

计算机视觉与应用

实践

练习5——图像视差匹配

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名:** | 林雅惠 | **学 号:** | 123106222874 |
| **学 院:** | 计算机科学与工程学院 | | |

**2024年5月13日**

目录

[一、 立体匹配实验原理 3](#_Toc29981)

[二、 基于归一化互相关的匹配算法 4](#_Toc7113)

[2.1实验原理 4](#_Toc9670)

[2.1.1窗口匹配方法 4](#_Toc18940)

[2.1.2归一化互相关 4](#_Toc8776)

[2.2实验方法 5](#_Toc26704)

[2.2.1NCC版本和高斯版本的视差匹配 5](#_Toc4194)

[2.2.2不同窗口值下的NCC视差匹配 6](#_Toc13535)

[2.3实验结果及分析 7](#_Toc4025)

[三、 基于置信传播的匹配算法 8](#_Toc2791)

[3.1实验原理 8](#_Toc2074)

[3.2实验方法 8](#_Toc20092)

[3.2.1计算数据代价 8](#_Toc25305)

[3.2.2计算能量 9](#_Toc19175)

[3.2.3消息更新 9](#_Toc20539)

[3.2.4消息图归一化 10](#_Toc16631)

[3.2.5计算置信度 10](#_Toc261)

[3.2.6立体匹配 11](#_Toc3274)

[3.3实验结果与分析 11](#_Toc15778)

# 立体匹配实验原理

立体匹配是计算机视觉中用于获取两幅图像之间的深度信息的一种技术。其中，视差图的是描述两幅图像中对应像素之间的偏移量的图像，立体匹配算法主要包括基于区域的匹配算法、基于特征的匹配算法、基于深度学习的匹配算法和基于光流的匹配算法。

立体匹配一般包含以下四个步骤：

1. 匹配代价计算

计算匹配代价，即计算参考图像上每个像素点IR(P) ，以所有视差可能性去匹配目标图像上对应点IT(pd)的代价值，因此计算得到的代价值可以存储在一个h∗w∗d(MAX)的三维数组中，通常称这个三维数组为视差空间图。

1. 代价聚合

通常全局算法不需要代价聚合，而局部算法需要通过求和、求均值或其他方法对一个支持窗口内的匹配代价进行聚合而得到参考图像上一点p在视差d处的累积代价CA(p,d)，这一过程称为代价聚合。通过匹配代价聚合，可以降低异常点的影响，提高信噪比进而提高匹配精度。代价聚合策略通常是局部匹配算法的核心，策略的好坏直接关系到最终视差图（Disparity maps）的质量。

1. 视差计算

局部立体匹配算法的思想，在支持窗口内聚合完匹配代价后，获取视差的过程通常采用‘胜者为王’策略（WTA，Winner Take All）,即在视差搜索范围内选择累积代价最优的点作为对应匹配点，与之对应的视差即为所求的视差。即P点的视差为d=argminCA(p,d)，但是这种方法可能导致噪声大，因为如果直接用点的像素值坐标匹配，风险太大，找到的匹配点不一定是正确的，这时可以采用窗口匹配方法。

1. 后处理

一般的，分别以左右两图为参考图像，完成上述三个步骤后可以得到左右两幅视差图像。但所得的视差图还存在一些问题，如遮挡点视差不准确、噪声点、误匹配点等存在，因此还需要对视差图进行优化，采用进一步执行后处理步骤对视差图进行修正。常用的方法有插值（Interpolation）、亚像素增强（Subpixel Enhancement）、精细化（Refinement）、图像滤波（Image Filtering）等操作。

# 基于归一化互相关的匹配算法

## 2.1实验原理

### 2.1.1窗口匹配方法

在立体匹配步骤中的视差计算中，如果使用点匹配方法的话，可能导致噪声大，因为直接找最匹配的点误差太大。这时可以采用窗口匹配方法，它的主要思路是：以当前点为中心，切出一个图像块，每个图像块依次做对比，找到最接近的图像块。这时就不只是拿单个点做对比了，可以有效减小误差。

