Experiment No.4
To study the Depth Estimation
Date of Performance:07/08/2023
Date of Submission: 14/08/2023

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

Capturing frames from a depth camera involves obtaining 2D images that represent the distance of objects in the scene from the camera, creating a depth map. Depth cameras, such as those based on time-of-flight or structured light technology, emit signals and measure the time taken for them to bounce back, allowing precise depth estimation.The process typically includes the following steps:

Initialization: Establish a connection with the depth camera hardware using appropriate software libraries or APIs. Ensure that the camera is correctly calibrated for accurate depth measurements.

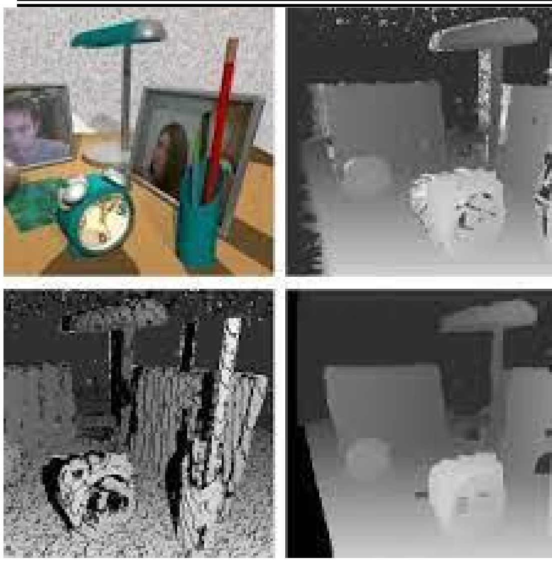
Frame Acquisition: Continuously capture frames from the depth camera. These frames contain pixel values that represent the depth information at each pixel, often stored as a grayscale image where brighter pixels correspond to objects closer to the camera.

Data Processing: Depending on the camera and application, post-processing may be necessary to convert the raw depth data into meaningful depth values or point clouds. This can involve calibration, noise reduction, and registration with RGB images if applicable.

Application-Specific Usage: Utilize the depth frames for various applications such as 3D reconstruction, object tracking, gesture recognition, or augmented reality.

Creating a mask from a disparity map

Creating a mask from a disparity map involves extracting regions in an image where depth information has been estimated using stereo vision techniques. A disparity map represents the differences in pixel positions between corresponding points in stereo images and provides a measure of depth or 3D information.



To create a mask from a disparity map:

Thresholding: Choose a suitable threshold value based on the desired depth range. Pixels with disparities below this threshold are considered invalid or unreliable and are set to zero.

Filtering: Apply additional filters or morphological operations like erosion and dilation to remove noise and fine-tune the mask.

Post-processing: Optionally, you can perform further processing like region growing or contour detection to identify distinct objects or regions with specific depth characteristics.

Mask Generation: The processed disparity map now serves as a mask, where non-zero values indicate valid depth information, and zero values represent areas without reliable depth data.

This mask can be used for various purposes, such as object segmentation, obstacle detection, or depth-based rendering in applications like 3D reconstruction, autonomous driving, or augmented reality.

Masking a copy operation

Masking a copy operation refers to selectively copying data from one source to another based on a specific criterion or pattern, often using a binary mask. A mask is a binary image where pixels are either on (1) or off (0), defining the areas to be copied (1) or excluded (0).

In image processing, masking can be useful for various purposes, like isolating specific regions, extracting features, or applying filters. For example, to copy a region of interest from one image to another, you'd create a mask that highlights the region, then apply the mask to the source image to copy only the selected area. This process ensures that the copied data conforms to the desired shape or pattern.

Masking can be done using bitwise operations, where the mask is used to filter the source data, allowing you to transfer only the relevant information. It's a powerful technique for customizing copy operations in image processing and data manipulation tasks, enabling precise control over the data transfer process.

Depth estimation with a normal camera-

Depth estimation with a regular camera, also known as monocular depth estimation, is the process of inferring the depth or distance of objects in a 2D image captured by a single camera.

