双目成像及双目相机标定原理整理

一、 双目测距原理

单目相机一般采用针孔成像模型（详见《相机成像模型及相机标定原理整理》），其不足之处在于无法得知像素所对应实际场景中物体的深度。相应地，基于单目相机的 SLAM 方案也无法确定场景尺度。测量像素深度的方式有很多，其中双目相机模仿人眼的视差原理，通过计算同一物体在左右相机中投影的视差来计算深度。其优势在于硬件成本相对较低，但是也有着算法复杂度较高、计算量大的缺点。
理想状态下，如图 1 所示，两摄像机的光轴平行，成像平面共面，焦距相等，所在对应行的像素精确对准。

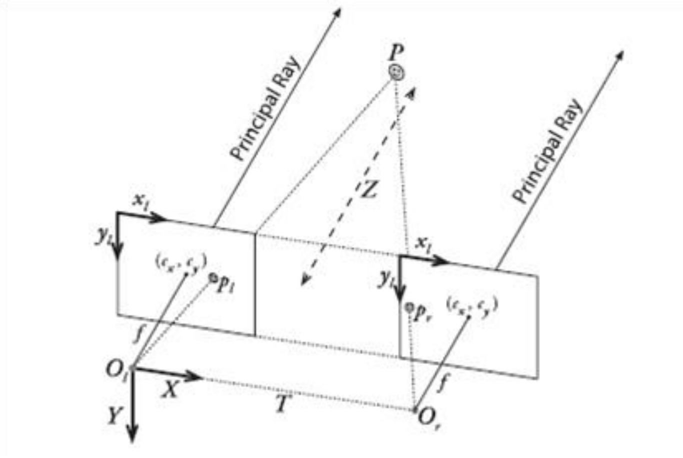

图 1 理想状态下双目测距

两光心的连线为基线B，即为左相机到右相机的位移$\boldsymbol{t}$。以左侧相机光心为原点，确立相机坐标系，有空间中一点$\boldsymbol{P}(X,Y,Z)$，其中Z即为其深度。该点在左右相机的投影及其成像平面坐标分别为$\boldsymbol{p}_l(x_l,y)$、$\boldsymbol{p}_r(x_r,y)$。可通过相似三角形原理求点的深度

$$\frac{Z}{\|\boldsymbol{t}\|}=\frac{f}{x_l-x_r}$$

即

$$Z=\frac{f\|\boldsymbol{t}\|}{x_l-x_r}$$

在实际测量中，得到的是点的像素坐标$\boldsymbol{p}_l(u_l,v)$、$\boldsymbol{p}_r(u_r,v)$；从标定好的相机内参中也无法直接得到焦距f，而是$\alpha=\frac{f}{dx}$。由

$$u=\frac{x}{dx}+u_0$$

可得

$$u_l-u_r=\frac{x_l}{dx}+u_0-\frac{x_r}{dx}-u_0$$
$$=\frac{x_l-x_r}{dx}$$

即

$$x_l-x_r=dx(u_l-u_r)$$

因此

$$
\begin{aligned}
Z &= \frac{f\|\boldsymbol{t}\|}{x_l - x_r} \\
&= \frac{f\|\boldsymbol{t}\|}{dx(u_l - u_r)} \\
&= \frac{\alpha\|\boldsymbol{t}\|}{(u_l - u_r)} \\
&= \frac{\alpha\|\boldsymbol{t}\|}{d}
\end{aligned}
$$

其中$d = u_l - u_r$，为特征在双目相机投影的视差（Disparity）。

从公式推导结果易知，特征深度与视差成反比。因此，只有当被测物体与相机间距离较近时，双目测距才会有较高的分辨率和精度。另一方面，增大基线B也可以提高测距分辨率。然而需要注意的是，受限于镜头视角，增大基线B的同时也会增大视觉盲区，影响最终建图效果。

二、 立体视觉模型

在实际的立体成像过程中，通常难以达到理想的共面、行对准条件。如图 2 所示，通常在确定相机畸变参数及左右相机之间的相对位姿变换之后，通过数学方法将左右图像标定为前向平行排列。

