EMBEDDED MACHINE VISION AND INTELLIGENT AUTOMATION

EXERCISE 4

NIKHIL DIVEKAR

Question 1.

Introduction:

Hough Transform is basically a computationally efficient procedure for line detection in the picture. The presented paper discusses how the use of angle-radius is better than slope-intercept parameters as it simplifies the computation. It is found that general method to detect the lines in an image formed by all pair of points is computationally time expensive.

Hough transform method involves transforming each of the figure points into straight line in a parameter space which is two-dimensional slope-intercept plane. An arbitrary straight line can be represented by single point in the parameter space. It is represented by angle theta of its normal and algebraic distance p from the origin. If we restrict theta from 0 to pi, we get unique point for every different line. This conversion into parameter space follows some properties. Point in picture plane corresponds to sine curve in parameter plane while point in the parameter plane corresponds to a line in picture plane.

Application and mechanism:

Consider we are provided with a certain image. This image is first pixelized to get the digitized form of the image. Next, a simple differencing operator is used to locate the significant intensity changes over the image. Further using theta and p, points in the parameter plane are stores in an accumulator array. When many number of points collinear, the entry for the line that fits them is accounted for. These results are sensitive towards the quantization of theta and p value.

Limitations and Quantization:

Since the results are pretty much dependent on quantization of theta and p, it may yield wrong result for over-quantization. Also, collinear points are found without regard to contiguity. Thus, position of line can be distorted with presence of collinear points in other part of the picture and so meaningless group of collinear points will be detected. Thresholding also plays a key role. If thresholding is pretty much reduced then we may miss certain lines. In general Hough transform can be viewed as computationally effective of scene analysis. The main importance of the Hough transform is to detect the occurrence of figure points lying on straight line and possessing some specified property. The Hough transform can also be extended to curves other than straight line. For examples, it can also be used to detect the number of circles

Summary:

Hough transform can be used to detect the curves like straight line or the circle in an image. It’s also explained how angle-distance approach is better than slope-intercept method. Further, four properties of image and parameter plane are explained. Then the advantages and limitation of Hough transform is discussed along with its methodology.

Question 2.

Name	Date modified	Type	Size
Arctic-Swan	7/21/2018 1:18 AM	PNG File	426 KB
hough	7/21/2018 1:18 AM	File	99 KB
hough.cpp	7/21/2018 1:18 AM	CPP File	2 KB
hough.o	7/21/2018 1:18 AM	O File	129 KB
Makefile	7/21/2018 1:18 AM	File	1 KB
q2	7/21/2018 1:18 AM	JPG File	242 KB