### 2.1.2归一化互相关

相似性测度函数用于度量参考图像中的匹配基元和目标图像中的匹配基元的相似性，即判断参考图像和目标图像中两点为对应匹配点的可能性，也称为匹配代价，这里记为C(x,y,d)，表示像素点(x,y)在视差为d情况下的匹配误差。常见的匹配代价包括绝对差值和、差值平方和、截断绝对差值和和归一化互相关NCC。

其中归一化互相关的原理是对与原始图像内任意一个像素点，构建一个n\*n的邻域作为匹配窗口，然后对于目标像素位置，同样构建一个n\*n大小的匹配窗口，对两个窗口进行相似度度量。对两幅图像来说，在进行NCC计算之前要对图像处理，也就是将两帧图像校正到水平位置，即光心处于同一个水平线上，此时极线是水平的，否则匹配过程只能在倾斜的极线方向上完成。其中NCC的计算公式如图1所示，其中 NCC(p,d)得到的值得范围将在[−1,1]之间。

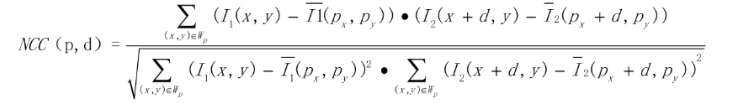


图1

本实验实现的是基于区域的匹配算法，比较图像中对应窗口之间的相似度来确定像素之间的视差。在匹配过程中，使用归一化互相关来衡量窗口之间的相似度，然后选择最高相关性对应的视差作为匹配结果。以下分别是基于归一化互相关的平面扫描匹配算法和基于高斯加权滤波器的平面扫描匹配算法的原理及步骤：

1. 基于归一化互相关的平面扫描匹配算法

该方法是立体匹配中最经典和常用的方法之一，原理是利用两幅图像之间的互相关来计算它们的相似度。通过对互相关结果进行归一化处理，可以使得匹配结果不受信号强度的影响，只受到信号形状的影响，从而提高匹配的准确性和鲁棒性。

1. 基于高斯加权滤波器的平面扫描匹配算法

该方法在基于归一化互相关的算法上进行了改进，加入了高斯加权滤波器来平滑图像，减少噪声的影响。通过对图像进行加权平均，可以减弱噪声的干扰，提高匹配的准确性。

## 2.2实验方法

### 2.2.1NCC版本和高斯版本的视差匹配

在本实验中，首先定义一个基于归一化互相关的平面扫描匹配方法，该方法中，输入为左右两幅灰度图像以及一些相关的参数，包括视差搜索的起始位置、搜索的步数、计算均值和方差的窗口大小。

1. 首先，计算图像块的平均值，对左右两幅图像进行平均滤波。
2. 然后，计算归一化图像，即原始图像减去均值。对于每一个视差值，计算互相关系数。
3. 最后，为每个像素选择最佳深度。对于每一个像素，选择具有最大归一化互相关系数的视差作为最佳视差。

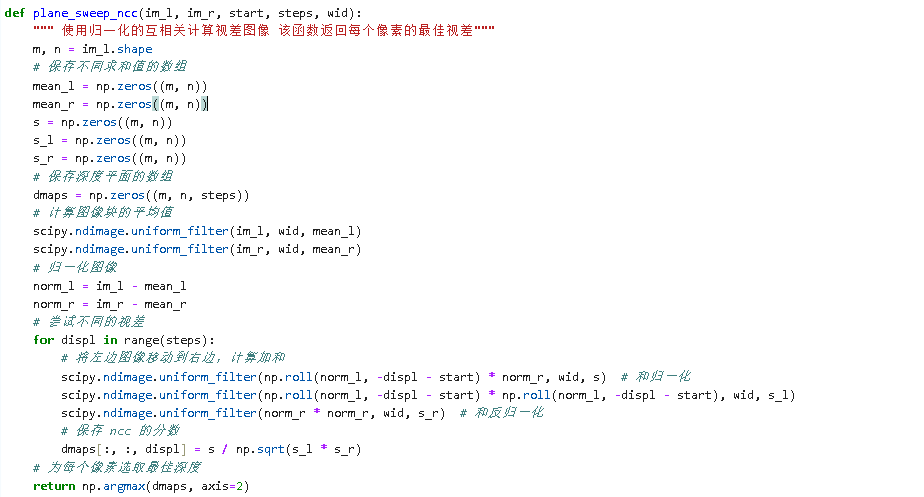


图2

以下函数实现了带高斯加权的归一化互相关方法，与之前的函数方法相似，只是在计算互相关时使用了高斯滤波器，高斯滤波器使得周围像素对当前像素的影响更加平滑，可以降低误差。

