



仪器科学与光电工程学院
SCHOOL OF INSTRUMENT SCIENCE & OPTO-ELECTRONICS ENGINEERING

图像传感技术综合实验
研究生专业实验
设计报告

2019 年 5 月



摘要

本文分为三个部分，第一部分为相机标定及单目视觉测量，第二部分为双目相机标定及测量，第三部分为基于 OpenCV 的增强现实实现。

第一部分中，采用张正友标定法对机械革命 X6Ti-S 笔记本电脑前置摄像头进行标定，并对该摄像机镜头进行畸变校正，获得摄像机内外参数。随后，在上述标定的基础上对 iPad 屏幕上绘制的半径 $R=20\text{mm}$ 的圆进行了测量，获得其半径大小。通过对图像进行边缘保留滤波，灰度化，采用 Hough 变换确定圆的大概区域，掩膜，二值化，之后采用 canny 算子在该区域内提取边缘，将边缘上所有的像素点投影到世界坐标系下，利用最小二乘法拟合圆。得到的测量结果为半径 $R=19.91\text{mm}$ ，误差约为 -0.09mm 。

第二部分为双目相机标定及测量。首先通过相机标定得到左右两个相机的内外参数和双目视觉系统的结构参数，使用该参数对图像进行立体校正，对左右视图消除畸变及共面行对准。随后，在图像坐标系下提取左右视图圆盘边缘。将边缘的顶端、底端、最左端、最右端作为特征点，将其投影到世界坐标系下，使用特征点计算拟合球和拟合平面，进而得到拟合圆的直径。实验测量的结果为 26.10mm ，误差为 1.04mm 。

第三部分分为有标记 AR 和无标记 AR。有标记的 AR 采用二维海明码作为标记，首先对图像中的标记进行解码，之后通过提取的角点计算外参矩阵，最后计算出要绘制的图像的图像坐标，将一棵树绘制在标记处。无标记的 AR 通过 ORB 算法提取目标的特征点，BF 进行特征点匹配，并用 RANSAN 算法对特征点进行随机采样，剔除冗余点，降低随机误差，实现目标检测。



目录

1	相机标定及单目视觉测量	1
1.1	相机标定原理	1
1.1.1	世界坐标系转化为相机坐标系	3
1.1.2	相机坐标系转化为理想图像坐标系（不考虑畸变）	3
1.1.3	理想图像坐标系转化为实际图像坐标系（考虑畸变）	4
1.1.4	实际图像坐标系转化为像素坐标系	4
1.1.5	张正友标定法求解内参和外参	5
1.1.6	求解畸变系数	7
1.2	相机标定实验	7
1.2.1	拍摄靶标及标定	8
1.2.2	精度评价	9
1.3	基于单目视觉的平面圆测量	10
1.3.1	原理	10
1.3.2	实验结果及分析	11
1.4	结论	12
2	双目相机标定及测量实验	14
2.1	双目测量原理	14
2.1.1	数学模型	14
2.1.2	极线几何	15
2.1.3	双目标定	16
2.2	双目相机测量实验结果及分析	16
2.2.1	实验流程	16
2.2.2	标定相机	16
2.2.3	立体矫正	17
2.2.4	图像预处理	18
2.2.5	特征点匹配与视差计算	19



2.2.6	计算圆盘的直径	19
2.3	结论	20
3	增强现实（AR）的 OpenCV 实现	21
3.1	算法原理	21
3.1.1	有标记 AR 原理	21
3.1.2	无标记 AR 原理	23
3.2	有标记 AR 的实现.....	23
3.2.1	标记检测.....	24
3.2.2	姿态估计.....	24
3.2.3	渲染绘图.....	25
3.3	无标记 AR 的实现.....	26
3.3.1	特征提取.....	26
3.3.2	特征匹配.....	27
3.3.3	目标检测.....	28



1 相机标定及单目视觉测量

在图像测量过程以及机器视觉应用中，首要任务就是要通过拍摄到的图像信息获取到物体在真实三维世界里相对应的信息，于是，建立物体从三维世界映射到相机成像平面这一过程中的几何模型就显得尤为重要，而这一过程最关键的部分就是要得到相机的内参和外参。在大多数条件下这些参数必须通过实验与计算才能得到，这个求解参数的过程就称之为相机标定（或摄像机标定）。无论是在图像测量或者机器视觉应用中，相机参数的标定都是非常关键的环节，其标定结果的精度及算法的稳定性直接影响相机工作产生结果的准确性。因此，做好相机标定是做好后续工作的前提，提高标定精度是科研工作的重点所在。

本文以 Matlab 为工具，使用张正友标定法对单目相机进行了标定，对标定结果进行畸变校正，并设计了单目视觉的平面圆尺寸测量实验，利用标定得到的相机内外参数基于 OpenCV 对平面圆进行尺寸测量。

1.1 相机标定原理

相机标定的目的之一是为了建立物体从三维世界到成像平面上各坐标点的对应关系，所以首先需要定义几个坐标系来为整个过程做好铺垫：

世界坐标系(world coordinate system): 用户定义的三维世界的坐标系，为了描述目标物在真实世界里的位置而被引入。单位为 m。

相机坐标系(camera coordinate system): 在相机上建立的坐标系，为了从相机的角度描述物体位置而定义，作为沟通世界坐标系和图像/像素坐标系的中间一环。单位为 m。

图像坐标系(image coordinate system): 为了描述成像过程中物体从相机坐标系到图像坐标系的投影透射关系而引入，方便进一步得到像素坐标系下的坐标。单位为 m。

像素坐标系(pixel coordinate system): 为了描述物体成像后的像点在数字图像上（相片）的坐标而引入，是真正从相机内读取到的信息所在的坐标系。单位为个（像素数目）。

如图 1 所示，世界坐标系： x_w, y_w, z_w ；相机坐标系： x_c, y_c, z_c ；图像坐标系： x_n, y_n ；像素坐标系： x_n, y_n 。其中，相机坐标系的轴与光轴重合，且垂直于图像坐标系平面并通过图像坐标系的原点，相机坐标系与图像坐标系之间的距离为焦距 f （也即图像坐标系原点与焦点重合）。像素坐标系平面 $u-v$ 和图像坐标系平面 $x-y$ 重合，像素坐标系原点

位于图中左上角（之所以这么定义，目的是从存储信息的首地址开始读写）。

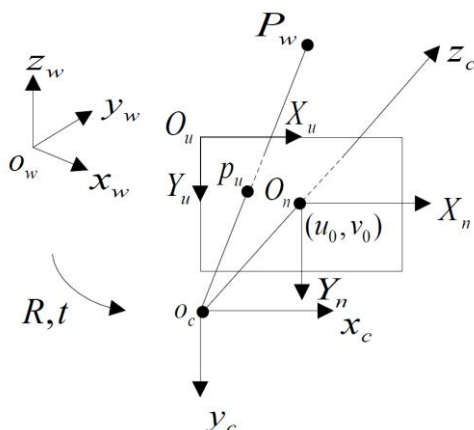


图 1. 摄像机成像针孔模型

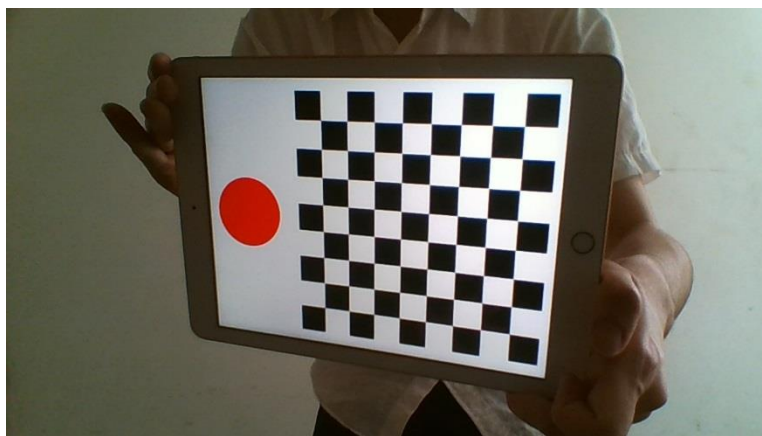


图 2. 棋盘格标定板

棋盘是一块由黑白方块间隔组成的标定板，作为相机标定的标定物（从真实世界映射到数字图像内的对象）。之所以用棋盘作为标定物是因为平面棋盘模式更容易处理（相对于复杂的三维物体），但与此同时，二维物体相对于三维物体会缺少一部分信息，于是需多次改变棋盘的方位来捕捉图像，以求获得更丰富的坐标信息。

