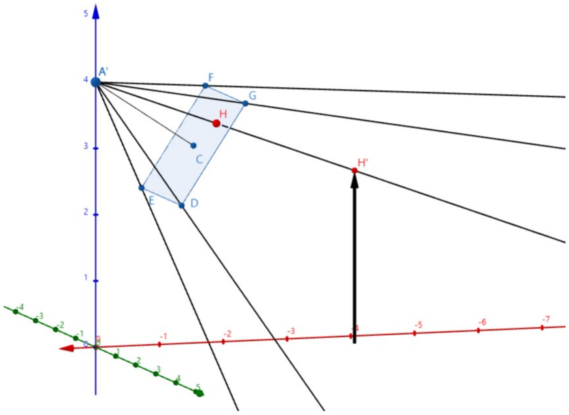


Fig. 1. The concept of camera geometry. H is the projection of point H' in world coordinate system onto image coordinate system. A' stands for the position of the camera and the square $DEFG$ represents the image coordinate system.

I. Introduction

With the advancement of communication technology and computing power, recent years have witnessed a surge in the research and development of smart home systems [1] and smart space systems based on the Internet of Things (IoT). Numerous companies and researchers are actively engaged in studying these areas. When implementing smart home, Human-Computer Interaction (HCI) is primarily utilized to facilitate interaction between humans and computing devices. This interaction is commonly achieved through various devices, including smartphones, keyboards, and remote controllers. However, these technologies suffer from a lack of convenience as they require physical control of the devices, which can be cumbersome for users.

Due to the recent advancements in metaverse technology, there has been a notable rise in research concerning the utilization of AR glasses to detect and display human hands and interfaces within virtual environments [2], enabling the implementation of HCI. This approach allows for direct interaction with the computer using the user's hands, eliminating the need for a separate controlling device. However, it is not considered a suitable method for HCI as it necessitates wearing AR glasses solely for HCI. Above all, the widespread adoption of this technology among the general public has been limited due to various issues, including high prices, limited display angles, lower resolution, short battery life, and heavy weight.To address these challenges, this paper proposes a novel method that eliminates the need for wearable devices by leveraging multiple cameras installed in the physical space.

In recent years, significant advancements have been made in high-performance object detection algorithms [3] and pose estimation techniques [4]. By employing these cutting-edge methods, it becomes feasible to extract the precise positions of key human keypoints from multiple camera images. Furthermore, traditional techniques such as camera geometry can be utilized to generate a digital representation, commonly known as a digital twin, within a virtual space. This digital twin accurately mimics the pose and movements of a person, providing a seamless and immersive interaction experience.

II. Background

A. Camera Geometry

This work utilizes the relationship between image coordinates and real-world coordinates, along with camera pa-

