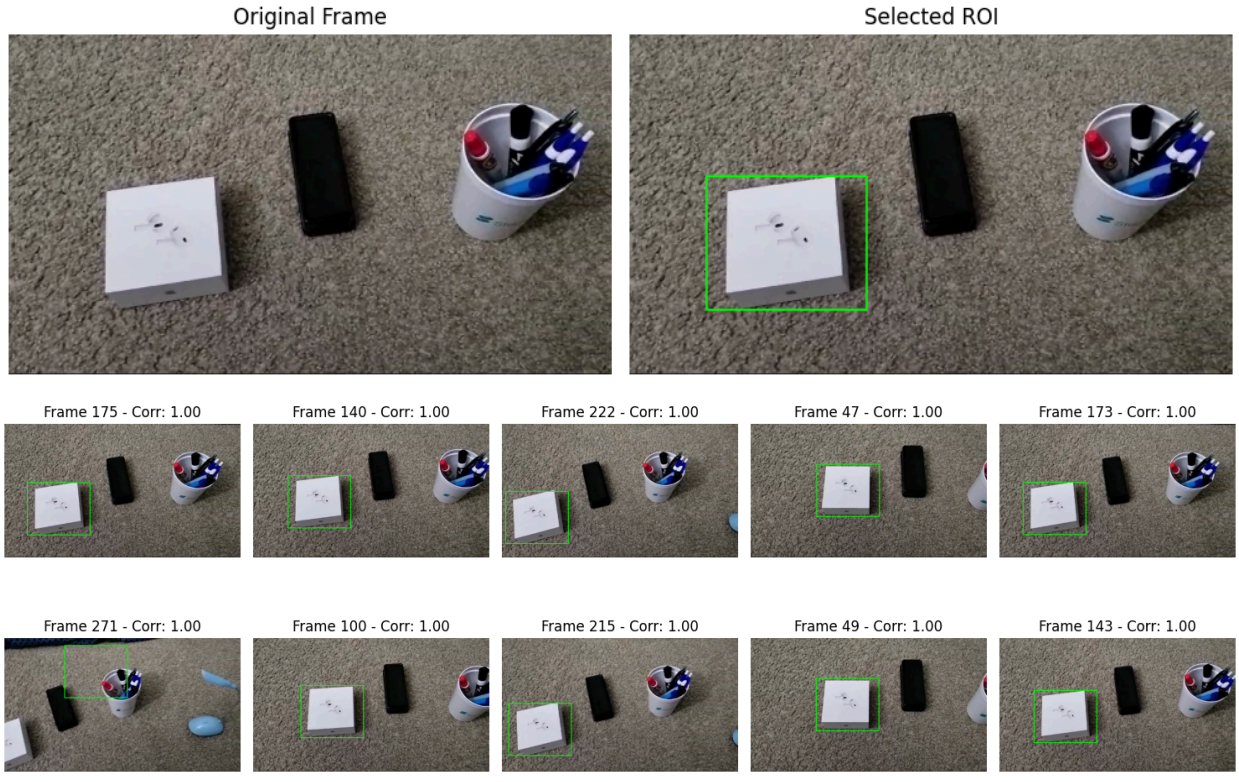
CSc 8830: CV Assignment-3 Solutions Report

1. **Capture a 10 sec video footage using a camera of your choice. The footage should be taken with the camera in hand and you need to pan the camera slightly from left-right or right-left during the 10 sec duration. Pick any image frame from the 10 sec video footage. Pick a region of interest corresponding to an object in the image. Crop this region from the image. Then use this cropped region to compare with randomly picked 10 images in the dataset of 10 sec video frames, to see if there is a match for the object in the scenes from the 10 images. For comparison use sum of squared differences (SSD) or normalized correlation.**

