OSTİM TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

EEE 309 – SIGNALS AND SYSTEMS

INSTRUCTOR: ASSOC. PROF. DR. AHMET GÜNGÖR PAKFİLİZ

HOMEWORK REPORT

CANAN KILIÇ

24.11.2024

INTRODUCTION

The aim of this project is to detect and track an airborne vehicle while it is flying in the sky. This aim is very important and useful, especially in the defense industry. It is also used at the borders of countries to detect any airborne devices that may pose a threat to national security. Tracking airborne vehicles can help improve surveillance and provide early warnings against possible risks. This technology can also be applied in search and rescue missions to locate missing aircraft or drones in critical situations.

TRACKING AIRBORNE VEHICLE CODE

```
%Airborne Vehicle Tracking Homework 220201037 Canan Kılıç

%Initialize Video Reader
airplane_video = 'resized_video3.mp4';
read_video_func = VideoReader(airplane_video);

%Initialize Video Writer
tracked_airplane_video = 'tracked_airplane_son.avi';
write_video = VideoWriter(tracked_airplane_video, 'Motion JPEG AVI');
write_video.FrameRate = read_video_func.FrameRate;
open(write_video);

%I used Kalman filter for tracking
kalman_filter = configureKalmanFilter('ConstantVelocity', [0, 0], [1.1, 1.1], [1.1, 1.1], 0.1);
isInitialized = false;

while hasFrame(read_video_func)
    frame = readFrame(read_video_func);

    %Convert to grayscale for easy detection
    gray_frame = rgb2gray(frame);

    dynamic_threshold = mean(gray_frame(:)) - 40;
    binary_mask = gray_frame < dynamic_threshold;

    %Detect regions
    regions_find = regionprops(binary_mask, 'Area', 'BoundingBox', 'Centroid', 'MajorAxisLength',
'MinorAxisLength');
    object_regions = []; %Initialize as an empty numeric array

    for i = 1:length(regions_find)
        region = regions_find(i);
        centroid = region.Centroid;
        bbox = region.BoundingBox;
        aspectRatio = region.MajorAxisLength / max(region.MinorAxisLength, 1);
        minAspectRatio = 1.2;
        maxAspectRatio = 6;
        minArea = 20;

        if centroid(2) < read_video_func.Height * 0.8 && ...
           aspectRatio > minAspectRatio && aspectRatio < maxAspectRatio && ...
```

```
            region.Area > minArea
                object_regions = [object_regions, i]; %Append the index
            end
        end

        if ~isempty(object_regions)
            [~, largestIndex] = max([regions_find(object_regions).Area]);
            largest_region = regions_find(object_regions(largestIndex));

            bbox = largest_region.BoundingBox;
            centroid = largest_region.Centroid;

            if isInitialized
                predict(kalman_filter);
                tracked_location = correct(kalman_filter, centroid);
            else
                tracked_location = centroid;
                isInitialized = true;
            end

            annotated_frame = insertShape(frame, 'rectangle', bbox, 'Color', 'red', 'LineWidth', 2);
            annotated_frame = insertMarker(annotated_frame, tracked_location, 'Color', 'red', 'Size', 10);
        else
            if isInitialized
                tracked_location = predict(kalman_filter);
                annotated_frame = frame;
            end
        end

        writeVideo(write_video, annotated_frame);
    end

    close(write_video);
```

BACKGROUND METHODS

Video Processing:

- Videos consist of frames. Analyzing the video frame by frame sequentially gives an opportunity to detect the airplane consistently.
- Converting RGB scale (red, green, blue) to grayscale helps the program differentiate other objects from the airplane. Since grayscale has only one-dimensional color information, the program can easily perceive differences.
- Thresholding is used to separate objects in the frame. It determines the outlines of objects to isolate them from the background.
- Bounding box (BBox) is used in the code to detect the airplane's outlines. The regionprops function calculates the bounding box dimensions for each detected region in the binary mask. This box surrounds the airplane and shows its position clearly. It helps the program focus on the airplane and ignore unrelated areas.
- The regionprops function is used in the code to measure certain properties, such as:

regionprops(binary_mask, 'Area', 'BoundingBox', 'Centroid', 'MajorAxisLength', 'MinorAxisLength');

Kalman Filtering:

- Kalman filtering predicts an object's future position based on its velocity and previous position. It is implemented in this project to predict the airplane's next position and eliminate static objects (e.g., in the video, there is a signboard).
- The syntax is:

kalmanFilter = configureKalmanFilter(MotionModel, InitialLocation, InitialEstimateError, MotionNoise, MeasurementNoise);

- In this project, the properties of the Kalman filter are:
  Initial state estimate: [0, 0] (position)
  Process noise: [1.1, 1.1]
  Measurement noise: [1.1, 1.1]
  Motion noise covariance: 0.1

