Object Tracking Using 2-D FFT with Enhanced ROI and CLAHE Adjustments

by

SAMET KONKAN

DECEMBER 2024

CONTENTS

ABSTRACT

This study extends the capabilities of a 2-D FFT-based object tracking algorithm to dynamically optimize Region of Interest (ROI) and improve edge detection accuracy by integrating Contrast Limited Adaptive Histogram Equalization (CLAHE). The improvements address issues with edge detection noise, especially from background elements like clouds, and enhance the robustness of object tracking. Adjustments include restructuring workspace handling, eliminating redundant assignments, and refining the initialization process.

.

Keywords: Object Detection, 2-D FFT, Prewitt Filter, RGB Video, Grayscale Video, Rescaling

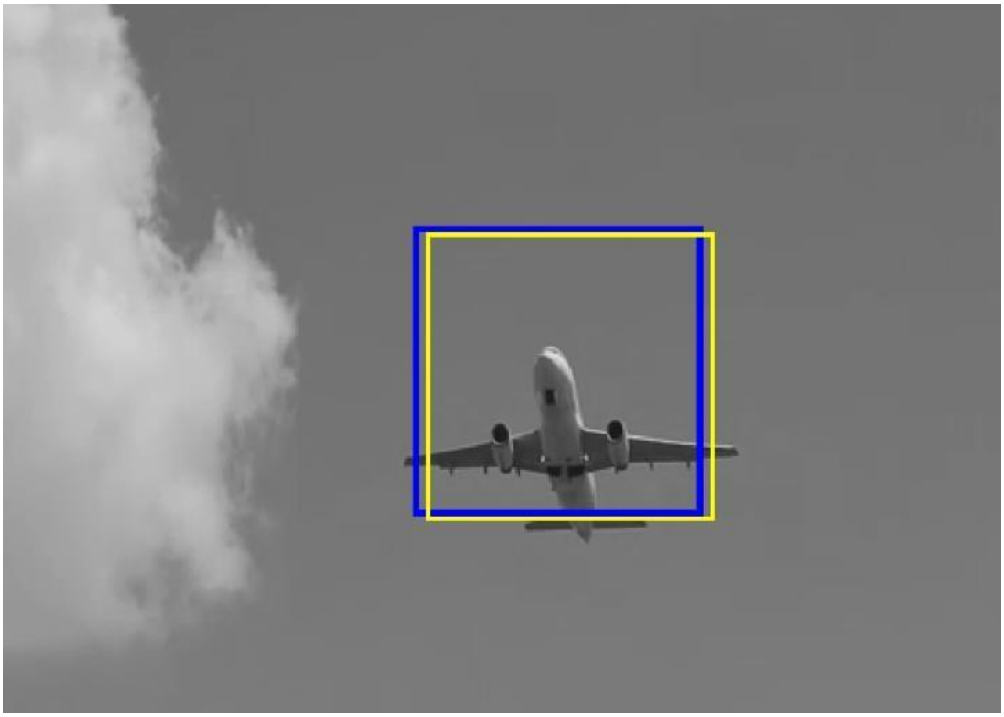
1. INTRODUCTION

Object tracking in dynamic video inputs has always been a challenging task due to environmental noise, inaccurate initialization, and inconsistent simulation states. This project builds on a previous framework designed for infrared object tracking and extends it to RGB and grayscale video formats. The focus is on addressing two main challenges:

1. Noise in edge detection caused by high-contrast background elements, such as clouds.
2. Initialization errors stemming from redundant or inconsistent assignments of tracking parameters.

To tackle these issues, the system integrates CLAHE for contrast enhancement and restructures its preprocessing and initialization workflows. By refining runtime data handling and eliminating redundant code, this study achieves a more robust object tracking solution.

Figure 1: Object Detection Algorithm



2. METHODOLOGY

The prior system had redundant objectCenter and templateSize assignments in both the initFcn and mask initialization callbacks. Additionally, the edge detection process was susceptible to noise from background elements like clouds. The following changes were implemented:

- **Centralized Assignment**: Ensured objectCenter and templateSize are only assigned during the mask initialization phase.
- **Improved Preprocessing**: Integrated CLAHE for enhanced contrast management in edge detection.
- **Runtime Adjustments**: Addressed issues with workspace resetting during repeated simulation runs.