下面将依次对刚体进行一系列变换，使之从世界坐标系进行仿射变换、投影透射，最终得到像素坐标系下的离散图像点，过程中会逐步引入各参数矩阵。

总的来说，标定的过程分为两个部分：第一步是从世界坐标系转为相机坐标系，这一步是三维点到三维点的转换，包括 R, t （相机外参，确定了相机在某个三维空间中的位置和朝向）等参数；第二部是从相机坐标系转为成像平面坐标系（像素坐标系），这一步是三维点到二维点的转换，包括 f_x, f_y, u_0, v_0 （相机内参,是对相机物理特性的近似）等参数。

1.1.1 世界坐标系转化为相机坐标系

刚体从世界坐标系转换到相机坐标系的过程，可以通过旋转和平移来得到，将其变换矩阵由一个旋转矩阵和平移向量组合成的齐次坐标矩阵来表示：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = [r_1 \ r_2 \ r_3 \ t] \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} = [r_1 \ r_2 \ 1] \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (1)$$

其中， R 为旋转矩阵， t 为平移向量，因为假定在世界坐标系中物点所在平面过世界坐标系原点且与 z_w 轴垂直（也即棋盘平面与 $x_w - y_w$ 平面重合，目的在于方便后续计算，也是进行单目测量平面圆的基本假设），所以 $z_w = 0$ ，可直接转换成（1）的形式。其中变换矩阵

$$\begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix}$$

即为外参矩阵，之所称之为外参矩阵可以理解为只与相机外部参数有关，且外参矩阵随刚体位置的变化而变化。

1.1.2 相机坐标系转化为理想图像坐标系（不考虑畸变）

这一过程进行了从三维坐标到二维坐标的转换，也即投影透视过程（用中心投影法将形体投射到投影面上，从而获得的一种较为接近视觉效果的面投影图，也就是使人眼看到景物近大远小的一种成像方式）。

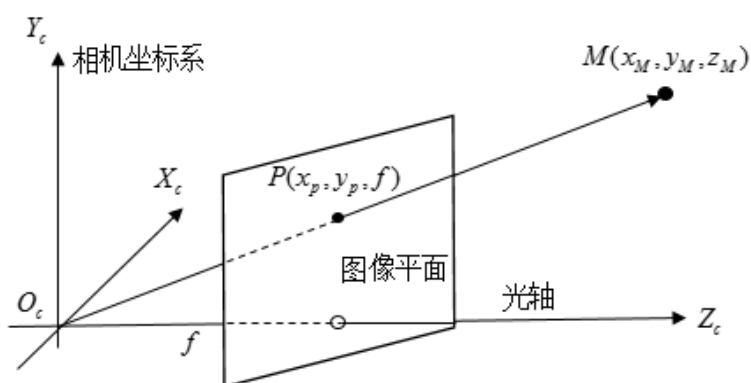


图 3

如图 3 所示，假设相机坐标系中有一点 M ，则在理想图像坐标系下（无畸变）的成像点 P 的坐标为（可由相似三角形原则得出）：

$$x_p = f \frac{x_M}{z_M}, y_p = f \frac{y_M}{z_M}$$

将上式化为齐次坐标表示形式为：

$$z_M \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ y_{11} \\ z_{11} \\ 1 \end{bmatrix} \quad (2)$$

1.1.3 理想图像坐标系转化为实际图像坐标系（考虑畸变）

透镜的畸变主要分为径向畸变和切向畸变。径向畸变是由于透镜形状的制造工艺导致。且越向透镜边缘移动径向畸变越严重。实际情况中常用 $r = 0$ 处的泰勒级数展开的前几项来近似描述径向畸变。矫正径向畸变前后的坐标关系为：

$$\begin{cases} x_{rcorr} = x_p(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{rcorr} = y_p(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases} \quad (3)$$

由此可知对于径向畸变，有 3 个畸变参数需要求解。

切向畸变是由于透镜和 CMOS 或者 CCD 的安装位置误差导致。切向畸变需要两个额外的畸变参数来描述，矫正前后的坐标关系为：

$$\begin{cases} x_{rcorr} = x_p + [2p_1x_py_p + p_2(r^2 + 2x_p^2)] \\ y_{rcorr} = y_p + [p_1(r^2 + 2y_p^2) + 2p_2x_py_p] \end{cases} \quad (4)$$

由此可知对于切向畸变，有 2 个畸变参数需要求解。

综上，一共需要 5 个畸变参数 $(k_1, k_2, k_3, p_1, p_2)$ 来描述透镜畸变

1.1.4 实际图像坐标系转化为像素坐标系

由于定义的像素坐标系原点与图像坐标系原点不重合，假设像素坐标系原点在图像坐标系下的坐标为 (u_0, v_0) ，每个像素点在图像坐标系 x 轴、y 轴方向的尺寸为： d_x, d_y ，且像点在实际图像坐标系下的坐标为 (x_c, y_c) ，于是可得到像点在像素坐标系下的坐标为：

$$u = \frac{x_c}{d_x} + u_0, v = \frac{y_c}{d_y} + v_0$$

化为齐次坐标表示形式可得：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_0 \\ 0 & \frac{1}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \quad (5)$$

若暂不考虑透镜畸变，则将 (2) 与 (5) 的转换矩阵相乘即为内参矩阵 M ：

$$M = \begin{bmatrix} \frac{f}{d_x} & 0 & u_0 \\ 0 & \frac{f}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

之所以称之为内参矩阵可以理解为矩阵内各值只与相机内部参数有关，且不随物体位置变化而变化。

1.1.5 张正友标定法求解内参和外参

引入单应性的概念。在机器视觉中的单应性被定义为从一个平面到另一个平面的投影映射关系，前文已经得到了像素坐标系和世界坐标系下的坐标映射关系：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad (7)$$

因此，定义相机标定的单应性矩阵（从物体平面到成像平面）为：

$$H = s \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad (8)$$

这里引入的 s 为任意尺度的比例系数。单应性矩阵描述了物体在世界坐标系和像素坐标系之间的相对位置关系（包含了内参和外参）。

先将 H 化为 $H = [h_1 \ h_2 \ h_3]$ ，再分解方程可得：

$$\begin{cases} h_1 = sMr_1 \\ h_2 = sMr_2 \\ h_3 = sMr_3 \end{cases} \text{ or } \begin{cases} r_1 = \lambda M^{-1}h_1 \\ r_2 = \lambda M^{-1}h_2 \\ t = \lambda M^{-1}h_3 \end{cases} \quad (9)$$

其中 $\lambda = s^{-1}$ 。

因为旋转向量在构造中是相互正交的，即 r_1 和 r_2 相互正交，由此我们就可以利用正交的两个含义，得出每个单应性矩阵（也即每个棋盘方位图像）提供的两个约束条件：

1. 旋转向量点积为 0（两垂直平面上的旋转向量互相垂直）：

$$r_1^T r_2 = 0 \quad (10)$$

替换和并化简可得：

$$h_1^T (M^{-1})^T M^{-1} h_2 = 0 \quad (11)$$

2. 旋转向量长度相等（旋转不改变尺度）：

$$r_1^T r_1 = r_2^T r_2 \text{ or } \|r_1\| = \|r_2\| = 1 \quad (12)$$

替换掉 r_1 和 r_2 可得：

$$h_1^T (M^{-1})^T M^{-1} h_1 = h_2^T (M^{-1})^T M^{-1} h_2 \quad (13)$$

不妨设

$$B = (M^{-1})^T M^{-1} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & -\frac{c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & -\frac{c_y}{f_y^2} \\ -\frac{c_x}{f_x^2} & -\frac{c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{31} & B_{33} \end{bmatrix} \quad (14)$$

则可将两个约束条件转化为:

$$\begin{cases} h_1^T B h_2 = 0 \\ h_1^T B h_1 = h_2^T B h_2 \end{cases} \quad (15)$$

由上式可知, 两约束中的单项式均可写为

$$h_i^T B h_j \quad (16)$$

的形式, 同时易知 B 为对称矩阵, 真正有用的元素只有 6 个 (主对角线任意一侧的 6 个元素)。于是可展开为如下形式:

