浙江大學

Project Proposal

Group 3

Group Members and IDs:

Pan Yue（3220100912）

Wang Haoyu（3220105740）

Hu Jiyuan（3220104116）

Date: October 1_{th}, 2024

Teacher: Wu Hongzhi

TA : Fei Fan

Table of Contents

1 Introduction

In the context of the rapid development of modern visual information processing technologies, object tracking has become a significant research direction in the field of computer vision. With the rapid advancements in technologies such as drones, intelligent surveillance, and autonomous driving, achieving efficient and stable object tracking has emerged as a key challenge. This project aims to develop an intelligent camera-assisted device that integrates the MPU6050 6-axis motion sensor with the Kernelized Correlation Filters (KCF) algorithm to achieve accurate and efficient object tracking.

By combining physical devices with visual information processing algorithms, our device will be capable of tracking moving objects in dynamic environments. This technology can be applied not only to areas such as security monitoring, logistics tracking, and drone navigation, but also provide foundational support for intelligent robots and augmented reality applications.

图 1: proposal effect demonstration

2 Related Work

2.1 Object Tracking Technology

- **KCF Algorithm**: KCF is an object tracking algorithm based on correlation filters. Its advantages include fast processing speed and robustness to scale changes of the target. The algorithm establishes a model of the target's appearance and continuously updates the target's position in real time.

- **Real-time end-to-end object detection**: YOLO is a real-time object detection and tracking algorithm that uses a single neural network to predict bounding boxes and class probabilities directly from full images. Its key advantages include high processing speed and the ability to detect multiple objects in a single pass, making it suitable for real-time applications. YOLO divides the input image into a grid and assigns detection tasks to each grid cell, allowing for efficient localization and classification of objects. The algorithm is capable of processing video streams in real time, making it ideal for applications in autonomous driving, surveillance, and robotics.

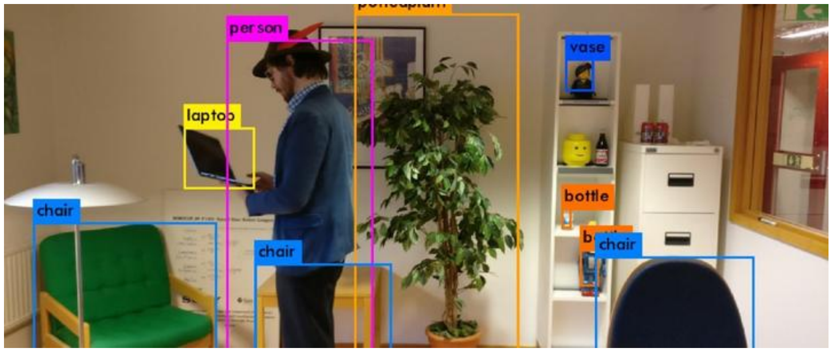

图 2: YOLO application

2.2 Sensor Integration

- **MPU6050 Sensor**: The MPU6050 is a high-performance 6-axis motion sensor that provides accelerometer and gyroscope data. It is widely used in fields such as drones and robotics.
- **Gimbal System**: A gimbal stabilizes the camera in dynamic environments, improving the stability of video capture. The design principles include control mechanisms with three degrees of freedom and motion compensation.

3 Our Approach

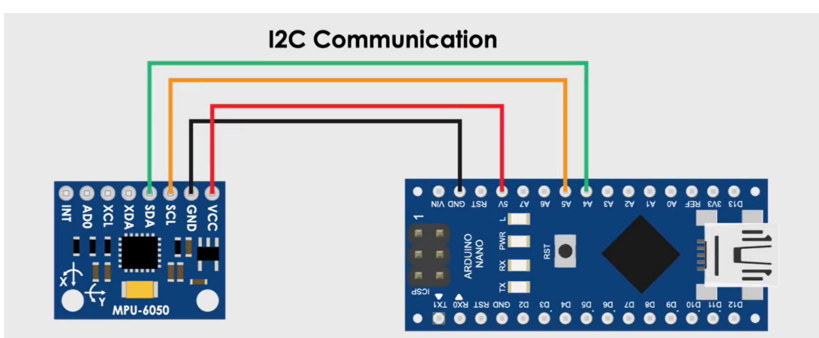

图 3: Integrating Arudino Nano with MPU-6050

3.1 Hardware

- **Basler Industrial Camera**: Used for capturing video streams, providing high-resolution imaging and fast frame rates essential for real-time tracking applications.

