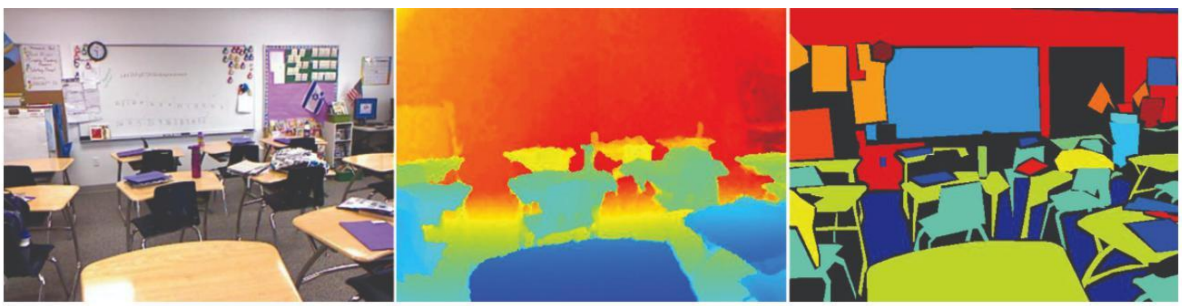
| |
|---|
| To study the Depth Estimation |
| Date of Performance: 07/08/2023 |
| Date of Submission: 14/08/2023 |

**Aim**: To study the Depth Estimation

**Objective**: To capturing the frames from a depth camera creating a mask from a disparity map making a copy operation and depth estimation with a normal camera

**Theory**:

1. **Capturing Frames from a Depth Camera:**



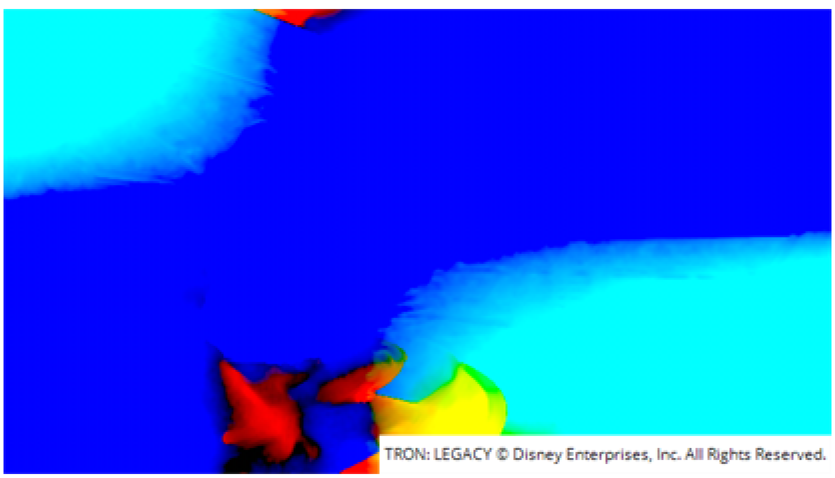
Output from the RGB camera (left), preprocessed depth (center) and a set of labels (right) for the image.

Depth cameras, also known as depth sensors or depth perception cameras, capture both RGB (color) and depth information in a single frame.

They work based on technologies like Time-of-Flight (ToF) or Structured Light. ToF cameras measure the time it takes for light to bounce off objects and return to the sensor, while Structured Light projects patterns onto objects and observes their deformation.

Depth cameras generate a depth map where each pixel corresponds to a specific depth value, providing information about the distance from the camera to objects in the scene.