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \quad (17)$$

由此, 两约束条件可等价于:

$$\begin{bmatrix} v_{12}^T \\ v_{11}^T - v_{22}^T \end{bmatrix} b = 0 \quad (18)$$

假设提供 K 个棋盘图像, 每个棋盘有 N 个角点, 于是有 $2KN$ 个约束方程。与此同时, 忽略畸变的情况下, 就要求解 4 个内参和 $6K$ 个外参, 也就是说, 只有当 $2KN \geq 4 + 6K$ 的时候, 也即 $K(N - 3) \geq 2$ 时, 才能求出内外参矩阵。同时, 无论在一张棋盘上检测到多少角点, 由于棋盘上角点的规则布置使得真正能利用上的角点只有 4 个 (在四个方向上可延展成不同的矩形), 于是有当 $N = 4$ 时, $K(4 - 3) \geq 2$, 即 $K \geq 2$, 也就是说, 至少需要两张棋盘在不同方位的图像才能求解出无畸变条件下的内参和外参。

实际上往往会在一张棋盘上布置更多的角点, 因为这样就可以通过最小二乘法求得最优解了, 同样地, 会需要至少 10 张以上的棋盘图像, 目的是考虑数值稳定性和提高信噪比, 得到更高质量的结果。

已经知道，棋盘图像数目满足就可求出内外参数，此时 \mathbf{b} 有解，于是由内参数 \mathbf{B} 的封闭解和 \mathbf{b} 的对应关系即可求解出内参数矩阵中的各个元素。得到内参数后，可继续求得外参数：

$$\begin{cases} r_1 = \lambda M^{-1} h_1 \\ r_2 = \lambda M^{-1} h_2 \\ r_3 = r_1 \times r_2 \\ t = \lambda M^{-1} h_3 \end{cases} \quad (19)$$

其中又由旋转矩阵性质有

$$\|r_1\| = \|\lambda M^{-1} h_1\| = 1 \quad (20)$$

则可得

$$\lambda = 1 / \|M^{-1} h\| \quad (21)$$

1.1.6 求解畸变系数

根据（3）和（4）可以得到矫正畸变后的坐标和矫正前的坐标关系为：

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} 2p_1 x_p y_p + p_2 (r^2 + 2x_p^2) \\ p_1 (r^2 + 2y_p^2) + 2p_2 x_p y_p \end{bmatrix} \quad (22)$$

有了上式的对应关系，又已知相应的内外参数，再利用大量给定的坐标数据即可进一步求得畸变系数

1.2 相机标定实验

常见靶标为印刷靶标，其印刷密度多为 300DPI。随着高像素密度电子显示技术的发展，电子屏幕可能代替传统印刷靶标以实现更加灵活和柔性的标定，因此用电子显示屏进行了如下标定对比实验。

为了保证靶标之于相机为连续图像，需要保证靶标显示器分辨率远高于待标定相机。具体实验配置如下：待标定摄像头选用机械革命 X6Ti-s 笔记本前置高清摄像头，分辨率为 1280×720 像素。靶标采用电子屏显示，显示设备为 iPad2018，9.7 英寸显示屏，分辨率为 2048×1536 像素，像素密度 264PPI。靶标图像由 Photoshop 绘制的 9×9 的黑白棋盘格靶标，每格边长为 15 mm。采用张正友标定法，由 Matlab R2019a 自带的计算机视觉工具箱 Camera Calibrator 进行标定，标定结果包含切向畸变，径向畸变包含 2 个待标定系数（一般只需要计算出 k_1, k_2, p_1, p_2 ，对于鱼镜头等径向畸变特别大的才需要计算 k_3 ）。

1.2.1 拍摄靶标及标定

将 iPad 靶标置于前置摄像头视场，进行标定实验。拍摄到的靶标图像如下示意：

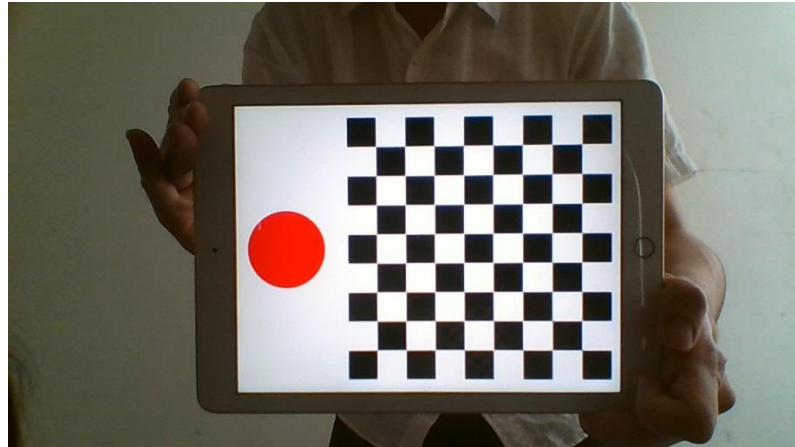


图 3 拍摄 surface 上的靶标图像

内参数标定结果如下所示：

Camera Intrinsics

```
IntrinsicMatrix: [3×3 double]
FocalLength: [923.9681 925.4082]
PrincipalPoint: [654.6625 404.1002]
Skew: -0.0154
RadialDistortion: [0.0633 -0.2022]
TangentialDistortion: [0.0012 -0.0011]
ImageSize: [720 1280]
```

重投影误差为 0.10 pixels。精度评价结果如下图所示：

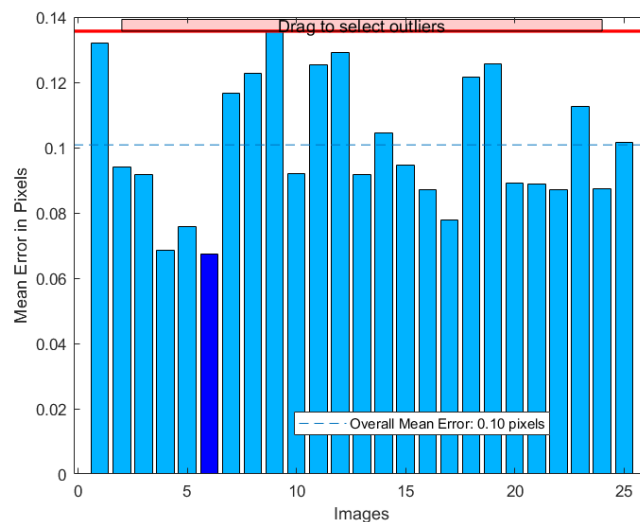


图 4 重投影误差

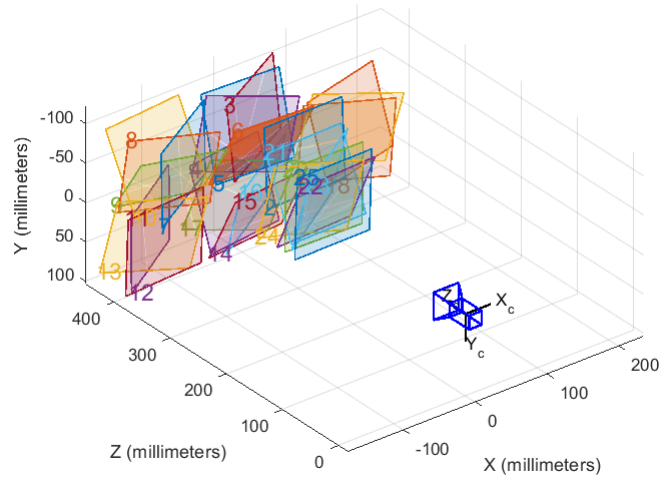


图 5 外参

1.2.2 精度评价

标定的精度有多种评价方法，其中以角点重投影（RMS）误差及归一化标定误差（NCE）为主，好的相机标定结果需要满足两者均尽可能低。本文采用角点重投影（RMS）误差进行精度评价，重投影误差为 0.10 pixels。

角点重投影（RMS）误差：选取一幅图的一系列三维角点 (x_{wi}, y_{wi}, z_{wi}) 在图像平面上的二维投影 $(\hat{x}_{di}, \hat{y}_{di})$ 。通过计算真实图像角点二维坐标 (x_{di}, y_{di}) （从图像中获得）与投影得到的角点二维坐标 $(\hat{x}_{di}, \hat{y}_{di})$ （通过摄像机模型获得）的差异评价标定精度。

$$\text{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_{di} - \hat{x}_{di})^2 + (y_{di} - \hat{y}_{di})^2]} \quad (23)$$

