获取视差图

首先读取两幅图片，然后得到图片的通道数，设置窗口的大小。然后利用opencv提供的全局匹配函数cv2.StereoSGBM\_create()函数创建stereo，然后再利用stereo.compute()函数计算得到视差图。

相关的参数设置在下方代码的cv2.StereoSGBM\_create()函数中体现。

*def* getDisparity():

*#读取彩色图片*

    imgr = cv2.imread("RIma-000023.png")

    imgl = cv2.imread("LIma-000023.png")

*#设置参数*

    window\_size = 7

    channels = imgr.shape[2]

*#计算视差图*

    stereo = cv2.StereoSGBM\_create(

*#参数设置*

*minDisparity* = 0, *#最小视差*

*blockSize* = block\_size, *#窗口大小*

*#以下参数设置为官方推荐*

*#P1与P2为动态规划有关参数*

*P1* = 8\*channels\*block\_size\*block\_size,

*P2* = 32\*channels\*block\_size\*block\_size,

*disp12MaxDiff* = 1, *#左视差图与右视差图的最大差异*

*uniquenessRatio* = 10, *#次低代价>=最低代价\*(1+uni/100)*

*speckleWindowSize* = 100, *#平滑区域最大尺寸*

*speckleRange* = 10 *#视差变化的阈值*

    )

    disparity = stereo.compute(imgl,imgr)

    return disparity

计算深度图

得到视差图之后，便可以根据公式得到深度，公式如下



对此，遍历视差图中每个像素，计算每个像素对应的深度，最终获得视差图。

*def* getDepth(*dis*):

    f = 7 *#焦距，单位为毫米*

    B = 500 *#基线， 单位为毫米*

    w\_of\_pixel = 0.0064 *#每个像素的宽度，单位为毫米*

    [row, col] = dis.shape *#获取形状*

*#定义空图片*

    depth = np.zeros((row, col))

    for r in range(row):

        for c in range(col):

*#计算视差实际值*

            D = dis[r][c] \* w\_of\_pixel

*#如果视差为0，说明深度无穷大*

            if D == 0.0:

                Z = *float*("inf")

            else:

                Z = (B \* f) / D

            depth[r][c] = Z

    return depth

实验结果



