Computer Vision Assignment - 3

Sathwik Chowda

Question - 1

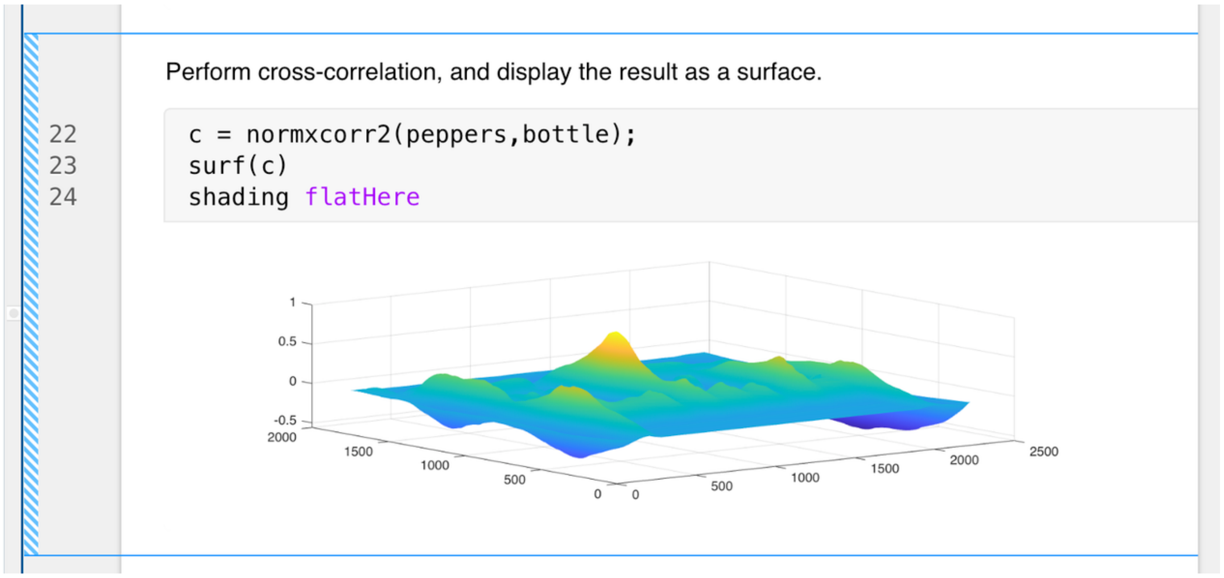
Capture a 10 sec video footage using a camera of your choice. The footage should be taken with the camera in hand and you need to pan the camera slightly from left-right or right-left during the 10 sec duration. Pick any image frame from the 10 sec video footage. Pick a region of interest corresponding 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.

Answer:

For comparition I have used Normalized Correlation,

1. Captured a 10 seconds video, with bottle in foreground and a black sofa in the background.
2.

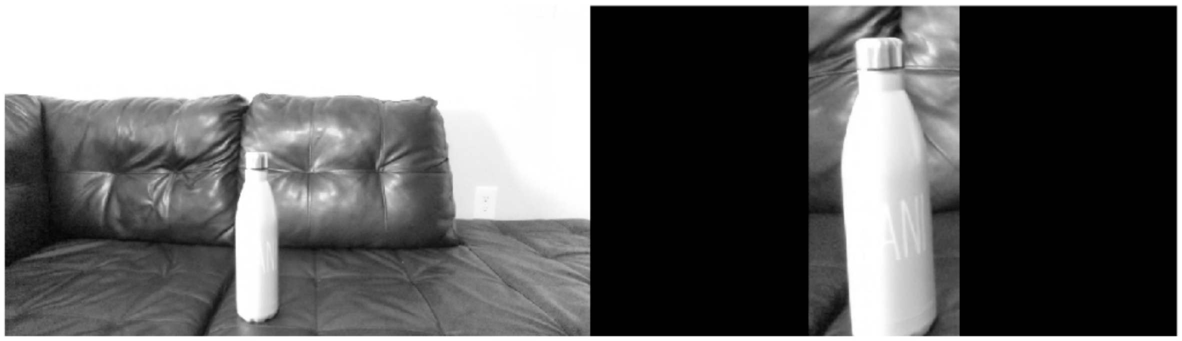
3. Now using matlab, I got all the frames from the video.
4. To begin the comparistion, it is very important to first convert the RGB image into a GRAY SCALE image.
5. Then using drawrectangle function, select your Region of Interest (ROI) in the frame.
6. Get the four-point coordinates for the rectangle, so that you can now crop your ROI and save it for further calculation.
7. Select another frame from the video and to perform Normalized correlation
8. If you try to plot the normalized correlation of these two pictures/frames, you will end up with a result similar to this in the following,

