Designing a Prototype for Smartphone Footage Stabilization

Original Scientific Paper

Bharathi R.B.*
Department of Electrical & Electronics Engineering,
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, Karnataka, India
bharathi.rb@manipal.edu

VijayaLaxmi
Department of Electrical & Electronics Engineering,
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, Karnataka, India
vijaya.laxmi@manipal.edu

Shashank Bhat
Department of Data Science and Engineering
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, Karnataka, India
shashank2.mitmpl2023@learner.manipal.edu

*Corresponding author

Abstract – *Hardware stabilizers are on high demand due to the increasing use of media capture devices such as smartphones and portable cameras in market. Hence an effort is made to develop a stabilizer prototype with Inertial measurement Unit (MPU 6050 sensor), arduino Uno microcontroller and servo motor. Using existing libraries of arduino, a code is developed to read the position of the smartphone by the sensor and is fed to arduino microcontroller. The microcontroller generates an appropriate PWM signal based on the output of sensor and sends it to two servo motors to nullify the effect of jitter produced by the movement of the smartphone along roll and pitch axes respectively. Stabilized footage is produced by smartphones compensating the movement along roll and pitch axes respectively. The cost of this developed prototype is very low compared to existing gimbal systems in the market.*

Keywords: *Hardware stabilizer, Inertial measurement Unit, servo motor*

Received: March 2, 2024; Received in revised form: May 2, 2024; Accepted: May 27, 2024

1. INTRODUCTION

Video stabilization is a method used to minimize blurring caused by camera movement during video recording [1]. Shaky videos have always been a problem for professional photographers. The rapid advancement of smartphone technology is closely linked to the development of photography and image processing. The camera is among the most used features. Poor quality results are produced by shocks and hand movements. A stabilizer for camera position is required to enhance the quality of camera images and video [14]. Using a 2- or 3-axis (roll, tilt, and yaw) gimbal system to separate the rotation of the UAV platform from the camera is the most popular method for video stabilization [2].

Full-frame films can be created using the technique suggested by Matsushita et. Al. [3], which works by locally matching the picture data of nearby frames to automatically fill in any missing image segments. Motion inpainting is suggested as a means of enforcing temporal and spatial consistency of the completion in both dynamic and static image sections.

Over the past 20 years, research on video stabilization (VS) has been ongoing. Guilluy et. Al. [4] focused particularly on the Video Stabilization Quality Assessment (VSQA) and introduced a new methodology that was influenced by the findings of research on Image Quality Assessment (IQA) in general. When image stitching techniques are directly applied to shake films, significant temporal and spatial distortions are frequently present [5]. To address this issue, Guo et. Al. [5] provides a unified framework that combines video stabilization and stitching.

There have been three phases of video stabilization: digital, optical, and mechanical [6]. By using an invisible tripod built into the camera to correct the recorded video and produce stabilized videos, a device called a gyroscopic stabilizer is used in mechanical stabilization. The use of optical stabilization is used in both high-end SLR cameras like Nikon and Canon and smartphones like Samsung, Apple, and others. Optical stabilization counteracts image vibration and achieves image stabilization by moving the lens group on a plane perpendicular to the optical axis. Without the need for hardware, digital stabilization directly determines the motion vectors of the subsequent frames. Motion correction and filtering are used to isolate and eliminate the shaky component. Ultimately, the picture warp reconstructs frames.

To increase the estimation efficiency, the suggested approach [7] employs differential global motion estimation with Taylor series expansion. The inter-frame error between successive frames is defined by the Affine Motion Model. Through the analytical solution of the derivatives of the inter-frame error, motion vectors have been computed. Gaussian kernel filtering has been used to smooth out computed motion parameters following motion estimation. To eliminate the rotation effect from the smoothed transformation chain, inverse rotation smoothing has been used. As a result, the accumulation error has been decreased by [7] and the missing image area has been much reduced.