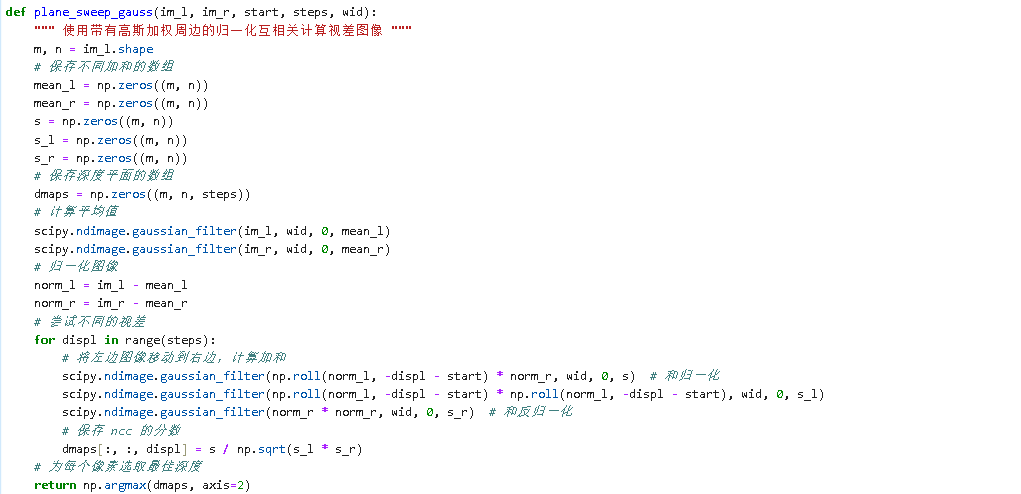


图3

主函数中读取两张图像，设置不同的参数，比如步长、宽度和起始视差，然后调用之前定义的两个函数进行视差计算，最后的将原始图像和视差估计的结果可视化输出。

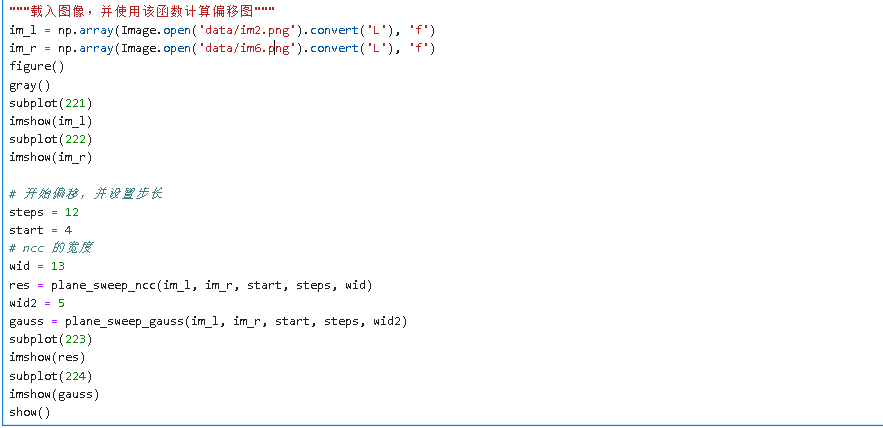


图4

### 2.2.2不同窗口值下的NCC视差匹配

本实验在实验一的基础上，设置多组不同窗口值的NCC计算，循环调用基于归一化互相关的平面扫描匹配方法，计算不同窗口值下的视差结果。



图5

## 2.3实验结果及分析

实验一得到结果如图６所示，在标准版本NCC中设置wid=13，在高斯版本中设置wid=5，左下方图像为基于归一化互相关的平面扫描匹配算法重建的视差图，右下方图像为基于高斯加权滤波器的平面扫描匹配算法重建的视差图。可以看到，与标准版本相比，高斯版本具有较少的噪声，但是相对缺少很多细节。