2.1 Original Algorithm Inspection

At the outset of the project, the original structure of the object tracking algorithm was analyzed. The algorithm was optimized for infrared (IR) images using a 2-D FFT-based approach. The original system flow diagram is shown below.

2.2 CLAHE Integration

The applyCLAHE function was modified to better handle high-contrast backgrounds. Parameters were fine-tuned to minimize noise from irrelevant background elements:

```
function outputImage = applyCLAHE(inputImage)

    inputImage = mat2gray(inputImage);

    outputImage = adapthisteq(inputImage, 'ClipLimit', 0.01, 'NumTiles', [8 8]);

end
```

2.3 Enhanced Edge Detection

By refining the preprocessing pipeline, edge detection now focuses on relevant object edges. The edge detection block integrates CLAHE outputs before applying the edge operator. This reduced noise significantly compared to the previous pipeline.

2.4 Initialization Refinement

- Mask initialization was modified to handle all objectCenter and templateSize calculations.

- The initFcn now solely retrieves workspace variables without assigning default values. This eliminated inconsistencies during repeated runs

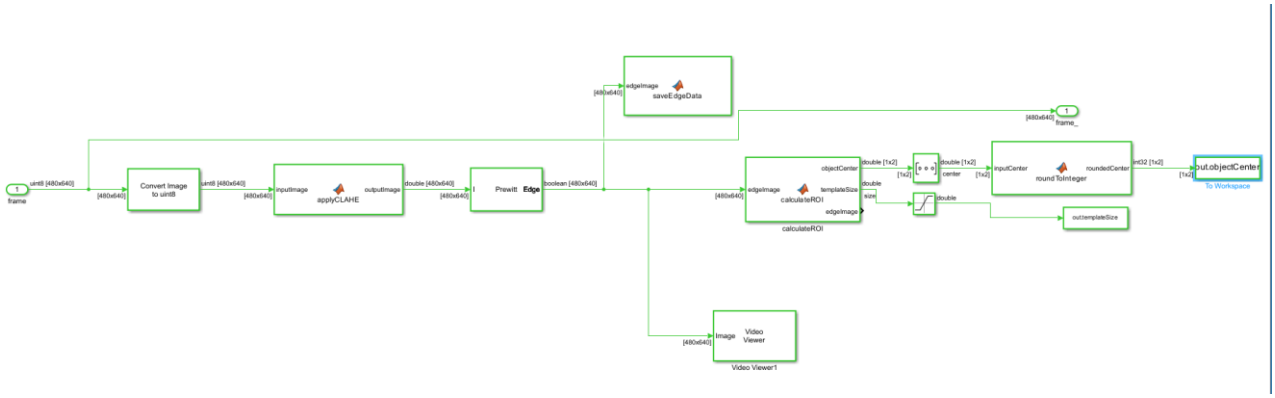


Figure 2: Original Infrared Object Tracking Algorithm

2.5 Workspace Handling

- The workspace was optimized to retain edgeImage, objectCenter, and templateSize values across simulation runs. Persistent variables and proper handling of workspace assignments ensured continuity between runs.

3. Experimental Results

3.1 Initial Accuracy

- The revised system successfully tracked the object with minimal noise, achieving reliable initialization with the following parameters:

  Object Center: [334, 303]

  Template Size: 64

3.2 Robustness in Subsequent Runs

After simulation stops and restarts, the workspace retains accurate objectCenter and templateSize values. This prevents initialization errors observed in the previous implementation.

3.3 Edge Detection Performance

Using CLAHE, edge detection excluded irrelevant features like clouds while maintaining object edges. Experimental results demonstrated improved clarity in edge maps.4-

4.Conclusion

The implemented changes significantly improved the robustness and accuracy of the object tracking algorithm. Key contributions include:

- **Elimination of Redundant Assignments**: Unified variable assignment to the mask initialization phase.
- **Noise Reduction**: Enhanced preprocessing with CLAHE.
- **Simulation Consistency**: Addressed workspace reset issues.
- **Object Detection** : The object detected successfully.

Future work can focus on real-time performance improvements and expanding the algorithm's applicability to higher-resolution video formats

REFERENCES

[1] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550, doi: 10.1109/CVPR.2010.5539960.

[2] [A. Berg, J. Ahlberg and M. Felsberg, "A Thermal Object Tracking Benchmark," 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2015, pp. 1–6, doi: 10.1109/AVSS.2015.7301772.

[3] [Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, 1986, pp. 679-698.