对于相机标定重投影误差来说，噪声及靶标自身精度都是潜在的影响因素。对于参数的估计误差（或者称作不确定度），靶标自身尺寸及光束平差过程也会影响，好的标定结果将保证解算参数具有较小的不确定性。其他标准包括相机 x 方向与 y 方向焦距的对称性及主点之于图像中心的偏移程度等，这些评价大多需要先验知识。由于相机光学系统具有旋转对称性，相机 x 方向与 y 方向的焦距应该尽可能相等。此外解算的主点应当符合图像比例。

1.2.2.1 精度提高

误差的高低是精度判断的依据，是否可接受依照应用需求而定。针对前述标定及精度评价，总结了几种方案可以用来改善测量精度：

- 采用高质量的靶标。需要保证靶标相对尺寸足以覆盖所需视场，每个小格的纵横比一致，且为矩形。靶标全局需要均匀对称。
- 修改参数设置。尝试更多的误差拟合系数、切向误差以及径向畸变，用更加复杂的模型进行计算。
- 良好的拍摄环境，尤其是光照条件。
- 拍摄更多的标定图像。靶标图像必须位于不同的 3D 朝向，并且保证全视场都有足够的关键点。特别注意，应该保证在边缘及角落处有角点，如此才能更好地估计畸变系数。剔除高重投影误差的图像，并重新校正。

拍摄图像需要保证在景深范围内尽量均匀。

1.3 基于单目视觉的平面圆测量

视觉测量技术具有非接触，精度高等特点。常见的视觉测量方法主要有单目视觉测量、双目视觉测量、结构光视觉测量。单目视觉测量过程中只使用一个摄像机，测量范围大，结构简单，广泛应用于机器人跟踪导航、飞行器对接、飞机姿态控制等多个领域。

本节使用了一种基于单目视觉的平面圆测量方法，通过对摄像机进行标定求出世界坐标系到图像坐标系转换的内外参数，进而实现对固定平面内物体的位置定位及尺寸测量。

1.3.1 原理



图 6 实验流程

测量流程如上图所示：

1、基于 Matlab Camera Calibrator 工具箱标定相机，获得相机的内部参数 A 和相对于圆盘所在平面的外部参数 R 、 t 。

2、将内外参数传至 OpenCV 中，注意两者旋转矩阵之间的求逆关系。矫正畸变得得到矫正后的内参矩阵以及矫正后的圆盘图像。

3、边缘保留滤波，滤除噪声的同时保留边缘信息，便于霍夫变换有效提取圆；转

为灰度图像便于后续处理。

- 4、霍夫变换提取圆，掩膜得到感兴趣区域。
- 5、采用自适应阈值二值化方法得到二值图像；Canny 算子提取待测平面圆边缘。
- 6、由前文可得

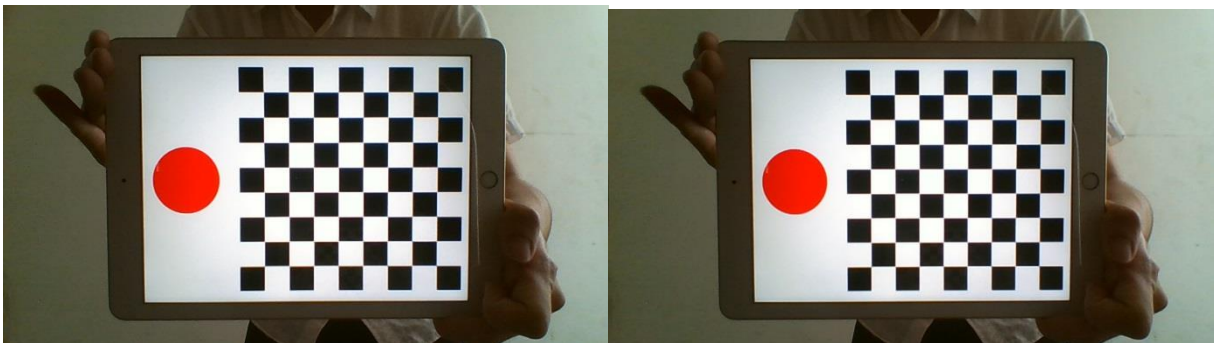
$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = sH^{-1} \begin{bmatrix} u \\ v \end{bmatrix} \quad (24)$$

由（24）可以根据图像坐标求解出物体的世界坐标。

- 7、最小二乘法拟合圆，测得平面圆半径。

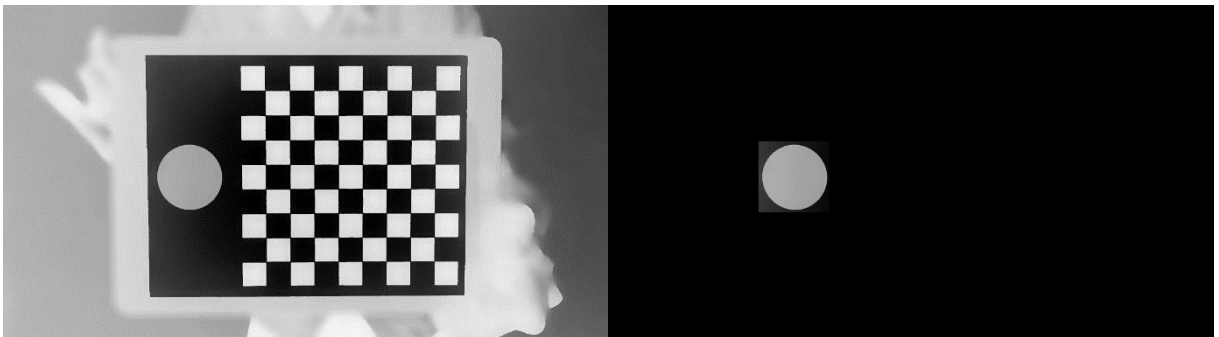
1.3.2 实验结果及分析

摄像头和标定设备与上一节所述相同，待测圆通过 Photoshop 生成，半径为 20.0mm。将待测圆绘制在靶标平面上，一同显示在显示器上，这样做的好处是待测圆和靶标严格处于同一平面上，标定与测量过程同时进行，有利于提高测量精度。经过图 6 所示的图像处理过程后，提取图像上圆的坐标。



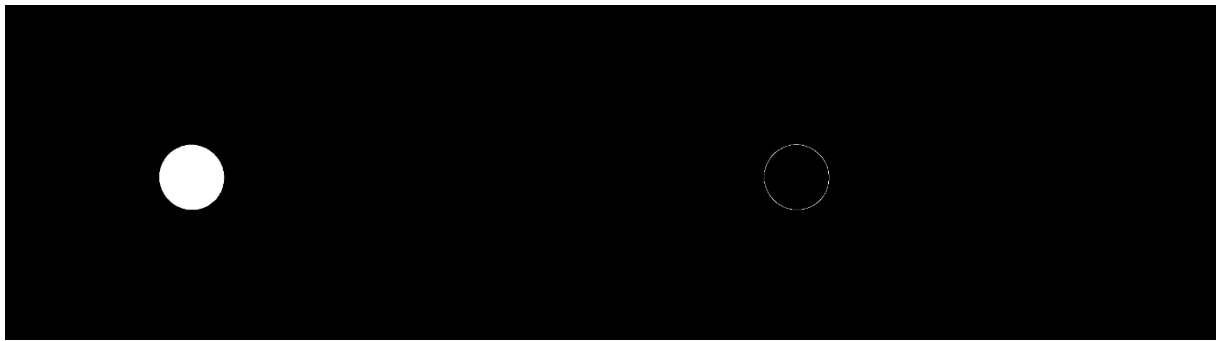
(a)原图

(b)矫正畸变



(c)灰度图

(d)掩膜 ROI



(e)二值化

(f)提取边缘

图 7 图像处理结果

内部参数和外部参数标定结果:

$$\text{camera matrix} = \begin{bmatrix} 923.968 & -0.015 & 654.662 \\ 0 & 925.408 & 404.100 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{distortion coefficients} = [0.063 \quad -0.202 \quad 0.001 \quad -0.001 \quad 0]$$

$$\text{rotation matrix} = \begin{bmatrix} 1.000 & -0.003 & 0.010 \\ 0.002 & 0.999 & 0.043 \\ -0.010 & -0.043 & 0.999 \end{bmatrix}$$

$$\text{translation vectors} = [-29.299 \quad -62.120 \quad 2.631]$$

```
In [48]: runfile('G:/Huang_Zhenkai/file/机器视觉/单目/measure.py', wdir='G:/Huang_Zhenkai/file/机器视觉/单目')
x0 = -69.02458136438108 y0 = 49.422534734775255 r = 19.908574276425615
```

