



Homework12 Stereo Vision

胡成成 2101210578

Question

计算下列两幅图的视差图（深度图）：

- (1) 比较BM和SGBM所获取的深度图的差别，并分析其原因；
- (2) 用实验分析BlockSize的大小对提取深度图算法的影响。



Answer

问题一回答

BM与SGBM代码实现

```
import cv2
import numpy as np

imgL = cv2.imread('im0.png')
imgR = cv2.imread('im1.png')

imgLG = cv2.cvtColor(imgL, cv2.COLOR_BGR2GRAY)
imgRG = cv2.cvtColor(imgR, cv2.COLOR_BGR2GRAY)

# stereo = cv2.StereoBM_create(numDisparities=32, blockSize=7)
stereo = cv2.StereoBM_create(numDisparities=64, blockSize=7)
disp = stereo.compute(imgLG, imgRG).astype(np.float32) / 16.0
```

```

cv2.namedWindow('BM disparity', cv2.WINDOW_FREERATIO)
cv2.imshow('BM disparity', (disp - 0) / 128)
cv2.waitKey(0)

# window_size = 7
# min_disp = 32 # 最小视差数
# num_disp = 288 - min_disp
# blockSize = window_size
# uniquenessRatio = 1 # 最佳比配代价优于次佳代价的比例
# speckleRange = 2 # 相连物体最大视差变化 *16 pixels
# speckleWindowSize = 3
# disp12MaxDiff = 200 # 左右最大允许视差 pixels
# P1 = 600 # 视差平滑控制参数1
# P2 = 2400 # 视差平滑控制参数2
#
# stereo = cv2.StereoSGBM_create(minDisparity=min_disp, numDisparities=num_disp, blockSize=window_size,
#                                uniquenessRatio=uniquenessRatio, speckleRange=speckleRange,
#                                speckleWindowSize=speckleWindowSize, disp12MaxDiff=disp12MaxDiff, P1=P1, P2=P2)

# disparity range is tuned for 'aloe' image pair
window_size = 3
min_disp = 16
num_disp = 192 - min_disp
stereo = cv2.StereoSGBM_create(minDisparity=min_disp,
                                numDisparities=num_disp,
                                blockSize=7,
                                P1=8 * 3 * window_size ** 2,
                                P2=32 * 3 * window_size ** 2,
                                disp12MaxDiff=1,
                                uniquenessRatio=10,
                                speckleWindowSize=100,
                                speckleRange=32
                                )

disp = stereo.compute(imgL, imgR).astype(np.float32) / 16.0
cv2.namedWindow('SGBM disparity', cv2.WINDOW_FREERATIO)
cv2.imshow('SGBM disparity', (disp - min_disp) / num_disp)
cv2.waitKey(0)

```

- BM算法计算结果



- SGBM算法计算结果



BM算法实现起来的优点就是快，缺点是深度图的效果不是很好。BM算法只能对8为灰度图像计算视差。BM算法原理是将两个摄像头的帧分成很多的小方块来机型匹配，通过移动小方块来匹配另一个图中的小方块，通过发现不同小方块在另一个图像中的像素点位置在结合两个摄像头的关系数据（标定的参数中的translate 和rotation矩阵）来计算出物体的实际深度从而生成相应的深度图。

