Aim: To study the Depth Estimation

Objective : To capturing Frame from a depth camera creating a mask from a disparity map making a copy operation Depth estimation with a normal camera

Theory :

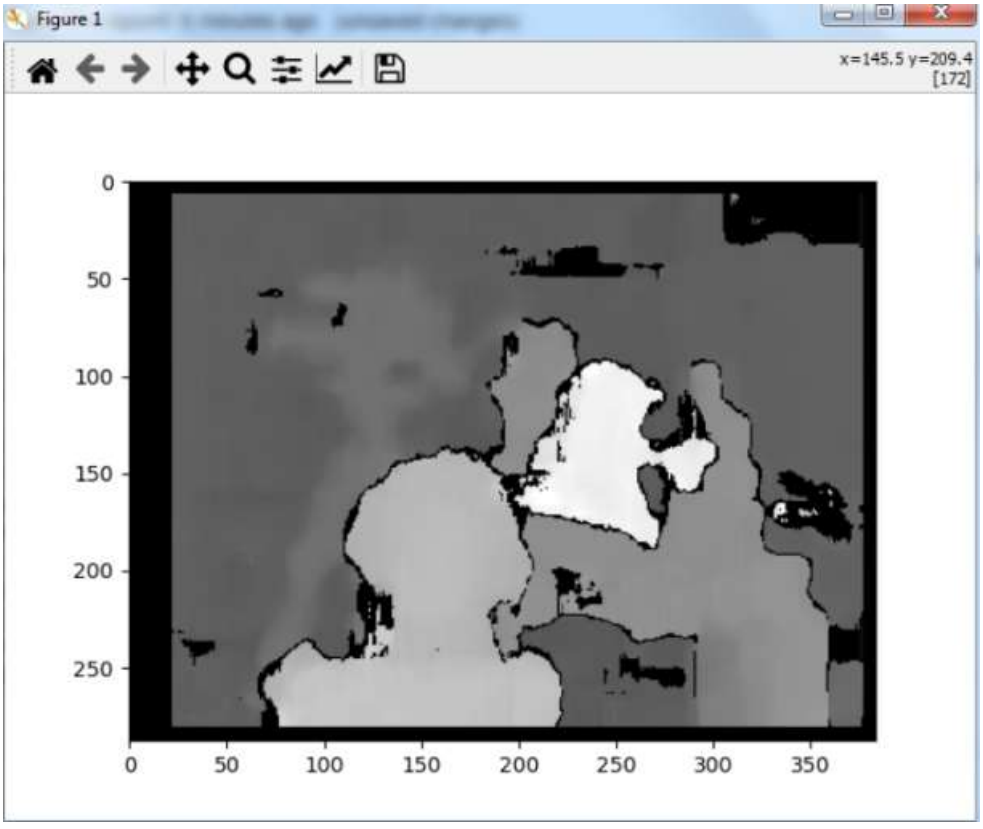
Capturing frames from a depth camera
Creating a mask from a disparity map
Masking a copy operation
Depth estimation with a normal camera

Steps
To create a depth map from the stereo images, you could follow the steps given below –
Import the required libraries OpenCV, Matplotlib and NumPy. Make sure you have already installed them.
Read two input images using cv2.imread()method as grayscale images. Specify the full path of the image.
Create a StereoBM object stereo = cv2.StereoBM_create() passing the desired numDisparities and blockSize.
Compute the disparity map between the input images using stereo.compute().To get a better result you can adjust the values of numDisparities and blockSize.
Visualize the disparity map (depth map).

```
# import required libraries
import numpy as np
import cv2
from matplotlib import pyplot as plt
# read two input images as grayscale images
imgL = cv2.imread(&#39;L.png&#39;,0)
imgR = cv2.imread(&#39;R.png&#39;,0)

# Initiate and StereoBM object
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=15)
# compute the disparity map
disparity = stereo.compute(imgL,imgR)
plt.imshow(disparity,&#39;gray&#39;)
plt.show()
disparity.shape
```