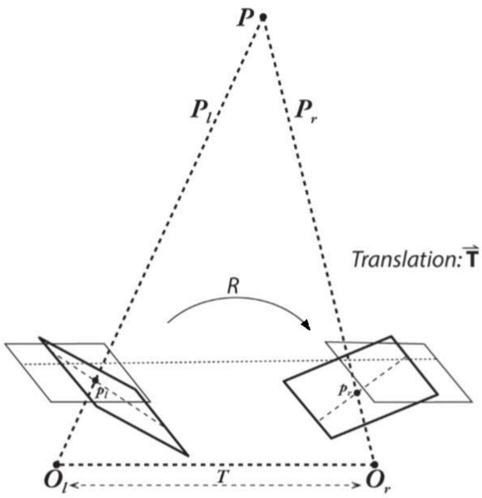

图 2 真实情况下的相机位姿关系及标定过程

右侧相机相对左侧相机的位姿可用旋转矩阵$\boldsymbol{R}$和平移向量$\boldsymbol{t}$表示。其位姿关系也可由基础矩阵（Fundamental Matrix）$\boldsymbol{F}$和本征矩阵（Essencial Matrix）$\boldsymbol{E}$表述，称左右相机光心的连线为基线（Baseline），基线与成像平面的交点$\boldsymbol{e}_l$、$\boldsymbol{e}_r$为极点（Epipoles），连线$\boldsymbol{p}_l\boldsymbol{e}_l$、$\boldsymbol{p}_r\boldsymbol{e}_r$（即极平面与两个成像平面的交线）可记作$\boldsymbol{l}_l$、$\boldsymbol{l}_r$，称为极线（Epipolar line）。一般通过对极几何的方式求出$\boldsymbol{E}$和$\boldsymbol{F}$，进而可确定左右相机的相对位姿。（详见《对极几何与三角测量整理》）

然而在实际中，OpenCV 的双目标定函数并未使用对极几何的方式计算$\boldsymbol{R}$和$\boldsymbol{t}$。在进行双目标定时，会首先对左右相机分别进行一次单目标定，初步得到其内外参数（内参$\boldsymbol{K}_l$、$\boldsymbol{K}_r$，及外参$\boldsymbol{R}_l$、$\boldsymbol{t}_l$、$\boldsymbol{R}_r$、$\boldsymbol{t}_r$）。设空间中一点$\boldsymbol{P}(X,Y,Z)$，在左侧相机坐标系下为$\boldsymbol{P}_l(X_l,Y_l,Z_l)$，右侧相机坐标系下为$\boldsymbol{P}_r(X_r,Y_r,Z_r)$。由相互间的变换关系，易知

$$\boldsymbol{P}_r = \boldsymbol{R}\boldsymbol{P}_l + \boldsymbol{t}$$

移项得

$$\boldsymbol{P}_l = \boldsymbol{R}^T(\boldsymbol{P}_r - \boldsymbol{t})$$

以及由单目标定结果可得

$$\boldsymbol{P}_l = \boldsymbol{R}_l\boldsymbol{P} + \boldsymbol{t}_l$$

$$\boldsymbol{P}_r = \boldsymbol{R}_r\boldsymbol{P} + \boldsymbol{t}_r$$

代入得

$$\boldsymbol{R}_l\boldsymbol{P} + \boldsymbol{t}_l = \boldsymbol{R}^T(\boldsymbol{R}_r\boldsymbol{P} + \boldsymbol{t}_r - \boldsymbol{t})$$

可化为

$$\boldsymbol{R}_l\boldsymbol{P} + \boldsymbol{t}_l = \boldsymbol{R}^T\boldsymbol{R}_r\boldsymbol{P} + \boldsymbol{R}^T\boldsymbol{t}_r - \boldsymbol{R}^T\boldsymbol{t}$$

由于$\boldsymbol{P}$为空间中任一点，因此等式恒成立的条件为

$$\boldsymbol{R}_l = \boldsymbol{R}^T\boldsymbol{R}_r$$

$$\boldsymbol{t}_l = \boldsymbol{R}^T\boldsymbol{t}_r - \boldsymbol{R}^T\boldsymbol{t}$$

