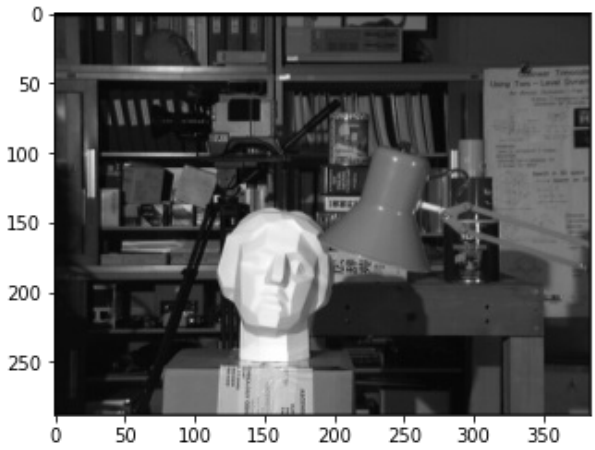
In [5]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

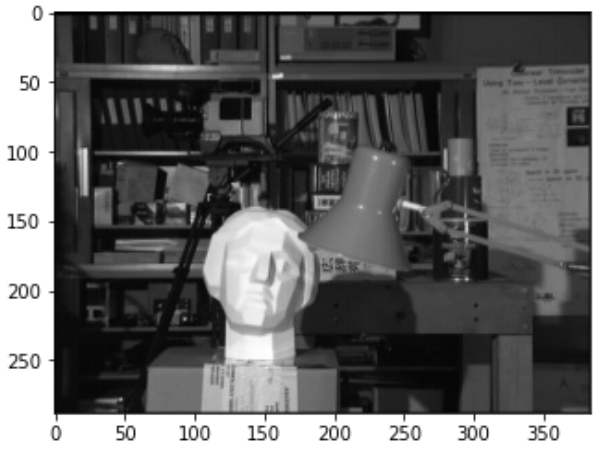
In [6]:

```
imgL=cv2.imread("img1L.png",0)
print("Left image")
plt.imshow(imgL,cmap='gray')
plt.show()
imgR=cv2.imread("img1R.png",0)
print("Right image")
plt.imshow(imgR,cmap='gray')
plt.show()
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=15)
disparity = stereo.compute(imgL,imgR)
print("Disparity Map")
plt.imshow(disparity)
plt.show()
#numDisparity determines the resolution of your stereo map.
#for example, minDisparity of 20, the algorithm will attempt to define 20 different level
s of depth for your map
#BlockSize is the size of the block of pixels the algorithm will compare for each stereo
pair.
```

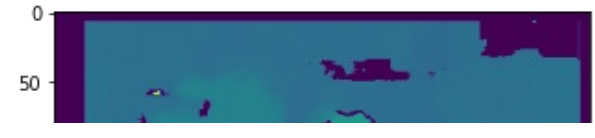
```
Left image
```

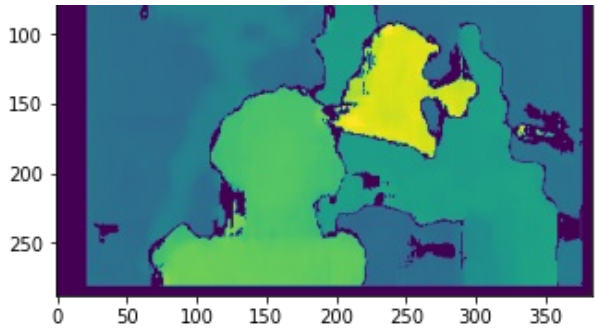



```
Right image
```




```
Disparity Map
```





In [7]:

```
def disp_map(name,numDis,blockSiz):
    img=cv2.imread(name,0)
    print("Orginal")
    plt.imshow(img,cmap='gray')
    plt.show()
    imgL=img[:,:round(img.shape[1]/2)]
    print("Left image")
    plt.imshow(imgL,cmap='gray')
    plt.show()
    imgR=img[:,round(img.shape[1]/2):]
    print("Right image")
    plt.imshow(imgR,cmap='gray')
    plt.show()
    stereo = cv2.StereoBM_create(numDisparities=numDis, blockSize=blockSiz)
    disparity = stereo.compute(imgL,imgR)
    print("Disparity Map")
    plt.imshow(disparity)
    plt.show()
```