979-8-3503-2845-5/23/$31.00 ©2023 IEEE

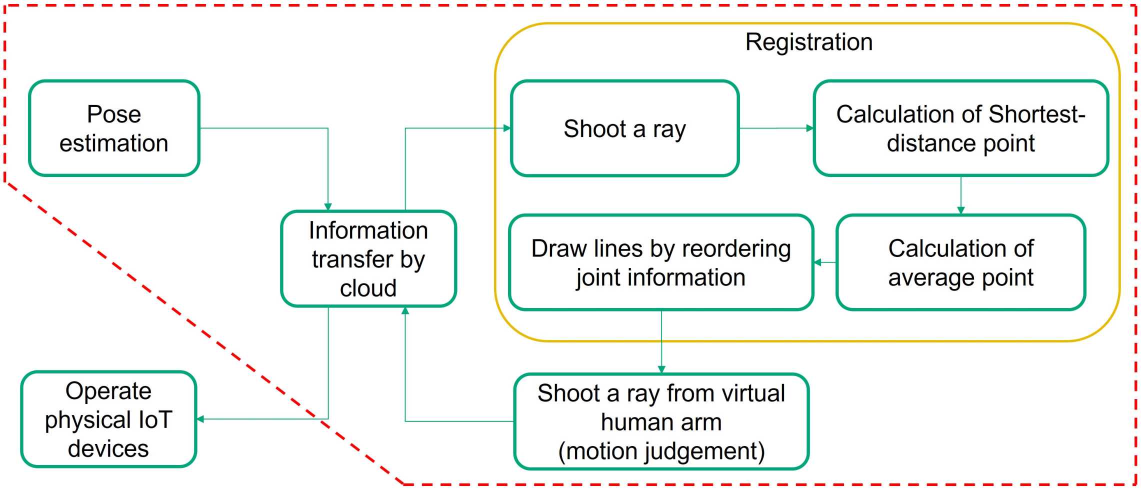

Registration
Pose estimation
Information transfer by cloud
Shoot a ray
Calculation of Shortest-distance point
Calculation of average point
Draw lines by reordering joint information
Shoot a ray from virtual human arm (motion judgement)
Operate physical IoT devices

Fig. 2. 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.

rameters, to reconstruct a three-dimensional object based on camera input. The image coordinate system refers to the coordinates of an object in the captured image, represented on a two-dimensional plane. On the other hand, the world coordinate system is an absolute coordinate system existing in the real world, independent of the camera. The origin of the image coordinate system is determined by various factors, including the camera's focal length, sensor size, and resolution. Fig. 1 illustrates the relationship between the image coordinate system and the world coordinate system.

B. Pose Estimation

Pose estimation is a computer vision task that involves 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 initially employs an object detection model to estimate the area in the image or video where the target object is located. Subsequently, pose estimation is performed within that localized 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 the keypoints present in the image or video and subsequently groups them into individual poses by associating the corresponding joints. This approach performs pose estimation directly without relying on a separate object detection model, which makes it faster in terms of computational efficiency. However, the accuracy of the bottom-up method is relatively lower compared to the top-down approach.

In this work, the top-down approach is employed to estimate the precise human pose due to its higher accuracy.

C. Human-Computer Interaction

Human-Computer Interaction (HCI) is a field dedicated to studying the interaction between humans and computers. Its primary objective is to enhance the user-friendliness and usefulness of computers, thereby improving convenience and utility for individuals. While HCI research has traditionally focused on hardware devices such as keyboards and smartphones, these devices can sometimes present an inconvenience for users. To overcome these challenges, wearable devices have been introduced; however, they still pose various obstacles, including limited battery life, high costs, and added weight. This paper seeks to address these aforementioned issues by proposing the implementation of HCI through the integration of computer vision and virtual reality technologies. By leveraging these advanced technologies, we aim to create a more seamless and immersive user experience, enhancing the efficiency and effectiveness of human-computer interactions.

III. Proposed Methods

Before the processes depicted in Fig. 2, this paper initially establishes a virtual space that closely resembles the real physical space. This virtual space aims to replicate not only the actual shape and layout of stationary objects, such as tables, refrigerators, and desktops, but also the camera parameters necessary 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. [5] 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.

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,

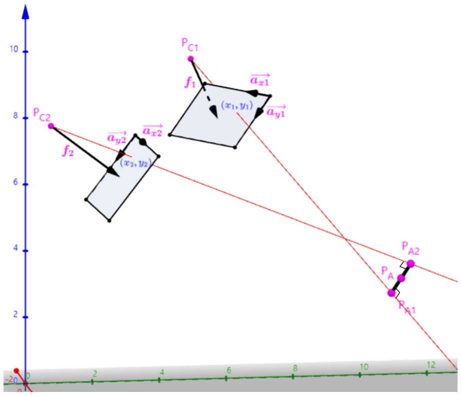

10
8
6
4
2
P_{C1}
P_{C2}
f_1
f_2
$\overrightarrow{a_{x1}}$
$\overrightarrow{a_{y1}}$
$\overrightarrow{a_{x2}}$
$\overrightarrow{a_{y2}}$
(x_1, y_1)
(x_2, y_2)
P_{A2}
P_A
P_{A1}
-20
0
2
4
6
8
10
12

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.