To address the issue of video stabilization, Liu et. Al. [8] suggests using an additional depth sensor, such as the Kinect camera. Even with its low resolution, noise, and incompleteness, the depth image helps with frame warping and camera motion estimation, making video stabilization a much better-posed problem.

A unique method for stabilizing videos is suggested by Yang et. Al. [9] and is based on the particle filtering framework. Yang et. Al. [9] added tracking of the projected affine model of the camera motions to the conventional application of particle filters in object tracking. Through simulated experiments, the algorithm's improved performance is shown by the author.

Image stabilization, whether software or hardware based, can significantly enhance images, particularly in hand-held photography, slower shutter speed shooting, and other unsteady conditions. Compared to SLR lenses smart phone cameras are far slower. They leave the shutter open for an extended period, which blurs any movement [10]. There are many techniques available in the present-day market that resolve this problem. They are Optical steady shot (OSS) by Sony, Vibration Reduction (VR) by Nikon and Image stabilizer (IS) by Canon [11]. Stabilization of footage is acquiring a greater significance day by day, and the form factor of the cameras that are used by the users is also rapidly decreasing [12].

Stabilizing footage either through specialized hardware or by software is now becoming an industrial norm for use either in personal vlogs or in commercial videography applications. With the increasing penetration of media capture devices such as mobile phones and portable cameras, the market for stabilizers in small appliances is increasing day by day [13]. Osmo mobile, manufactured by DJI is one of the solutions for the same, currently costing nearly 130 dollars. Thus, the primary goal of the research is to reduce the cost of the prototype.

Hence, a hardware prototype is designed that can be used for video stabilization which builds upon the existing technologies that are already present in the camera.The smartphone video stabilizer has Arduino microcontroller, gyro sensor MPU-6050 [14] and 2 servo motors for stabilization along two axes roll and pitch respectively. The change in angular velocity or displacement of the phone is detected by the MPU-6050 sensor and this data will be transmitted to the microcontroller. The servo motor then rotates the base on which the smartphone is placed by the angle detected by the gyro sensor, hence, compensating for the motion. Therefore, resulting in a stabilized footage. A servo motor is used to stabilize the smartphone for smoother footage. Fig. 1 summarizes the block diagram of the proposed solution.

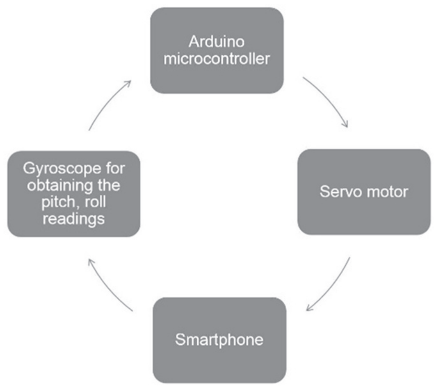

Fig. 1. Block Diagram of the proposed solution

2. METHODOLOGY

The algorithm of the working of the prototype is summarized below:

Step 1: Wake up the MPU 6050 and initiate communication via I2C protocol using address 0x68.

Step 2: Assign the servo motors to pins 9,10 (PWM pins) of the Arduino controller. Initialize the Pulse width values corresponding to 0 and 180 degrees as obtained from calibration to separate variables.

Step 3: Send a command to the Arduino to obtain the first values from the accelerometer and the gyroscope.

Step 4: Calculate pitch and roll and assign them as initial values to the complementary filter.

Step 5: Wait for an interrupt and hence obtain the raw values from the sensor again. Calculate values of roll and pitch.

Step 6: Update the complementary filter with the new values of roll and pitch.

Step 7: Use this filtered value and map it to obtain the value of pulse width based on the values obtained from the calibration code.

Step 8: Send the appropriate pulse width to the motors to compensate for the motion caused by the jitter.