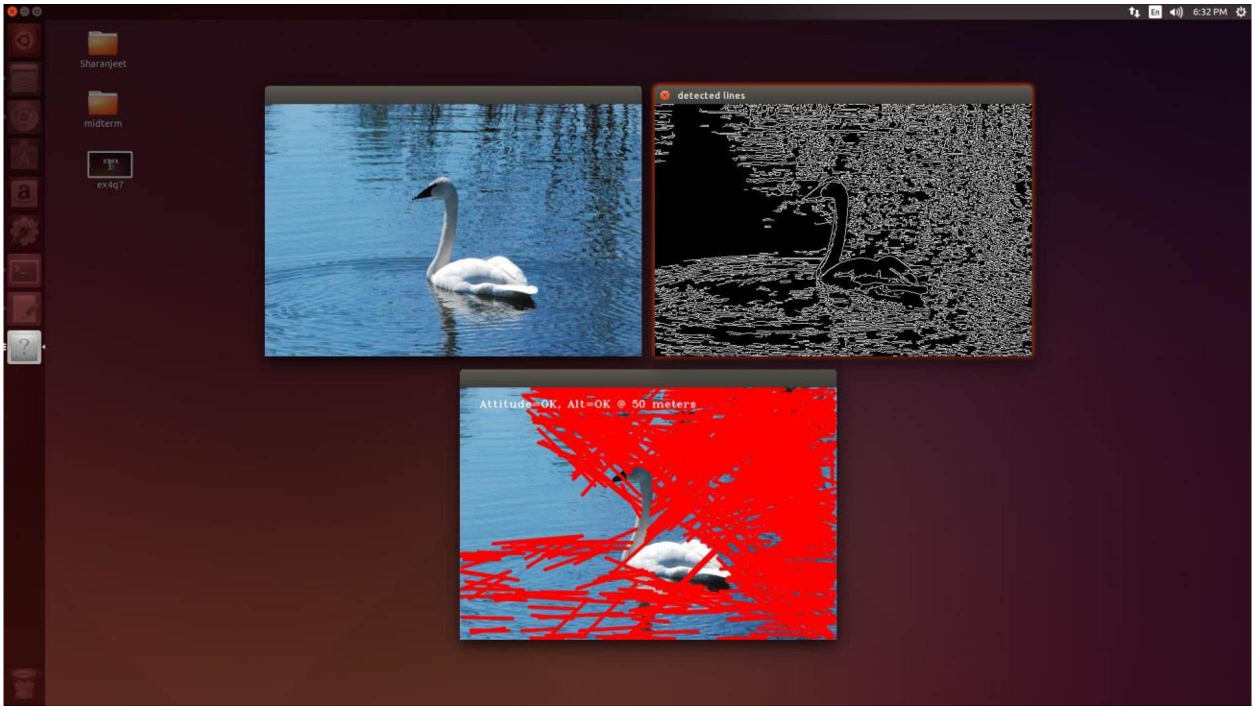
Executable called hough is created which proves code builds successfully and images below that code runs successfully and image is transformed successfully.

Image of make command:

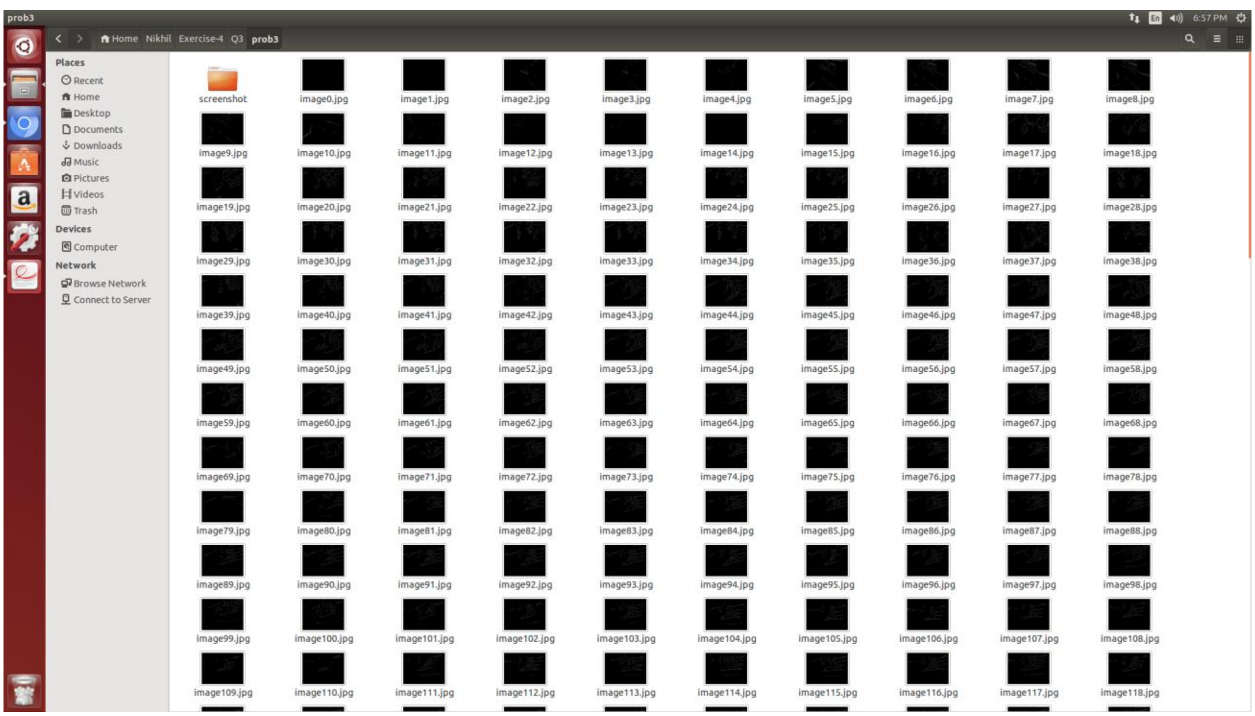
```
g++ -O0 -g   -c hough.cpp
g++  -O0 -g   -o hough hough.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopencv_flann -lopencv_video
```

