Intuitive User Control of a Remotely Steered Camera Mount System

Glen Merritt, Yashui Zuo, and Xuefeng Wang

Department of Mechanical Engineering
University of Alabama
Tuscaloosa, Alabama 35401, USA
xwang201@eng.ua.edu

***Abstract* - Common user control interfaces are exemplified by translations of user interaction into an unnatural medium, such as joystick controllers or vehicle controls. These devices require experience to use properly, limiting user interaction and carryover to different control devices. In this work we propose a control device based on intuitive user motion interaction, where this motion is replicated on a robotic system instead of an intermediate device control. The device control consists of a common camera phone, where the motion of the user is replicated on a 2 degree-of-freedom camera mount system. To achieve this concept, we begin by building and testing the accuracy of two different types of prediction algorithm, one based on images taken by the camera phone, and one using the sensors contained in the phone itself. The accuracy of the angular rotational estimation is evaluated on two different experiments, one using slower rotation, more suited to the image-based approach, and one using a faster rotation speed, where a sensor-based approach is more suitable. The results of each experiment are then evaluated, indicating that a combined approach for prediction is suitable for a range of rotational velocities.**

I. INTRODUCTION

Intuitive human robot interaction, where human subjects use natural modes of speech or gesture, remains a relatively unexplored area of robotic control. Traditional industrial robotic systems rely on constrained settings and precise closed loop programming to accomplish work tasks, even with recent advancements [1, 2, 3]. Another familiar form of robot are those that operate in the domestic domain [4, 5, 6], such as robotic cleaners. Typically, these robots can only be passively programmed to avoid stationary obstacles or stop working whenever they encounter uncertain environments, and thus cannot actively interact with or be controlled by humans in any meaningful way. In a broader sense both of these robots belong to the closed loop autonomous category, where they are assigned some task which cannot be dynamically redefined or solve new issues in their task space.

More recent advancements in robotic control and interaction rely on input from human users to determine the system heuristics. In contrast, traditional user system controls are simplified versions of human input, such as buttons or joysticks, which are limited to the non-intuitive actions that correspond to these inputs. The main issue with such systems is the required training and adjustment to control designs, as well as the lack of carryover from system to system, or game to game [7]. Intuitive control instead depends on natural motion that allows the human to operate in ways that they would understand relatively quickly with a little training or testing with the system in order to be able to utilize properly. This is of great interest in fields such as virtual and augmented reality, where intuitive control can correlate directly to realistic action. Such systems utilize natural gestures as opposed to buttons or joysticks or other forms of simplified control to make the experience of the user feel natural. Gesture-based human robot interaction is a developing field [8] of great interest to roboticists and human robot teams. In the future, robotic systems and users must be able to collaborate intuitively and simply in the evolving workplace [9]. Recently many different types of available new gesture-based technologies, inspired by the augmented or virtual reality systems [10, 11], where capturing natural user movement is fundamental to the system. These types of user control systems often appear as sensor harnesses where different sensors are attached to the user and the different readings are recorded and calculated to determine user intent.

Accelerometer gyroscope combinations such as those utilized in [12, 13] are the most common type of sensory systems that utilize human motion to determine user intent. By sensing the acceleration and rotational velocity experienced by each sensor, the underlying system mechanics and motion can be determined. Very precise accelerometers and gyroscopes can determine these readings with a very high degree of accuracy. Because such sensor outputs are based on actual mechanics, the motion can often be more explicitly modelled when compared to other methods. One drawback to this kind of system is it has no reference to the environment, only relative motion, and drift can play a large part in the error of the motion estimation.

When used with other inputs, accelerometer gyroscope combinations can render accurate renderings of motion and intent, especially when measuring muscular signals either mechanically or electrically. Electromyographic (EMG) sensors are another type of common application in human robotic interaction [14, 15]. EMG sensors measure the actual voltage output for muscle groups where they are attached and these measurements can be translated into input signals for user control, often using machine learning methods to estimate the system mechanics. Due to differences in user body composition, motion tendencies, and muscular utilization, it can be difficult to get precise or consistent outputs from these types of sensors, and often they are utilized in tandem with other types, after the effort of calibrating them to a particular user [16, 17].