A brief description of the components used in the development of the prototype is below:

2.1. MPU 6050 SENSOR

To stabilize the platform on which the smartphone was placed, an Inertial Measurement Unit (IMU) was required to identify the degree to which the setup deviated from its initial position [13]. The MPU-6050 is an advanced chip that has two on-board sensors, a gyroscope and an accelerometer, each with its advantages [14]. The individual components of the IMU (MPU 6050) are extremely susceptible to noise. The gyroscope component is accurate for short duration of time but tends to drift after time (because it is integrating), while the accelerometer is sensitive to forces and gives inaccurate reading for short duration of time. The gyroscope gives a high quality signal in a short term and the accelerometer is good for measuring changes in speed. The complementary filter is a powerful and simple tool to improve the shortcomings of MPU 6050 sensor. Combining the accelerometer and gyroscope together gives the basic complementary filter. A complementary filter combines a high-pass and a low-pass filter. In this case, a high-pass filter is used on the gyroscope readings, and a low-pass filter is applied over the accelerometer readings. A complementary filter eliminated the gyroscope drift and reduced the accelerometer sensitivity. Thus, the output becomes more robust against noise factors [15]. The coefficients of the high-pass and low-pass filters add up-to one, and hence filter is said to as complimentary. To appropriately calibrate the filter, the coefficients of filter are changed by trial-and-error method. The high-pass filter design with a coefficient of 0.98 and low-pass filter with a coefficient of 0.02 is found to be appropriate for the given application. Therefore, 98% of the values for each update come from the gyroscope, and 2% come from the accelerometer.

The MPU-6050 is tested at the initial stage using an external I2C library developed by Jeff Rowberg and has been interfaced by a self-authored code using a complementary filter. The library had in-built calls for obtaining values from the MPU-6050 and made use of the on-board DMP (data management platform) to do the same. The values from the accelerometer, gyroscope stored individually on the MPU 6050 on different registers. Arduino is open source and is used for a large variety of projects. The filtered values from sensor are read by Arduino using the I2C communication protocol. The I2C communication address used by the sensor is a 7-bit address 0x68. The sensor supports a communication speed of 400Khz. Every time the data is ready, the sensor makes the interrupt (INT) pin high. MPU-6050 sensors are placed in the unstabilized portion of the setup, and the degree of instability, as communicated by the sensor, is then sent to the Arduino controller after filtering to control the servo motor accordingly [14]. The Sultana et. Al. [16] uses the MPU 6050 with a complementary filter for the stabilization of the video. The performance of various wearable cameras in the surgical setting is carried out and has concluded that IMU based sensors are appropriate in the operating room to objectively quantify camera motion [17]. The results are validated by comparing the positional sensing of the device to a geared tripod head that allows for fine, measured manipulations of the sensor in three orthogonal axes.

2.2. SERVO MOTOR

The servo motor is connected to the output of the microcontroller and has metal gears with a torque of 15.5kg cm to 17 kg cm. It has a built-in gearbox, position feedback mechanism, and motor controller. The servo motor can be controlled to move to any position by using PWM using in-built resistive feedback. This motor has a three-wire interface, one for control and two for power supply. The weight of the motor is 77 grams and has an operating voltage of 4.8V-6V. The biggest advantage of using a servo motor is the presence of robust pre-existing library functions for controlling the servo motor [18]. The servo motor is also extremely lightweight and cost-effective.

For controlling the position of the servo motor, a PWM signal is to be sent to it. The width of this PWM signal determines the angle at which the shaft of the servo motor moves. Based on the manufacturer, the width of the PWM signal that is required to move the shaft from 0 to 180 degrees varies. To calibrate the servo motor to move exactly by 180 degrees, a pre-existing Arduino library for controlling servo motors was used to send a PWM signal of a width starting from an extremely low value of 100 microseconds. At 540 microseconds, it was noticed that the shaft of the motor started to move. This was marked as the pulse width corresponding to 0 degrees. A paper was placed below the servo motor and this position was marked. Then at 2140 microseconds pulse width, it was noticed that the shaft moved by exactly 180 degrees. A mapping function was then used to linearly map 540 to 2140 microseconds pulse width to -90 to +90 degrees, to match the output obtained from the MPU-6050. The motors drew approximately 400mA of current and a peak current of 4A. The use of servo motors, on account of their easier PWM-based position control, originated directly from the Arduino website itself [19].

