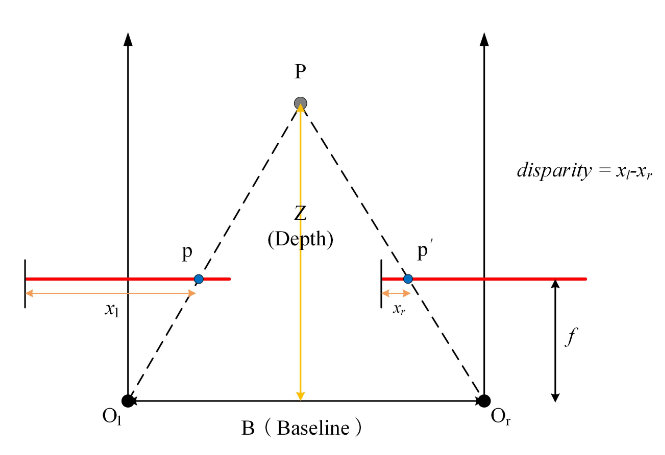
图像视差匹配实验报告

1. **实验目标**

图像视差匹配,通过立体匹配( Stereo Matching）得到两张图像的视差图,需要详细的实验过程和结果分析。

1. **实验原理**
   1. 立体匹配基本概念

立体视觉是指利用物体或场景的平面信息得到三维信息，即实现“立体”的结果。维度信息的变换决定了其复杂性。因此，得到精确的结果需要经过一系列严格的步骤。



**视差（disparity）**

视差等于同名点对在左视图的列坐标减去在右视图上的列坐标，是像素单位，其中



立体视觉里，视差概念在极线校正后的像对里使用。也就是上图中，两张图片只有水平方向的移动时候的情况。

**深度（depth）**

深度D等于像素在该视图相机坐标系下Z坐标，是空间单位。深度并不特在校正后的图像对里使用，而是任意图像都可获取深度图。

##### **[视差图](https://so.csdn.net/so/search?q=%E8%A7%86%E5%B7%AE%E5%9B%BE&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/rs_lys/article/details/_blank)（disparity map）**

视差图指存储立体校正后单视图所有像素视差值的二维图像。

1.视差图是一张二维图像，和原图等大小

2.视差图每个位置保存的以像素为单位的该位置像素的视差值

3.以左视图视差图为例，在像素位置p的视差值等于该像素在右图上的匹配点的列坐标减去其在左图上的列坐标

**深度图（depth map）**

深度图指存储单视图所有像素的深度值的二维图像，是空间单位，比如毫米。

1.深度图是一张二维图像，和原图等大小，也就和视差图等大小

2.深度图每个位置保存的是该位置像素的深度值

3.深度值就是相机坐标系下的Z坐标值

**深度图与视差图的转化**

视差图是立体匹配算法的产出，而深度图则是立体匹配到点云生成的中间桥梁。视差图和深度图中间，有着一对一的转换公式：



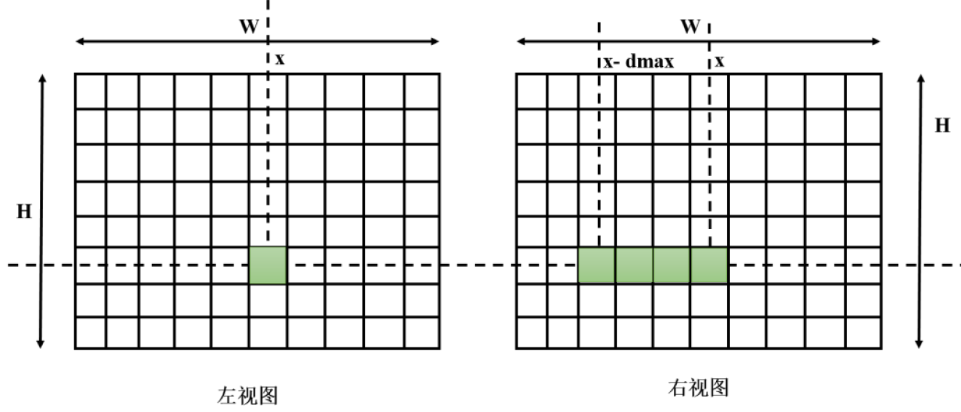
其中，D为深度，d为视差，B为基线长度，f为焦距（像素单位)，x分别为左右视图主点的列坐标。

* 1. 立体匹配基本流程

立体匹配是立体视觉的中间关节，也是核心环节。采集的图像在经过摄像机标定，图像校正等预处理步骤后，立体匹配直接对预处理结果进行处理。

根据立体匹配基础理论可知，立体匹配就是寻找同一点在不同投影面的投影点，根据极线形成的约束可知，某一投影点对应的匹配点必定位于该点对应的极线上。

在同一场景中，左摄像机拍摄得到的视图整体偏右，右摄像机拍摄得到的视图整体偏左，因此根据左视图寻找右视图对应的匹配点的寻找方向应偏左。通常设定视差搜索范围 d m a x，搜索过程在[ 0 , d m a x ]范围内进行。



