**2018年第4次课习题答案**

**——Vance吴方熠**



答：

图像去畸变过程：

1. 用书中（5.14）式算出无畸变图各像素点对应的理论坐标x\_ud,y\_ud；
2. 计算坐标点到原点的距离r2 **=** x\_ud**\***x\_ud **+** y\_ud**\***y\_ud；
3. 根据参数，由（5.13）式算出各点的畸变坐标；
4. 再由（5.14）式算出各坐标对应的畸变像素坐标（即在畸变图上的位置，不一定为整数）
5. 校正图像，校正后的图像各像素点的灰度值等于该像素点对应的畸变像素坐标的灰度值。（注意可用最近邻差值法解决像素坐标不为整数的问题）

自编代码部分如下所示：

|  |  |
| --- | --- |
| 30  31  32  33  34  35  36  37  38  39  40 | // start your code here  double x\_ud **=** **(**u **-** cx**)/**fx**;**  double y\_ud **=** **(**v **-** cy**)/**fy**;**  double r2 **=** x\_ud**\***x\_ud **+** y\_ud**\***y\_ud**;**  double r4 **=** r2**\***r2**;**  double x\_d **=** x\_ud**\*(**1 **+** k1**\***r2 **+** k2**\***r4**)** **+** 2**\***p1**\***x\_ud**\***y\_ud **+** p2**\*(**r2 **+** 2**\***x\_ud**\***x\_ud**);**  double y\_d **=** y\_ud**\*(**1 **+** k1**\***r2 **+** k2**\***r4**)** **+** p1**\*(**r2 **+** 2**\***y\_ud**\***y\_ud**)** **+** 2**\***p2**\***x\_ud**\***y\_ud**;**  u\_distorted **=** fx**\***x\_d **+** cx**;**  v\_distorted **=** fy**\***y\_d **+** cy**;**  // end your code here |

程序运行的结果截图如下所示：

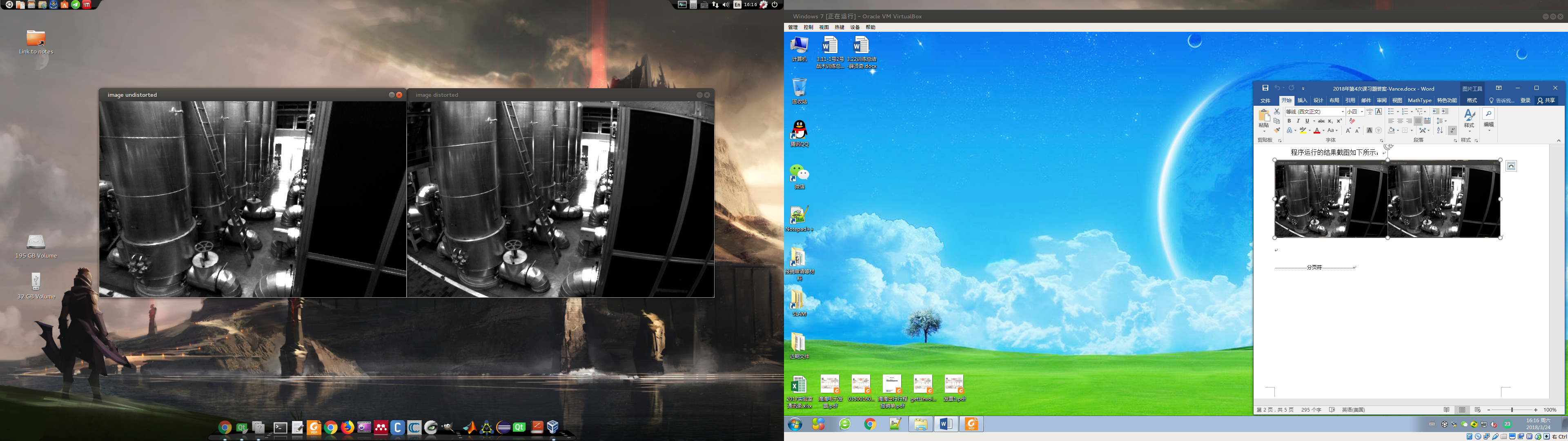
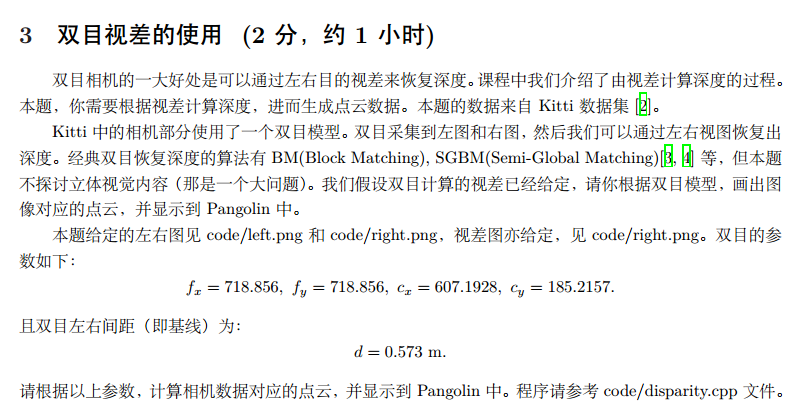
****