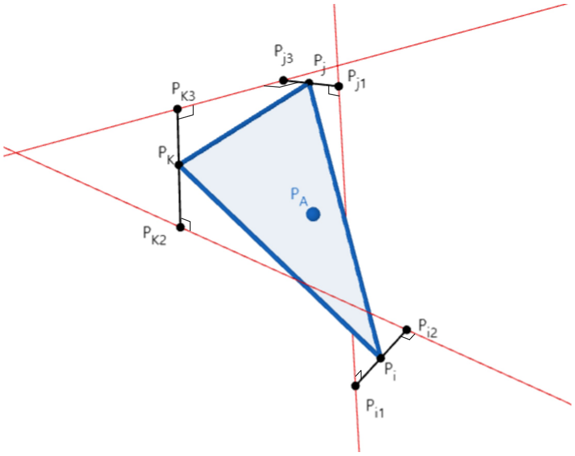


Fig. 4. Point determination with three cameras. Three rays are emitted by cameras. P_A represent the determined point by weighted mean. Subscripts (i, j, k) indicate the points about closest distance between two rays.

classical key-points or feature extraction methods such as [6] 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 joints. By estimating 17 joints through pose estimation, it becomes possible to determine which points correspond to the same location in any video captured from any cameras. 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 position in 2D image coordinate into the 3D. The location 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.

Fig. 3 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 Eq. 1.

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

The coordinates of Eq. (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, Eq. (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 Fig. 3, the purple vectors are all based on the world coordinate system, so Eq. (2) holds true. P_{im} represents a 3-dimensional vector obtained by transforming the coordinates of Eq. (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, as illustrated in Fig. 3.

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}, Eq. (3) and (4) can be used to represent the equation of the line, and as shown in Fig. 3, 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 Eq. (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.

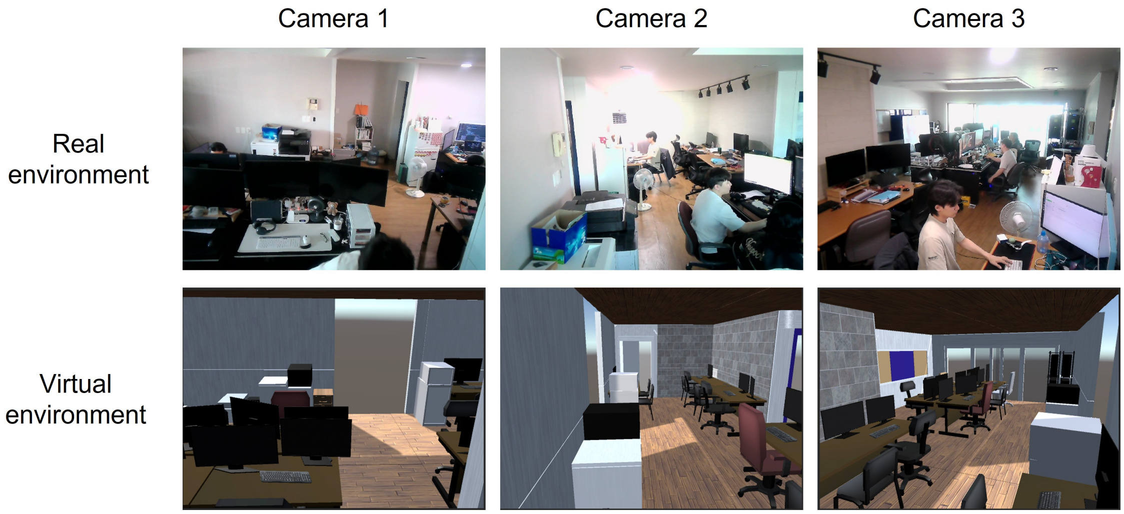


Fig. 5. Experiment environment with three cameras. The first raw shows the real environment from each camera and the second raw shows the virtual environment.