In robotic control, image-based techniques allow systems to make use of the environment in the surrounding feature information, similar to how human vision operates, identifying objects or specific points in those scenes and using those to track objects or surroundings. Many computational image processing techniques rely on simpler versions of these, such as reducing tracking to specific keypoints, and sometimes processing these keypoints in applications such as facial recognition [18]. In more recent modern systems deep learning approaches are utilized instead [19]. The disadvantage to this type of deep learning approach is that excessive processing can severely hinder real time application. Despite this flaw, these programs have the capability of tracking entire objects and separating them from the surrounding environment. Simple keypoint detection, by contrast, allows fast computation and nearly real time application, and these keypoints can be tracked from frame to frame.

The system introduced in this work attempts to create a system where a camera mounted on a remote vehicle moved by two servo motors, and the motion of these motors replicates the natural motion of the human operator, who can in the camera phone view remotely see what the mounted camera sees. Instead of tracking an object through keypoints with the camera view, these same keypoints can instead be used to estimate the motion of the camera itself. These keypoints in this system can be tracked from frame to frame and used to estimate the angular change between subsequent views.

The simplified version of this system introduced here consists of three main components: a camera phone which is used to capture the motion of the user, a computer which receives the signals from the phone and estimates the motion, and a 2 degree-of-freedom (DoF) mount operated by servo motors which replicates the angular rotation of the user with the camera phone. In summary, the remotely mounted camera should move as the user with the camera phone does, creating an intuitive, remote video stream system.

The robotic system makes use of multiple different sensing types and once combined they shall operate in a manner that reduces overall error in the motion estimation and robust estimation in multiple different movement and environment types, such as rotation speed or poor lighting or keypoint existence. For example, image processing works well with slower rotational movement, where the image resolution can render a high degree of accuracy with keypoint tracking. In other situations, such as faster rotation and poor lighting, sensors such as accelerometers and gyroscopes can render a better estimation of movement. Generally, image data is accurate enough to predict angular rotation in isolation, and thus this aspect would be weighted more heavily in a combined system if its sampling rate is fast enough. It is robust against noise by selecting carefully the keypoint detection scheme, and yields directly the angular difference, so does not encounter the same drift issues.

This work is divided into the following sections: I. Introduction to the system goals and underlying concepts, II. Design of the system and Methods utilized, III. The experimental setup, procedure, and results of the experiment, and finally IV. Concluding remarks.

II. SYSTEM DESIGN AND METHODS UTILIZED

2.1 Overview of the System

The main function of the design is to serve as a type of remote point of view system, what this means physically is that the motion of the user with the camera phone is re-created on the camera mount system which is controlled by two servo motors along two different axes, a vertical axis and the lateral yaw axis. The entire system consists of three different components, the first of which, the camera phone, serves as the interaction point between the user and the system. The computer potion of the system serves to process the incoming data and to send the commands to the camera mount portion. The camera mount component, which is attached to a control board, serves as the actualization element, which reproduces the user motion using the two servo motors. A simple visualization diagram for the system can be viewed in Fig. 1. The phone utilized in this system is the LGG 810 Q device which operates on an android platform. The computer is an ASUS GU502G Windows PC utilizing python scripts to process the incoming image and sensor data and sends the calculated angle to the Botboarduino controller, which then moves the two Hitec HS-85BB servo motors to replicate the motion of the user.