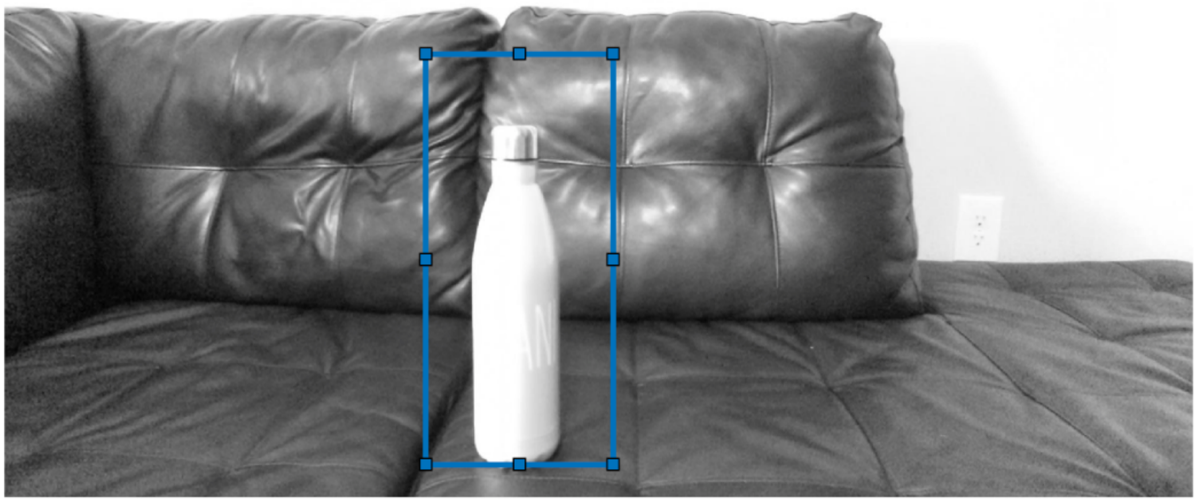


Perform cross-correlation, and display the result as a surface.

```
c = normxcorr2(peppers,bottle);
surf(c)
shading flatHere
```

8. In the above plot, now we need to find the peak and store it in xpeak and y peak
9. We need to store offset values of x and y, by removing height and width of ROI.
10. Then by plotting a rectangle on the selected frame, using xoffset, yoffset, height and width of ROI, we can match the foreground, i.e. the bottle in our example





Above picture are Region of Interest and the matched result after correlation.
NormalizedCorrelation.mlx

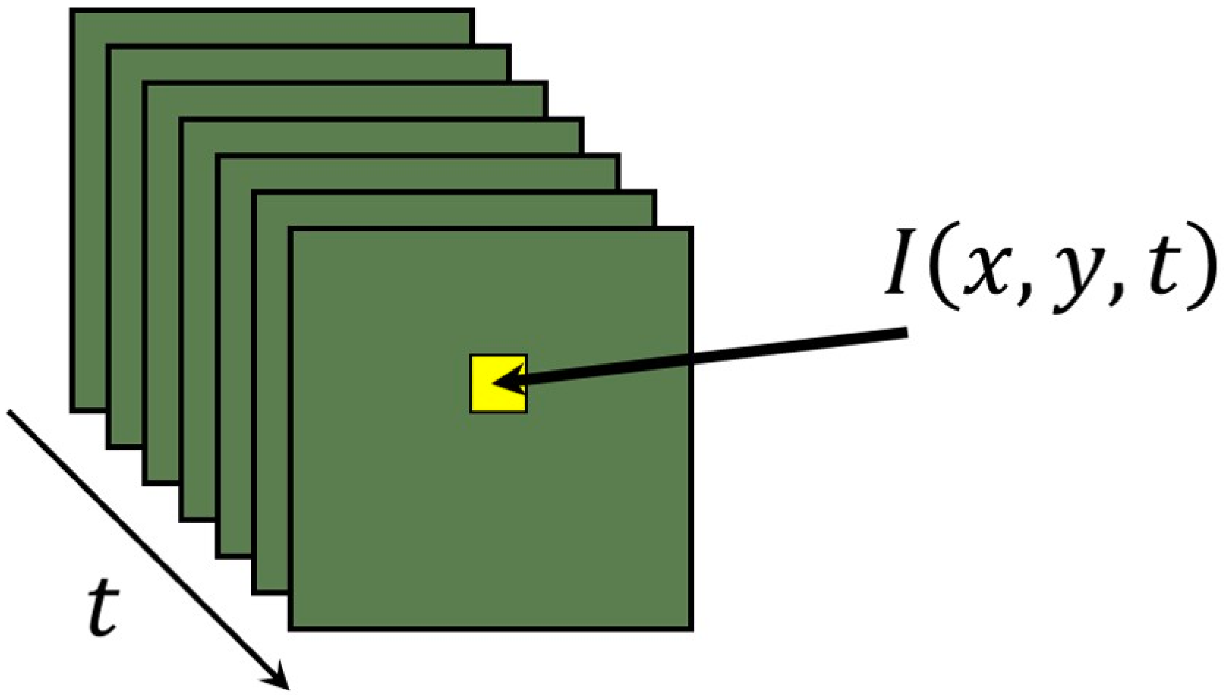
Question - 2

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.