SGBM算法，作为一种全局匹配算法，立体匹配的效果明显好于局部匹配算法，但是同时复杂度上也要远远大于局部匹配算法。

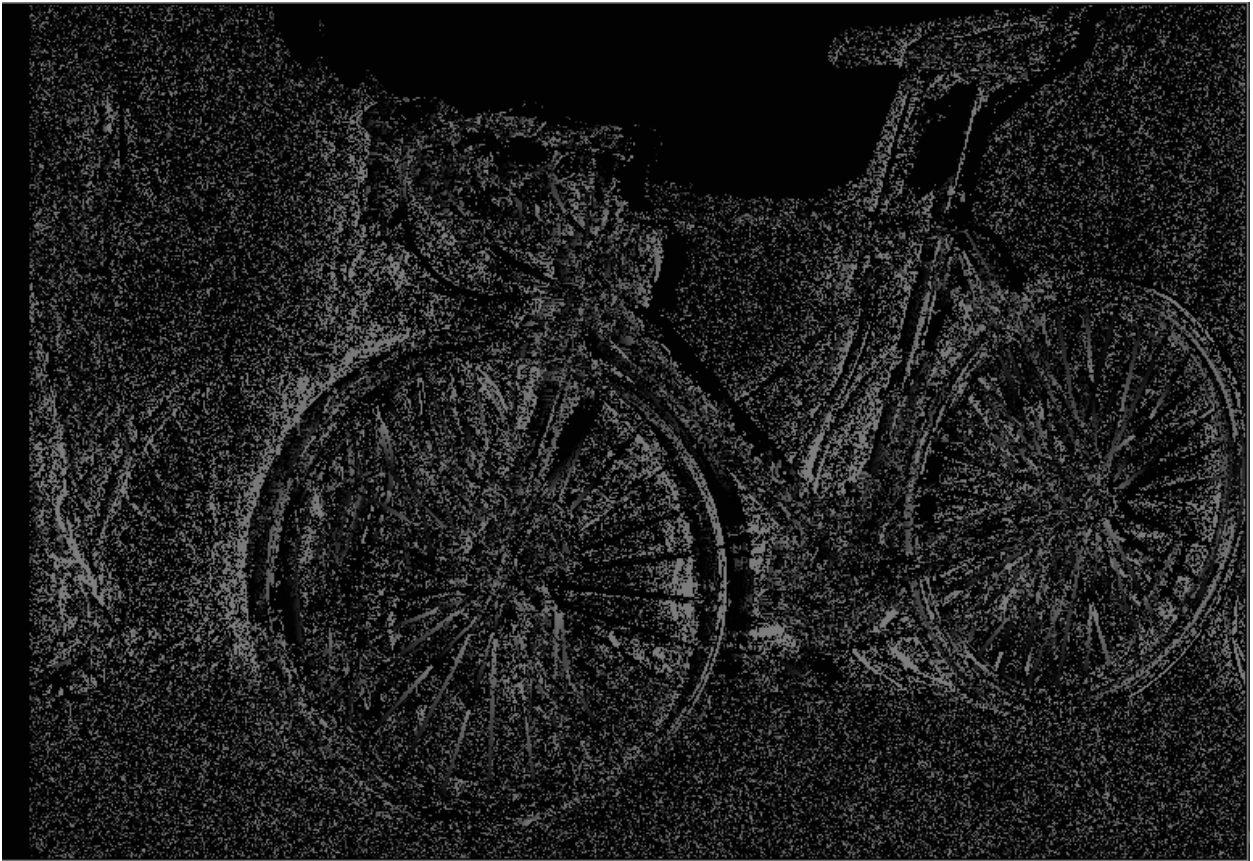
因此：

- SGBM的效果相比于BM算法的效果更好，因为SGBM算法是一种全局算法，能够综合到图片整体的信息，不会将图片因为分块导致的一些局部块信息失去联系。BM算法也正是因为分块匹配，只利用块的信息计算，而忽视了全局的信息
- 但从另一个方面考虑，BM算法的计算速度明显快于SGBM算法。因为SGBM从全局上计算图片，其复杂度远远大于BM算法。