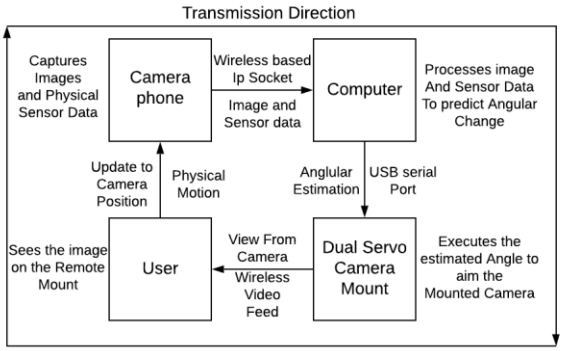

Fig. 1: System visualization

For communication between the various components, the Android phone and laptop communicate using TCP wireless protocol while both are connected to the same network using server-client sockets. The laptop and Botboarduino communicate using a serial port USB connection. The incoming sensor data consists of the linear accelerometer, gravitational vector, and gyroscope data, passed as a string which is decoded by the laptop. The laptop sends a request signal and a received signal that to keep the data from being interpreted only in single sets. Similarly, the image data is passed in JPEG format as a series of bytes and read and then processed by the laptop. The phone begins the process of taking the next image and decoding it into bytes until the

signal from the laptop for the next image is received, then the process starts again.

2.2 Image-Based Prediction

For the image-based approach to angular prediction, the keypoint detection is utilized Traditional forms of key points include edge detection and corner detection, but these key points do not work well when changes in lighting or viewing angle occur. Therefore, the scale invariant form of keypoints such as what was originally developed in the SIFT[20] and SURF[21] methods is performed on the image to use different image resolutions to define key points in multiple scales. Images are converted to grayscale before processing, as often. These scale invariant keypoints are more easily detectable and robust without regards to lighting or viewing angle, and in this application more easily trackable. Here we make use of the ORB algorithm [22], due to the reduced requirement of system resources as compared to other methods, which is important for faster tracking of motion. The results of this type of detection can be seen in Fig. 2, where the keypoints and magnitude and orientation descriptors are drawn on a red panda image.

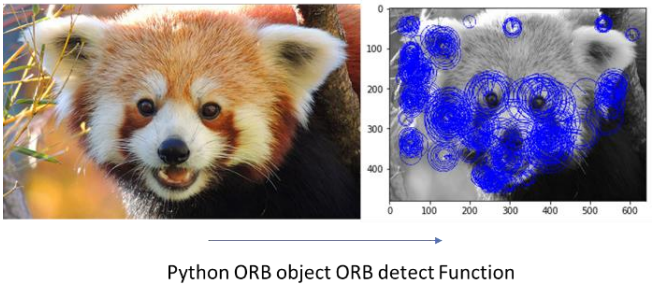

Fig. 2 Keypoint detection and description.

For sequential images, the keypoints are matched using the BFMatcher (brute force matcher) object and the relative motion in frame of the best 5 keypoints is calculated. The average motion of the keypoints is computed as the angular rotation when measured within the field of view of the camera, using Eq. (1):

$$\Delta\theta = \frac{Fov}{res}\frac{1}{N}\Sigma\Delta x \qquad (1)$$

where field of view (*Fov*) is divided by the image resolution (*res*) and multiplied by the average pixel movement (Δx). An example of this process can be seen in Fig. 3. This method works well even when the image resolution is somewhat limited.

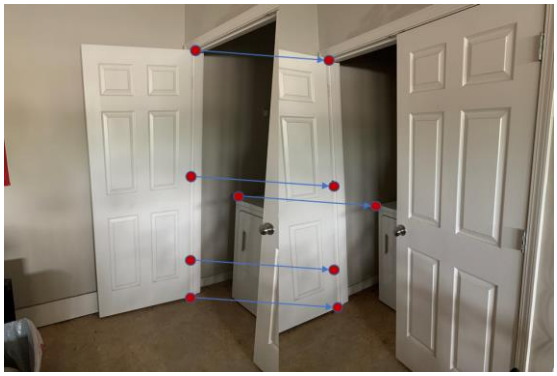

Fig. 3 An Example of keypoint matching.