图 8 测量结果

由提取到的图像坐标, 根据 (24) 计算世界坐标系下的圆盘坐标。再用最小二乘法拟合圆曲线得到圆盘半径 $R=19.91\text{mm}$, 误差约为 0.09mm 。这表明测量精度较高, 主要有两个原因: 一方面是相机标定精度高, 可以满足高精度的测量; 另一方面是由于圆盘与靶标由同一个显示器显示, 待测圆盘与已知平面共面性好。

1.4 结论

本文对机械革命 X6Ti-s 笔记本前置高清摄像头进行了标定并进行了精度评价, 通过分析发现了提高标定精度的有效措施; 提出了一种基于单目视觉的平面内目标的测量方法, 可用于测量位于固定平面内物体的位置及长度等。实验结果误差较小 (约 0.09mm), 验证了本方法的有效性。针对平面内目标提出的单目视觉位置测量方法,



应用前景广阔。例如，该方法可应用于道路交通中基于视觉的车辆速度估计，还可应用于建筑物视频监控中的目标位置与速度估计等。

2 双目相机标定及测量实验

2.1 双目测量原理

双目立体视觉是基于视差，由三角法原理进行三维信息的获取，即由两个摄像机的图像平面（或单个摄像机在不同位置的图像平面）和被测物之间构成一个三角形。已知两摄像机之间的位置关系，便可以获取两摄像机公共视场内物体的三维尺寸及空间物体特征点的三维坐标。双目立体视觉系统一般由两个摄像机或由一个运动的相机组成。

2.1.1 数学模型

如图 1 所示，设左相机 $O\text{-}xyz$ 位于世界坐标系的原点处且无旋转，图像坐标系为 $O\text{-}X_lY_l$ ，有效焦距为 f_l ；右摄像机坐标系为 $O_r\text{-}x_r y_r z_r$ ，图像坐标系为 $O_r\text{-}X_r Y_r$ ，有效焦距为 f_r ，由摄像机透视变换模型有

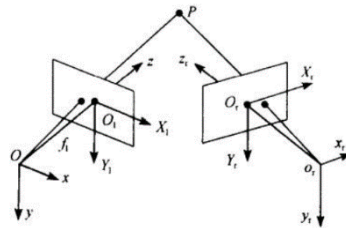


图 9 双目立体视觉中的三维重建

$$s_l \begin{bmatrix} X_l \\ Y_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_l & 0 & 0 \\ 0 & f_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (25)$$

$$s_r \begin{bmatrix} X_r \\ Y_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_r & 0 & 0 \\ 0 & f_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} \quad (26)$$

而 $O\text{-}xyz$ 坐标系与 $O_r\text{-}x_r y_r z_r$ 坐标系之间的相互位置关系可通过空间转换矩阵 M_{lr} 表示为

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = M_{lr} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, M_{lr} = [R|T] \quad (27)$$

其中, R 和 T 分别为 O -xyz 坐标系与 O_r - $x_r y_r z_r$ 坐标系之间的旋转矩阵和原点之间的平移变换矢量。由 (25) ~ (27) 可解出空间点三维坐标

$$\begin{cases} x = zX_l / f_l \\ y = zY_l / f_l \\ z = \frac{f_l(f_r t_x - X_r t_z)}{X_r(r_7 X_l + r_8 Y_l + f_l r_9) - f_r(r_1 X_l + r_2 Y_l + f_l r_3)} \end{cases} \quad (28)$$

因此, 已知焦距 f_l 、 f_r 和空间点在左右摄像机中的图像坐标, 只要求出旋转矩阵 R 和平移矢量 T 就可以得到被测物体点的三维空间坐标。

2.1.2 极线几何

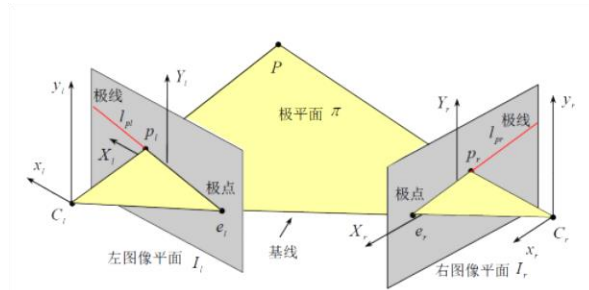


图 10 极线几何

极线约束是双目匹配中极重要的约束条件。如图所示, 称直线 $\overline{e_l p_l}$ 为图像 I_l 上对应于 p_r 点的极线, 直线 $\overline{e_r p_r}$ 图像 I_r 上对应于 p_l 点的极线。若已知 p_l 在左图像上的位置, 则右图像上 p_r 一定在直线 $\overline{e_r p_r}$ 上, 反之亦然, 这被称为极线约束。

两相机的投影方程可表示为

$$\begin{cases} s_l p_l = M_l X = M_{l1} X + m_l \\ s_r p_r = M_r X = M_{r1} X + m_r \end{cases} \quad (29)$$

其中, p_l 和 p_r 分别为空间点在左右相机中的图像坐标, M_l 、 M_r 分别为左右相机投影矩阵, 消去 X 后可得极线约束方程

$$p_r^T F p_l = 0 \quad (30)$$

其中 $F = [m]_x M_r M_l^{-1}$, 被称为基本矩阵。

2.1.3 双目标定

在常规标定法中, 由标定靶标对两个摄像机同时进行摄像标定, 以分别获得两个摄像机的内外参数, 同时标定出双目立体视觉系统的结构参数。通过张正友标定法分别得到左右相机的内外参数, 其中外部参数可表示为 R_l 、 T_l 与 R_r 、 T_r 。对任意点, 它在世界坐标系, 左相机坐标系和右相机坐标系下的坐标分别为 x_w 、 x_l 和 x_r , 则

$$x_l = R_l x_w + T_l, x_r = R_r x_w + T_r \quad (31)$$

消去 x_w 可得 $x_r = R_r R_l^{-1} x_l + T_r - R_r R_l^{-1} T_l$ 。因此, 双目立体视觉系统的结构参数可表示为

$$R = R_r R_l^{-1}, T = T_r - R_r R_l^{-1} T_l \quad (32)$$

至此就完成了双目立体视觉系统的标定。

2.2 双目相机测量实验结果及分析

2.2.1 实验流程



图 11 双目测量流程

2.2.2 标定相机

利用 Matlab Stereo Camera Calibrator 工具箱进行双目标定得到了两个相机各自的内参矩阵, 畸变向量以及右相机到左相机的平移矢量和旋转矩阵。值得注意的是, 在 OpenCV 调用 Matlab 标定得到的参数时, OpenCV 中定义的旋转矩阵与 Matlab 定义的

旋转矩阵之间有一个求逆关系。

内参矩阵:

$$A_l = \begin{pmatrix} 2749.308 & 0.186 & 835.439 \\ 0 & 2749.156 & 617.821 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A_r = \begin{pmatrix} 2748.435 & -0.319 & 838.256 \\ 0 & 2748.455 & 628.644 \\ 0 & 0 & 1 \end{pmatrix}$$

平移矢量和旋转矩阵

$$T = (-185.184 \quad -1.885 \quad 44.167)$$

$$R = \begin{pmatrix} 0.911 & 0.014 & -0.411 \\ -0.009 & 1.000 & 0.015 \\ 0.411 & -0.010 & 0.911 \end{pmatrix}$$

畸变:

$$dist_l = (-0.101 \quad 0.432 \quad -0.0001 \quad -0.0001 \quad 0)$$

$$dist_r = (-0.081 \quad 0.210 \quad -0.0005 \quad -0.0002 \quad 0)$$

2.2.3 立体矫正

立体校正,把实际中非共面行对准的两幅图像,校正成共面行对准,提高匹配搜索的效率,将二维搜索变为一维搜索。(共面行对准:两摄像机图像平面在同一平面上,且同一点投影到两个摄像机图像平面时,应该在两个像素坐标系的同一行)。