   `A 10 sec video has been taken from a mobile phone camera`



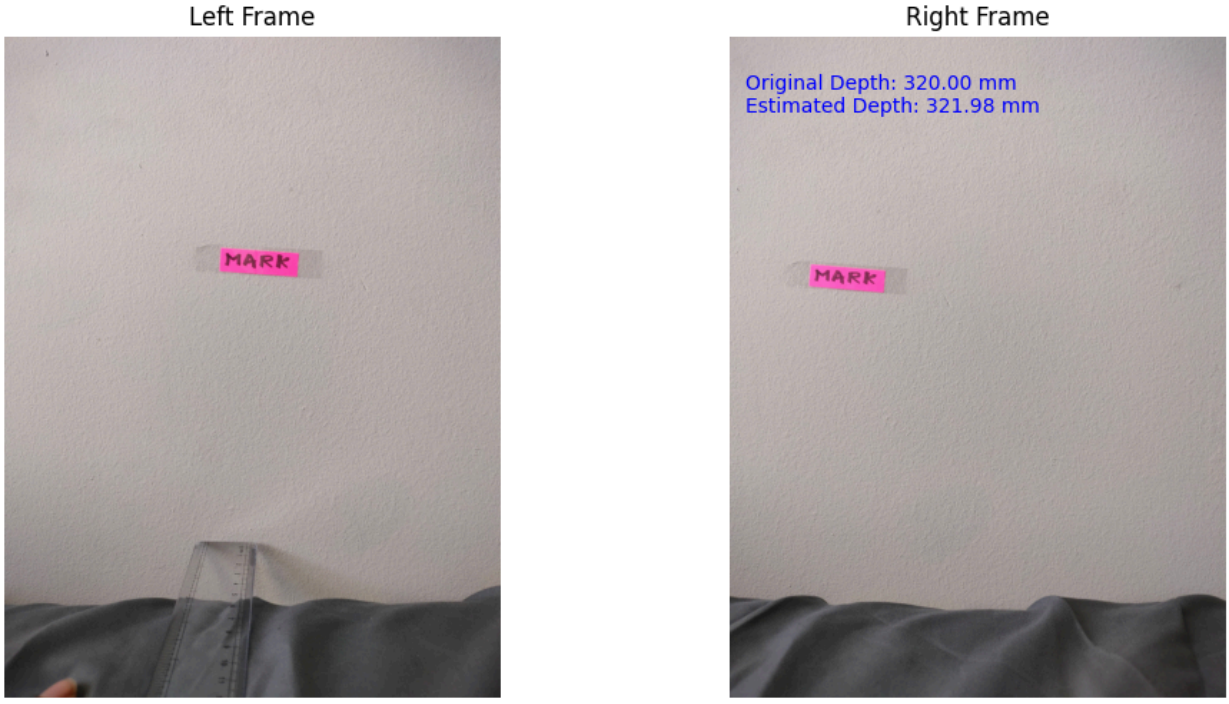
2. Solve the following by hand (on paper or typed: Do not just copy it from literature)

(a) Derive the motion tracking equation from fundamental principles. Select any 2 consecutive frames from the set from problem 1 and compute the motion function estimates.

(b). Derive the procedure for performing Lucas-Kanade algorithm for motion tracking when the motion is known to be affine: u(x,y) = a1*x + b1*y + c1; v(x,y) = a2*x + b2*y + c2 (the numbers are subscripts, not power)

Attached Solution PDF

3. Fix a marker on a wall or a flat vertical surface. From a distance D, keeping the camera stationed static (not handheld and mounted on a tripod or placed on a flat surface), capture an image such that the marker is registered. Then translate the camera by T units along the axis parallel to the ground (horizontal) and then capture another image, with the marker being registered. Compute D using disparity based depth estimation in stereo-vision theory. ***(Note: you can pick any value for D and T. Keep in mind that T cannot be large as the marker may get out of view. Of course this depends on D)***