2.2 Sensor-Based Prediction

Gyroscope sensors are widely used in modern smartphones, which can track the motion of phones. In this paper, the gyroscope sensor is set in the GAME delay mode, i.e., output at 50Hz. All gyroscope data are relative to the phone frame shown in Fig. 4, which changes when the phone moves. To reduce the influence of noise, a classic Butterworth lowpass filter is applied on the sensor data before they are used for pose calculation. During each sampling interval, rotation velocity of a phone is considered constant. Given initial direction of the phone, the relative final direction can be calculated in the following way:

$$\boldsymbol{\theta}_1 = \dot{\boldsymbol{\theta}}_0 \Delta t \qquad (2)$$

where $\dot{\boldsymbol{\theta}}_0$, a three-dimensional (3D) vector, are initial rotation velocities, acquired from the gyroscope sensor, and Δt is the corresponding sampling interval. For consecutively sampling process, initial position of the process is defined as the absolute reference frame, i.e., $\boldsymbol{\theta}_0 = \mathbf{0}$. When the first data comes in, Eq. (2) can be used to calculate the initial pose of the next sampling interval. For the second sampling interval, pose of the phone relative to the first sampling interval can be calculated using cardan angles (Z (ψ_1)-X (θ_1)-Y (φ_1)):

$$\boldsymbol{R'}_1 = \begin{bmatrix} C_z C_y - S_z C_x S_z & -C_z S_y - S_z C_x C_y & S_z S_x \\ S_z C_y - C_z C_x S_y & -S_z S_y + C_z C_x C_y & -C_z S_x \\ S_x S_y & S_x C_y & C_x \end{bmatrix} \qquad (3)$$

where $S_x = sin\theta_1$, $C_x = cos\theta_1$, $S_y = sin\varphi_1$, $C_y = cos\varphi_1$, $S_z = sin\psi_1$, and $C_z = cos\psi_1$.Thus, the ending pose of the second sampling interval can be defined in the absolute reference frame as $\boldsymbol{R_1} = \boldsymbol{R'_1 R_0}$. Due to the fact that $\boldsymbol{\theta}_0 = \mathbf{0}$, $\boldsymbol{R}_0$ is an identity matrix. For the third sampling interval, the pose of the phone can be defined as $\boldsymbol{R_2} = \boldsymbol{R'_2 R_1}$. Where $\boldsymbol{R'_2}$ is the pose of the phone relative to the previous sampling interval. Iteratively, pose of the n-th sampling interval can be obtained by:

$$\boldsymbol{R_n} = \boldsymbol{R'_n R_{n-1}} = \cdots = \boldsymbol{R'_n R'_{n-1}} \cdots \boldsymbol{R'_1 R_0} = \begin{bmatrix} C_z C_y - S_z C_x S_Z & -C_z S_y - S_z C_x C_y & S_z S_x \\ S_z C_y - C_z C_x S_y & -S_z S_y + C_z C_x C_y & -C_z S_x \\ S_x S_y & S_x C_y & C_x \end{bmatrix} (4)$$

where $S_x = sin\theta'_n$, $C_x = cos\theta'_n$, $S_y = sin\varphi'_n$, $C_y = cos\varphi'_n$, $S_z = sin\psi'_n$, and $C_z = cos\psi'_n$. The prime indicates incremental of the angles between two samples. Based on Eq. (3), cardan angles $\boldsymbol{\theta_n}' = [\theta'_n \quad \varphi'_n \quad \psi'_n]^T$ of each sampling interval relative to the absolute reference frame can be calculated, which represent the current pose of the phone relative to the absolute initial pose. In addition, these angles are also exactly the absolute rotation angles of our actuator.

Fig. 4 Definition of phone frame.

III. EXPERIMENTAL SETUP AND RESULTS

3.1 Experimental Setup

The key parameter tested for the system introduced in this work is the accuracy in the prediction of the rotation angle. In order to test the accuracy and capability of the prediction algorithm a special mount was developed for the phone system to standardize the angular rotation that the phone experiences. To achieve this a standard servo motor (HiTec HS645MG) was attached to an adjustable phone mount. The vertical angle of rotation can be adjusted offline while the lateral rotation is controlled by the attached servo to generate continuous motion, as seen in Fig. 5. The Botboarduino controlling the servo mount receives angular commands through the USB serial cable connected to the PC. The designed experimental setup allows the angle of the servo motor to directly act as the angular input. In summary, the main tested parameters for the system are the prediction accuracy concerning the angle amplitude and the angular velocity. Because the vertical rotation of the camera phone is easy to estimate from the gravity vector, this work focuses on recreating the more difficult yaw rotation, the axis of which is parallel to the gravitational direction.

