图像处理第三次作业

蒋康

June 16, 2024

1

1.1

双目立体相机主要是模拟人眼的工作原理，立体相机系统由两台相机组成，通常放置在一定距离的水平线上，模拟人类的双眼。通过同时拍摄同一场景的两张图像，我们可以通过比较这两张图像中的对应点（同一个物体在两张图像中的位置）来估算深度信息。视差是指同一物体在两张图像中的位置差异，设左相机图像中的一个点的坐标为 (x_L, y)，右相机图像中的对应点的坐标为 (x_R, y)，则视差 d 可以表示为 $d = x_L - x_R$，通过 $Z = \frac{B \cdot f}{d}$ 来计算，其中 B 是两相机的间距。但是由于相机的安装位置和角度可能不完全一致，直接拍摄的两张图像往往不能直接用于视差计算。为了简化视差计算，需要对图像进行校正，使得两张图像中的对应点在同一行上：通过相机标定（Calibration）将图像重新投影到一个公共的平面，使得每对对应点行对齐。

1.2

1. 按照算法中的假设得到光流约束方程: $I_x u + I_y v + I_t = 0$
2. 根据方程的要求计算 I 对 x,y,t 的偏导数: $I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}$，在代码实现中 I_x, I_y 可以通过 numpy 的 sobel 或者 gradient 函数计算，I_t 可以通过两个帧作差得到
3. 在一个窗口内得到多个光流约束方程构成方程组如下：

$$\begin{aligned} I_x u_1 + I_y v_1 &= -I_t \\ I_x u_2 + I_y v_2 &= -I_t \\ &\cdots \\ I_x u_n + I_y v_n &= -I_t \end{aligned} \tag{1}$$

表示为：$A\mathbf{v} = \mathbf{b}$

4. 窗口大小肯定大于 2，所以方程是超定的，用最小二乘法求解得 $\mathbf{v} = (A^T A)^{-1} A^T \mathbf{b}$

5. 移动窗口求解，求得整个图像的估计光流场。

1.3

pyramid representation 通过对图像进行逐级降采样，生成一系列分辨率不同的图像，可以在 Lucas-Kanade 方法中起到如下作用：

- 处理大范围运动
  在低分辨率的图像中，大范围的运动会变得相对较小，因此可以在较低分辨率图像上进行初步估计。这样一来，即使是原始图像中很大的运动，在低分辨率图像中也可能变得足够小，以便 Lucas-Kanade 方法能够处理。

- 避免局部最小值问题
  如果直接在高分辨率图像上进行大范围运动估计，容易陷入局部最小值。通过金字塔表示法，在低分辨率图像上进行初步估计，然后逐层细化，可以有效地避免这个问题。

- 提高计算效率
  低分辨率图像包含的像素更少，计算速度更快。因此，先在低分辨率图像上进行估计，可以减少整体计算量。

2

2.2

(a)

实现 compute_error 之前，最后有效的 keypoints：

Figure 1: error_before

实现 compute_error 之后，最后有效的 keypoints：

Figure 2: error_after

在实现 compute_error 之前，播放运动过程发现很多 keypoints 会飘起来，实现 compute_error 之后，这些点直接被过滤掉了。

FeedBack

在实现计算 disparity map 时要求中有写到：Note that you need to detect and mask out occluded and dis-occluded area in the disparity map, in which the estimated disparity values are not reliable.

我搜索了一下网上的资料都是要通过 dispartity left 和 disparity right 来算，没有办法直接在 disparity map 中进行操作，看了一下 ipynb 文件中也没有需要我编写相关操作的地方，所以我不太确定这里需不需要我完成这个。