Input Image:

Hough Transform Image (Detected lines) and transformed image along with input image:



Question 3.

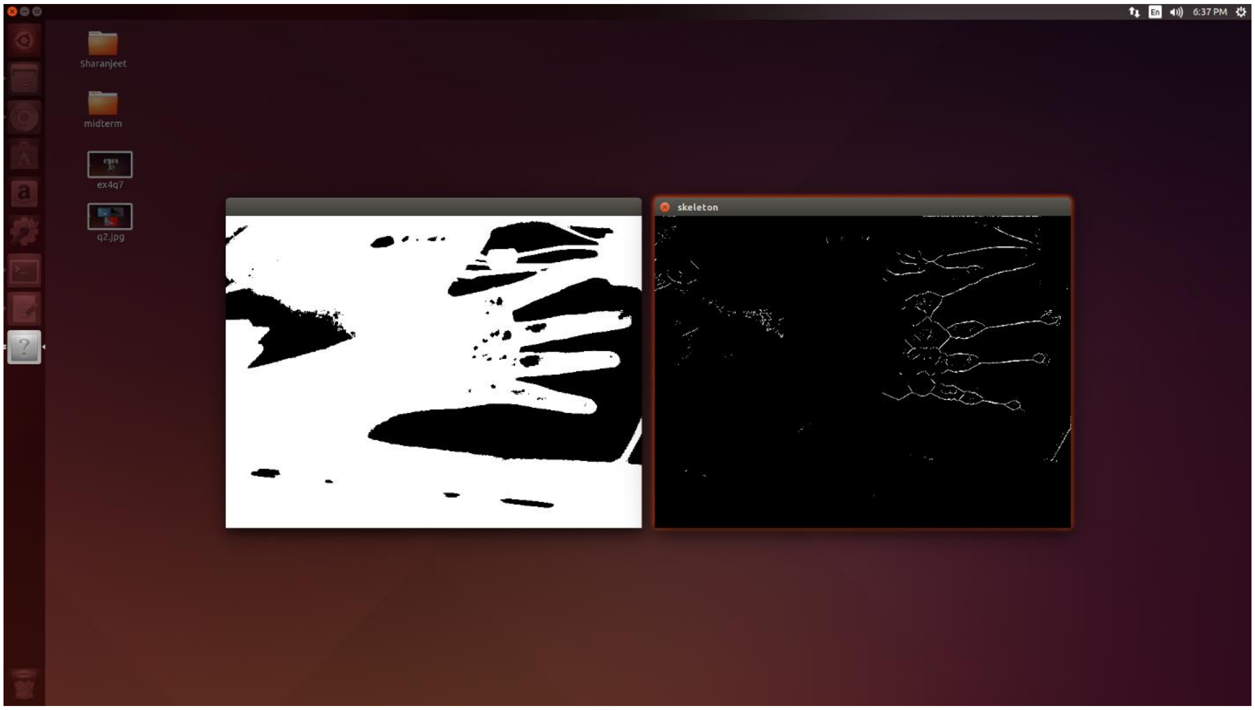


Frames are created separately after running the code which proves that code is built and run. The video is re-encoded through these frames.