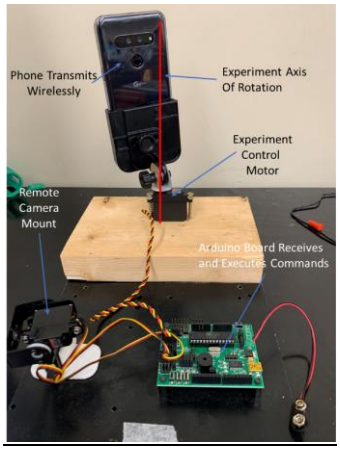

Fig. 5 Experimental Setup.

3.2 Experiment Design

For the experiment, the two different tests correlate to two different rates of angular motion for the rotation of the mount holding the camera phone. For the first set of given data, the measurements were taken with a rotation speed of 3.45s per 30 degrees of rotation, favoring the image-based algorithm, and for the second a rate of 2.09s per 90 degrees, favoring the sensor-based algorithm. The first dataset taken for each rotation speed consists of a simple motion from 0 degree to the -45 degree angle, then stopping. This can be seen in the figures as the simplest graph. The next two datasets at each speed consisted of a motion from 0 degree to the -60 degree angle, next to the positive 60 degree angle, and then back to the 0 degree angle, the maximum amplitude, and representing the longest continuous motion. The next motion consists of movement from 0 degree to the -45 degree angle, then to the 0 degree, 45 degree and back to the 0 degree angles, a smaller motion amplitude but greater command density. The final dataset consists of motion from 0 degree to the -60 degree angle, motion in 30 degree increments to the 60 degree angle, and finally motion back to the 0 degree angle, the largest command density set, used to test issues associated with estimation drift.

3.3 Results and Discussion

The results of the eight different motion sequences are pictured in Fig. 6 and Fig. 7, the first four of which correspond to the slow rate of motion, about 3.45s per 30 degrees, and the next four correspond to the fast rate of motion, about 2.09 seconds per 90 degrees. For this slower rate of rotational velocity the image-based approach clearly better estimates the given command angle in the experimental setup. The estimation error is given as the difference between the input to the experimental setup and the algorithmically estimated angle. The maximum absolute estimation error from any given command was only 2.66 degrees and occurred in the densest command dataset. The average absolute error on all commands was 0.98 degrees and the maximum residual error, or error remaining between the input angle and estimated angle at the end of the trial, in any trial was 1.14 degrees, indicating that many commands would have to be given to the

camera mount before residual error from motion estimation would be at all significant.

The estimation is based only on minimizing distortion of the image data, so that increasing accuracy can be achieved by increasing the streaming and processing speed. The sensor-based approach as can be seen above is highly suspect to noise and thus drift at a slower rate of motion, as well as any vibration from the motor itself. To minimize the noise and vibration error, i.e., to increase the signal-to-noise ratio (SNR), higher rotation speeds as in the second experimental trial can lead to more accurate angular estimation, a compliment to the necessarily slower image data method.

