**Final PROJECT PROPOSAL MAE6291: IoT for Engineers**

**Spring 2025, Instructor: Prof. Kartik Bulusu (MAE Department, GWU)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Final**  **PROJECT TITLE** | Autonomous 6-Axis Gimbal for Face Tracking and Web Upload | **Category: Circle One** | Grad  Undergrad |
| **NAME or NAMES** | Eliot Hunter, Shota Kakiuchi | **DATE** | 3/20/25 |

|  |
| --- |
| 1. **INTRODUCTION** Central problem being addressed; Topic of study related to problem **[1.0 Points]** |
| This project aims to develop a 6-axis gimbal system that autonomously recognizes a human face, centers a camera on the detected face, and captures an image. The system will then upload the final image to a webpage, enabling remote monitoring and data access.  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. |
| 1. **BACKGROUND AND SIGNIFICANCE** Problem details; Rationale; Problems addressed; Research methods and sources. **[1.0 Points]** |
| 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 face 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.   Our research will focus on optimized real-time processing to enhance gimbal accuracy and minimize latency. [1] |
| 1. **LITERATURE REVIEW** Cite, Compare, Contrast, Critique, Connect **[2.0 Points]** |
| 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. |
| 1. **PROJECT DESIGN AND METHODS** Research operations and result interpretation methodology argument; Potential obstacles**[1.5 Points]** |
| 1. System Architecture  The project follows a three-layer IoT architecture:   * Perception Layer (Sensing & Tracking):   + **Camera module** (Raspberry Pi Camera)   + Face detection via OpenCV & Haar cascades   + **3x servos** for sensor for gimbal stabilization [2]   + **MPU6050** for camera stabilization * Edge Computing Layer (Processing & Control):   + Raspberry Pi handles face recognition, motor control, and web communication   + PID control for gimbal stabilization   + Image processing using OpenCV & TensorFlow Lite * Application Layer (Web Upload & Display):   + Captured images are sent to a locally hosted webpage   + Web interface displays the latest captured image, perhaps also servo angles   2. Motion Control Strategy   * Servo-driven 6-axis gimbal for pan, tilt, and roll control * Possible PID tuning for smooth movement and face centering * Face tracking adjusts gimbal position to keep the face in the center   3. Image Processing & Uploading   * Face recognition using Haar cascades and/or dlib * Image compression to optimize upload speed * Web-based image repository using Flask   4. Obstacles Proposed Solutions  Latency in real-time tracking Optimize face detection with multi-threading  Servo jitter affecting stability Use Kalman filtering for noise reduction  Slow image uploads Implement image compression & caching |
| 1. **PRELIMINARY SUPPOSITIONS AND IMPLICATIONS** Task division; Identify who is doing what if you are in a team of two **[2.0 Points]** |
| |  |  |  | | --- | --- | --- | | **Eliot Hunter:** | **Hardware Engineer** | **Gimbal design, motor control, IMU integration** |  |  |  |  | | --- | --- | --- | | **Shota Kakiuchi:** | **Software Developer** | **Face recognition, OpenCV integration, web API** | |
| 1. **SUMMARY** Why is this problem worth addressing; Why this problem is unique and how it advances existing knowledge **[1.0 Points]** |
| 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 - Seamless IoT integration for cloud-based image sharing  Successful execution of this project will advance face-tracking applications in surveillance, photography, and robotics, offering an affordable, customizable, and scalable solution. |
| 1. **CITATIONS** References; Bibliography **[1.5 Points]** |
| 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. |