移项得

$$\boldsymbol{R} = \boldsymbol{R}_r\boldsymbol{R}_l^T$$

$$\boldsymbol{t} = \boldsymbol{t}_r - \boldsymbol{R}\boldsymbol{t}_l$$

这样就可以由单目标定所得的外参$\boldsymbol{R}_l$、$\boldsymbol{t}_l$、$\boldsymbol{R}_r$、$\boldsymbol{t}_r$计算得出左右相机的相对位姿关系$\boldsymbol{R}$和$\boldsymbol{t}$。一般标定时都会拍摄几十组图片，而由于误差的存在，每组图片得出的$\boldsymbol{R}$、$\boldsymbol{t}$的值会有微小的差别。因此标定时会以各组$\boldsymbol{R}$、$\boldsymbol{t}$的中位数作为初值，构造以$\boldsymbol{R}$、$\boldsymbol{t}$为参数的重投影误差函数，并使用 L-M 法进行非线性优化，以使重投影误差最小。

三、 立体校正

为便于测量特征深度，在得到$\boldsymbol{R}$、$\boldsymbol{t}$后，可使用数学方法，使左右相机的成像平面达到理想状态，即立体校正。立体校正算法有很多，按照是否得到左右相机的内参数，可分为非标定算法和已标定算法两大类。在 OpenCV 中，实现了两种校正算法，分别是非标定的 Hartley 校正算法和已标定的 Bouguet 校正算法。

3.1.Hartley 算法

Hartley 校正算法作为一种非标定算法，无需计算相机内参，主要思路为在确定了左右图像的**基础矩阵**和**极点坐标**后，寻求射影变换$\boldsymbol{H}_l$和$\boldsymbol{H}_r$，将极点映射到无穷远处，使极线与图像水平轴平行，进而实现重采样，使图像行对准。值得注意的是，Hartley 算法虽然不需要摄像机内参，但是**默认图像没有径向畸变**，即该算法的输入图像应首先进行单目的去畸变处理。

相对于关于点的映射$\boldsymbol{H}_l$和$\boldsymbol{H}_r$，可以有对应的关于极线$\boldsymbol{l}_l$、$\boldsymbol{l}_r$的线映射

$$\boldsymbol{H}_l^*\boldsymbol{l}_l = \boldsymbol{H}_r^*\boldsymbol{l}_r$$

即，$\boldsymbol{H}_l^*$和$\boldsymbol{H}_r^*$将对应极线映射到同一水平线上。

在得到左右图像的基础矩阵$\boldsymbol{F}$、以及极点坐标$\boldsymbol{e}_l$、$\boldsymbol{e}_r$后，首先需要确定右侧图像$\boldsymbol{J}_r$的射影变换$\boldsymbol{H}_r$。寻求矩阵$\boldsymbol{T}$，将图像中心$(u_0, v_0, 1)^T$变换到像素坐标系原点

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

旋转矩阵$\boldsymbol{R}$将$\boldsymbol{J}_r$的极点$\boldsymbol{e}_r = (u_r, v_r, 1)^T$映射到$(f, 0,1)^T$，即图像水平轴上

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

其中$\theta = \tan^{-1}(v_r/u_r)$为旋转角度。

矩阵$\boldsymbol{G}$将$(f, 0,1)^T$映射到水平方向的无穷远点$(f, 0,0)^T$

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

整体的作用过程如图 3 所示。

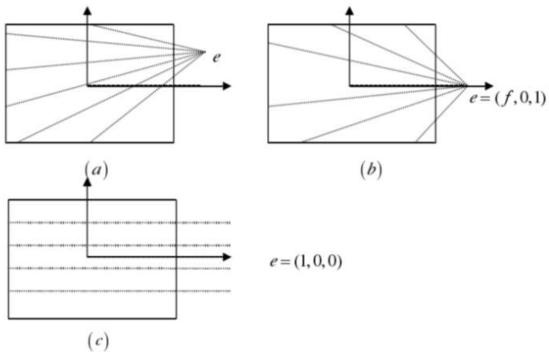

图 3 变换的作用过程

由此可以得出射影变换矩阵

$$\boldsymbol{H}_r = \boldsymbol{GRT}$$

在得到$\boldsymbol{H}_r$后，可以得出$\boldsymbol{H}_l$的一组候选值

$$\boldsymbol{H}_l = (\boldsymbol{I} + \boldsymbol{H}_r\boldsymbol{e}_r\boldsymbol{a}^T)\boldsymbol{H}_r\boldsymbol{M}$$