2.3. ARDUINO UNO BOARD

An Arduino Uno was used in the development of the prototype. The I2C communication protocol was used to communicate with the MPU-6050 sensor, while two PWM pins of the Arduino Uno microcontroller were used to communicate with two motors. The software implementation was performed in the Arduino IDE.

2.4. OVERALL CIRCUIT

Connection of MPU 6050 with arduino Uno:

- Vcc to +5V
- Gnd to Gnd of Arduino
- SDA to Analog pin A4 (I2C communication)
- SCL to Analog pin A5 (I2C communication)
- INT to Digital pin 2

Connections of Servo motors with Arduino Uno:

- Red wire to positive terminal of external supply
- Black wire to negative terminal of external supply
- Yellow wire of motor B to Digital pin 9 of Arduino (PWM)
- Yellow wire of motor A to Digital pin 10 of Arduino (PWM)
- GND of Arduino to negative terminal of external supply

The overall circuit connection is shown in Fig. 2.

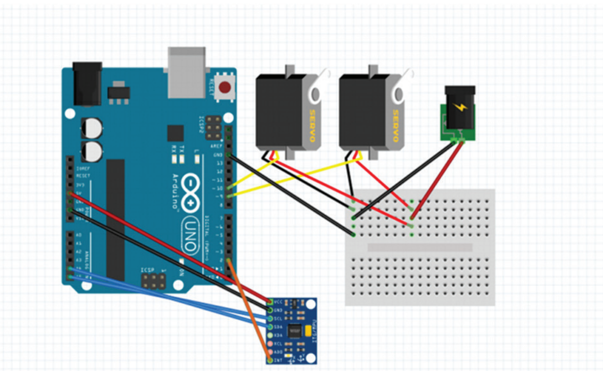

Fig. 2. Overall Circuit Diagram

The implementation of the chassis and the software is discussed below:

2.5. CHASSIS DESIGN

The servo motor (A), which is stabilizing along the X axis. The IMU is guaranteed to be on the unstabilized section of the chassis because the user is holding this motor. On one side of motor A, the servo motor (B) for Y-axis stabilization is attached, and on the other, the rotating shaft of this motor is connected to the platform holding the smartphone. A servo horn drilled into a wooden block has been used to mount Motor (A) on Motor (B), which is subsequently attached onto Motor (A) using epoxy resin. Additionally, Motor (B) is attached to a servo horn that is drilled into a wooden block that is positioned 90 degrees from the initial block. The revolving shaft of this motor is attached to the platform holding the smartphone. To provide smartphone stabilization, motors (A) and (B) work in tandem.

The chassis design of the prototype is indicated in Fig. 3.