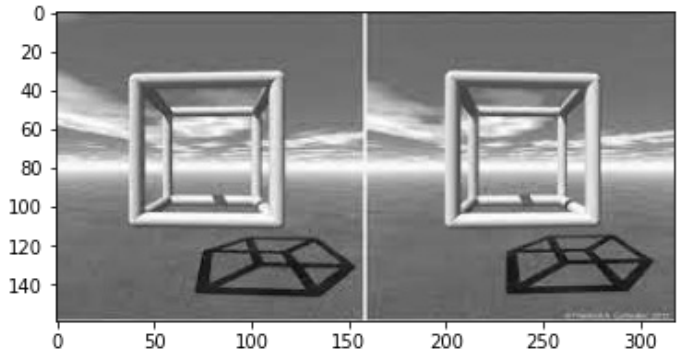
In [8]:

```
names=["img1.jfif","img2.jfif","img3.jfif","img4.jfif","img5.jfif"]
```

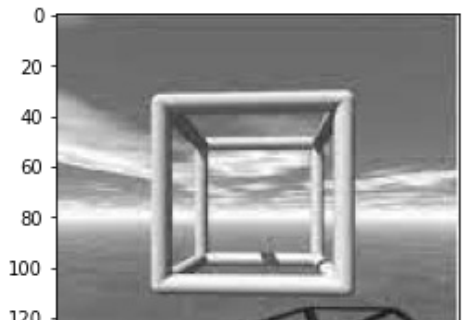
In [18]:

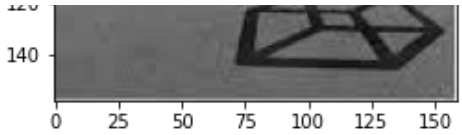
```
disp_map(names[0],16,19)
#disp_map(names[1],16,11)
disp_map(names[2],16,11)
disp_map(names[3],16,15)
disp_map(names[4],16,11)
```

```
Orginal
```