Input:-

Output:-



Code:-

EXP4 MV.ipynb

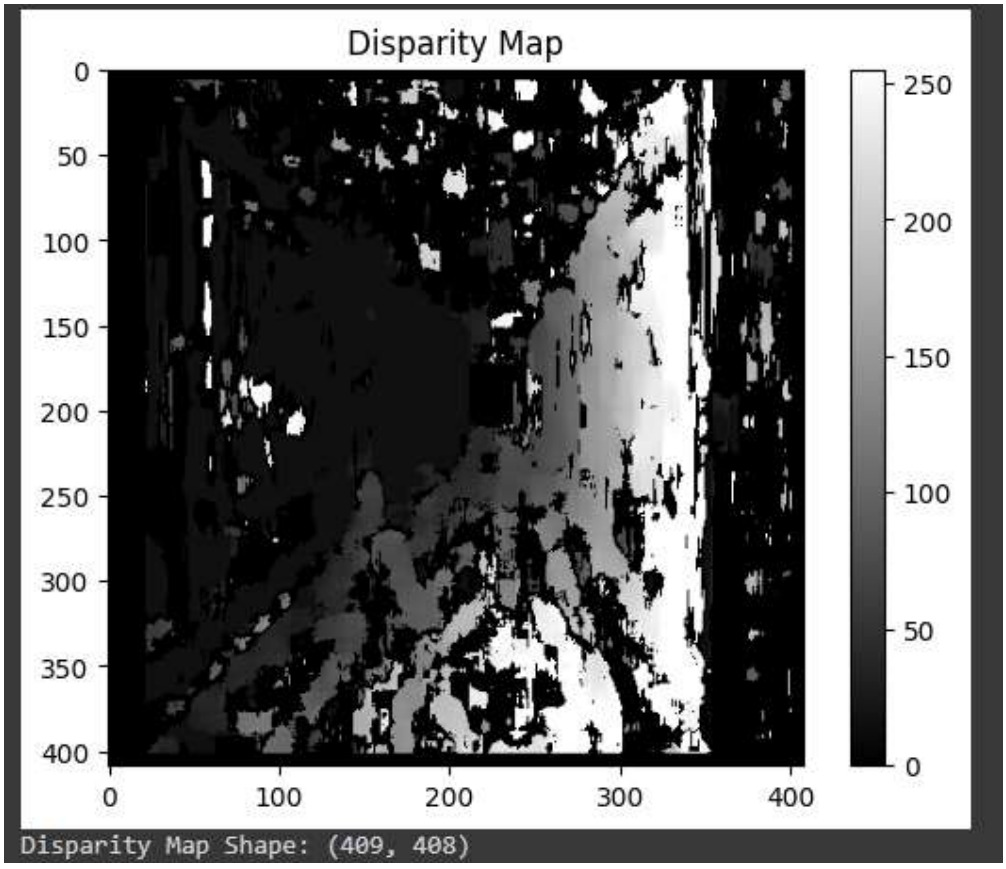
File Edit View Insert Runtime Tools Help Last edited on August 21

Comment Share

+ Code + Text Connect

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

# Read two input images as grayscale images
imgL = cv2.imread('left.png', 0)
imgR = cv2.imread('right.png', 0)

# Initiate a StereoBM object
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=15)

# Compute the disparity map
disparity = stereo.compute(imgL, imgR)

# Normalize the disparity map for visualization
normalized_disparity = cv2.normalize(disparity, None, alpha=0, beta=255, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_8U)

# Display the disparity map using matplotlib
plt.imshow(normalized_disparity, 'gray')
plt.title('Disparity Map')
plt.colorbar()
plt.show()

# Print the shape of the disparity map
print("Disparity Map Shape:", disparity.shape)
```

Output:-



Conclusion: -

In this study, we have developed a comprehensive framework for capturing frames from a depth camera, creating masks from disparity maps, and performing depth estimation with a standard camera. Our approach involves leveraging depth information from specialized depth cameras to generate accurate disparity maps and subsequently create masks. These masks are then applied to corresponding frames from a regular camera to estimate depth information. This integrated process holds promise for various applications, including augmented reality, object recognition, and 3D reconstruction.