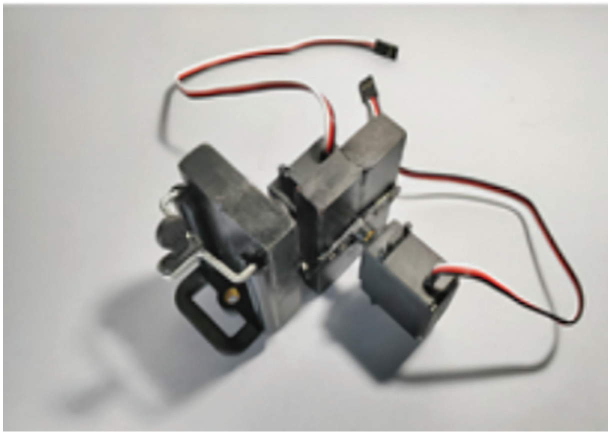

Fig. 3. Chassis design of the proposed solution

3. RESULTS AND DISCUSSION

The prototype designed will compensate for the motion along pitch and roll axes by employing the sensor along with two servo motors. The Gyroscope +accelerometer sensor is interfaced with the Arduino microcontroller. After calibrating the servo motor, they are placed on the chassis as described in Fig. 3. Upon running the code, the IMU sensor continuously monitors for any change in angle. If there is a change in the angular position of the motor, caused by a movement in the user's hand or by jitter, this change is filtered using the complementary filter sent to the Arduino microcontroller for better stabilization. If unfiltered readings are sent to the motor, the motor shall not be able to stabilize as efficiently as with the filtered readings. The change in angle is mapped to the corresponding value of pulse width by the Arduino microcontroller. This pulse width modulation signal is fed to the servo motor and the motor then moves to the corresponding change in angle. This process is carried on iteratively. Note that this is an open-loop control system.

A graph of unfiltered values of Roll & pitch vs filtered values of Roll & pitch are explained in plotted in Fig. 4 and Fig. 5 respectively. The illustrated values include the angle in pitch (Y-axis) and roll (X-axis), both of which are required to achieve stabilization. Fig. 4 and Fig. 5 denote the results in the self-authored code, with the X-axis illustrating time and the Y-axis illustrating angle in degrees along the pitch and roll direction respectively. The MPU-6050 sensor is moved randomly in space, and the resulting values are illustrated on the graph. The blue curves denote the unfiltered values while the red curves denote the filtered values.

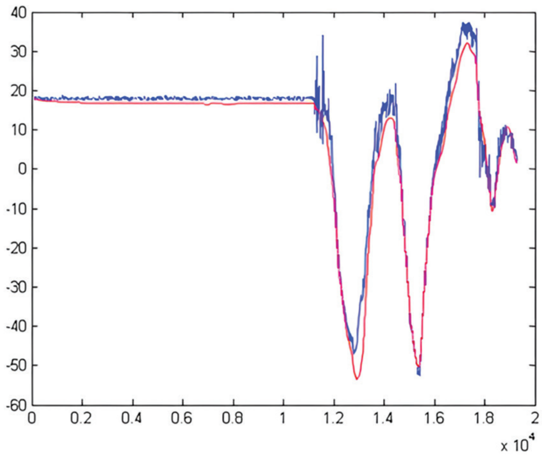

Fig. 4. Self-authored code with the obtained values for pitch (Y-axis)

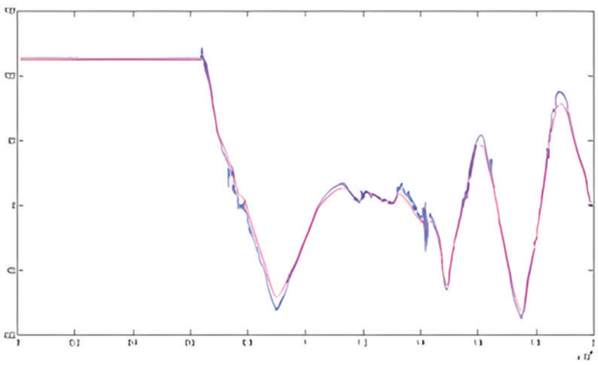

Fig. 5. Self-authored code with the obtained values for roll (X-axis)

To demonstrate the stabilization, the setup is moved first along the X axis (roll) and the resulting stabilization is shown in Fig 6. The motion of the setup is caused along Y-axis (pitch) is shown in Fig. 7. A snapshot from a video showing the stabilization along both axes is shown in Fig. 6 and Fig. 7 respectively. Hence, the movement of the person holding the chassis shall not matter, the platform housing the smartphone will remain stable.