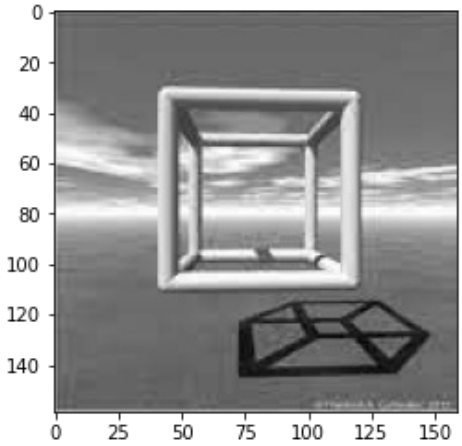



```
Left image
```

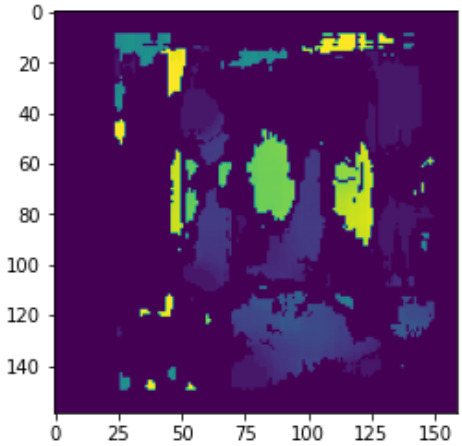




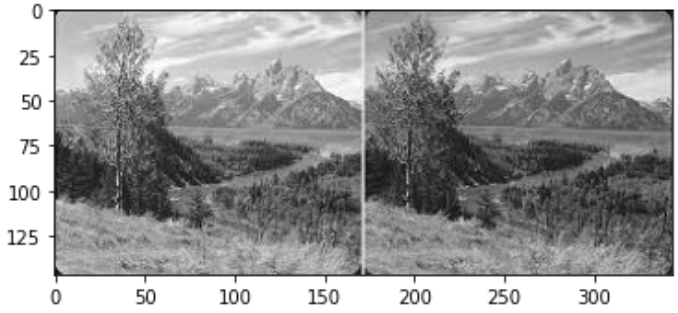
Right image



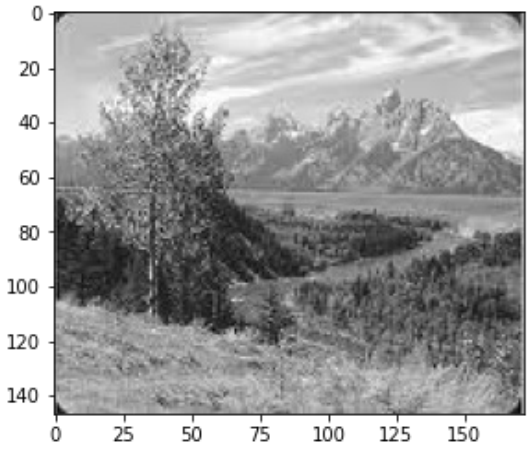
Disparity Map



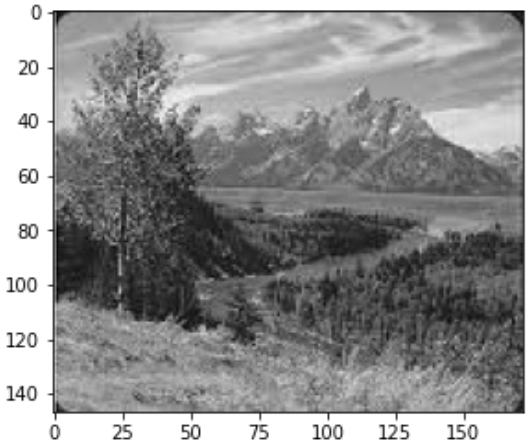
Orginal



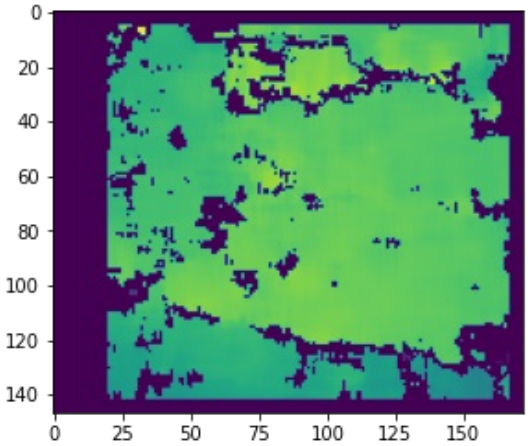