Answer:

If we split the video into a series of frames and consider two consecutive frames, both the frames have similar features in them. This feature mapping from frame to frame can be used in motion tracking and estimating the direction and speed of the object in the frame. This is also called Optical Flow and it is primarily based on an algorithm called the Lucas Kanade method.

In a video sequence instead of using just space coordinates, we also need to track change of intensity, therefore we also need to consider one additional variable, time.

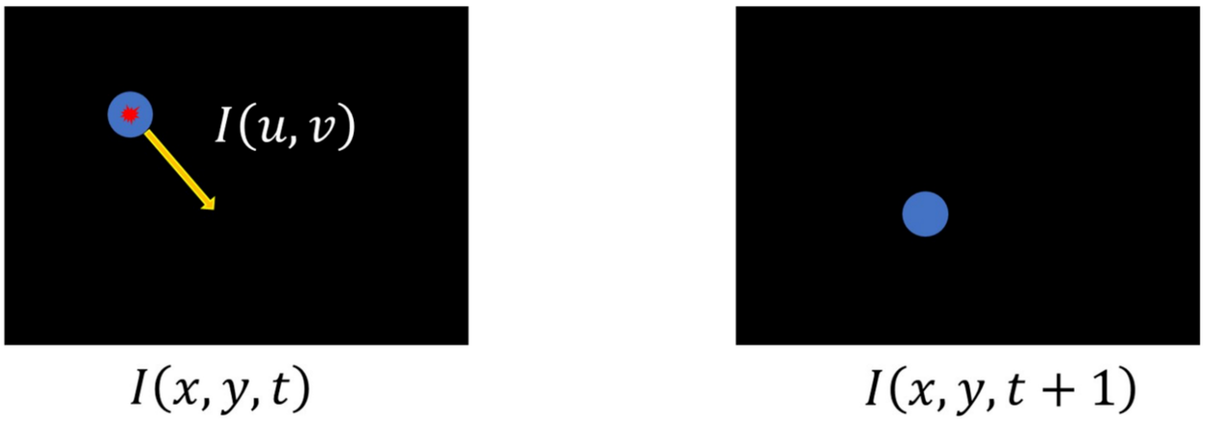


source: https://datahacker.rs/calculating-sparse-optical-flow-using-lucas-kanade-method/

In Lucas Kanade's method, an assumption was made that there is a fixed light condition in every frame and it is constant.

So if I(x,y,t) represents pixel intensity at time t at location (x,y). I(x+u,y+v,t+1) represents pixel intensity at t+1 is basically consecutive frame and u,v are displacements.

For this example lets consider small motion, because it is form consecutive frames. So there is no necessity to change x and y, but because t changes 1 frame, so (t+1)



source: https://datahacker.rs/calculating-sparse-optical-flow-using-lucas-kanade-method/

Now the equation can be written as,
□(□,□,□)=□(□+□,□+□,□+1)
Substituting the □(□+□,□+□,□+1) with □(□,□,□+1) because of **small motion and add derivate of x direction and y direction.**

Then we get,
0≈[□(□,□,□+1)−□(□,□,□)]+□□□+□□□, where □□=∂□/□ and □□=∂□/□
The above equation can also be written as **0≈□□+□□u+□□**v

LUCAS KANADE'S METHOD :

1. Compute the Image x and Image y derivatives
2. Compute difference between Image It = Image 1 - Image 2, Because of consecutive frames, it will be 1 in our case
3. Smoothen the image components Ix, Iy, It
4. Solve the linear equations for each pixel and calculate the eigen values
5. Plot the optical Flow vectors

This is the brief procedure to find out Optical Flow estimates from a fundamental perspective.