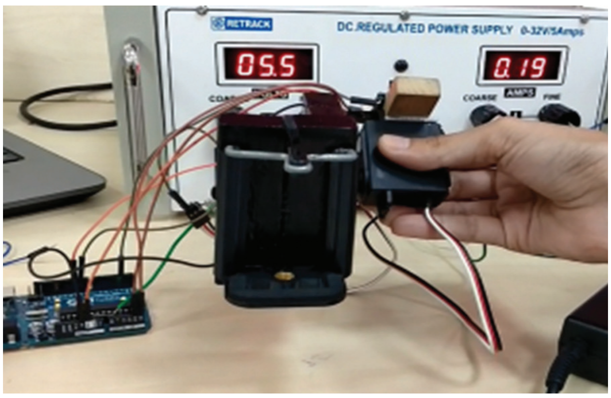

Fig. 6. Stabilization along X-axis (roll)

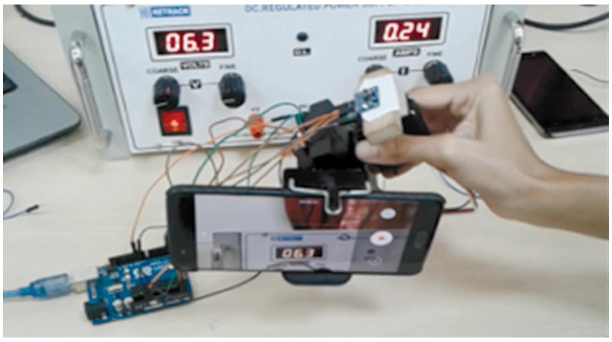

Fig. 7. Stabilization along Y-axis (pitch)

To illustrate the movement of the motor upon sending the required final angle through the Arduino code, an app developed by Google called 'Science Journal' was used as indicated in Fig. 8. This app used the in-built compass of a smartphone to graphically illustrate the current angle as read from the phone's compass. Further phone is attached to the top of the servo motor to obtain these readings. Thus, the command was given to the motor to move by different angles, and the readings obtained are shown in Fig. 8.

Fig. 8. Motor angle with respect to time, as obtained using the compass sensor of a smartphone, using the science journal app by Google

The angle detected by the compass sensor of a smartphone is around 16° Fig. 8 (part A) and the command was given to the motor to provide compensation is 15° as indicated in Fig. 8 (part B). Hence the error found to be around 1°.

Rafiq [14] and others develop a gimbal smartphone, which is made simple and less money consuming by utilizing microcontroller and MPU 6050 sensor. This MPU 6050 sensor is optimized to detect sway in axis X, Y, and Z or roll, pitch, and yaw. Gyroscope and accelerometer provide input to the microcontroller, which will process output on 3 servomotors that function to maintain the camera's position at a specified set point. The results show that MPU 6050 sensor can response angle reading error of 1.34° of roll, 0.25° of pitch, and 0.78° of yaw. Error in maximum servomotor movement is 1.5° [14]. The performance of the designed prototype cannot be compared with the reference [14] in terms of performance as the present design works on;y roll and pitch axes.

4. CONCLUSIONS

Smartphone are embedded with software stabilizer (like OSS and VR), but hardware stabilizers are more accurate specially for photographic and good quality video recording purposes. In this work,a prototype is developed to compensate the jitter in shaky videos along roll and pitch axes. The values from a gyroscope and an accelerometer of MPU-6050 are sent to complementary filter. The arduino receives the filtered values and this will generate a PWM signal. The width of this PWM signal determines the angle at which the shaft of the servo motor moves. Two PWM signals are used to control the position of motor along 2 axes. The prototype explains a new method with less cost to improve the quality of image from smartphone. The error will be introduced by the sensor and DC motor operation. The proposed solution can be scaled up to stabilize along the third axis yaw (Z axis). Stabilization along all three axes will ensure better and good quality footage. The total cost of the developed prototype including the component cost is around Rs. 3630.00.