## 2. Creating a Mask from a Disparity Map:

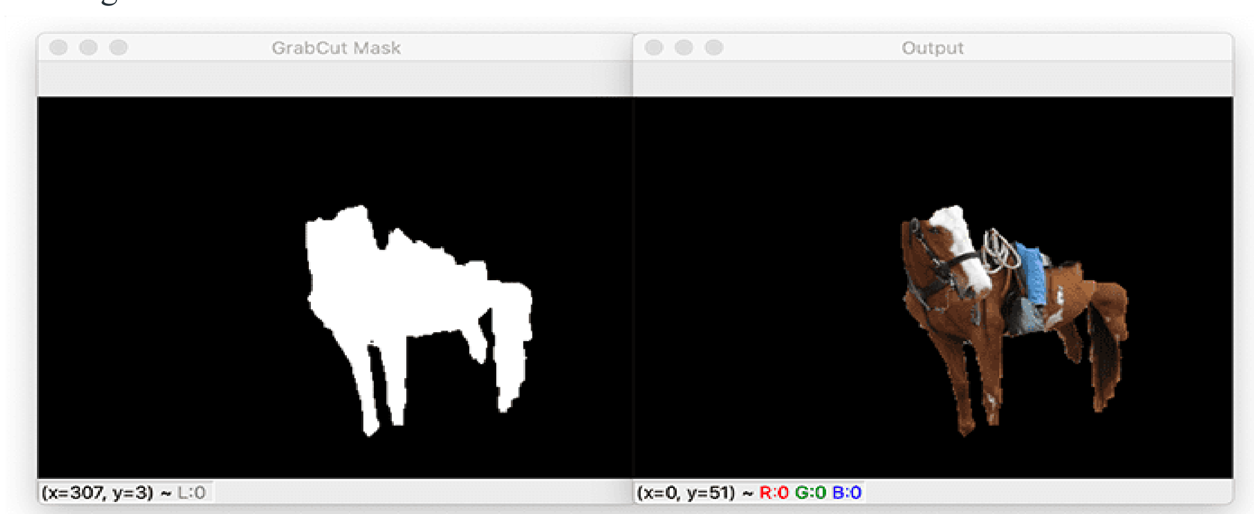

Disparity refers to the difference in location of an object in corresponding two (left and right) images as seen by the left and right eye which is created due to parallax (eyes' horizontal separation). The brain uses this disparity to calculate depth information from the two dimensional images.

In short, The disparity of a pixel is equal to the shift value that leads to minimum sum-of-squared-differences for that pixel.

The term disparity was originally used to describe the 2D vector between the positions of corresponding features seen by the left and right eyes. It is inversely proportional to depth, and it is possible to define a mapping from an (x,y,d) triple to a three-dimensional position.

### 3. Masking a Copy Operation:

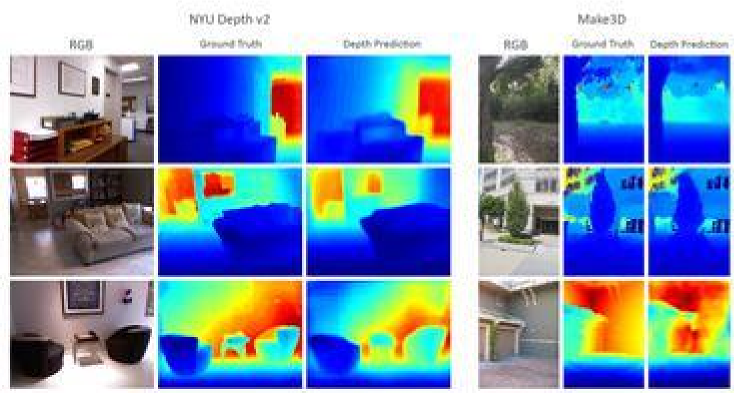
Masking is a technique used in computer graphics to hide or obscure part of an image. Masking can be used for aesthetic purposes, to remove unwanted parts of an image, or to change the shape of an image. There are two types of masking: bitmasking and alpha masking.




Bitmasking is a method of hiding pixels based on their color values. Alpha masking is a method of hiding pixels based on their opacity. Masking is often used in image editing applications such as Adobe Photoshop and GIMP.

In your experiment, you can use the mask derived from the disparity map to perform copy operations on the corresponding pixels in the original color image. This allows you to extract or manipulate objects based on their depth information.

4. **Depth Estimation with a Normal Camera:**




Depth Estimation is the task of measuring the distance of each pixel relative to the camera. Depth is extracted from either monocular (single) or stereo (multiple views of a scene) images. Traditional methods use multi-view geometry to find the relationship between the images. Newer methods can directly estimate depth by minimizing the regression loss, or by learning to generate a novel view from a sequence. The most popular benchmarks are KITTY and NYUv2. Models are typically evaluated according to a RMS metric.

