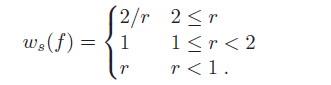
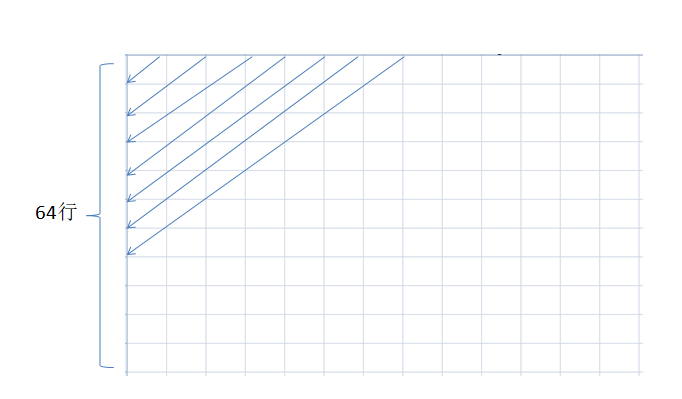
1. 选neighbor
2. 对于每张图像，找到他与参考图像的所有公共稀疏点。对于每个公共稀疏点，计算一个wAngle和一个wScale，把wAngle\*wScale作为这个稀疏点的权重，把所有稀疏点的权重累加求和，得到这幅图像的score。然后分别计算稀疏点覆盖参考图像和该图像的比例area1,和area2，用其中较小者乘以score,得到最终的score。
3. wAngle计算过程：计算它到两个相机中心的向量的夹角fAngle，wAngle=(MINF(POW(fAngle/fOptimAngle, 1.5f), 1.f))。当fAngle<fOptimAngle时，wAngle变小，与参考图像视差很小的图像的score就会变小。
4. wScale计算过程：对于每个公共稀疏点，把这个点在X方向偏移一个位移得到一个新的点，计算这两个点在参考图像上的投影坐标的距离footprint1，这两个点在这幅图像上的投影坐标的距离footprint2，r= footprint1/ footprint2，根据下面分段函数确定wScale。这样做是为了防止参考图像和neighbor图像的分辨率相差太大。



1. 深度图初始化
2. 把这张图像上观察到的每个稀疏点投影在图像上，得到图像坐标（x,y）,然后把图像坐标和这个点的深度放在一起组成一个三维点（x,y,d），然后把所有的这些三维点进行Delaunay三角化。
3. 三角化之后得到若干三角形，对于每个三角形，把三个顶点的(x,y)坐标取出来，这三个（x,y）坐标是图像上三个像素点的坐标，这三个顶点的Z分量是各自的深度值，利用这几个坐标可以计算出这三个点在空间中对应的三个三维点（相机坐标系下），计算出这三个顶点所在的平面P。由图像上的三个点组成一个三角形，这个三角形覆盖若干像素，对每个像素，根据内参可以确定出这个像素在空间中对应的一条射线，这条射线与平面P的交点，认为是这个像素点在空间中对应的三维点，这个三维点的Z坐标就是这个像素点的初始深度，平面P的法向量就是这个点的法向量。
4. 深度图估计
5. 初始化置信度map，也就是每个像素的NCC score。把深度为零或者法向量Z值大于等于零的像素点重新赋一个随机的深度值和法向量。然后计算每个像素的ncc score，计算方法如下：对于这个图像的一个neighbor，用这一点的深度和法向量，计算单应矩阵，用单应矩阵计算NCC窗口内的像素在neighbor图像上的对应像素，计算两组像素的NCC，1-ncc等于这个neighbor的score，然后对深度值与周围像素相似的像素进行奖励，使得它的score变小一点（score越小越好）。取第idxScore个neighbor的score作为这个像素的置信度。
6. 进行三次空间传播和随机优化。
7. 空间传播。遍历每个像素，遍历顺序如下图（每64行分一块，块内从左上到右下）。对于每个像素，用它的4个相邻像素的深度值和法向量计算新的置信度，如果新的置信度小于旧的置信度，就把深度值和法向量替换为这个相邻像素的深度值和法向量。



1. 随机优化。在当前深度和法向量的邻域之内随机选择一个深度和邻域，用它计算新的置信度，如果新的置信度小于当前置信度，就替换掉深度和法向量。
2. 删掉score大于阈值的深度估计。
3. 深度图优化
   1. 删掉小的连通块。把所有相邻（4邻域）且深度值相似的像素放在同一个联通块之内，如果联通块内像素个数小于阈值，就把它们的深度值全部置为零。
   2. 深度图缝隙的插值。对于每一列里面连续的一段深度值为零的像素，如果个数小于阈值，就把这个区间前面一个像素和后面一个像素的深度值均匀分割，赋给这段区间。对每一行也进行相同处理。
4. 深度图过滤
5. 把每幅图像的n个neighbor的深度图恢复成三维点，然后投影到这幅图像（如果多个三维点投影到同一个像素点，只保留深度值最小的一个），得到n个深度图。
6. 方法一：对于这幅图像深度图上的每个像素的深度值d，检查n个深度图上相同像素点的深度值有几个与d非常相似，如果个数小于阈值，则把这个点的深度值置为0。然后用n个深度图上，这个像素点上下左右四个邻域的深度值与d比较，计算有几个与d相似，如果个数小于阈值，则置为零。
7. 方法二：对于这幅图像深度图上的每个像素的深度值d，检查n个深度图上相同像素点的深度值有几个与d非常相似，假设有n1个，如果n1小于阈值，则把这个点的深度值置为0，否则，把n1个图像在该点的深度值按置信度进行加权平均，把平均值作为这个像素点新的深度值，同时修改这个点的置信度。
8. 深度图融合
9. 把所有图像按照neighbor数量从大到小进行排序
10. 对每张图像，遍历每个像素。对每个像素，如果它没有被使用过，则，利用像素坐标和这个点的深度值，计算出它对应的空间三维点。然后把这个三维点转换到每个neighbor 的相机坐标系，得到这个点在这个相机下的深度d，然后投影到这个neighbor图像，得到投影像素P，从这个图像的深度图中得到像素P对应的深度depthB。检查d和depthB是否相近，如果相近，就把这幅图像添加到这个三维点的view，并且标记这个像素P为已经使用过的像素点。如果d和depthB不相近，且d<depthB，那么认为这个像素点P的深度是错误的，并且做好标记。把每个neighbor遍历完之后，如果这个三维点的view数量小于阈值，就把这个三维点删掉，同时修改相应的标记。
11. 注意事项
12. Openmvs选neighbor基本思想是根据两张图像公共稀疏点的数目多少和公共稀疏点覆盖图像的比例来排序，但是对每个稀疏点加了权重，来保证参考图像和neighbor图像视差不能太小，分辨率相差不能太大。
13. 在计算NCC窗口内的像素均值时，用到了积分图，可以很快获取这个窗口内像素值的和。