5. REFERENCES:

[1] F. Vella, A. Castorina, M. Mancuso, G. Messina, "Robust digital image stabilization algorithm using block motion vectors", Proceedings of the International Conference on Consumer Electronics, Los Angeles, CA, USA, 18-20 June 2002, pp. 234-235.

[2] M. Odelga, N. Kochanek, H. H. Bülthoff, "Efficient real-time video stabilization for UAVs using only IMU data", Proceedings of the Workshop on Research, Education and Development of Unmanned Aerial Systems, Sweden, 3-5 October 2017, pp. 210-215.

[3] Y. Matsushita, E. Ofek, X. Tang, H.-Y. Shum, "Full-frame video stabilization", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20-25 June 2021, pp. 50-57.

[4] W. Guilluy, L. Oudre, A. Beghdadi, "Video stabilization: Overview, challenges and perspectives", Signal Processing: Image Communication, Vol. 90, 2021.

[5] H. Guo, S. Liu, T. He, S. Zhu, B. Zeng, M. Gabbouj, "Joint video stitching and stabilization from moving cameras", IEEE Transactions on Image Processing, Vol. 25, No. 11, 2016, pp. 5491-5503.

[6] Y. Wang, Q. Huang, C. Jiang, J. Liu, M. Shang, Z. Miao, "Video stabilization: A comprehensive survey", Neurocomputing, Vol. 516, 2023, pp. 205-230.

[7] H.-C. Chang, S.-H. Lai, K.-R. Lu, "A robust and efficient video stabilization algorithm", Proceedings of the IEEE International Conference on Multimedia and Expo, Taipei, 27-30 June 2004, pp. 29-32.

[8] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, J. Sun, "Video stabilization with a depth camera", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16-21 June 2012, pp. 89-95.

[9] J. Yang, D. Schonfeld, M. Mohamed, "Robust video stabilization based on particle filter tracking of projected camera motion", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 7, 2009, pp. 945-954.

[10] D. Har, Y. Son, S. Lee, "SLR digital camera for forensic photography", Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications, Vol. 5301, 2004, pp. 276-284.

[11] R. Furchgott, "Camcorders allow a steady shot, even without a steady hand", The New York Times, 2000.

[12] I. Rodriguez-Padilla, B. Castelle, V. Marieu, D. Morichon, "A simple and efficient image stabilization method for coastal monitoring video systems", Remote Sensing, Vol. 12, No. 1, 2019.

[13] K. N. Truong, S. N. Patel, J. W. Summet, G. D. Abowd, "Preventing camera recording by designing a capture-resistant environment", Proceedings of the 7th International Conference UbiComp, Tokyo, Japan, 2005, pp. 73-86.

[14] A. A. Rafiq, W. N. Rohman, S. D. Riyanto, "Development of a simple and low-cost smartphone gimbal with MPU-6050 sensor", Journal of Robotics and Control, Vol. 1, No. 4, 2020, pp. 136-140.

[15] S. Ortega, F. Javier, "Microelectromechanical Systems Inertial Measurement Unit As An Attitude And Heading Reference System", Open Access Theses & Dissertations, 2017, https://scholarworks.utep.edu/open_etd/546 (accessed: 2024)

[16] J. M. Sultan, N. H. Zani, M. Azuani, S. Z. Ibrahim, A. Md Yusop, "Analysis of inertial measurement accuracy using complementary filter for MPU6050 sensor", Jurnal Kejuruteraan, Vol. 34, No. 5, 2022, pp. 959-964.

[17] T. J. Saun, T. P. Grantcharov, "Design and validation of an inertial measurement unit (IMU)-based sensor for capturing camera movement in the operating room", HardwareX, Vol. 9, 2021.

[18] Instructables, “Making a brushless DIY gimbal with Arduino”, http://www.instructables.com/id/DIY-Brushless-Gimbal-with-Arduino/ (accessed: 2024)

[19] J. A. Langbridge, “Arduino sketches: tools and techniques for programming wizardry”, John Wiley & Sons, 2015.