- **MPU6050 Sensor**: A high-performance 6-axis motion sensor that provides accelerometer and gyroscope data. This data is crucial for obtaining the device's orientation and compensating for any unwanted movement during tracking.

- **Microcontroller**: Options include Arduino or Raspberry Pi, which are utilized to process sensor data and control the gimbal mechanism. The microcontroller interfaces with the MPU6050 to read motion data and with the camera for synchronized tracking operations.

- **Gimbal Stabilization System**: This system integrates with the microcontroller and camera to ensure smooth, stable video capture even in dynamic environments. It adjusts the camera's position in real-time based on the MPU6050's feedback, maintaining the target within the frame.

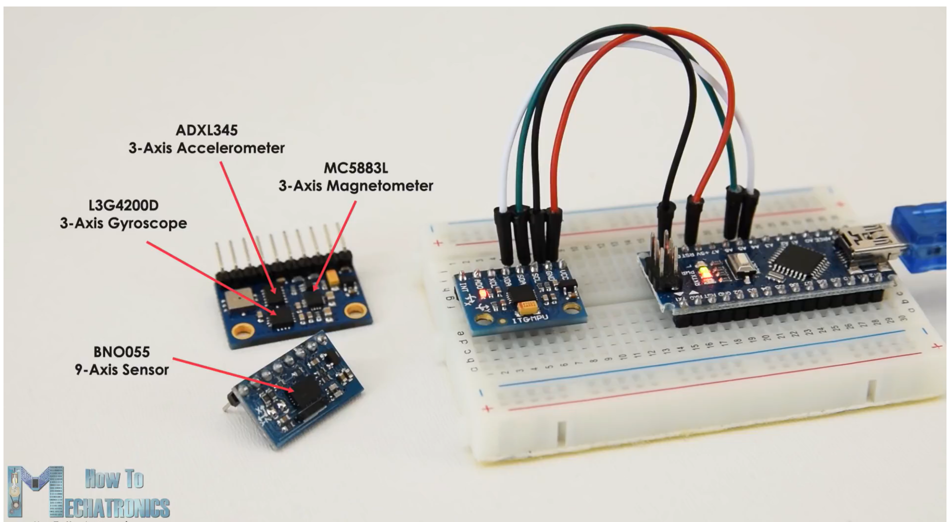

图 4: Details of the MPU-6050 chip

3.2 Algorithm Implementation

3.2.1 KCF Algorithm

1. **Initialization**: In the initial frame, identify the target's bounding box and extract its HOG features. Generate positive and negative samples through cyclic shifting of the entire image, creating a relevant training dataset for the correlation filter.

2. **Frame Processing**: For each subsequent frame, treat the entire image as the search area. Extract HOG features and compute a response map by multiplying the detected sample with the correlation filter template. Use inverse Fourier transform to locate the optimal position, updating the target's appearance model accordingly.

3. **Prediction and Update**: Utilize the updated filter classifier template to predict the target' s position in the following frames. Repeat this process until all video sequences are processed.