立体矫正得到消除畸变的左右相机旋转矩阵 R_1 、 R_2 , 以及投影矩阵 P_1 、 P_2 , 由此对待测图像进行校正得到图 13。

此外立体矫正得到重投影矩阵 Q , 实现了世界坐标系和左视图图像像素坐标系之间的转换。

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 1 & 0 & -\frac{1}{T_x} & \frac{c_x - c'_x}{T_x} \end{bmatrix} \quad (33)$$

其中 c'_x 表示右图像的主点, 当校正正确时, $c_x = c'_x$

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (34)$$

式中 d 为视差，有 (33) (34) 可得三维坐标：

$$(X/W, Y/W, Z/W) = \left(-\frac{x - c_x}{d} T_x, -\frac{y - c_y}{d} T_x, -\frac{f}{d} T_x \right) \quad (35)$$



图 12 左右相机的原始图像

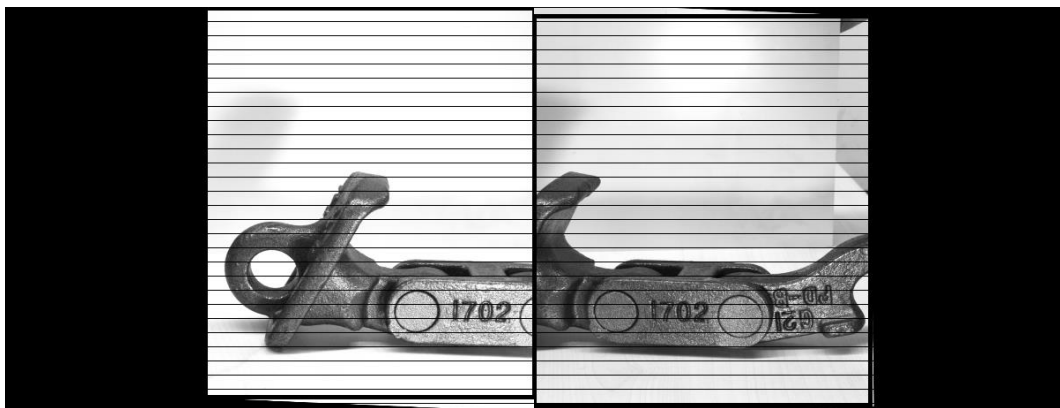


图 13 立体矫正后左右图像

2.2.4 图像预处理

对左右视图分别进行处理：

- 1、边缘保留滤波，滤除噪声的同时保留边缘信息，便于霍夫变换有效提取圆；转为灰度图像便于后续处理。
- 2、霍夫变换提取圆，掩膜得到感兴趣区域。
- 3、采用自适应阈值二值化方法得到二值图像。
- 4、进行形态学闭操作填补边缘空洞，填平窄缝使物体边缘更平滑。
- 5、Canny 算子提取待测左右视图中圆的边缘。

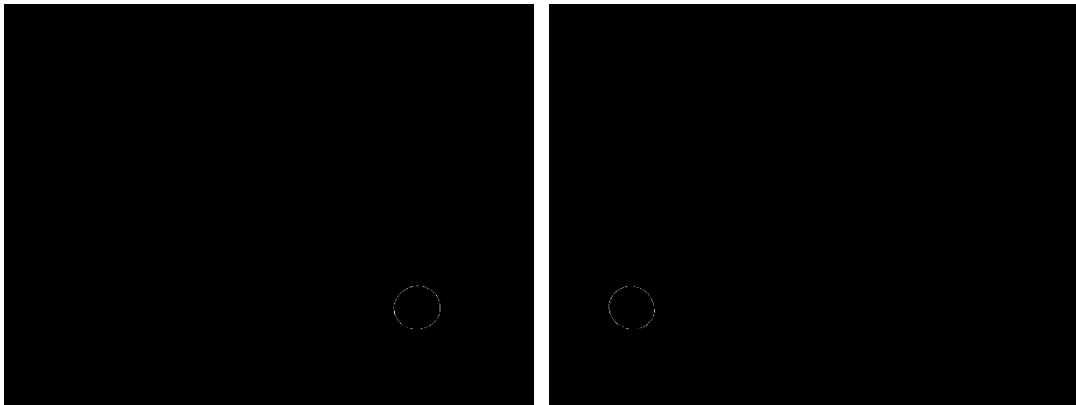


图 14 左右视图待测圆边缘

2.2.5 特征点匹配与视差计算

在上述步骤得到的边缘信息中，在左视图中选边缘的顶端、底端、最左端和最右端为待匹配特征点，根据极线约束在右视图进行搜索。理论上，在立体矫正后，左右视图共面行对准，因此其在右视图的对应匹配点也为右视图边缘的顶端、底端、最左端和最右端特征点。因此共找到四对特征点。

计算视差：

$$d = x_{left} - x_{right} \quad (36)$$

2.2.6 计算圆盘的直径

将上述步骤中提取四对特征点代入(35)，将这四对特征点投影回世界坐标系下，得到八个特征点的世界坐标。求这些点的拟合球以及拟合平面，则这二者的交线就是要测量的三维圆。设球心到平面距离为 l ，球半径为 r_{ball} ，则圆盘直径为：

$$d = 2\sqrt{r_{ball}^2 - l^2}$$

最终求解的结果为 26.10mm，通过游标卡尺测得工件上的圆的直径为 25.06mm，误差为 1.04mm。

```
In [3]: runfile('G:/Huang_Zhenkai/workspace/github/machine-vision/双目/measure2.py',
wdir='G:/Huang_Zhenkai/workspace/github/machine-vision/双目')
a = 0.0004704360948323444
b = -7.393697697464354e-05
c = -0.0019253147244190402
x0 = 6.189126774302395
y0 = 121.09739836221875
z0 = 509.0649352195793
r_ball = 14.802148078920796
d = 26.103956616334802
```

图 15 计算结果



2.3 结论

本节对双目相机进行标定，以及对复杂工件上的圆盘进行测量。首先通过相机标定得到左右两个相机的内外参数和双目视觉系统的结构参数，使用该参数对图像进行立体校正，对左右视图消除畸变及共面行对准。随后，在图像坐标系下提取左右视图圆盘边缘。将边缘的顶端、底端、最左端、最右端作为特征点，将其投影到世界坐标系下，使用特征点计算拟合球和拟合平面，进而得到拟合圆的直径。实验测量的结果为 26.10mm，误差为 1.04mm。



3 增强现实（AR）的 OpenCV 实现

增强现实（AR）是当前最热门的趋势之一。增强现实的概念可以定义为对现实进行改进，其中现实世界的视图通过叠加的计算机生成元素（例如图像、视频或 3D 模型等）得到增强。AR 主要分为两种不同类型的技术，分别是基于位置和基于识别的方法。

基于位置和基于识别的增强现实是增强现实的两种主要类型。AR 系统的关键在于如何将扩增物件与实际环境结合。AR 算法软件必须要从入设备中的影像取得真实世界的坐标，再将扩增物件叠合到坐标。这两种类型都试图计算出用户正在观察的位置。这一信息是增强现实过程中的关键，并依赖于精确地摄像机姿态估计。为了完成这一任务，以下简要描述了这两种方法的原理。

3.1 算法原理

基于位置的增强现实依赖于通过从多个传感器读取数据来检测用户的位置和方向，这些传感器在智能手机设备（如 GPS、数字罗盘和加速度计）中非常常见，以得出用户正在查看的位置。此信息用于在屏幕上叠加计算机生成的元素。

另一方面，基于识别的增强现实使用图像处理技术来获得用户正在查看的位置。从图像中获取相机姿势需要找到环境中已知点之间的对应关系及其相应的投影关系。主要有两种方法：有标记姿态估计和无标记的姿态估计。

基于有标记姿态估计：这种方法依赖于使用平面标记(主要是基于正方形标记的方法)从四个角计算相机的姿态。姿态估计时使用方形标记的一个主要缺点是它依赖于标记的四个角的精确确定。在有遮挡的情况下，这个任务可能非常困难。然而，一些基于标记检测的方法也可以很好地处理遮挡。

基于无标记的姿态估计：当场景无法使用标记进行姿态估计时，图像中自然存在的对象可用于姿态估计。在计算出一组 N 个二维点及其对应的三维坐标后，通过解(PNP)问题来估计相机的姿态。由于这些方法依赖于点匹配技术，然而输入数据难免有异常值（噪声）。因此需要在姿态估计过程中可以使用异常值的鲁棒技术（例如 RANSAC）。