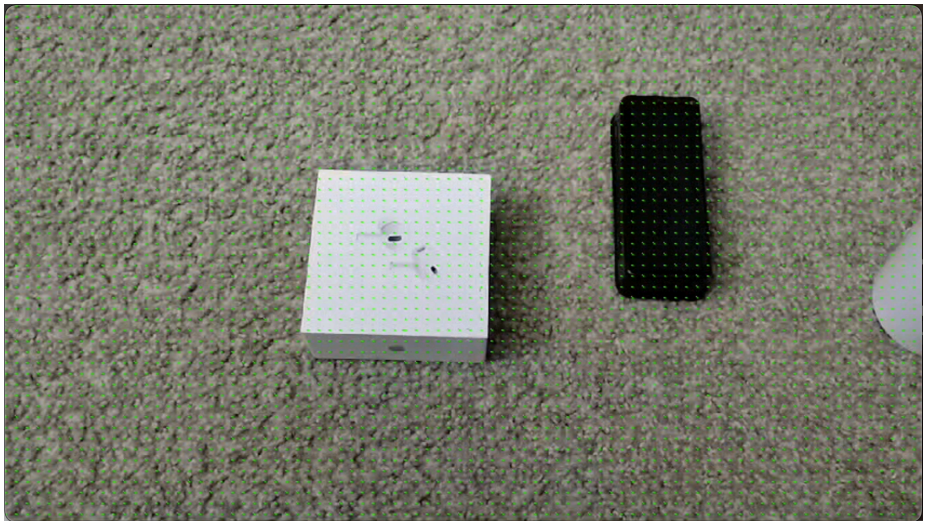



```
Original distance D: 320.0 mm
Estimated distance D: 321.98240319726756 mm
```

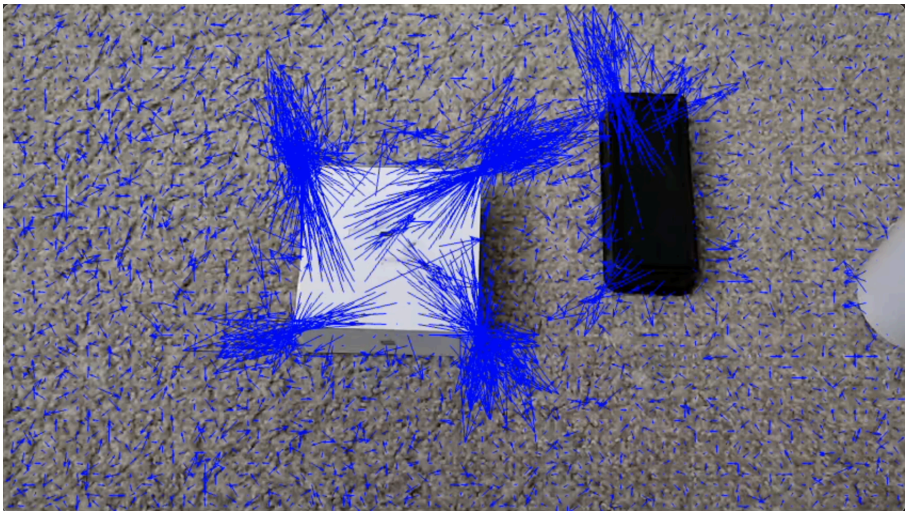
4. For the video (problem 1) you have taken, plot the optical flow vectors on each frame using MATLAB's optical flow codes. (i) treating every previous frame as a reference frame (ii) treating every 11th frame as a reference frame (iii) treating every 31st frame as a reference frame

This question is solved using both Python & Matlab

Python:

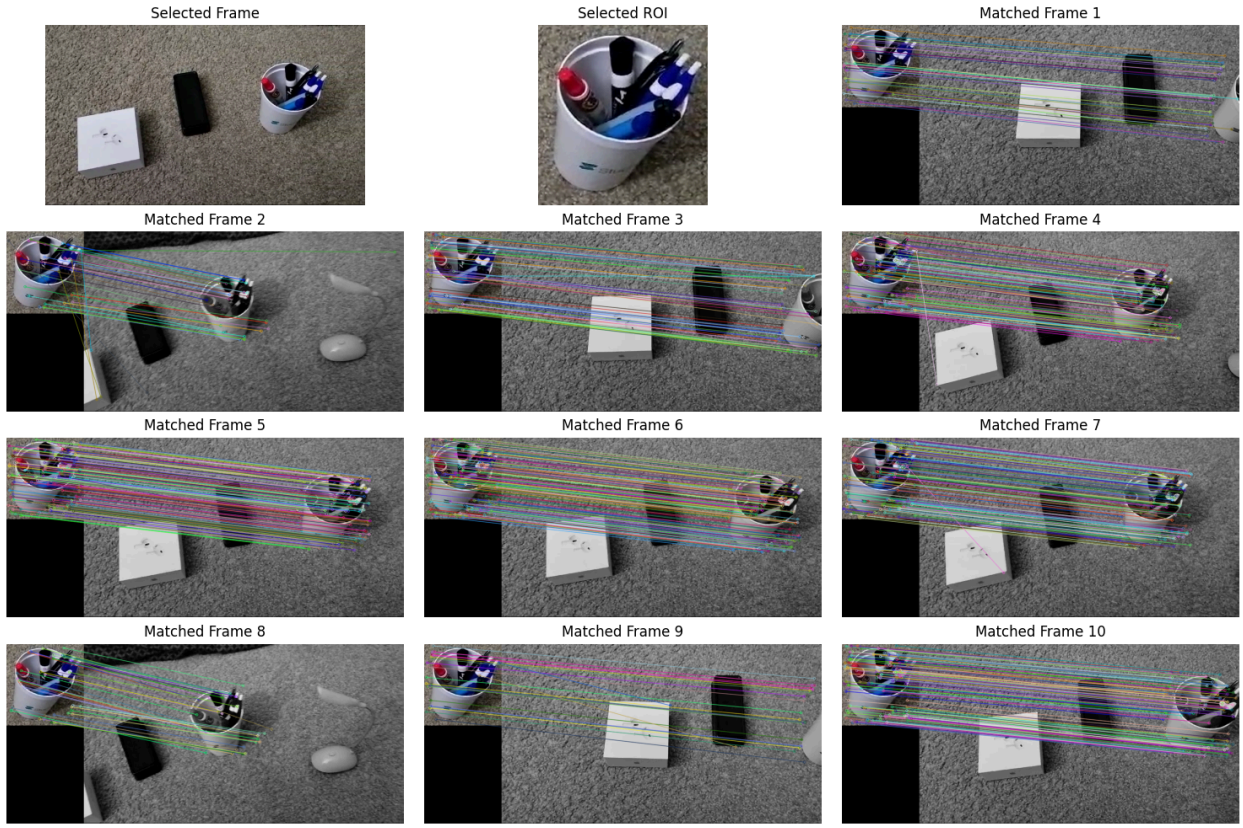


Matlab:



5. Run the feature-based matching object detection on the images from problem (1). ***MATLAB (not mandatory for this problem)*** *Tutorial for feature-based matching object detection is available here:* ***https://www.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html***