Image of make command:

```
g++ -O0 -g   -c problem3.cpp
g++  -O0 -g   -o problem3 problem3.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopencv_flann -lopencv_video
```

Screenshot of the binary thresholded video and thinned video:



The complete thinned video of arm gesture has been uploaded to D2L in separate folder.

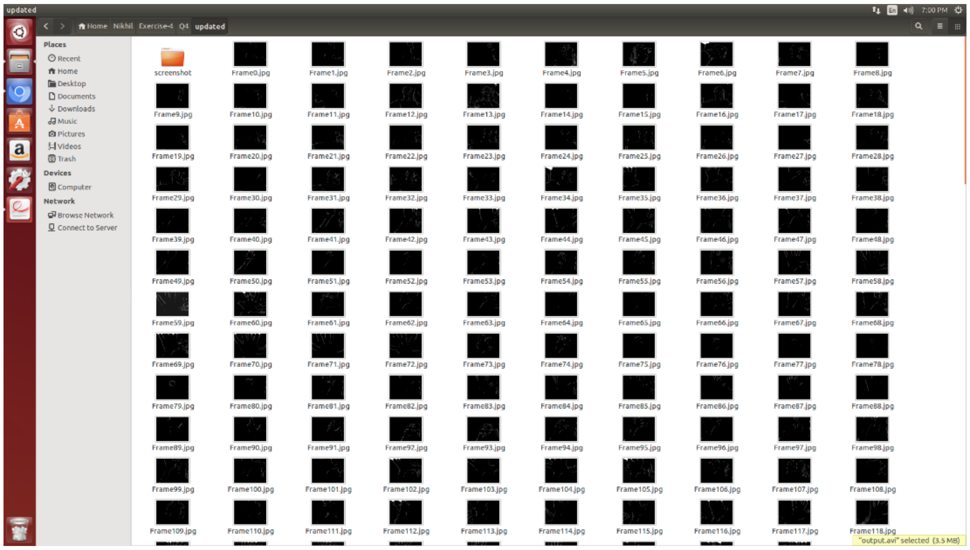
Link to the video: https://drive.google.com/open?id=1lM-qVGiHaVsTMM6yEARynQaT_8eNxk8C

Question 4.

Frames are created separately after running the code which proves that code is built and run. The video is re-encoded through these frames.

Image of make command:

```
g++ -O0 -g   -c q4.cpp
g++  -O0 -g   -o q4 q4.o `pkg-config --libs opencv` -L/usr/lib -lopencv_core -lopencv_flann -lopencv_video
```



Screenshot of binary thresholded video and thinned video using bottom-up algorithm:

The complete thinned video of arm gesture has been uploaded to D2L in separate folder.

Link to the video:

https://drive.google.com/open?id=1L6baiia7uV6vAH59lpSwraxM_ipGeKgE

Comparison between top-down and bottom-up approach:

1. The Top-down approach provides better quality thinning than bottom-up approach since it uses in-built OpenCV APIs.
2. The bottom-up approach however is faster in speed than the top-down approach.

Question 5.

Introduction:

This paper discusses a method for extracting distinctive features from image which performs reliable matching between different views of object or scene. These features are robust enough to withstand change in 3D viewpoint, addition of noise, etc. The paper also discuses method of using these features for object recognition also using fast-nearest neighbor algorithm. This is then followed by Hough transform to identify cluster belonging to single object. The features are highly distinctive and thus allows a single feature to be correctly match with high probability. This is the approach used in object or scene recognition. Cascading filter approach is used where in the more expensive operation are applied only at locations which pass an initial test.

Major stage of computation used to generate set of images are as follows.

1. Scale space extrema deviation using difference-of-Gaussain function.
2. Keypoint localization.
3. Orientation assignment.
4. Keypoint descriptor.