图1 图像去畸变（左）和未去畸变（右）的效果



答：

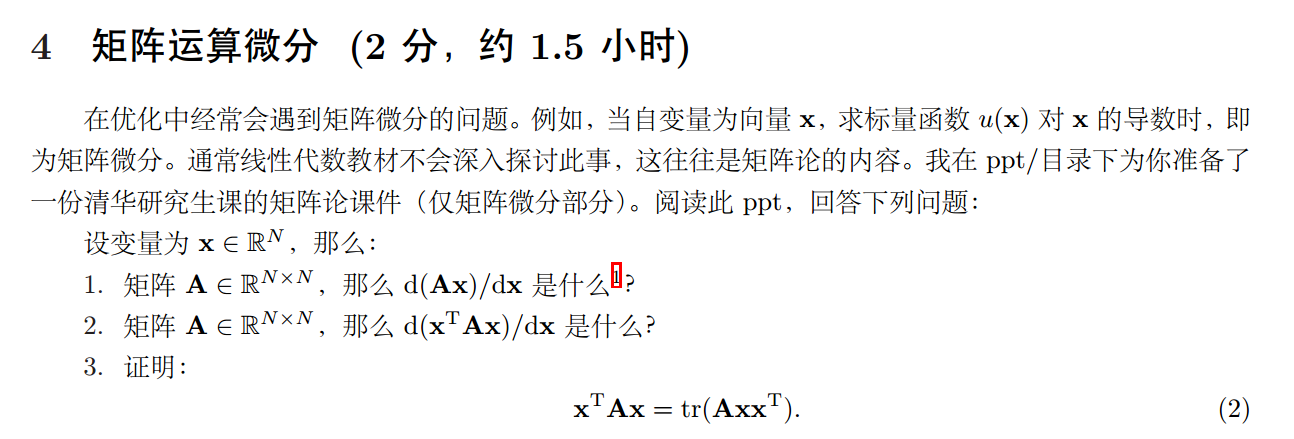
自编代码部分如下所示：

|  |  |
| --- | --- |
| …  26  27  …  43  44  45  46  47  48  49  50  51  … | ……  // 间距baseline  double b **=** 0.573**;**  ……  // start your code here (~6 lines)  // 根据双目模型计算 point 的位置  point**[**2**]** **=** fx **\*** b **/** disparity**.**at**<**uchar**>(**v**,** u**);**  point**[**0**]** **=** **(**u **-** cx**)** **\*** point**[**2**]** **/** fx**;**  point**[**1**]** **=** **(**v **-** cy**)** **\*** point**[**2**]** **/** fy**;**  pointcloud**.**push\_back**(**point**);**  // end your code here  **}**  …… |