其中$\boldsymbol{a}$为任意向量，$\boldsymbol{M}$为由$\boldsymbol{F}$得到的非奇异矩阵

$$\boldsymbol{F} = [\boldsymbol{e}_r]_\times \boldsymbol{M}$$

由于$\boldsymbol{a}$为任意向量，因此$\boldsymbol{H}_l$的候选值不唯一，可通过最小二乘的方式得出$\boldsymbol{H}_l$的最优解（$\boldsymbol{u}_i$、$\boldsymbol{u}_i'$为对应点）

$$\min \sum_i d(\boldsymbol{H}_l\boldsymbol{u}_i, \boldsymbol{H}_r\boldsymbol{u}_i')^2$$

其中$\boldsymbol{H}_r$将$\boldsymbol{e}_r$映射到无穷远点$(1,0,0)^T$，因此

$$\boldsymbol{I} + \boldsymbol{H}_r\boldsymbol{e}_r\boldsymbol{a}^T = \boldsymbol{I} + (1,0,0)^T\boldsymbol{a}^T$$

$$= \boldsymbol{A} = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

令$\boldsymbol{H}_0 = \boldsymbol{H}_r\boldsymbol{M}$，则有$\boldsymbol{H}_l = \boldsymbol{A}\boldsymbol{H}_0$。令$\widehat{\boldsymbol{u}}_i = \boldsymbol{H}_0\boldsymbol{u}_i$，$\widehat{\boldsymbol{u}}_i' = \boldsymbol{H}_r\boldsymbol{u}_i'$，所求最小二乘可化为

$$\min \sum_i d(\boldsymbol{A}\widehat{\boldsymbol{u}}_i, \widehat{\boldsymbol{u}}_i')^2$$

令$\widehat{\boldsymbol{u}}_i = (\hat{u}_i, \hat{v}_i, 1)^T$，$\widehat{\boldsymbol{u}}_i' = (\hat{u}_i', \hat{v}_i', 1)^T$，$\boldsymbol{H}_r$和$\boldsymbol{M}$已知，那么最小二乘可化为

$$\min \sum_i (a\hat{u}_i + b\hat{v}_i + c - \hat{u}_i')^2 + (\hat{v}_i - \hat{v}_i')^2$$

其中第二项由对应点的坐标决定，与所求矩阵无关，因此可通过求解下式最小值来得出$\boldsymbol{H}_l$的最优解

$$\min \sum_i (a\hat{u}_i + b\hat{v}_i + c - \hat{u}_i')^2$$

之后由$\widehat{\boldsymbol{u}} = \boldsymbol{H}\boldsymbol{u}$进行对应坐标的重采样，并插值即可。

3.2. Bouguet 算法

对于已通过标定得出相机内外参数的情况而言，Bouguet 校正更为适用。Bouguet 算法通过

标定得出的相对位姿$\boldsymbol{R}$、$\boldsymbol{t}$，在减小重投影误差的同时尽量最大化共同投影面积，来实现图像的行对准。
将$\boldsymbol{R}$拆分为$\boldsymbol{r}_l$和$\boldsymbol{r}_r$，使得

$$\boldsymbol{R} = \boldsymbol{r}_r^{-1}\boldsymbol{r}_l$$
$$\boldsymbol{r}_r \cdot \boldsymbol{r}_l = 1$$