History:

Image matching can be dated back to work of Moravec in 1981 on stereo matching using corner detector. It was improved by Harris and Stephens in 1988 to make it more repeatable near edges. Initial applications included short-range motion tracking but this formed basic platform for further development. Zhang in 1995 further improved the algorithm using correlation. Torr's algorithm further found application in long range motion tracking where geometric constraints were used to remove outliers for rigid objects within image. Schimd and Mohr in 1997 showed that feature matching can be extended for image pattern recognition. The approach discussed in this paper is robust against changes in 3D viewpoint and addition of noise. Other advantage being ability to extract large number of features. The features use image contours or region boundaries which avoids disruption by clustered backgrounds.

Methodology:

The approach is known as Scale invariant feature transform (SIFT) algorithm. Image of dimension 500x500 pixels can give rise to 2000 stable features. More the number of features, better is the object recognition. In SIFT algorithms, stable features are extracted and stored in database. When working with new image, the features are compared with those already present in the database and finding candidates based on Euclidean distance. Keypoint descriptor allow single feature to find its correct match but in cluttered image, many features from background cannot be match with those in database. Determination of clustered can be performed using hash table implementation of Hough transform.

Each cluster of 3 or more features that agree on an object are valid for further inspection. This process is iterated over and over. Finally, a detailed computation is made of probability that particular set of features indicates the presence of object. Object matches that satisfies all these tests can be considered correct.

The first stage of keypoint detection is to identify location and scales that can be repeatedly assigned using different vies of same object. This can be accomplished by searching for stable features. Scale-space of an image is defined as the convolution of variable-scale Gaussian with an input image.

Next step is local extrema detection. This is done by comparing a certain pixel to all of its 8 neighboring pixels. The pixel is selected if it is either larger or smaller than all its neighboring pixels.

Each image is then subjected to number of transformation like rotation, scaling, etc. If we pre-smooth the image before extrema detection then we lose high spatial frequency. Hence, we double the size of an input image using linear interpolation. If original image had blurring of sigma = 0.5, now the blurring equals sigma=1 in expanded image. This increase stable keypoints by factor of 4.

Once key-point has been located the next step is perform detailed fit to nearby data for location and scale. The next step is to eliminate edge responses. The difference-of-Gaussian function will have strong response along the edges and is unstable towards the noise. Orientation is then assigned to each keypoint based on local image properties thereby forming an oriental histogram. The highest peak is detected and then any other local peak within 80% of highest peak is used to create keypoint with that orientation. Then we need to compute descriptor for local image region. First image gradient magnitudes and orientations are sampled around keypoint location. Then the coordinates of descriptor and gradient orientations are rotated relative to keypoint orientation.

Summary:

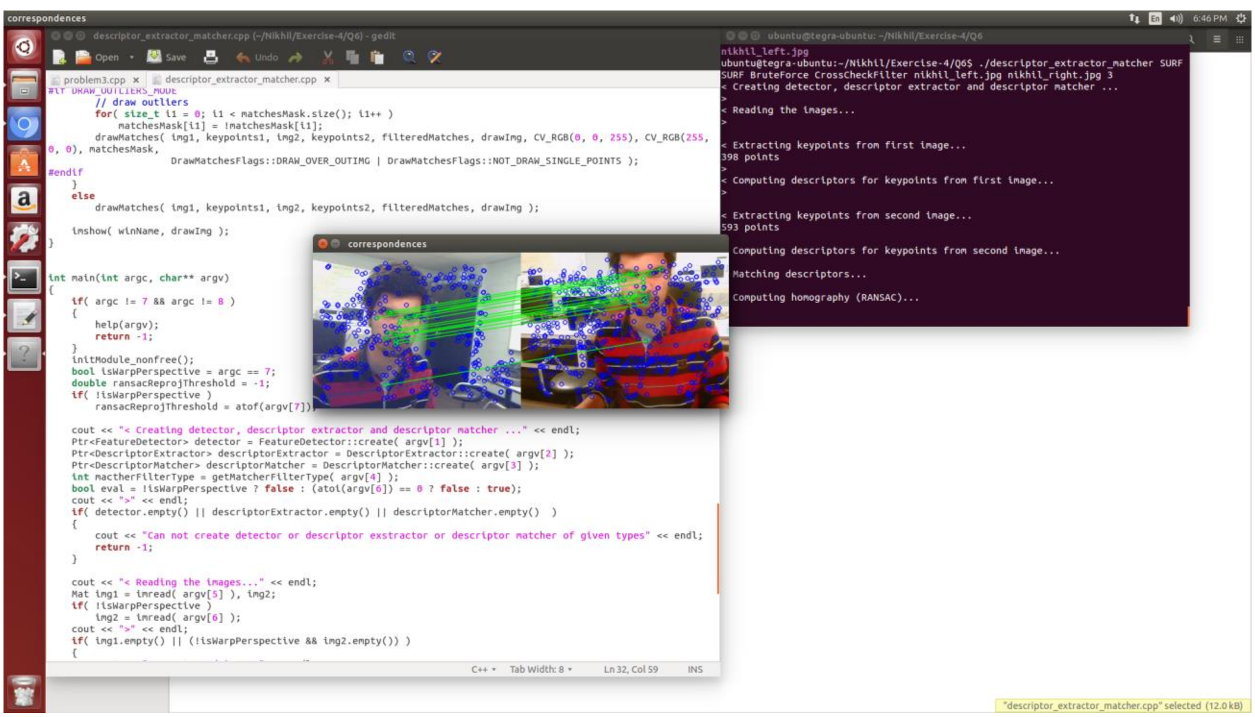
Object recognition is performed by matching keypoints to those already present in the database based on training data. Clusters of 3 or more features that agree on the objects are identified. Then each cluster is checked using geometric fit. The best candidate for each keypoint is identified by nearest neighbors in database based on minimum Euclidean distance. Best-bin-first algorithm is applied to detect the closest neighbor with highest probability of feature matching. To maximize the performance, the clusters of features are applied Hough transform for better results. These features are then passed through verification called least-squares solution wherein outliers can be removed mostly. The final decision to accept or reject the location depends on the probability of feature matching.

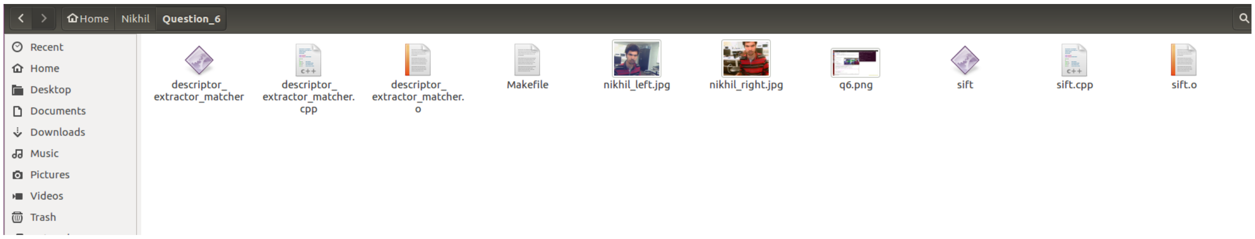
Question 6.