Left image



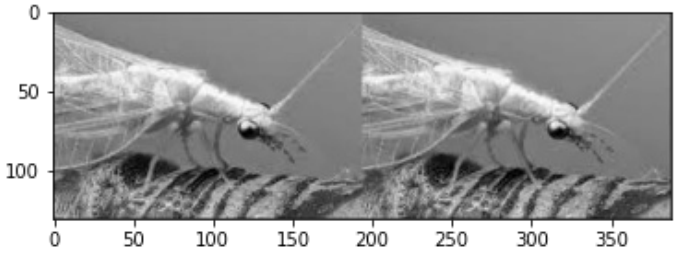
Right image



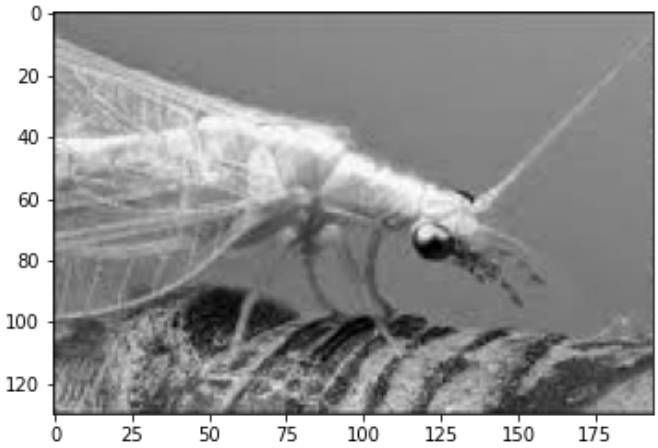
Disparity Map



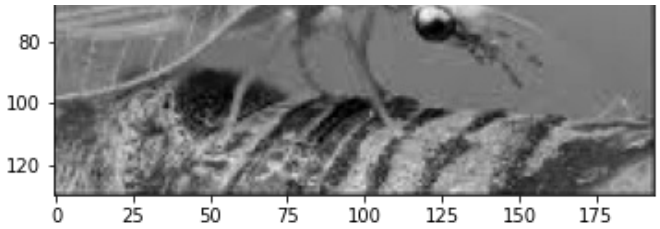
Orginal



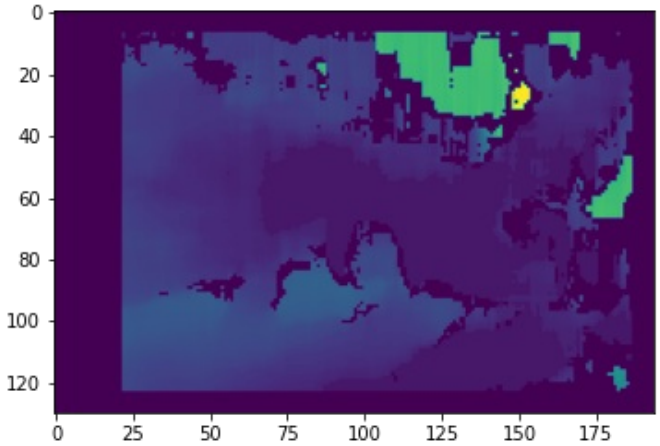
Left image



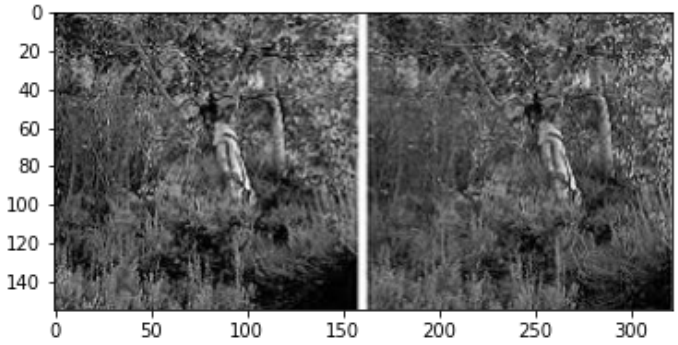
Right image



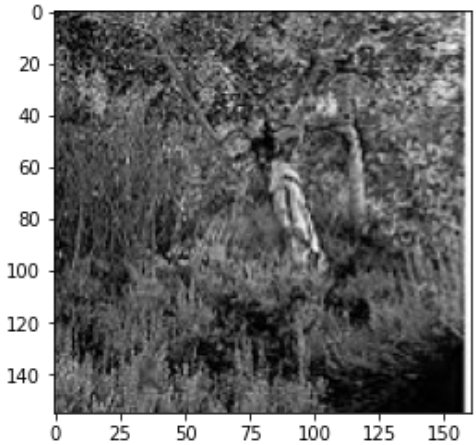