This is Solved using Python

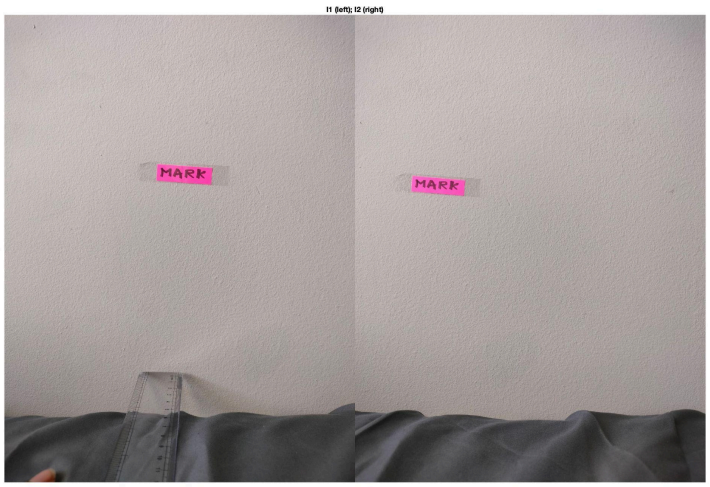


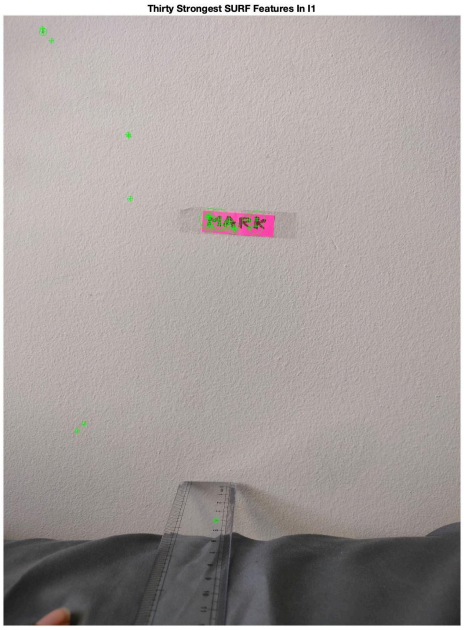
6. Refer to the Bag of Features example MATLAB source code provided in the classroom's classwork page. In your homework, pick an object category that would be commonly seen in any household (e.g. cutlery) and pick 5 object types (e.g. for cutlery pick spoon, fork, butter knife, cutting knife, ladle). Present your performance evaluation.

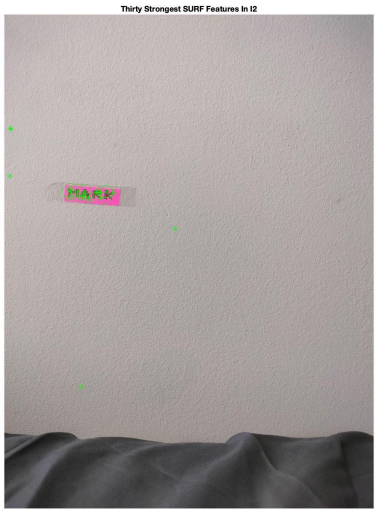
```
Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.

Encoding images using Bag-Of-Features.
----------------------------------------
* Encoding an image...done.
Accuracy: 1
```

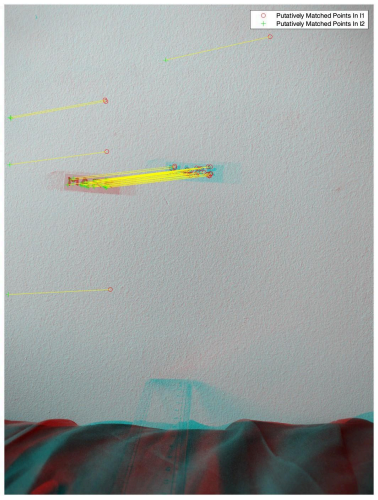
7. Repeat the image capture experiment from problem (3), however, now also rotate (along the ground plane) the camera 2 (right camera) towards camera 1 position, after translation by T. Make sure the marker is within view. Note down the rotation angle. Run the tutorial provided for uncalibrated stereo rectification in here: https://www.mathworks.com/help/vision/ug/uncalibrated-stereo-image-rectification.html (MATLAB is mandatory for this exercise). Exercise this tutorial for the image pairs you have captured. You can make assumptions as necessary, however, justify them in your answers/description. ***(Note: you can print out protractors from any online source and place your cameras on that when running experiments: http://www.ossmann.com/protractor/conventional-protractor.pdf).***

Rotation Angle: 15 degrees





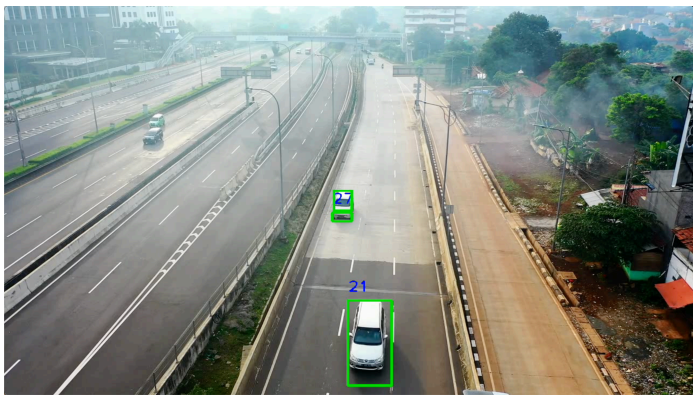




8. Implement a real-time object tracker (two versions) that (i) uses a marker (e.g. QR code or April tags), and (ii) does not use any marker and only relies on the object

i)

ii)



Repo:
https://github.com/ShaikNagurShareef/CSc8830-Computer-Vision/tree/main/Assignment-3