Output Video:

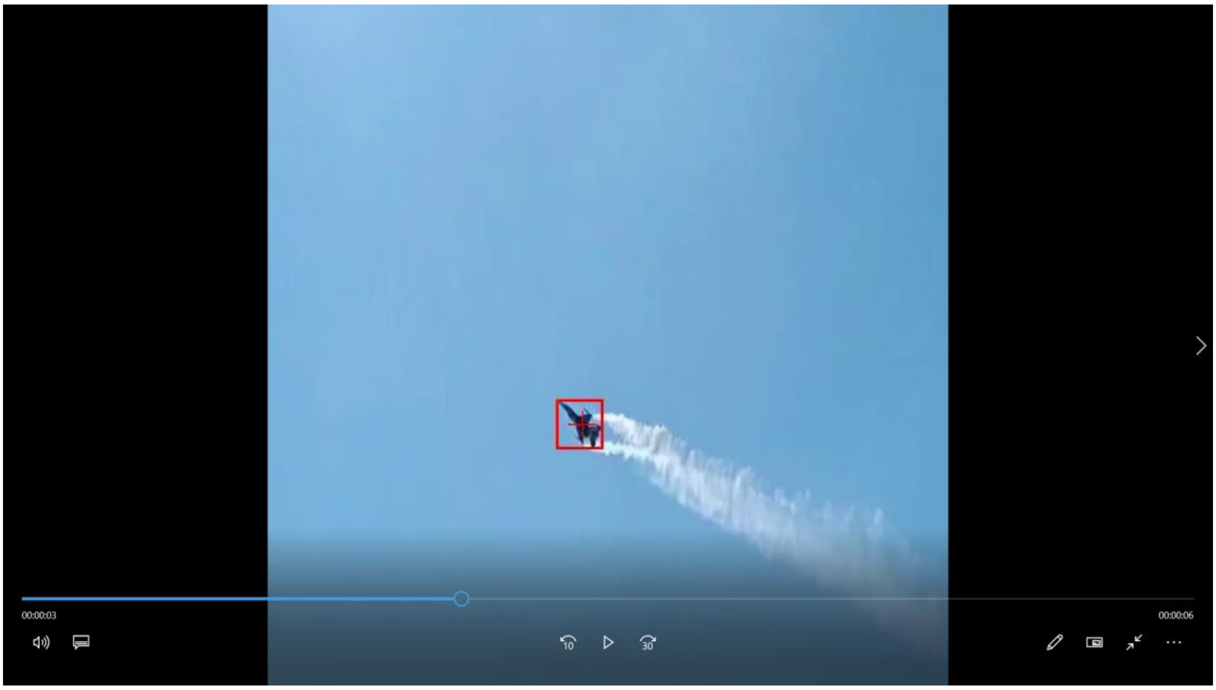
- The output video highlights the detected airplane's location in each frame with a red rectangular box. This visual feedback makes it easy to track the airplane throughout the video.

RESULTS

In this project, potential airplane regions were detected in each frame using Kalman Filtering. It tracks the airplane throughout the video.

A screenshot of the tracked airplane is included in this report. In the frame, there are an airplane, clouds, and a signboard. The tracking is successful even though there are other objects presents in the frame. The program successfully separates the airplane from constant (not moving) objects like the signboard and moving objects like the clouds.The red rectangular box marks the airplane's position clearly, and its location updates accurately in every frame. The tracking remains consistent even when the airplane moves or changes position in the frame. This result shows that the method works well for detecting and tracking airborne vehicles.


00:00:00
00:00:09


00:00:03
00:00:06

REFERENCES

MathWorks. Regionprops Function. Available at: https://www.mathworks.com/help/images/ref/regionprops.html. Accessed November 2024.

MathWorks. ConfigureKalmanFilter Function. Available at: https://www.mathworks.com/help/vision/ref/configurekalmanfilter.html. Accessed November 2024.

GeeksforGeeks. Sobel Edge Detection vs. Canny Edge Detection in Computer Vision. Available at: https://www.geeksforgeeks.org/sobel-edge-detection-vs-canny-edge-detection-in-computer-vision/. Accessed November 2024.

LogicMonitor. Static Thresholds vs. Dynamic Thresholds. Available at: https://www.logicmonitor.com/blog/static-thresholds-vs-dynamic-thresholds#h-when-to-choose-dynamic-thresholds. Accessed November 2024.

Stack Overflow. Change Movie Dimensions in MATLAB. Available at: https://stackoverflow.com/questions/13121889/change-movie-dimensions-in-matlab. Accessed November 2024.

OpenAI. ChatGPT Tool. Available at: https://chatgpt.com. Accessed November 2024.

Claude AI. Generative AI Tool. Available at: https://claude.ai. Accessed November 2024.