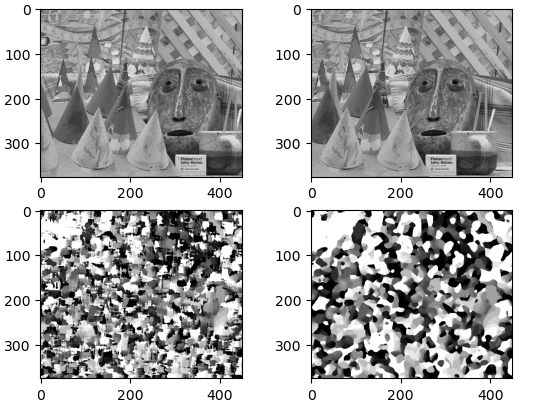


图6

实验二得到结果如图7所示，可以看到窗口值越小，能得到更多的细节，当窗口值值越小时，进行滤波时考虑的因素少，只受某像素点自身影响以及周围少量像素点的影响，所有保留下来更多的自身特征，但是同时噪声也大。当窗口值增大时，鲁棒性增强，噪声明显减少，但是图像的细节也在逐渐减少。

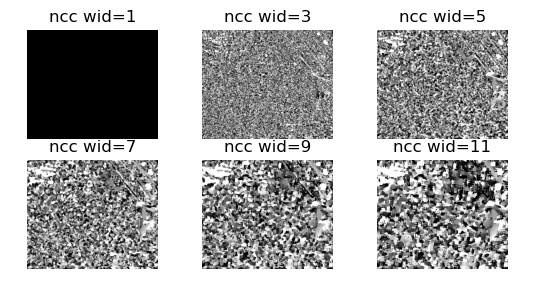


图7

# 基于置信传播的匹配算法

## 3.1实验原理

基于置信传播的立体匹配算法的原理如下：

1. 能量函数建模：首先，将立体匹配问题建模为一个能量最小化的问题，引入能量函数，其中包括数据项（代价函数）和正则项（平滑项），目标是最小化整体能量以获得最佳的视差图。
2. 消息传递：利用置信传播算法，通过相邻像素之间传递消息来更新每个像素点的置信度。具体地，将视差图分解为一组未归一化的消息，然后通过迭代传递消息来更新这些消息。
3. 置信度计算：基于更新后的消息，计算每个像素每个视差值的置信度，可以采用最大后验概率估计或其他方法来计算置信度。
4. 最大后验估计：根据计算得到的置信度，采用最大后验估计来确定每个像素点的最终视差值，即选择具有最高置信度的视差值作为该像素的视差值。
5. 迭代更新：重复进行消息传递、置信度计算和最大后验估计的步骤，直到收敛或达到预设迭代次数。

## 3.2实验方法

### 3.2.1计算数据代价

函数接收左视图、右视图、视差值的数量和阈值参数，对每个视差值进行循环处理，使用NumPy函数np.roll进行图像的平移操作，计算数据代价并使用np.minimum函数将其与阈值进行比较取小。

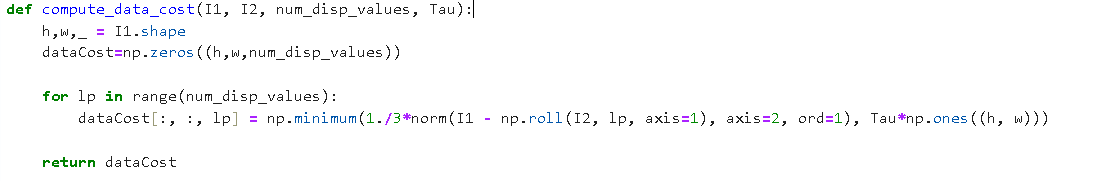


图8

### 3.2.2计算能量

函数实现一个计算能量函数，用于评估立体匹配算法生成的视差图像的拟合程度。先根据输入的数据代价和视差图像计算出所有像素的代价值相加，作为能量值的一部分。然后，通过比较相邻像素的视差值，计算相邻像素之间的交互成本，并将其添加到能量中。

