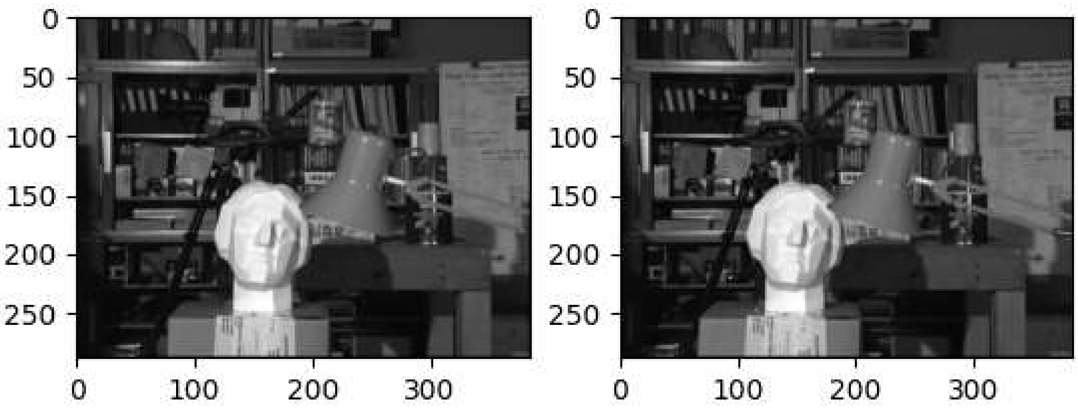
使用Opencv进行视差估计

In [ ]:

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
# 读取左右图像
left_image = cv2.imread('tsukuba_l.png', cv2.IMREAD_GRAYSCALE)
right_image = cv2.imread('tsukuba_r.png', cv2.IMREAD_GRAYSCALE)
plt.subplot(1,2,1)
plt.imshow(left_image,'gray')
plt.subplot(1,2,2)
plt.imshow(right_image,'gray')
plt.show()
```



基于视差图的立体匹配（Stereo Matching based on Disparity Maps）：

In [ ]:

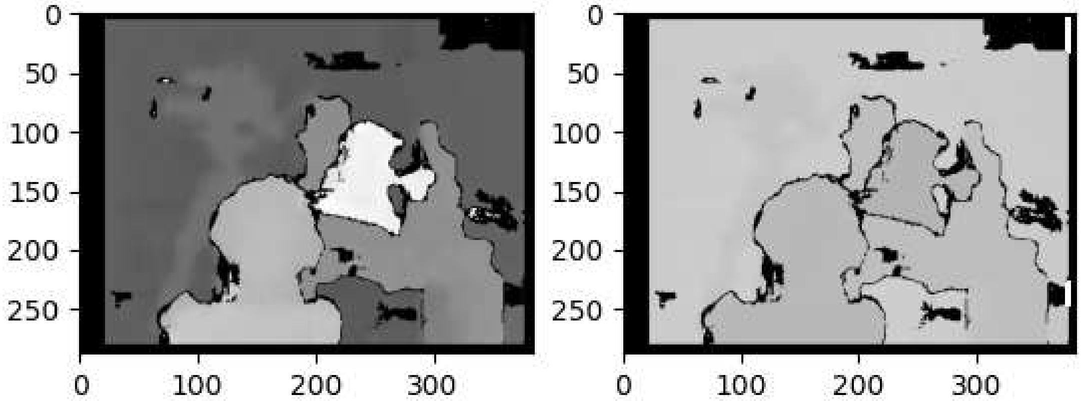
```
# 创建立体匹配对象
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=15)

# 计算视差图
disparity_map = stereo.compute(left_image, right_image)
plt.subplot(1,2,1)
plt.imshow(disparity_map,'gray')
# 可选：视差图后处理，如滤波、归一化等

# 将视差图转换为深度图
focal_length = 0.8  # 焦距
baseline = 0.1  # 立体摄像机基线
depth_map = baseline * focal_length / (disparity_map)

# 显示深度图
plt.subplot(1,2,2)
plt.imshow(depth_map,'gray')
plt.show()
```

```
C:\Users\10037\AppData\Local\Temp\ipykernel_17120\757759566.py:13: RuntimeWarni
ng: divide by zero encountered in divide
  depth_map = baseline * focal_length / (disparity_map)
```



使用Semi-Global Matching (SGM) 算法进行视差估计：

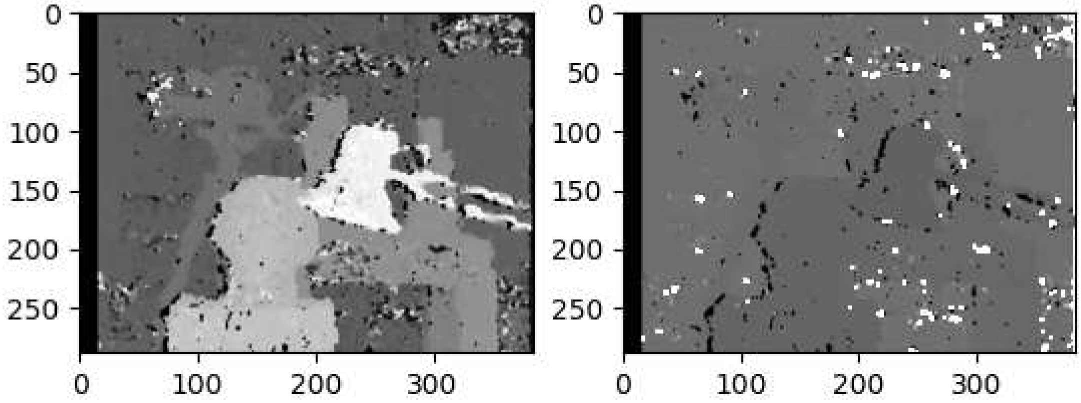
```
In [ ]: # 创建立体匹配对象
stereo = cv2.StereoSGBM_create(minDisparity=0, numDisparities=16, blockSize=3)

# 计算视差图
disparity_map = stereo.compute(left_image, right_image)
plt.subplot(1,2,1)
plt.imshow(disparity_map,'gray')
# 可选: 视差图后处理, 如滤波、归一化等

# 将视差图转换为深度图
focal_length = 0.8  # 焦距
baseline = 0.1  # 立体摄像机基线
depth_map = baseline * focal_length / (disparity_map)

# 显示深度图
plt.subplot(1,2,2)
plt.imshow(depth_map,'gray')
plt.show()
```

```
C:\Users\10037\AppData\Local\Temp\ipykernel_17120\628456969.py:13: RuntimeWarni
ng: divide by zero encountered in divide
  depth_map = baseline * focal_length / (disparity_map)
```


0
50
100
150
200
250
0
100
200
300
0
50
100
150
200
250
0
100
200
300