CMSC 691
Proj 2 Due Mar 25th
Feature Point Matching

Overview

Your objective is to create an interactive application using feature point matching to:

1. display a video (shown as a series of frames) as well
2. allow the user to “drag a box” on the video
3. Display the feature points (using any feature point detector in opencv 4.2)
4. recalculate for each frame (using feature point matching / RANSAC) the orientation of the box

You must write the application in c++

You are allowed to use opencv 4.2

Requirements

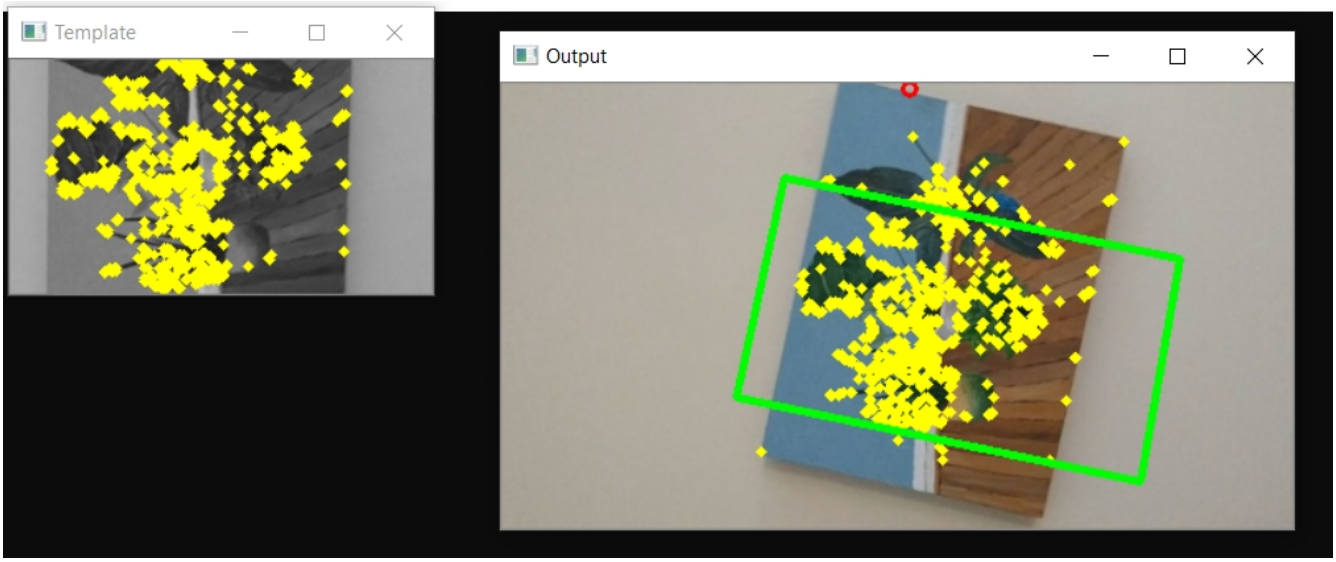
r.1 Your output must reasonably resemble the “proj2_sample_output.avi” video

r.2 Your program must run using the images from the “pictureframes” dataset

r.3 Your program must display the video, output box, and feature points in a named window called “Output”

r.4 Your program must interactively allow the user to “drag a box” over the output image in order to snag a “template” image for matching

r.5 Your program must display the “snagged template” as well as featurepoints over the snagged template in a separate named window called “Template”



(note: this is not shown in the sample video, example screenshot is attached)

r.6 Your program must identify the orientation of the pictureframe so long as the pictureframe is fully in view of the scene.

r.7 Your program must run at an interactive framerate on an i5 processor

Grading Rubrick

50 points for output, 50 points for algorithm

Output (50 points)

-50 Code does not compile and cannot be easily fixed
-20 Code does not compile but can be fixed easily
-35 Output does not reasonably match sample output in ability to detect picture frame most of the time once template is snagged
-25 Output does not correctly identify the orientation of the picture frame and draw this orintation with an affine transformed box
-25 Output does not display feature points over main window
-25 Output does not allow user to drag a box to snag a template
-25 Output does not display visualization window with feature points

Algorithm (50 points)

-20 Algorithm does not attempt use feature points
-15 Algorithm does not use RANSAC
-15 Algorithm does not properly identify affine transformation

Late Policy
If the project is submitted late there will be a 25 point deduction for each week that the project is late.
This deduction is effective immediately once the project is past due, and the submitter will have exactly one week before an additional point deduction is placed upon the project.