Disparity Map



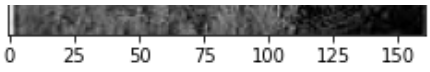
Orginal



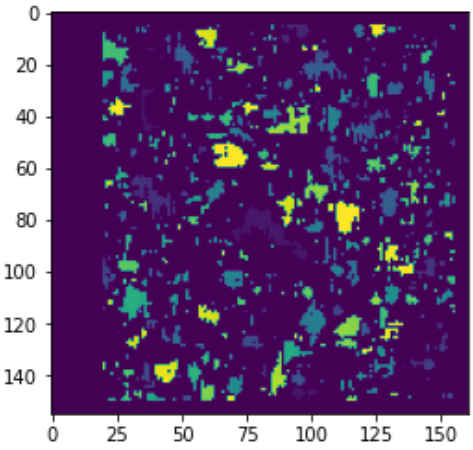
Left image



Right image



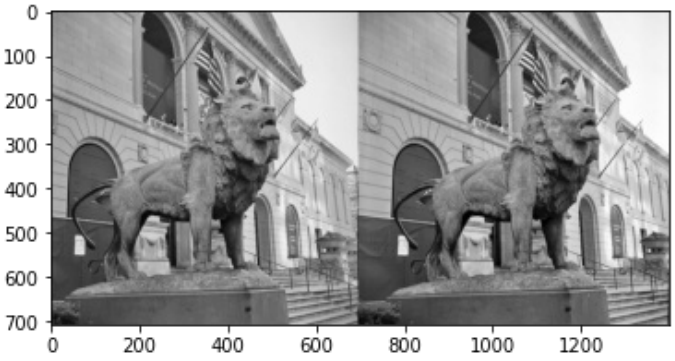
Disparity Map



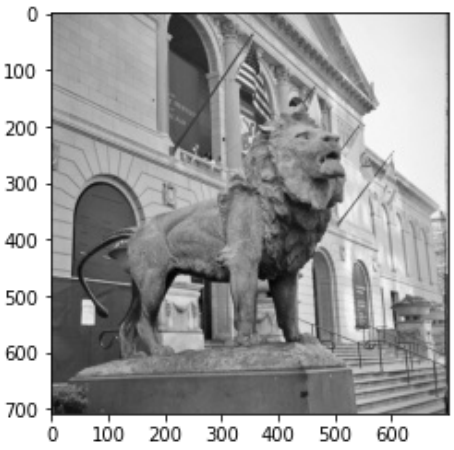
In [10]:

```
disp_map("img.jpeg",16,11)
disp_map("img2.jpg",16,15)
```