This is a challenging computer vision task with various practical applications like autonomous navigation, augmented reality, and image editing.One common technique for monocular depth estimation is using deep learning models, particularly convolutional neural networks (CNNs). These models are trained on large datasets containing paired images with ground truth depth information. During training, the CNN learns to extract visual features and establish relationships between objects' sizes, positions, and their corresponding depths.The trained model can then be used to estimate depth from a single image. However, since monocular depth estimation is an inherently ambiguous problem (the same 2D image can correspond to multiple 3D scenes), the accuracy of such methods may be limited, especially in complex scenes. To improve results, additional sensors like LiDAR or stereo cameras can be used in combination with a regular camera to enhance depth estimation accuracy.

Steps

Steps

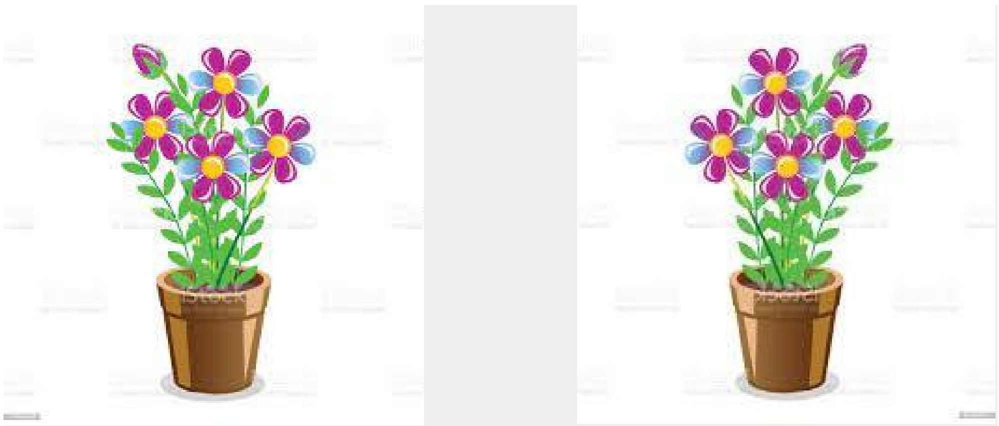
To create a depth map from the stereo images, you could follow the steps given below −

- 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).

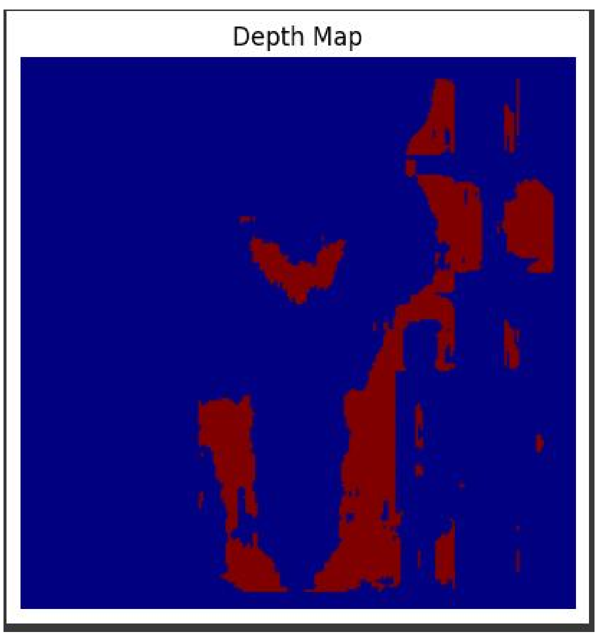
Code:

```
# 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('Left.png',0)
imgR = cv2.imread('Right.png',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,'gray')
plt.show()
disparity.shape
```



Output:



Conclusion:

The study of depth estimation plays a pivotal role in computer vision and perception. It enables machines to perceive and understand the three-dimensional structure of the world, facilitating applications like 3D reconstruction, object recognition, and autonomous navigation. Through techniques like stereo vision, monocular depth cues, and advanced deep learning models, researchers and engineers are continually advancing our ability to accurately estimate depth from 2D images or video streams. The pursuit of improved depth estimation methods is crucial for enhancing the capabilities of robotics, augmented reality, and other fields reliant on precise spatial understanding.