程序运行结果如下所示：



图2 由双目视差计算出的深度图效果



解：

1.常数矩阵**A**与向量的微分

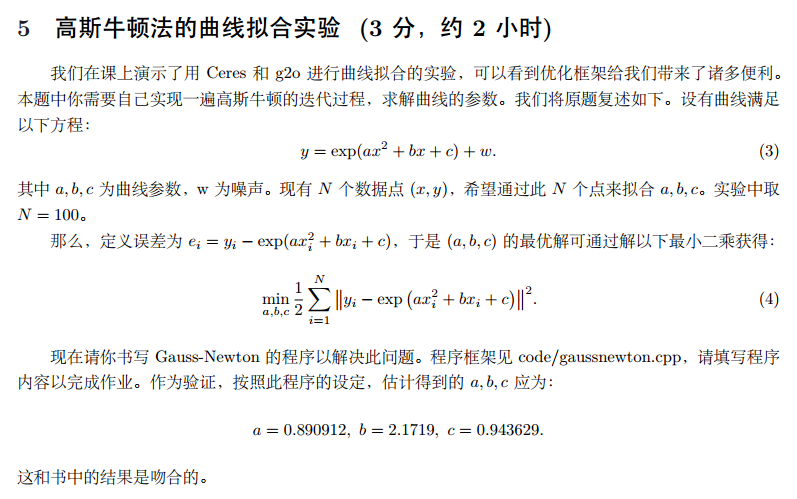
2.二次型运算

3.证明：

将两侧同时对常数矩阵A取导，有：

因为求导的A为常数矩阵，故有

得证。

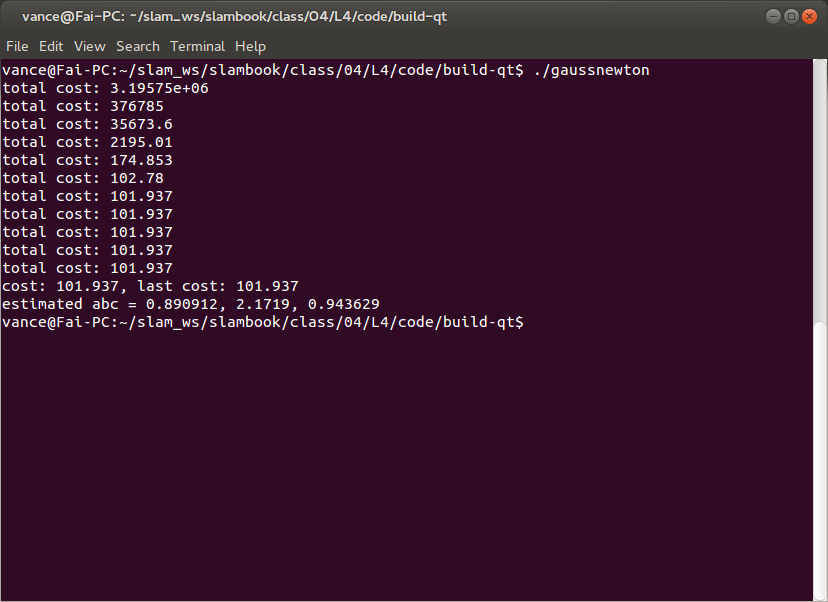


答：

自编代码部分如下所示：

|  |  |
| --- | --- |
| …  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  … | ……  **for** **(**int i **=** 0**;** i **<** N**;** i**++)** **{**  double xi **=** x\_data**[**i**],** yi **=** y\_data**[**i**];** // 第i个数据点  // start your code here  double error **=** 0**;** // 第i个数据点的计算误差  error **=** yi **-** exp**(**ae**\***xi**\***xi **+** be**\***xi **+** ce**);** // 填写计算error的表达式  Vector3d J**;** // 雅可比矩阵  J**[**0**]** **=** **-**exp**(**ae**\***xi**\***xi **+** be**\***xi **+** ce**)\***xi**\***xi**;** // de/da  J**[**1**]** **=** **-**exp**(**ae**\***xi**\***xi **+** be**\***xi **+** ce**)\***xi**;** // de/db  J**[**2**]** **=** **-**exp**(**ae**\***xi**\***xi **+** be**\***xi **+** ce**);** // de/dc  H **+=** J **\*** J**.**transpose**();** // GN近似的H  b **+=** **-**error **\*** J**;**  // end your code here  cost **+=** error **\*** error**;**  **}**  // 求解线性方程 Hx=b，建议用ldlt  // start your code here  Vector3d dx**;**  dx **=** H**.**ldlt**().**solve**(**b**);**  // end your code here  …… |

程序运行结果如下所示：



结果和答案吻合。



解：

参考《State Estimation for Robotics》一书第3.1节，有：

1.

2.

3. 存在唯一解，解为：