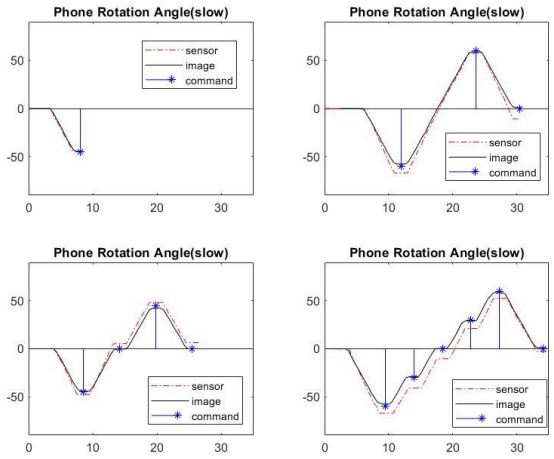

Fig. 6 Results of the Slow Rotation Experiment

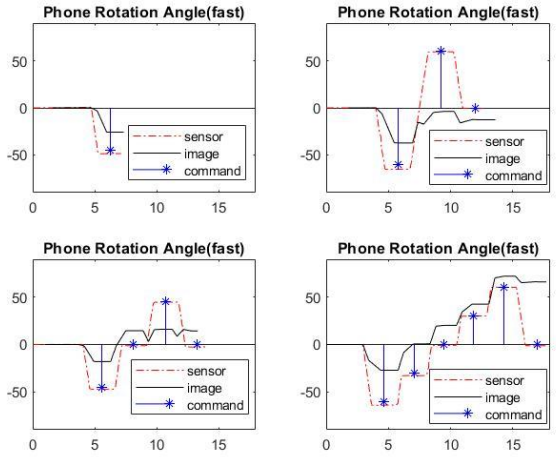

Fig. 7 Results of the fast rotation experiment.

From the above figure the advantage of the sensor-based approach at higher rotational speed becomes apparent, as the image-based approach becomes unable to match keypoints between image views with any significant degree of accuracy. Thus, the image keypoint based approach fails at sufficiently high speed. The sensor-based approach in contrast performs well at higher rotational velocities since it has a higher SNR, where in this experiment the average absolute error was only 1.84 degrees, high accuracy for a numerical approach to noisy data. The maximum residual error occurred during the simplest trial, where a lack of motion in the reverse direction creates a larger drift error. Finally, the maximum estimation error occurred where the largest initial rotation occurred, indicating larger motions are more subject to slight errors in sensor data.

IV. Conclusion and Future Work

In this work, a system meant to capture user motion using a camera phone and recreate this motion in a remote camera mount system was developed and the accuracy of the system in predicting angular motion was tested, by way of both sensor-based angular estimation and image-based angular estimation, each of which proved to be accurate with respect to different angular rotation rates, as seen in Fig. 8, where motion was begun at the slow rate and then duplicated at the fast rate of rotation.

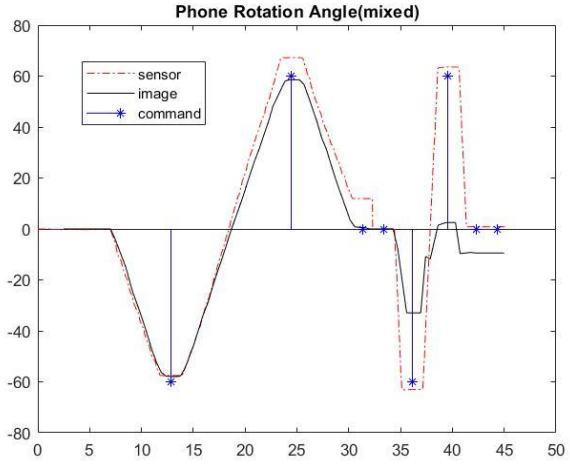

Fig. 8 Example of both slow and fast experiment.

Both data processing approaches were reset to zero angular change in the middle, and the clear difference in accuracy of each based on rotational velocity can be seen. The experimental data indicate that high accuracy in combined data can be achieved over a range of rotational velocities.

In future iterations of the system design, a higher communication speed through adjustment of the network communication scheme will be implemented to aid in increasing the accuracy and speed of the image-based approach to angular rotation prediction. Further improvements include the combining of the two forms of angular estimations, first by way of experimental weighting parameters of the two estimations, and then by using statistics-based methods similar to those of the Kalman or particle filters to create an accurate, continuous speed-based estimation weighting model. Finally, the remote camera mount component will hold a wireless video camera, which will transmit its image feed back to the user interacting with the camera phone.

REFERENCES

[1] Rubio, F., Llopis-Albert, C., Valero, F. and Suñer, J.L, 2016, "Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory". Robotics and Autonomous Systems, 86, pp.106-112.