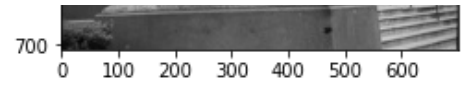
Orginal



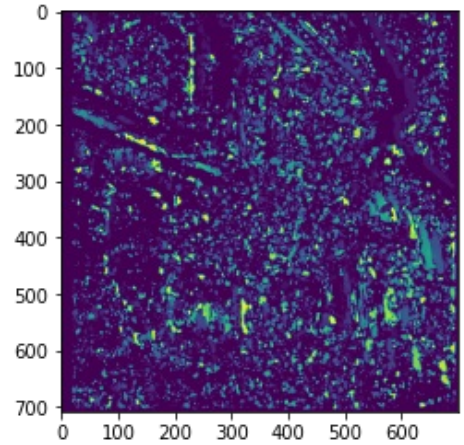
Left image



Right image

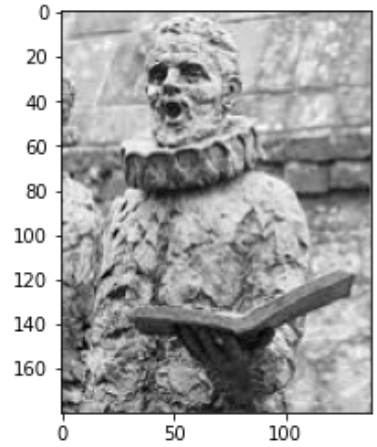


Disparity Map

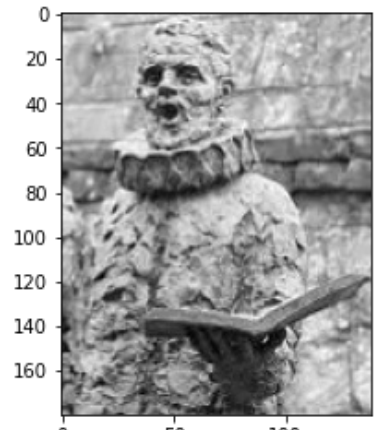


Orginal

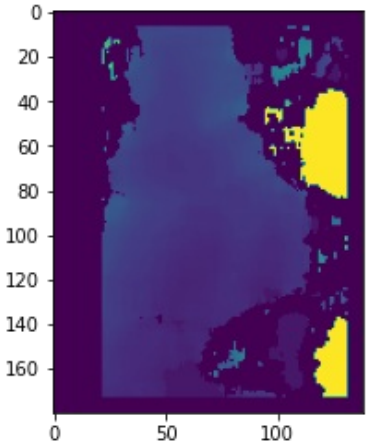
Left image



Right image



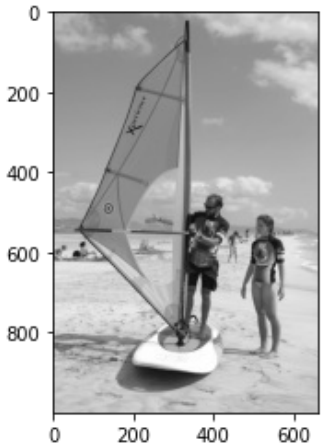
Disparity Map



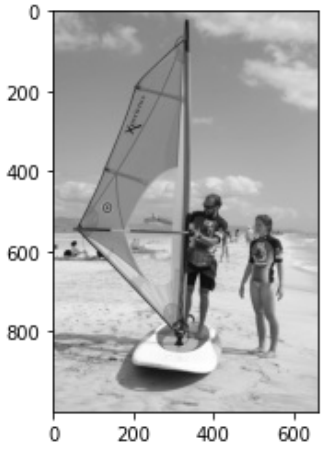
In [15]:

```
imgL=cv2.imread("left8.jpg",0)
print("Left image")
plt.imshow(imgL,cmap='gray')
plt.show()
imgR=cv2.imread("right8.jpg",0)
print("Right image")
plt.imshow(imgR,cmap='gray')
plt.show()
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=15)
disparity = stereo.compute(imgL,imgR)
print("Disparity Map")
plt.imshow(disparity,cmap="gray")
plt.show()
```

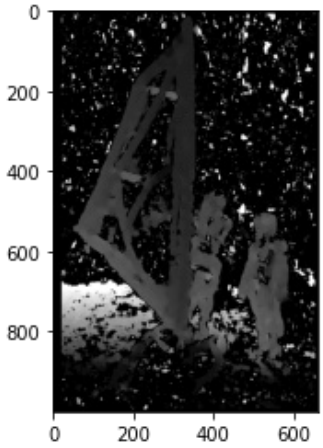
Left image



Right image



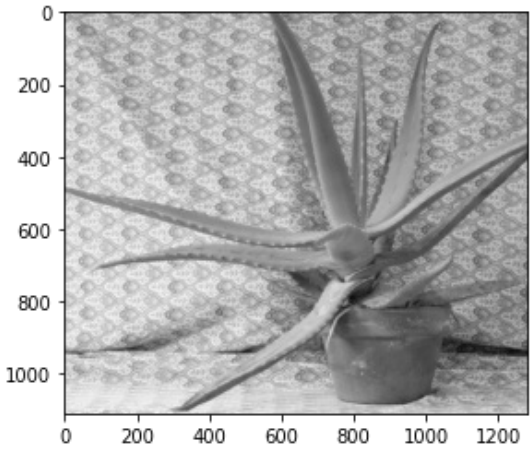
Disparity Map



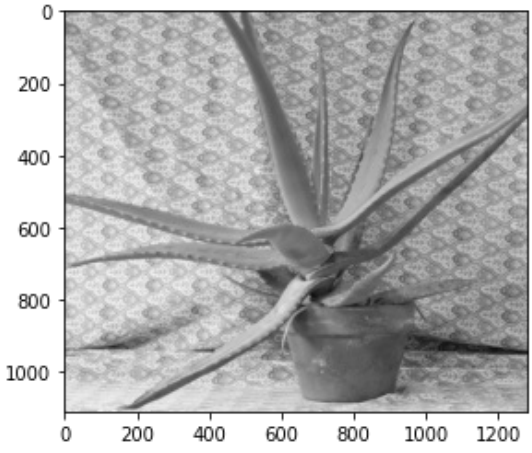
In [12]:

```
imgL=cv2.imread("img2L.jpg",0)
print("Left image")
plt.imshow(imgL,cmap='gray')
plt.show()
imgR=cv2.imread("img2R.jpg",0)
print("Right image")
plt.imshow(imgR,cmap='gray')
plt.show()
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=5)
disparity = stereo.compute(imgL,imgR)
print("Disparity Map")
plt.imshow(disparity,cmap="gray")
plt.show()
```

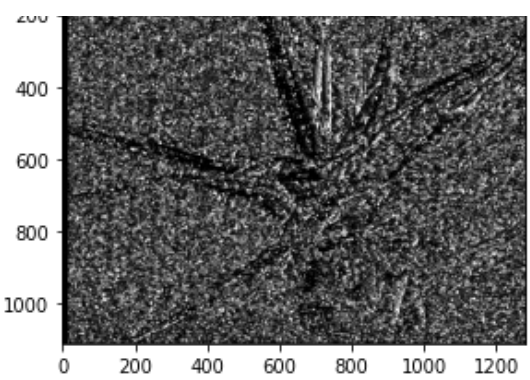
```
Left image
```




```
Right image
```




```
Disparity Map
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: