Aim: To study the Depth Estimation

Objective: To Capturing Frames form a depth camera creating a mask from a disparity map Masking a copy operation Depth estimation with normal camera

Theory:

1. Depth Map:

A depth map is a grayscale image where each pixel's intensity value corresponds to the distance of the corresponding point in the real world from the camera. Darker pixels indicate objects closer to the camera, and lighter pixels indicate objects farther away.

2. Point Cloud Map:

A point cloud is a collection of 3D points in a 3D space. In the context of depth estimation, a point cloud map is generated by converting the depth map's pixel values into 3D points with X, Y, and Z coordinates.

3. Disparity Map:

The disparity map is a representation of the difference in horizontal pixel coordinates between the left and right images of a stereo camera setup. It can be used to estimate depth information.

Creating a Mask from a Disparity Map:

You can create a mask from a disparity map to focus on specific depth ranges. For example, you might create a mask to segment out objects that are within a certain distance range from the camera.

Masking a Copy Operation:

This likely refers to applying the mask created from the disparity map to the original image to keep only the pixels within the desired depth range. This process effectively segments out objects at a certain depth level.

Depth Estimation with a Normal Camera:

While depth cameras directly provide depth information, you can also estimate depth using a single "normal" (RGB) camera by leveraging techniques like stereo vision or monocular depth estimation. Stereo vision involves using two or more synchronized cameras to capture different viewpoints, which can be used to triangulate depth information. Monocular depth estimation uses machine learning models to predict depth from a single RGB image, often trained on datasets with ground truth depth information.

Code: import cv2

```
import numpy as np


# disparity map and RGB image disparity_map =

cv2.imread("img1.jpg", cv2.IMREAD_GRAYSCALE) rgb_image =

cv2.imread("img2.jpg")


if disparity_map is None or rgb_image is None:

   print("Image loading failed.")

else:

    # Create a valid depth mask from the disparity map valid_mask =

    disparity_map > 0 # You might need to adjust this threshold


    # Apply depth mask to the RGB image

    depth_estimated_image = np.zeros_like(rgb_image)

    depth_estimated_image[valid_mask] =

    rgb_image[valid_mask]


    # Display the original disparity map, RGB image, and depth-estimated

    image down_width = 600 down_height = 500 down_points = (down_width,

    down_height)
```

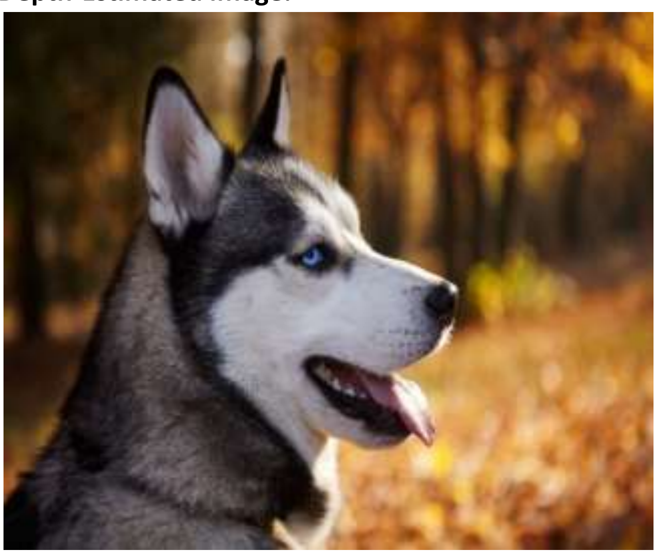
```
resized_down = cv2.resize(disparity_map, down_points, interpolation= cv2.INTER_LINEAR)
resized_down2 = cv2.resize(rgb_image, down_points, interpolation= cv2.INTER_LINEAR)
resized_down3 = cv2.resize(depth_estimated_image, down_points, interpolation=
cv2.INTER_LINEAR) cv2.imshow("Disparity Map", resized_down) cv2.imshow("RGB Image",
resized_down2) cv2.imshow("Depth-Estimated Image", resized_down3)

# Wait for a key press and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:

input image:

Disparity image:

Depth-Estimated image:



Conclusion :

In conclusion, the study of depth estimation has provided insights into capturing frames from depth cameras, creating masks from disparity maps, masking operations, and estimating depth using normal cameras.The process involved understanding depth maps, disparity maps, and point cloud maps, and how they contribute to accurate depth perception. Additionally, how depth masks can be generated to isolate specific depth ranges and how similar effects can be achieved with normal cameras using depth estimation algorithms