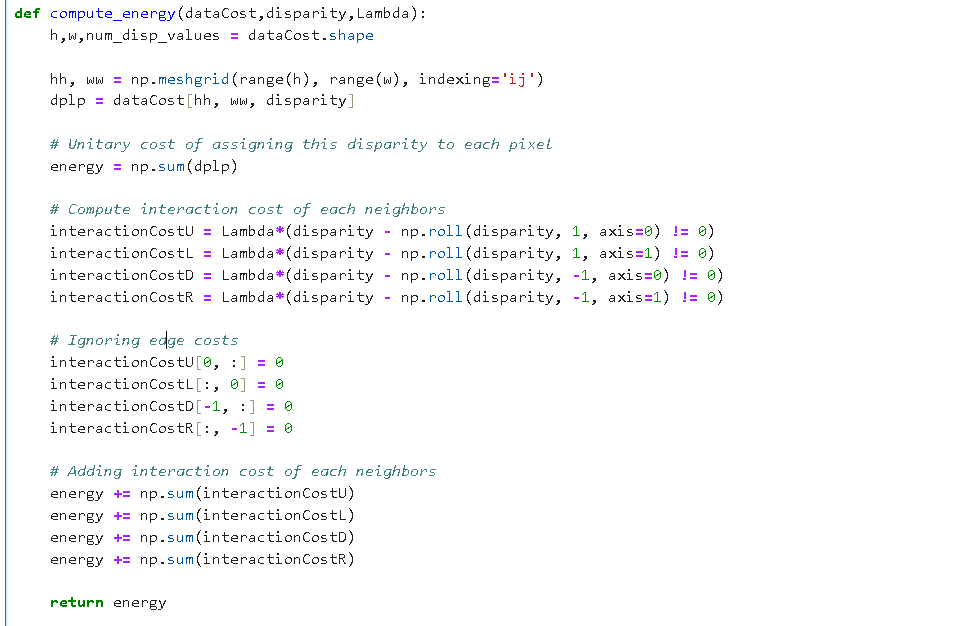


图9

### 3.2.3消息更新

函数用于更新消息图，用于后续的立体匹配算法中。函数接收旧的消息图和恶数据代价作为输入，根据旧消息图计算出各个方向上传入的消息。然后，根据最小化操作更新每个方向上的消息图，得到更新后的消息图，用于下一轮的消息传递。

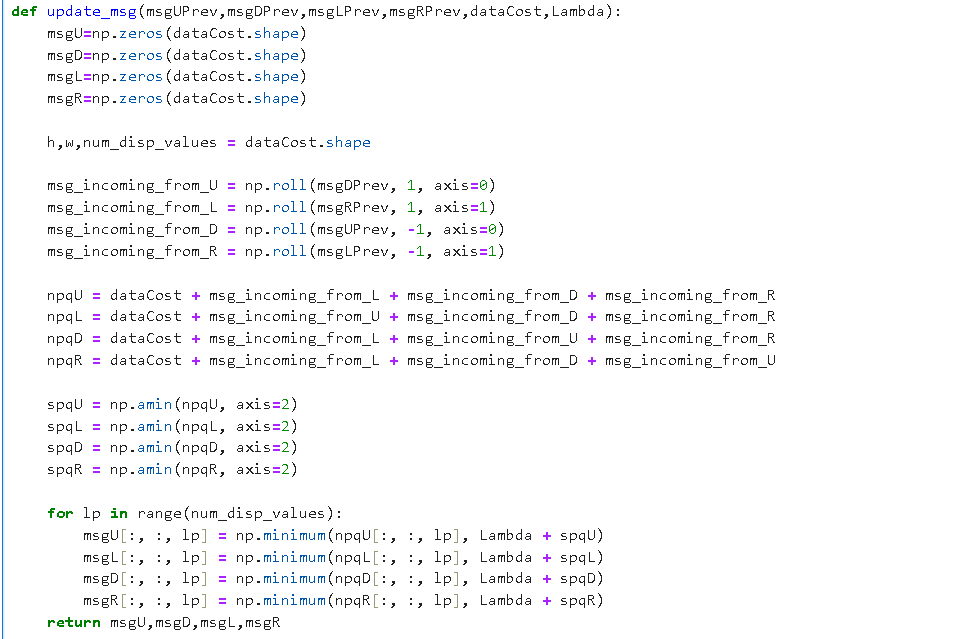


图10

### 3.2.4消息图归一化

函数用于对传入的消息图进行归一化处理。函数接收四个方向上的消息图作为输入，首先计算出每个像素点在深度维度上的平均值。然后，将每个像素点的消息减去对应的平均值，以实现消息图的归一化，得到归一化的消息图。

