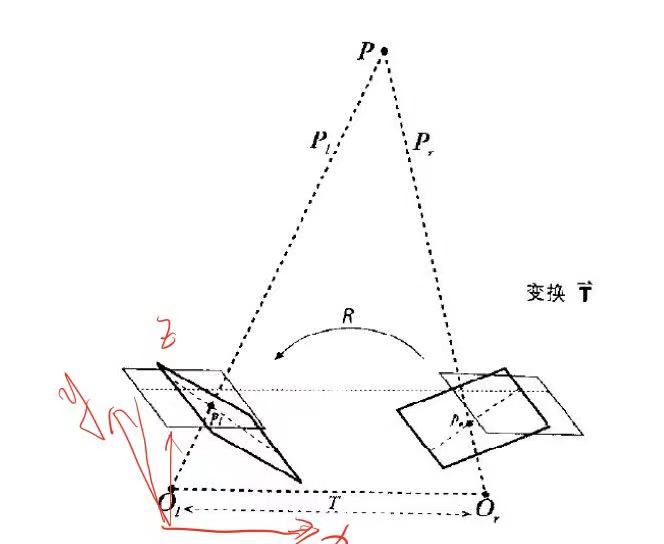
Openmvs



# 极线校正： 将相机在数学上对准到同一观察平面上，使得相机上像素行是严格对齐的， 校正目的：对两幅图像的二维匹配搜索变成一维，节省计算量，排除虚假匹配点

左相机外参Rl， tl， 右相机外参Rr， tr

*openmvs中全部采用的是旋转和光心坐标的方式来表示相机外参p\_c=R\*p\_w+t， p\_c在图像上的坐标，p\_w世界坐标， t平移，相机中心p\_c=[0,0,0]，则p\_w=-R\_invert\*t，p\_w就表示光心坐标C*

## 1.1 首先是X轴的旋转，将X轴旋转到与基线相同的方向，则旋转向量

## 1.2 然后确定新的y轴，只要满足与e1正交即可，选择与旧的主光轴和新的X轴垂直的方向

，这里code中没有tl，平移不会影响旋转之后的方向

## 1.3 新的z轴与新的x轴和y轴互相垂直

坐标转换的旋转矩阵 Rnew=[e1, e2, e3]

左右相机新的旋转矩阵

Rlnew = RnewRlt;

Rrnew = RnewRrt;

校正之后新的基线的计算

1. 先求出未校正之前，右相机光心在左相机坐标系下的坐标
2. 然后在求解左相机光心在右相机校正后的坐标系下的坐标（也等于与右相机校正后的坐标下光心[0, 0, 0]的距离）

（左图到右图的变换）

Tx = t.x，其为负数

# 视差图与深度图转换，矫正后坐标系下的视差图与校正前的坐标系下的深度图的互相转换

## 根据投影方程，先将校正后的视差图转成校正后的深度图

## 2, 写成矩阵形式

## 3, 将校正后的坐标系下的深度图转到校正之前的相机坐标系下

校正之前的相机坐标系下（x, y, z），校正之后的相机坐标系下的坐标（x’,y’,z’）

则有

## 在将校正前相机坐标系下的位置转换成图像坐标可得

## 将上面合并可得校正后坐标系下的视差图（u’， v’， disp）与校正前的坐标系下深度图（u，v, depth）=（X，Y，Z）的转换

# 3 深度估计

1.ComputeDepthMapsMaps

OpenMVS源码中流程

4. EstimatePointNormals

3. EstimatePointColors

2. FuseDepthMaps

DenseReconstruction

输入：图像，位姿， 输出：depth，点云