[2] Slamani, M., Nubiola, A. and Bonev, I., 2012. "Assessment of the positioning performance of an industrial robot". Industrial Robot: An International Journal.

[3] Rubio, F., Valero, F., Sunyer, J. and Mata, V., 2009. "Direct step-by-step method for industrial robot path planning". Industrial Robot: An International Journal.

[4] Cakmak, M. and Takayama, L., 2013, March. "Towards a comprehensive chore list for domestic robots." In 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (pp. 93-94). IEEE.

[5] Prassler, E., Munich, M.E., Pirjanian, P. and Kosuge, K., 2016. *Domestic robotics. In Springer handbook of* robotics (pp. 1729-1758). Springer, Cham.

[6] Palacin, J., Salse, J.A., Valgañón, I. and Clua, X., 2004. "Building a mobile robot for a floor-cleaning operation in domestic environments." IEEE Transactions on instrumentation and measurement, 53(5), pp.1418-1424.

[7] Sutter, C., Oehl, M. and Armbrüster, C., 2011. "Practice and carryover effects when using small interaction devices". Applied ergonomics, 42(3), pp.437-444.

[8] Liu, H. and Wang, L., 2018. "Gesture recognition for human-robot collaboration: A review". International Journal of Industrial Ergonomics, 68, pp.355-367.

[9] Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A. and Mulanda, D., 2004. "Humanoid robots as cooperative partners for people". Int. Journal of Humanoid Robots, 1(2), pp.1-34.

[10] Kaufman, R.E., 2007, March. "A family of new ergonomic harness mechanisms for full-body natural constrained motions in virtual environments". In 2007 IEEE Symposium on 3D User Interfaces. IEEE.

[11] Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L. and Di Marca, S., 2006. "Accelerometer-based gesture control for a design environment". Personal and Ubiquitous Computing, 10(5), pp.285-299.

[12] Luinge, H.J., Veltink, P.H. and Baten, C.T., 1999, October. "Estimation of orientation with gyroscopes and accelerometers". In Proceedings of the First Joint BMES/EMBS Conference. 1999 IEEE Engineering in Medicine and Biology 21st Annual Conference and the 1999 Annual Fall Meeting of the Biomedical Engineering Society (Cat. N (Vol. 2, pp. 844-vol). IEEE.

[13] Favre, J., Jolles, B.M., Siegrist, O. and Aminian, K., 2006. "Quaternion-based fusion of gyroscopes and accelerometers to improve 3D angle measurement". Electronics Letters, 42(11), pp.612-614.

[14] Zhang, X., Chen, X., Wang, W.H., Yang, J.H., Lantz, V. and Wang, K.Q., 2009, February. „Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors". In Proceedings of the 14th international conference on Intelligent user interfaces (pp. 401-406).

[15] DelPreto, J. and Rus, D., 2020, March. "Plug-and-Play Gesture Control Using Muscle and Motion Sensors". In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (pp. 439-448).

[16] Benko, H., Saponas, T.S., Morris, D. and Tan, D., 2009, November. "Enhancing input on and above the interactive surface with muscle sensing". In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (pp. 93-100).

[17] Liu, Y.H., Huang, H.P. and Weng, C.H., 2007. "Recognition of electromyographic signals using cascaded kernel learning machine". IEEE/ASME Transactions on Mechatronics, 12(3), pp.253-264.

[18] Mian, A.S., Bennamoun, M. and Owens, R., 2008. "Keypoint detection and local feature matching for textured 3D face recognition". International Journal of Computer Vision, 79(1), pp.1-12.

[19] Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. "Object detection with deep learning: A review". IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.

[20] Lowe, D.G., 2004. "Distinctive image features from scale-invariant keypoints". International journal of computer vision, 60(2), pp.91-110.

[21] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., 2008. "Speeded-up robust features (SURF)". Computer vision and image understanding, 110(3), pp.346-359.

[22] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011, November. "ORB: An efficient alternative to SIFT or SURF". In 2011 International conference on computer vision (pp. 2564-2571). IEEE.