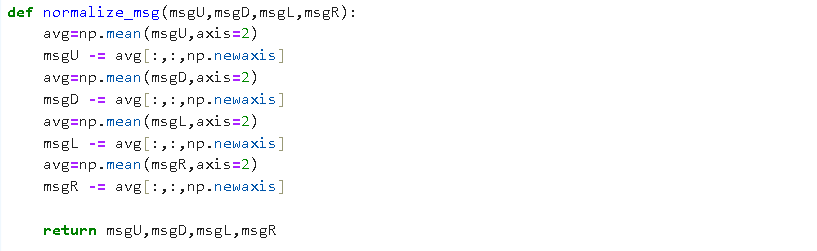


图11

### 3.2.5计算置信度

函数用于根据传入的置信度图，进行最大的后验标签分配。函数接收一个置信度图作为输入，该置信度图包含每个像素点对应不同标签的置信度值，通过找到每个像素点在深度维度上置信度值最小的标签，来确定最终的标签分配结果。最后，返回一个与输入置信度图相同尺寸的标签图，其中每个像素点对应着最可能的标签值。



图12

### 3.2.6立体匹配

函数用于从输入的左右视图中计算出视差图。函数接收左右视图以及一些算法参数如视差值数量、平滑项权重、数据项阈值和迭代次数等。首先，调用计算数据代价函数计算出数据项代价。然后，初始化消息传递过程中用到的消息图，并在每次迭代中更新消息图、归一化消息图、计算置信度图并进行最大后验标签分配。最后，返回计算得到视差图以及每次迭代的能量值。

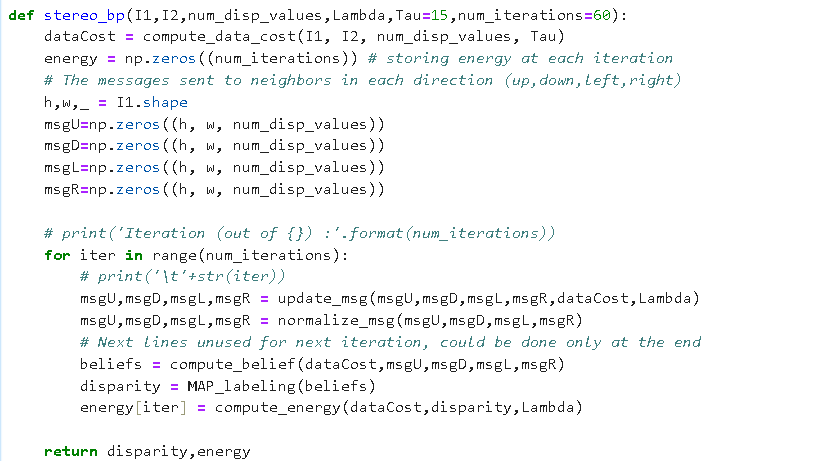


图13

## 3.3实验结果与分析

每次迭代过程中的能量值如图所示，在迭代的过程中，能量逐渐减小。

视差匹配的结果如图15所示，可以观察到距离相近越近的物体亮度越亮，因为前景的位移比背景的位移更多，越靠近摄像机的目标， 它在左右视图位移的距离会更多。

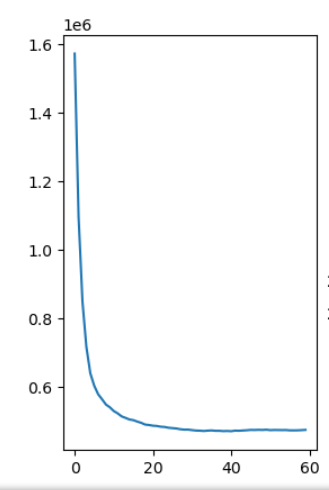


图14

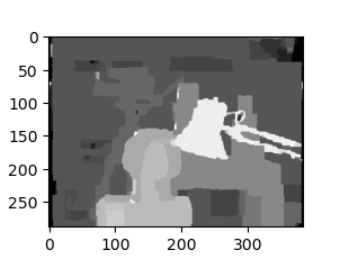


图15