这样可以视为将左右相机各自旋转一半的角度，方向相反，使两个成像平面共面，此时的图像共面而并未对准。之后再通过矩阵$\boldsymbol{R}_{rect}$，将图像修正为行对准，即极点映射到无穷，极线映射为水平。另一种思路则如图 4 所示，视为通过矩阵$\boldsymbol{R}_{rect}$，将并非平行的图像旋转到平行。

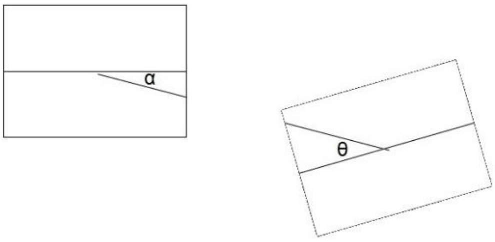

图 4 旋转到共面之后的图像，之后将其映射到行对准

令

$$\boldsymbol{R}_{rect} = \begin{bmatrix} \boldsymbol{e}_1^T \\ \boldsymbol{e}_2^T \\ \boldsymbol{e}_3^T \end{bmatrix}$$

那么有

$$\boldsymbol{e}_1 = \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|}$$

其中$\boldsymbol{t}$为平移向量，$\boldsymbol{t} = [t_x \quad t_y \quad t_z]^T$。要求$\boldsymbol{e}_2$与$\boldsymbol{e}_1$正交，则取$\boldsymbol{e}_2$为主光轴$[0 \quad 0 \quad 1]^T$与$\boldsymbol{e}_1$的叉积

$$\boldsymbol{e}_2 = [-t_y \quad t_x \quad 0]^T \cdot \frac{1}{\sqrt{t_x^2 + t_y^2}}$$

要求$\boldsymbol{e}_3$与$\boldsymbol{e}_1$、$\boldsymbol{e}_2$都正交，则取$\boldsymbol{e}_3$为二者叉积

$$\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2$$

那么总体的摄像机旋转为

$$\boldsymbol{R}_l = \boldsymbol{R}_{rect}\boldsymbol{r}_l$$
$$\boldsymbol{R}_r = \boldsymbol{R}_{rect}\boldsymbol{r}_r$$

之后为便于进行图像坐标转换，有左右相机的校正后投影矩阵$\boldsymbol{P}_l$、$\boldsymbol{P}_r$及重投影矩阵$\boldsymbol{Q}$。其中$\boldsymbol{P}$为校正后相机内参矩阵$\boldsymbol{K}_{rect}$结合投影矩阵$\boldsymbol{P}'$而得

$$\boldsymbol{P}_l = \boldsymbol{K}_{rect\_l} \cdot \boldsymbol{P}_l' = \begin{bmatrix} f_{rect\_l} & 0 & u_{0l} \\ 0 & f_{y\_l} & v_{0l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{P}_r = \boldsymbol{K}_{rect\_r} \cdot \boldsymbol{P}_r' = \begin{bmatrix} f_{rect\_r} & 0 & u_{0r} \\ 0 & f_{y\_r} & v_{0r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

校正后投影矩阵将相机系下的 3D 齐次坐标映射到校正后的左右图像中，即

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

投影后的像素坐标为

$$u = x/w$$
$$v = y/w$$

重投影矩阵$\boldsymbol{Q}$为

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & -u_{0l} \\ 0 & 1 & 0 & -v_{0l} \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/t_x & (u_{0l} - u_{0r})/t_x \end{bmatrix}$$

重投影矩阵将得到视差后的像素坐标映射到相机坐标系中，即

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ W_c \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix}$$

参考

[1] 高翔, 张涛, 刘毅, 等. 视觉SLAM十四讲[M]. 2017年3月第1版. 电子工业出版社, 2017.
[2] Kaehler A, Bradski G. 学习 OpenCV3[M]. 阿丘科技团队, 译. 北京:清华大学出版社, 2018.
[3] Hartley R I. Theory and Practice of Projective Rectification[J]. International Journal of Computer Vision, 1999, 35(2): 1-16.
[4] 陈宏洋. 双目视觉深度信息提取及其关键算法研究[D].电子科技大学,2016.
[5] 李杰. 基于双目视觉的深度重建技术的研究和应用[D].吉林大学,2016.