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

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

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

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

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. The Fig. 4 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 Eq. (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 experiments are conducted in an environment similar to the real laboratory. The Fig. 5 is shown the scene from each camera about real world and virtual world. A virtual environment is created using Unity, and three cameras with different perspectives are placed. The Fig. 6 illustrates the structure of the laboratory, and the camera parameters are provided in Table I.

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

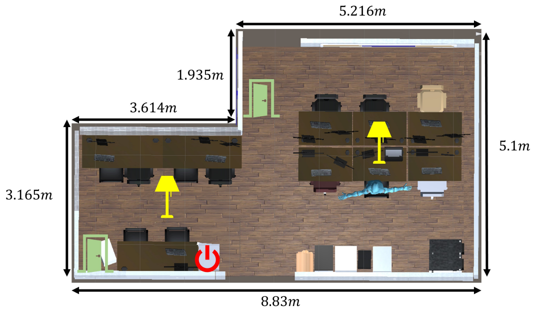
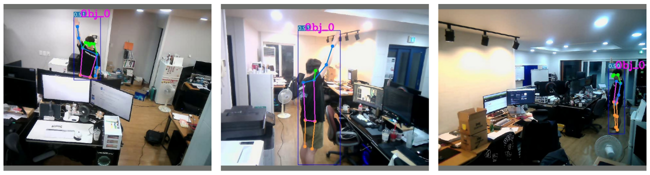


Fig. 6. Top view of laboratory. The IoT device used in the virtual environment is set up as a refrigerator door, an entrance door, lights.

B. Pose Estimation

In this paper, [7] is used for the pose estimation. [7] is a top-down pose estimation algorithm based on YOLOv7 [8]. Fig. 7 shows the extracted 17 key-points from each camera



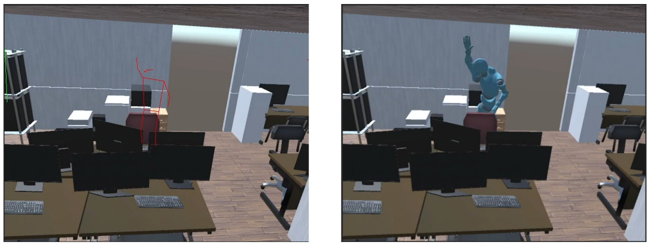
(a) Camera 1 (b) Camera 2 (c) Camera 3

Fig. 7. Pose estimation from each camera

using The x and y coordinates, along with the confidence of the extracted key-points, are transmitted to the cloud through Firebase [9].

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 [10] and a low-pass filter are utilized to compensate for errors caused by the performance of pose estimation. The Fig. 8 shows a digital twin generated at the same viewpoint as the Fig. 7, demonstrating the effectiveness of the aforementioned techniques.



(a) (b)

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

D. Experiment result of interaction

The furniture and electronic devices controlled through HCI in the experiments 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 II presents the accuracy for each interaction measured when using all three cameras and two cameras. All experiments are conducted 15 times for each number of cameras and each interactions, and it is considered success if they operated normally within 3 seconds. When using two cameras instead of three, the average accuracy of interaction decreased by 21.33%. Based on this result, it is 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 cameras	46.67	60	66.67	46.67	86.67
Three cameras	80	66.67	86.67	80	100

V. CONCLUSION

This paper suggests how to implement HCI using human key-points information from multiple cameras. Using Unity 3D program, a realistic environment is created, and experiments are conducted on five interactive scenarios that can be applied in daily life. The experimental results are compared between using two cameras and using three cameras. When using three cameras, an average accuracy of 82.67achieved, it is 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] 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.

[6] 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.

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

[8] 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.

[9] 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.

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