立体匹配算法通常分为四个基本步骤来研究。

**（1）代价计算：**根据图像属性特征，如像素值，色彩等信息，计算出点和点之间的相似度。两个点越相似，说明是匹配点的可能性越大。

**（2）代价聚合：**代价聚合是立体匹配中十分重要的一个步骤。在初步的代价计算后，得到的结果不够准确，因此需要考虑邻近像素间的关系，涉及的范围不同，累加的方式不同，立体匹配的效率及效果都会不同。

**（3）视差计算：**对于选定的视差范围，会对应着不同的匹配代价值，从中选取最小的值作为最优解，这就是对应的匹配点。

**（4）视差优化：**确定视差后，可以根据邻近视差值通过曲线拟合方式细化视差，剔除匹配错误点，然后根据相邻像素灰度值相似原理进行填充，让视差图更加平滑。

**3.实验过程**

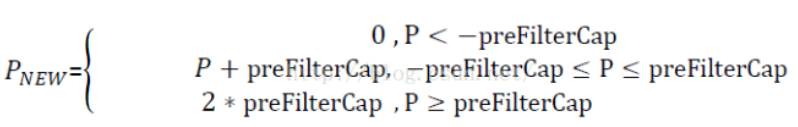
**3.1预处理**

第一步：SGBM采用水平Sobel算子，把图像做处理，公式为:

Sobel(x,y)=2[P(x+1,y)-P(x-1,y)]+ P(x+1,y-1)-P(x-1,y-1)+ P(x+1,y+1)-P(x-1,y+1)

第二步：用一个函数将经过水平Sobel算子处理后的图像上每个像素点(P表示其像素值）映射成一个新的图像:PNEw表示新图像上的像素值。

映射函数为：



预处理实际上是得到图像的梯度信息。经预处理的图像保存起来，将会用于计算代价。

**3.2代价计算**

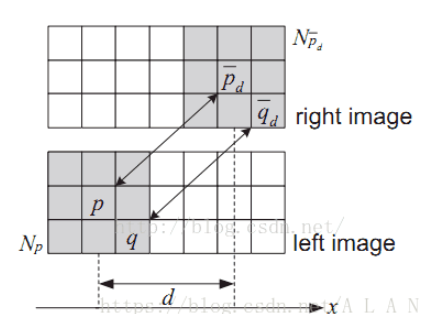
代价有两部分组成:

1、经过预处理得到的图像的梯度信息经过基于采样的方法得到的梯度代价。

2、原图像经过基于采样的方法得到的SAD代价。

上述两个代价都会在SAD窗口内进行计算。SAD公式如下：

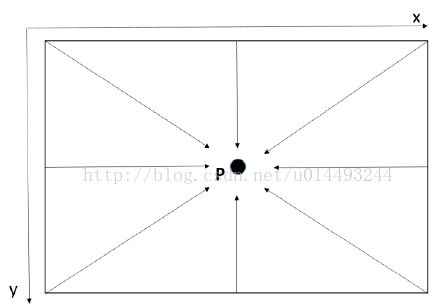




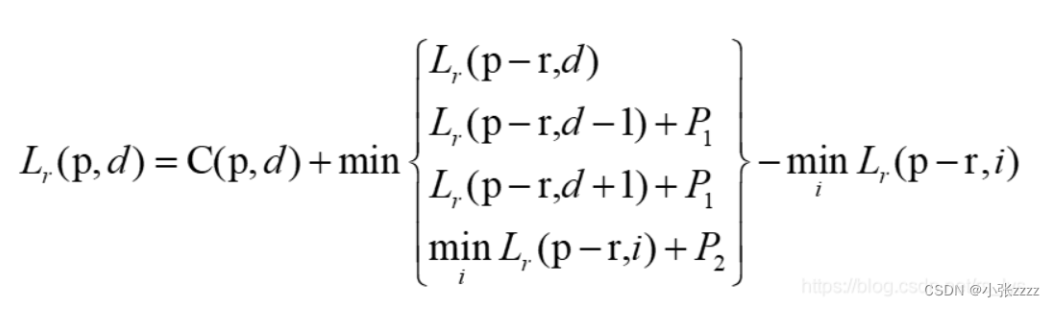
**3.3动态规划**

动态规划算法本身存在拖尾效应，视差突变处易产生错误的匹配，利用态规划进行一维能量累积累，会将错误的视差信息传播给后面的路径上。半全局算法利用多个方向上的信息，试图消除错误信息的干扰，能明显减弱动态规划算法产生的拖尾效应。

半全局算法试图通过影像上多个方向上一维路径的约束，来建立一个全局的马尔科夫能星方程，每个像素最终的匹配代价是所有路径信息的叠加，每个像素的视差选择都只是简单通过WTA (Winner Takes All）决定的。多方向能量聚集如下图所示:



在每个方向上按照动态规划的思想进行能量累积，然后将各个方向上的匹配代价相加得到总的匹配代价，如下式所示：



**3.4后处理**

opencvSGBM的后处理包含以下几个步骤：

Step1：唯一性检测：视差窗口范围内最低代价是次低代价的(1 + uniquenessRatio/100)倍时，最低代价对应的视差值才是该像素点的视差，否则该像素点的视差为0。其中uniquenessRatio是一个常数参数。

Step2：亚像素插值：





Step3：左右一致性检测：误差阈值disp12MaxDiff默认为1，可以自己设置。

1. **结果展示**

****

最终得出得视差图如下所示：