Left Image

Right Image

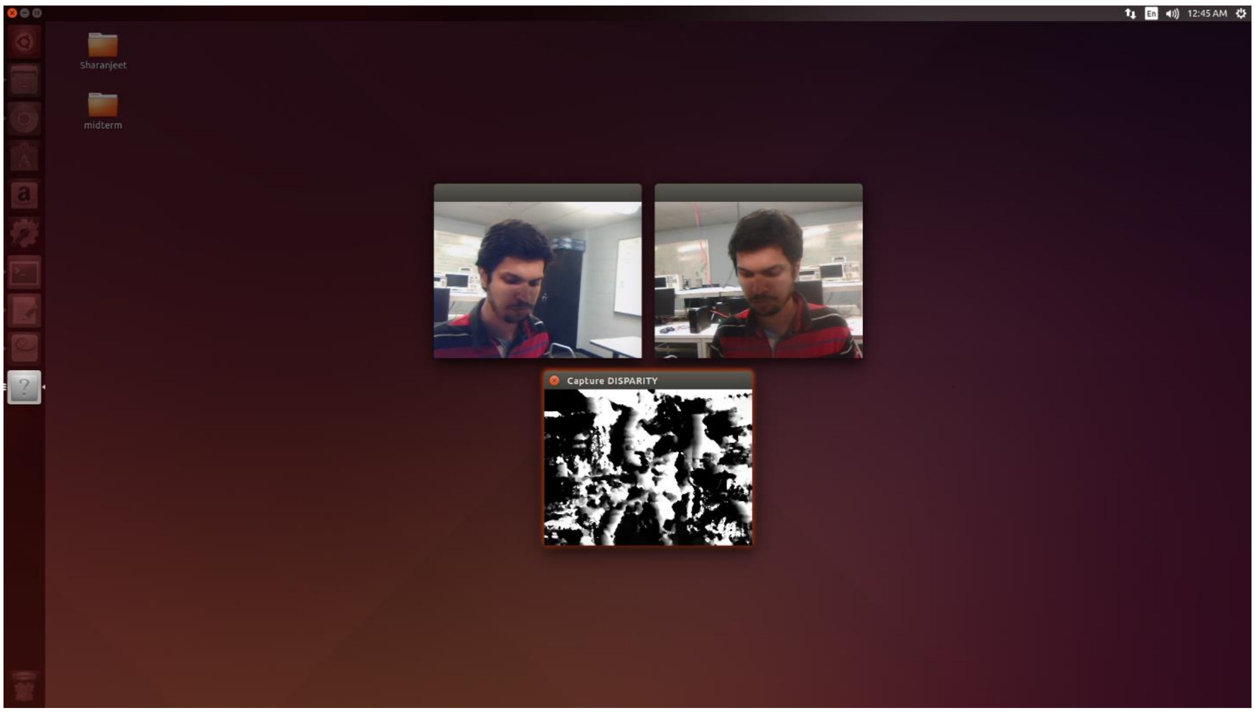
Correspondence Image:





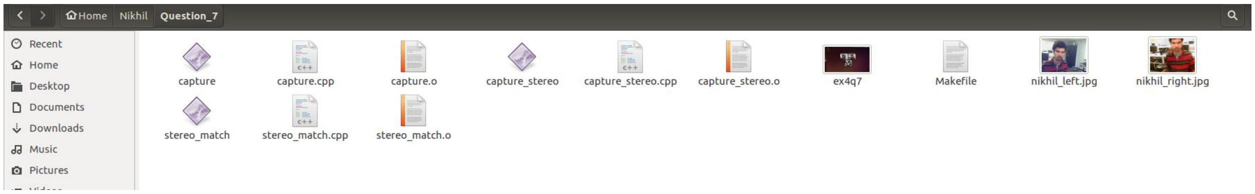
Executable file created. Proof of build and compile of the program.

Question 7:



Frame from two cameras and Disparity map.

First of all, we take into consideration the calibration of the camera and extract all the corner points from the image and then calculate extrinsic and intrinsic parameters. Next image rectification is performed in which is projecting two or three image in a common plane. Image distortion is removed. And image preprocessing to remove unwanted noise. The difference between the rectified right and left image is called disparity map. This is the actual procedure performed while performing disparity map. However, in the code key parameters like camera calibration, considering intrinsic and extrinsic parameter and image rectification is not done. These key parameters can be performed to improve the efficiency of disparity map.



Executable file created. Proof of build and compile of the program.