Project Report: Autonomous 6 Axis Face Tracking

Eliot Hunter [1], Shota Kakiuchi [2]
Department of Mechanical and Aerospace Engineering
The George Washington University
Washington, D.C, US
ehunter25@gwu.edu [1], skakiuchi@gwu.edu [2]

ABSTRACT

The autonomous 6 axis face tracking system aims to develop a low cost, modular, open-source approach for Internet of Things (IoT) automation and security. The motivation of the research stems from widespread usage of facial recognition in current security, including on The George Washington University campus. The research presented utilizes minimal hardware, including a Raspberry Pi unit, camera, and positioning servos, as well as software from facial recognition libraries such as OpenCV. The final product yielded a working six-axis, servo-driven gimbal that has the ability to detect the nearest human face, dynamically re-center the camera on to the target, and upload the camera's video feed to a web interface. This technology has a broad range of potential applications and represents a significant contribution to security solutions, particularly in the context of emerging challenges posed by generative AI and deep fakes. The project successfully integrates computer vision, motion control, and IoT connectivity to achieve an autonomous security application that can be further researched in the creation of low-cost security solutions.

KEYWORDS

IoT, Edge Computing, Computer Vision, Security

1. Introduction

Autonomous facial recognition systems exist in nearly all modern security systems, as their implementation requires nothing more than a single camera and computer. There are various applications of these facial recognition systems, including security, defense, personnel verification, and more. However, there exists the common theme of prohibitive costs due to the companies and proprietary systems that are used for these systems. The motivation of the autonomous 6 axis face tracking system aims for a lower cost to remove barriers the security industry faces for the application of face tracking technology. This research aims to create a low cost, modular, and open-source approach the is replicable by those with lower budgets and little funding.

The device takes visual inputs from the camera and runs the OpenCV library to detect human faces, controlling the servo through calculations and centering the face, while simultaneously uploading the video to a local webpage. The project incorporates computer vision, motion control, and IoT connectivity to achieve autonomous operation. The primary challenge lies in real-time face recognition, camera alignment, and stable image capturing while ensuring smooth gimbal motion.

The future applications of the autonomous 6 axis face tracking system rely on the device's capability to automatically identify and following human subjects, with positive results potentially reshaping the way security, telepresence, and media are approached.

1.1. Background and Literature review

With advancements in automation and surveillance technology, face-tracking gimbals are crucial in applications such as security systems, autonomous photography, and assistive robotics.
This project addresses the challenge of dynamic facial tracking and seamless image upload using an IoT-based architecture. The system will leverage machine learning (ML) for facial recognition, active motion control for stabilization and centering, and a hosted webpage for image sharing/access.
Potential applications include:

- Home security - Intelligent cameras that follow intruders.
- Photography and videography - Hands-free framing and auto-tracking.
- Telepresence robots - Autonomous video conferencing.

This project will focus on optimized real-time processing to enhance gimbal accuracy and minimize latency. [1]

Several existing studies have explored face recognition with OpenCV and gimbal motion stabilization. Most commercial face-tracking gimbals rely on proprietary software with limited customization. [3][4]
Key comparisons:

- Traditional vs. AI-based tracking - Earlier systems relied on color/histogram tracking, while modern systems use deep learning models like Haar cascades, MTCNN, and YOLO.
- Hardware implementations - Commercial gimbals primarily use brushless motors and IMUs for stabilization. We will explore cost-effective servo-based alternatives.
- IoT integration - While security cameras often store data locally, IoT systems enable real-time image sharing on the edge.

Our system aims to bridge gaps in affordability, real-time response, and open-source flexibility by integrating low-cost sensors, Python-based ML models, and Raspberry Pi-based IoT processing.

2. Methods and Implementation

The autonomous 6 axis face tracking system is designed to utilize computer vision and mechanical movements to create a device capable of constant facial tracking.

2.A. Hardware and Software

The hardware utilized in the device includes a Raspberry Pi 4 Model B for computation, Raspberry Pi Camera Module for visual input and image acquisition, three servos for motion control, and a 3D Printed Mount. Overall, the "thing" consists of simple and inexpensive hardware and demonstrates the use cases in IoT and security applications.

The software utilized in the device has three layers consisting of facial recognition, motor control, and web application. The first layer of the system operates using the OpenCV python library and the Haar Feature-based Cascade Classifiers which is a machine learning based facial detection algorithm. Taking the output of the OpenCV library and applying a simple algorithm to control the servo it can aim the physical camera to the center of the face. The final layer takes the video feed and the algorithm and displays a live feed using the Flask micro web framework.

2.B. Implementation

The implementation of the device is simple, consisting of only a few components. The 3D printed mount acts as the connective frame, allowing various components to be attached and integrated. For the device to move, the servos must be installed onto the mount and connected to the GPIO pins of the Raspberry Pi. Each servo receives 5 volts for power and pulse width modulation signals for commands. The camera is installed on the mount in the central location, allowing it to rotate in six axes, i.e., pitch, roll, and yaw. All these components are integrated with one another using various inexpensive nuts, bolts, and screws.

The Raspberry Pi then acts as the computational brain of the device and allows for the software to control and determine the location of the faces and what duty cycle PWM signal must be supplied to the servo to create the necessary combination of angular movements to track the face. The Raspberry Pi is constantly running a python script that includes the OpenCV facial recognition library, an algorithm that takes the location of the face and determines how much the servo must adjust, and a section controlling and updating the Flask webpage.

2.C. IoT Layer / Structure

For the design, the 5-layer IoT model was used, as seen below.