Question - 4

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)

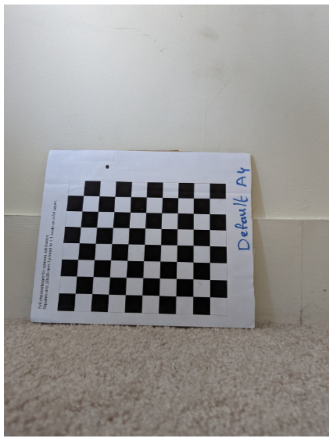
Answer:

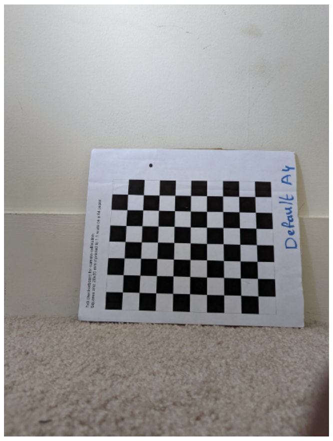
The formula to calculate distance using disparity is,

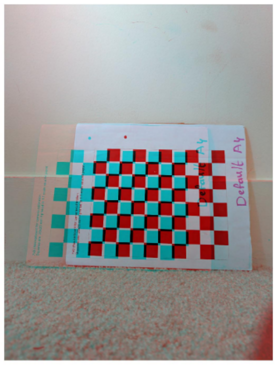
```
D:= Distance of point in real world,
T:= base offset, (the distance *between* your cameras)
f:= focal length of camera,
d:= disparity:

D = T*f/d
```

disparity.m

So to mimic the stereo camera effect in our camera, we need to slide the stationed camera exactly parallel to the flat vertical object. As below,







The stereo anaglyph Image of the above two pictures is the one which you are seeing on the rightmost. Now the disparity value let's say **'d'** is simple, the distance between the blue dot and red dot you are seeing above the chess pattern is the disparity (d)

The code I used to do this,

```
L = imread('/Users/sathwikchowda/Desktop/L.jpg');
R = imread('/Users/sathwikchowda/Desktop/R.jpg');
stereoAnaglyph_Image = stereoAnaglyph(L, R);
figure; imshow(stereoAnaglyph_Image);
```

Coming to the actual solution of finding out the distance,

- Distance is currently unknown
- T = 90 mm (approximate value)
- f = 26 mm (focal length of the pixel 6 camera)
- As mentioned below, the distance/disparity between marked points is 410.18 pixels which is around 108 mm.
- Note: the disparity range for a good stereo camera can only be between 0, and 128. Because I did not take pictures with a stereo camera, the d_max which is the maximum disparity is taken as 410, i.e, d = 108 mm
-

Substituting values into the formula:

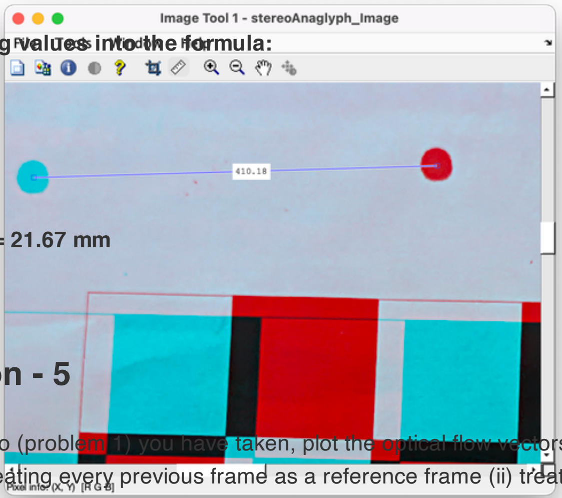


Image Tool 1 - stereoAnaglyph_Image
File Tools Window Help
410.18
Pixel info: (X, Y) [R G B]

T = 90 mm
f = 26 mm
d = 108 mm

D = (T*f)/d = 21.67 mm

Question - 5

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

Answer:

1) In the first part the requirement is to take every previous frame as a reference frame. That means, the comparition should be performed between two consecutive frames continuously, until all the frames completes. For this to happen, we need to change two variables in the code,

```
referenceFrame = 01, % So the loop starts from 1st frame
```

```
stepFrame = 1 % So the logic does not skip considering frame
```

2 and 3rd)

For both second and third parts of this question, we need to change stepFrame as 11 and 31, so each 11 or 13th frame will be considered for comparition.

```
stepFrame = 11 or 31 % So the logic takes each 11 or 31st frame
```

ComputeAndDisplayOpticalFlowExample.mlx

Question - 6

Run the feature-based matching object detection on the images from problem (1). *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

Answer:

Attached mlx document, please refer it for solution FeatureMatching.mlx

Question - 7

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.

Answer:

Please find the solution in attached document: 7.mlx and 7.pdf

Question - 8

Repeat the image capture experiment from problem (4), 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 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).*

Answer:

I have attached the solution in the below,
Matlab Script: 8.mlx
PDF: 8.pdf

Now, after running this experiment, I realized that we can also track location without the help of geo_locations. I felt really excited when I get to know that, OpticalFlow estimates can be so helpful in tracking location.

DJI drones, which are very popular, have four cameras placed in a rectangle composition. I did not realize its importance until now. Now it make sense, because those are nothing but stereo cameras, helping drone to know its home path, even without the help of GPS.