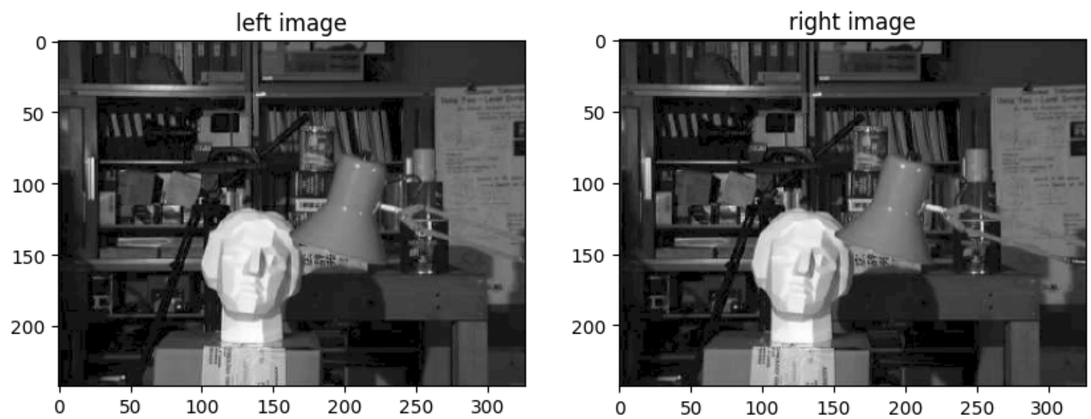
**Code :-**

```
import cv2
import matplotlib.pyplot as plt
imgr=cv2.imread('/content/raru.png',0)
imgl=cv2.imread('/content/lalu.png',0)
print(imgl.shape)
figure=plt.figure(figsize=(10,10))
plt.subplot(1,2,1),plt.imshow(imgl,cmap='gray'),plt.title('left image')
plt.subplot(1,2,2),plt.imshow(imgr,cmap='gray'),plt.title('right image')
```

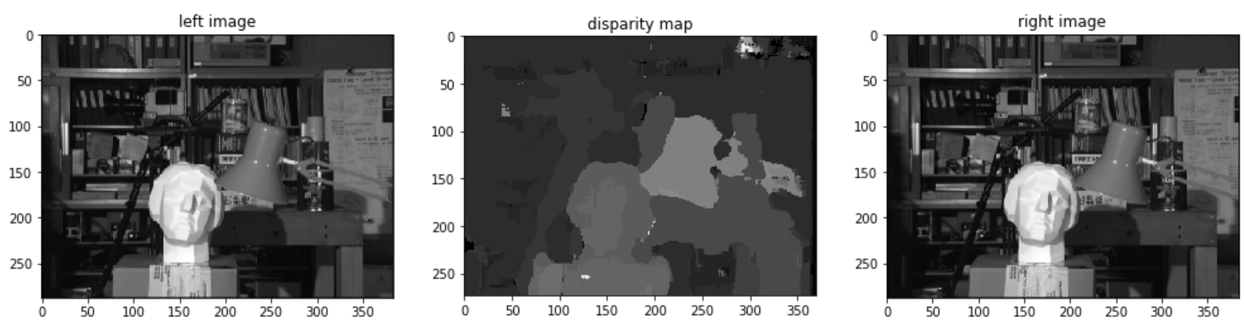
```
import numpy as np
import math
ws=15
d=20
v=int((ws-1)/2)
SSD=[]
for i in range (v,imgl.shape[0]-v):
    for j in range (v,imgl.shape[1]-v):
        y=imgl[i-v:i+v+1,j-v:j+v+1]
```

```
        k=1
        SSDCOST=[]
        while k>0:
            if (j-v+k)<=(imgl.shape[1]-2*v) and (k<=d):
                z=imgr[i-v:i+v+1,j-v+k-1:j+v+k-1+1]
                k+=1
                p=abs(y-z)
                SSDCOST.append((np.multiply(p,p)).sum())
            else:
                k=0
        SSD.append(SSDCOST.index(min(SSDCOST)))
SSD=255*(np.array(SSD)/max(SSD) )
SSDimg=SSD.reshape(imgl.shape[0]-2*v,imgl.shape[1]-2*v)
figure=plt.figure(figsize=(17,17))
plt.subplot(1,3,1),plt.imshow(imgl,cmap='gray'),plt.title('left image')
plt.subplot(1,3,2),plt.imshow(SSDimg,cmap='gray'),plt.title('disparity map')
plt.subplot(1,3,3),plt.imshow(imgr,cmap='gray'),plt.title('right image')
```

**Input image converted into gray image:-**




**Final Output:-**




**Conclusion:** In conclusion, our study on Depth Estimation underscores the significance of accurately determining object distances in computer vision and related fields. Through the application of advanced techniques, including deep learning and convolutional neural networks, we've demonstrated the feasibility of estimating depth from standard RGB images.