问题二回答

对于BM算法

- BlockSize=5



- BlockSize=7



- BlockSize=9



- BlockSize=11



- BlockSize=21



对于SGBM算法

- BlockSize=5



- BlockSize=7



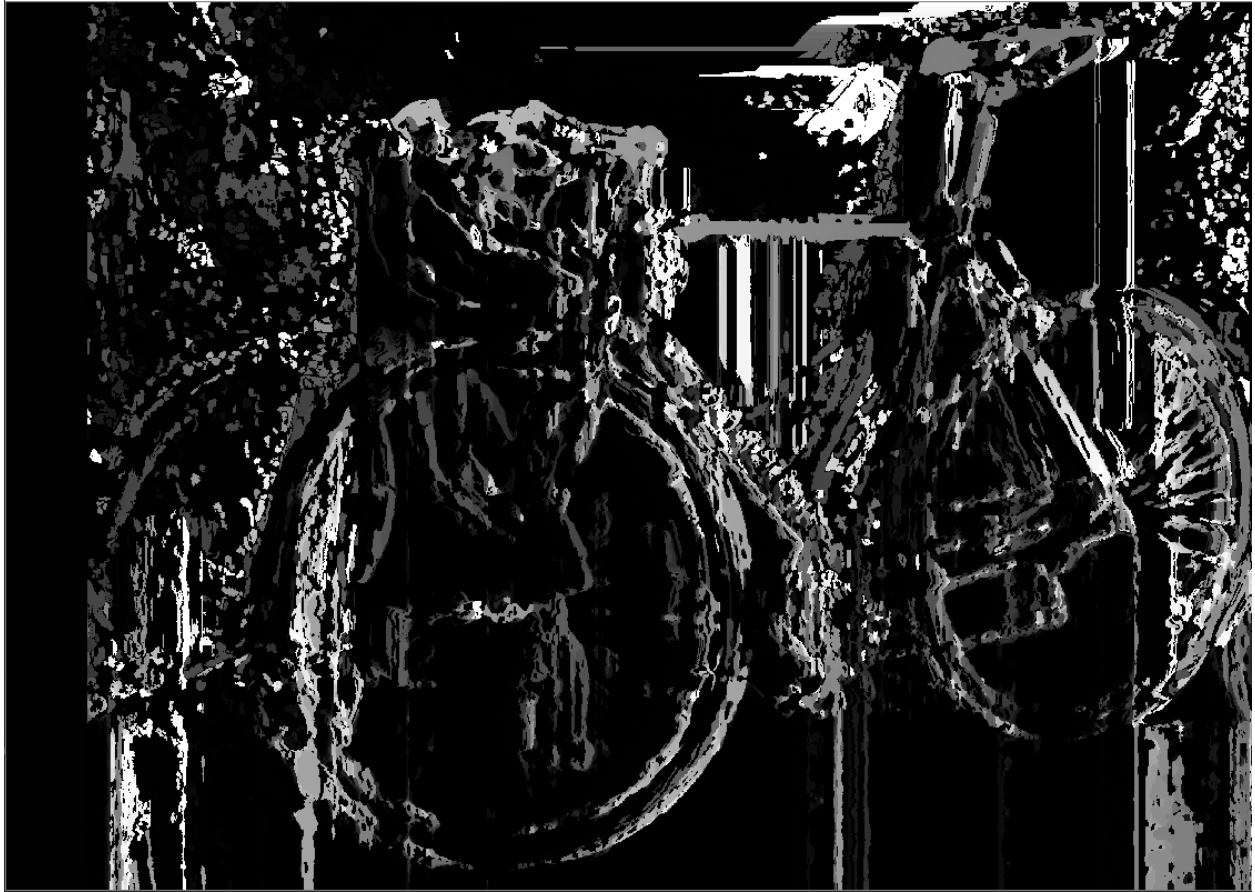
- BlockSize=9



- BlockSize=11



- BlockSize=21



- 总结：综合BM和SGBM算法的BlockSize对比，发现BlockSize越大，图片看上去越光滑，噪点也会越少，但是会丢失一些细节纹理。
- 分析原因：可能是BlockSize越大，参与计算的Block越大，参与计算的方块大小变大了，计算的像素点变多了，一些细粒度的噪声就会被均匀掉，因此图片看上去也会越平滑。