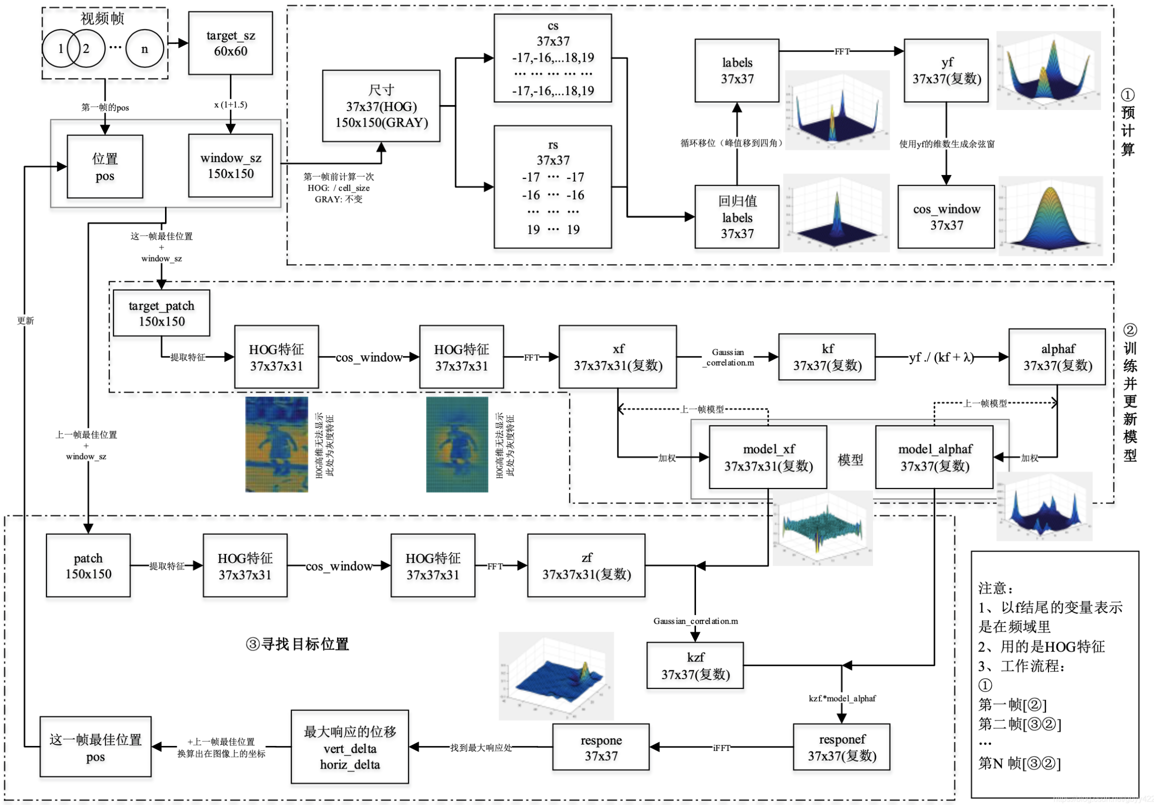

图 5: KCF algorithm specification

3.2.2 Data Fusion

- **Sensor Integration**: Combine the MPU6050's orientation data with image data to compensate for camera shake in dynamic environments.
- **Kalman Filtering**: Implement Kalman filtering for improved prediction of the target's motion trajectory.

3.3 Gesture Control

- **Gesture Recognition**: Use computer vision libraries (e.g., OpenCV, MediaPipe) to implement real-time hand gesture recognition for starting and stopping tracking.
- **Target Selection**: Recognize gestures (e.g., pointing) to select and track specific targets.

3.4 Functionality Expansion

- **Multi-Target Tracking**: Develop a module to enable simultaneous tracking of multiple objects within the same frame.
- **Object Recognition**: Integrate deep learning models (e.g., YOLO or SSD) to enhance the system' s ability to recognize and track objects intelligently.

3.5 Testing and Evaluation

- **Experimental Conditions**: Conduct tests under various environmental conditions, including outdoor, indoor, and low-light scenarios, to assess system stability and accuracy.
- **Performance Analysis**: Record and analyze the device's performance in complex backgrounds to optimize algorithms and hardware configurations for better reliability and efficiency.

4 Milestone/Timeline

Week 5: Conduct literature research to understand the current state of tracking algorithms and sensor integration.

Week 6: Design and assemble the hardware, including the camera, gimbal, and MPU6050.

Week 7: Implement the real-time object tracking algorithm and integrate it with the hardware.

Week 8: Perform system testing, calibration, and final evaluation, adjusting algorithm parameters as needed, followed by preparation of the project report.

5 References

[1] Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2544-2550. https://doi.org/10.1109/CVPR.2010.5539960

[2] Gadd, J. (2015). Design and implementation of a gimbal stabilizer. *International Journal of Robotics Research*, 34(4-5), 627-643. https://doi.org/10.1177/0278364914558494

[3] Lee, J., & Kim, Y. (2016). Sensor fusion for MPU6050. *Journal of Sensors*, 2016, Article ID 8341672. https://doi.org/10.1155/2016/8341672

[4] O. M. Y. S. (2018). Multi-target tracking using deep learning techniques. *Pattern Recognition Letters*, 120, 83-88. https://doi.org/10.1016/j.patrec.2018.01.010

[5] HowToMechatronics. (n.d.). Arduino and MPU6050 accelerometer and gyroscope tutorial. Retrieved from https://howtomechatronics.com/tutorials/arduino

[6] Ultralytics. (2023). YOLOv10: Latest enhancements in object detection using YOLO architecture. Available at: https://github.com/ultralytics/yolov10.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, Mar. 2015. https://doi.org/10.1109/TPAMI.2014.2345390.