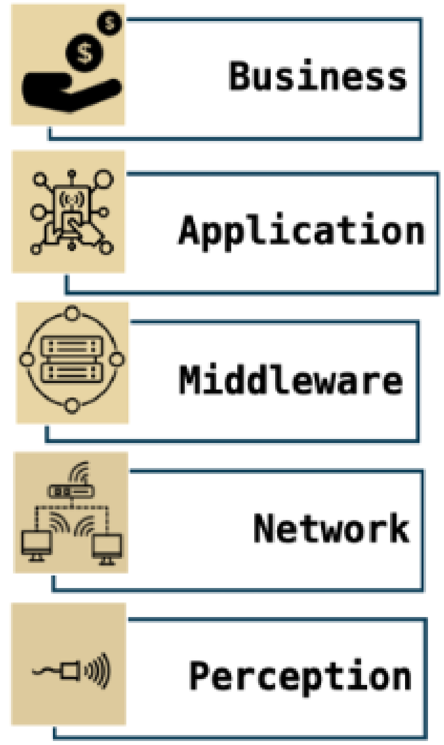

Fig. 1. IoT layer/architecture model (Created by Professor Bulusu)

The model consists of the perception layer, network layer, middleware layer, application layer, and business layer. The perception layer for this application involves the use of Raspberry Pi camera module and three servo motors for the sensing and actuation of the "thing". The network layer involves communication between devices using local Wi-Fi and HTTP/REST API for uploading the video feed and data to a Flask-based web server. The network layer should ensure seamless communication between the hardware, processing unit, and remote server while transmitting processed image data from the edge to the web interface. The middleware layer is responsible for data processing, taking the acquired image to obtain real-time facial recognition, controlling servo motor angles using algorithms and pulse width modulation, and image preprocessing. The application layer provides user interfacing by creating a Flask-based server which has a live video feed and a constantly updated webpage for image display. Finally, the business layer relates the entire project to real life applications such as the importance of this device in real-time autonomous surveillance, security, telepresence, and other industries. Overall, the 5-layer model demonstrates the steps taken to create the device and how it demonstrates the outlined low-cost, modular, open-source approach for IoT automation and how the insights gained may guide future product development or research in assistive robotics.

3. Critical Discussion of Results

Testing was done using the prototype where the created algorithm was run on the Raspberry Pi, requiring all computing to be done "at the edge". The demonstration proved a working model, where once the device determines the largest face (usually the closest) on the camera and runs an algorithm to determine how much the servo needs to be moved to center the face on the camera. The data generated from the camera image feed is then compared to machine learning models for determining faces. Once a face has been positioned in 2D space, the algorithm calculates the servo's new optimal angle, and supplies the corresponding PWM signal, centering the face. As a parallel process, the Flask website is created and hosted on the local server and takes the data of the camera feed to display a live video feed for those granted access.

The result of this prototype is an automatic face tracking device that has a dedicated webpage to display the video feed taken in the data gathering step. It was concluded that with minimal hardware and lightweight programs, one can create a device capable of automated facial recognition and movement tracking. The device utilizes edge computing, as each device can collect, process, and transmit the data in a completely self-contained environment. The overall findings of the project were the capabilities of low cost IoT devices in recreating high end security systems.

The main results and findings were as follows:

- Application: Low cost, modular, open-source approach for IoT security applications.
- Actionable data: Facial recognition software that can be scaled.
- Real time tracking: Device capable of tracking the nearest/largest target.
- Scalability: Can be installed in any location for any application at low cost.

Some issues that can be addressed in future versions of this device are latency in real-time tracking, servo jitter affecting stability, and slow image uploads. Some solutions to these issues are to optimize face detection with multi-threading, use Kaman filtering for noise reduction, and implement image compression & caching.

4. Conclusions

The proposed autonomous 6 axis face tracking device addressed the prohibitive costs and non-modularity found in modern day security devices. The IoT based system can continuously monitor any area and track face movement to help stabilize the image quality, which becomes blurry and has limited range with regular cameras.

This project presents an innovative IoT-enabled face-tracking gimbal that autonomously centers a camera on a detected face and uploads the image to a web platform. Unlike traditional face-tracking systems, our design emphasizes low-cost implementation, using open-source tools, real-time processing via edge computing, and seamless IoT integration for cloud-based image sharing. The purpose of this project is to advance face-tracking applications in surveillance, photography, and robotics, offering an affordable, customizable, and scalable solution.

New research questions presented after the completion of the prototype and initial testing revolved around the improvement of the system by adding more features as well as the application and tradeoffs introduced during the system's development.

The main conclusion from this device is the tradeoff of large-scale, dependent commercial solutions. This device is self-contained, low-cost, and focuses on edge computing to remain fully self-sufficient. Other important benefits are the use of cheap hardware such as smaller cameras and common servos so that the device can be recreated easily. While this device aims to create a cheaper alternative to a modular security system, it also introduces a variety of ethical concerns and compliance/regulatory issues, which should be included in future research.

ACKNOWLEDGMENTS

We are deeply grateful for the support provided by Dr. Kartik Bulusu, for his guidance, mentorship, and providing of the materials used in this project. His support has granted us the opportunity to pursue this project and guided us to understand the impact we can have. In addition, we would like to extend our appreciation to our fellow classmates who have provided insight and support. Without the full support of the George Washington University, this project could not have been completed successfully.

REFERENCES

[1] He, Kaiming, et al. "Deep residual learning for image recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, https://doi.org/10.1109/cvpr.2016.90.

[2] "SunFounder Focuses on Steam Education with Open-Source Robots." SunFounder, www.sunfounder.com/. Accessed 20 Mar. 2025.

[3] Szeliski, Richard. *Computer Vision: Algorithms and Applications*. Springer, 2011.

[4] Viola, P., and M. Jones. "Rapid object detection using a boosted cascade of Simple features." *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, https://doi.org/10.1109/cvpr.2001.990517.