3.1.1 有标记 AR 原理

基于标记的 AR 依赖于实现制作好的标记（Marker）。二维标记为一个平面，假定场景的几何约束为平面约束，并由此进行世界坐标解算。考虑标记在不同场景中的辨识度

以及图形的复杂程度，如下图所示，本项目采用的标记为二维海明码。

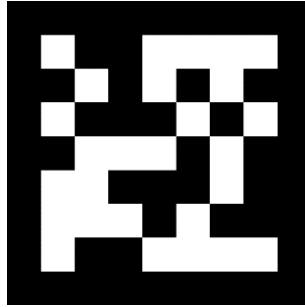


图 3.1 二维海明码

海明码的优势是其特征的鲁棒性。中心的 7×7 网格区域包含该二维码的 ID 信息，黑色表示 0，白色表示为 1，则标记内部每一行为 1 个 7 bit 字符串，字符串以海明码方式编码，3 bit (1,2,4) 用于校验，4 bit 用于存放数据。这种规则编码方式具有高鲁棒性以及旋转不变性，根据该编码特征即可判断候选区域是否为预先设置的标记区域，并且识别该标记的旋转方向。

第一步是相机标定，获得相机内参矩阵，不赘述。第二步是姿态估计。标记是规则的平面正方形，具有很强的几何约束。假定世界坐标系建立在标记平面，且标记中心为世界坐标系，XY 轴与正方形的边平行。

根据摄影几何模型，从图像坐标系到世界坐标系的射影变换如下。其中 $u-v$ 为图像坐标系坐标， $X_m Y_m Z_m$ 为标记平面世界坐标系坐标。A 为 3×3 的相机内参矩阵，由相机标定获得。R 为世界坐标系至相机坐标系的旋转矩阵，T 为平移矢量，[R T] 组成一个待求解的 3×4 的外参矩阵。当标记识别出后，最少用 4 个角点即可联立解算外参矩阵，角点通常选取为矩形角点。

$$\rho \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R \ T] \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

首先根据摄像头采集到的图像进行标记特征识别，之后根据前述原理判断标记角点在图像坐标系坐标及其对应的世界坐标系坐标，由于假定平面约束， Z_m 分量均为 0，由此便可解算出外参矩阵。

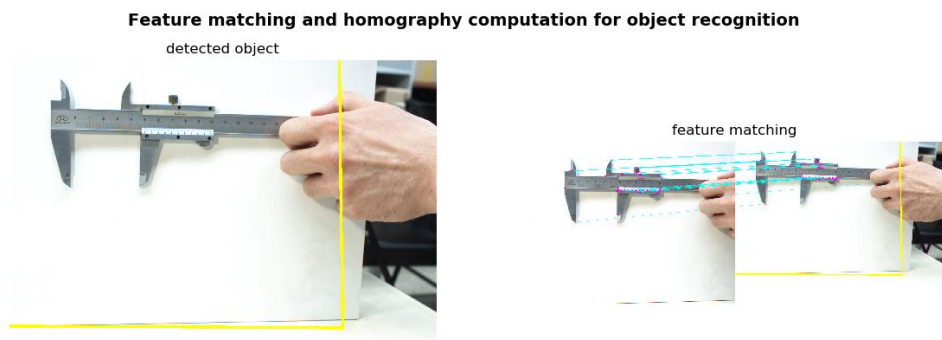
最后一步是扩展物体绘制。扩展物体的绘制是上述过程的逆过程。已知扩展物体每个三维点在标记世界坐标系的坐标，带入上述公式可求解图像坐标系坐标，由此便可以在相机采集的图像中实时绘制。

3.1.2 无标记 AR 原理

无标记增强现实与有标记增强现实的步骤基本相同，都属于有场景几何先验的增强现实，两者区别在与特征角点提取方法。

无标记的 AR 与有标记 AR 区别在于基准标记并非规则形状，依赖于目标纹理的空间分布特征，比如 SURF、ORB 特征等特征检测算法。特征纹理中的特征点，包括边缘，角点以及斑点等。这些特征方向和幅值两个分量。特征点的方向通过计算局部梯度的统计信息获得，且具有旋转不变性。特征点的大小可以固定大小，也可以自适应或者尺度不变。本项目采用 ORB 特征。

图 3.2 优选后的目标



ORB 特征及其匹配

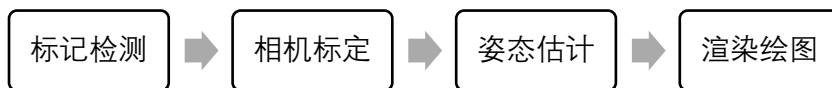
无标记增强现实需要首先采集平面目标的图像，之后检测每一帧图像中的 ORB 特征，并且与第一帧平面目标的 ORB 特征进行匹配，从而寻找特征的对应关系，并且剔除离群点。之后根据图像 ORB 特征在图像中的空间分布情况，并假设世界坐标系在目标平面即可确定外参矩阵。这一解算与有标记基本相同，区别在于无标记筛选的特征数量往往远大于 4 个，因此为非线性优化求解。求得外参矩阵后，扩展物体的显示算法与有标记相同。

3.2 有标记 AR 的实现

基于标记的增强现实，有许多库、算法或包可以用来生成和检测标记。我们选择使用 ArUco 工具包。因为，ArUco 能自动检测标记并更正可能的错误。

如前所述，姿态估计是增强现实应用中的一个关键过程。姿态估计可以基于标记进

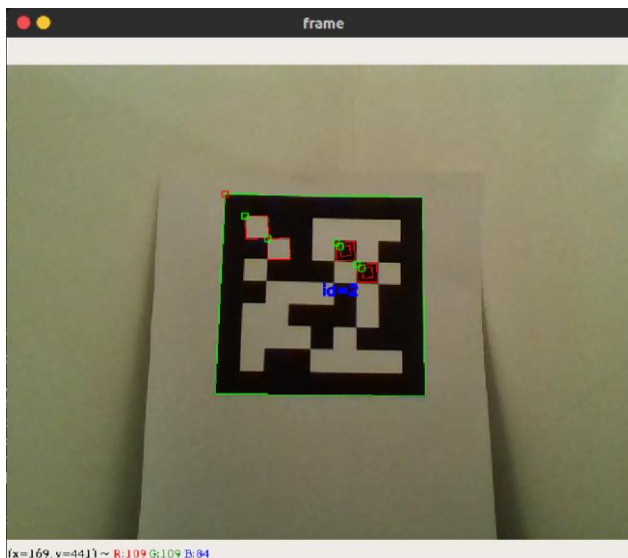
行。该方法的主要优点是能够高效、鲁棒地检测出图像中标记点的位置，从而准确地提取出标记点的四个角。最后，通过之前计算出的标记的四个角可以得到相机的姿态。处理流程如下：



3.2.1 标记检测

使用 `cv2.aruco.detectMarkers()` 函数检测图像中的标记。输入要检测标记的灰度图像。函数返回以下信息：返回检测到的标记的角列表。对于每个标记，返回其四个角（左上、右上、右下和左下）；返回检测到的标记的标识符列表。返回被拒绝的候选标记列表，该列表由找到的所有方块组成，但没有适当的编码。这个被拒绝的候选标记列表对于调试很有用。每个被拒绝的候选标记也由四个角组成。

`aruco_detect_markers.py` 从摄像头检测标记。首先，使用上述 `cv2.aruco.detectMarkers()` 函数检测标记，然后使用 `cv2.aruco.drawDetectedMarkers()` 函数绘制检测到的标记和拒绝的候选标记。



标记用绿色边框绘制。此外，您还可以看到两个带有红色边框的被拒绝候选标记。

3.2.2 姿态估计

在进行相机标定获得相机内参矩阵和畸变系数后开始估计摄像机姿态，使用 `cv2.aruco.estimateposeSingleMarkers()` 函数，估计单个标记的姿态。姿态由旋转和平移

矢量组成。函数为每个检测到的标记返回 `rvecs`(旋转向量)、`tvecs`(平移向量)和 `_objPoints`(所有检测到的标记角的对象点数组)。

标记坐标系集中在标记的中间。因此,标记的四个角的坐标(在其自己的坐标系中)如下:

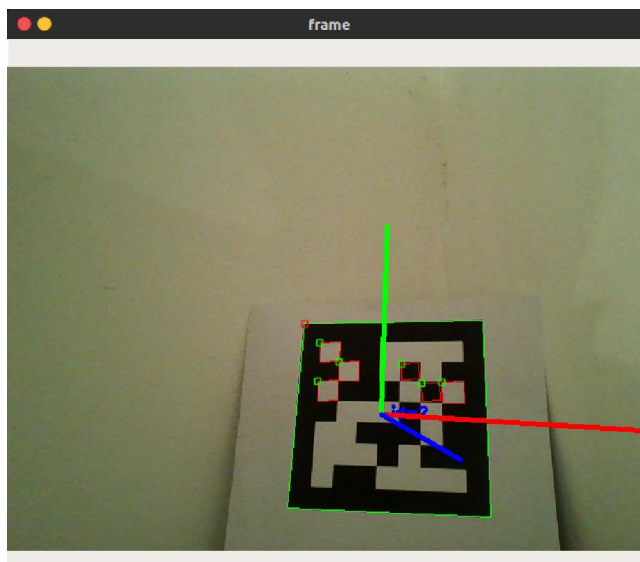
$(-markerLength/2, markertlength/2, 0)$

$(markerLength/2, markertlength/2, 0)$

$(markerLength/2, -markertlength/2, 0)$

$(markerLength/2, -markertlength/2, 0)$

最后,用 `cv.aruco.drawAxis()` 函数,为每个检测到的标记绘制坐标轴。



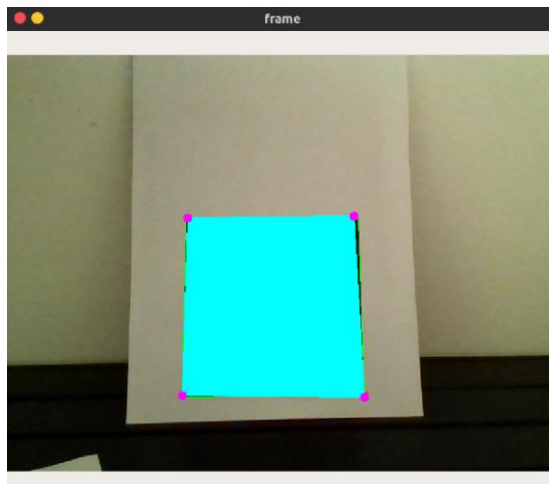
可以看到只检测到一个标记,而且该标记的系统轴也已绘制。

3.2.3 渲染绘图

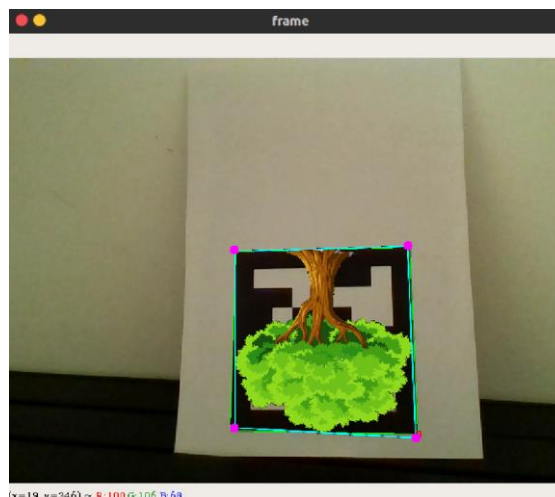
此时,可以覆盖一些图像、形状或三维模型,以便看到一个完整的增强现实应用程序。在第一个示例中,我们将用标记的大小覆盖一个矩形。

第一步是定义要覆盖图像的点。标记上覆盖矩形的坐标是 $[[-1/2, 1/2, 0], [1/2, 1/2, 0], [1/2, -1/2, 0], [-1/2, -1/2, 0]]$ 。在标记坐标系中定义这些坐标。第二步,使用 `cv2.projectPoints()` 函数投影这些点。最后绘制出来。

如图,可以看到青色矩形覆盖在检测到的标记上。此外,您还可以看到矩形的四个角,它们以洋红色绘制。



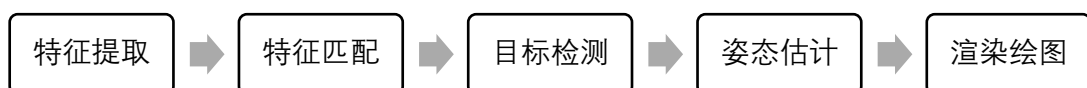
接下来在标记上绘制一棵树。



要覆盖更复杂和高级的 3D 模型，可以使用 OpenGL。开放式图形库（OpenGL）是一个用于渲染二维和三维模型的跨平台 API。

3.3 无标记 AR 的实现

摄像机姿态估计可以从图像中得到，以找出环境中已知点与其摄像机投影之间的对应关系。主要是从图像中提取特征来获得相机姿态。



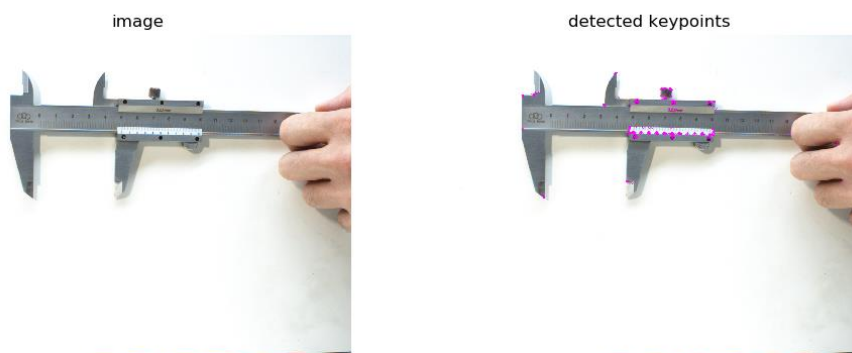
3.3.1 特征提取

图像特征应该对图像缩放、旋转和照明都是不变的（尽可能多）。这样，可以从同一场景的不同图像中以不同的视角检测到相同的特征。因此，好的特征应该具有可重复性和精确性（应能从同一对象的不同图像中提取相同的特征）。

OpenCV 提供了许多算法和技术来检测图像中的特征。其中包括以下内容: Harris 角点检测; Shi-Tomasi 角点检测; SIFT; SURF; FAST; BRIEF; ORB 等。

我们使用 ORB 进行图像中的特征检测和描述。ORB 基本上是 FAST 特征点检测算法和 BRIEF 特征描述子的组合, 通过修改提高了性能。第一步是检测特征点。ORB 使用修改后的 FAST-9(半径为 9 像素的圆, 并存储检测到的特征点方向)检测特征点。一旦检测到特征点, 下一步是计算特征描述子, 以获得与每个检测到的特征点相关联的信息。ORB 使用修改过的 BRIEF-32 描述子来获取每个检测到的特征点的描述。

ORB keypoint detector



结果显示了由 ORB 特征点检测器检测到的特征点。

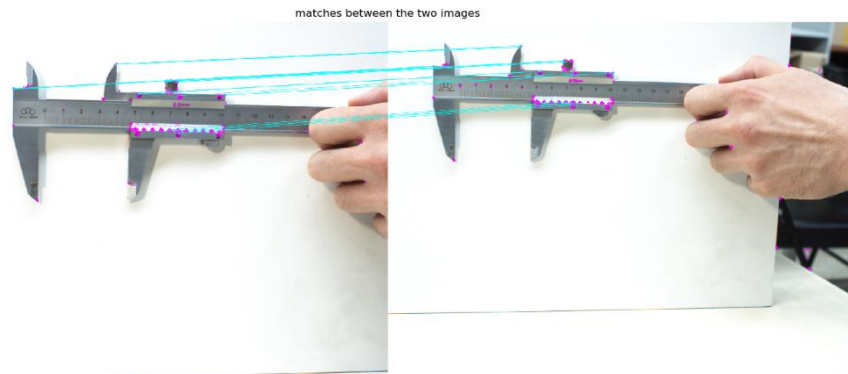
3.3.2 特征匹配

匹配检测到的特征。OpenCV 提供两个匹配器, 如下所示:

Brute-Force (BF) matcher: 此匹配器获取第一组中每个特征点的描述子, 并与第二组中的所有其他描述子匹配。最后, 返回最近距离的匹配对。本实验使用 BF matcher 进行特征匹配。

FLANN matcher: 对于大型数据集, 此匹配器比 BF matcher 工作得更快。它包含用于最近邻搜索的优化算法。

ORB descriptors and Brute-Force (BF) matcher

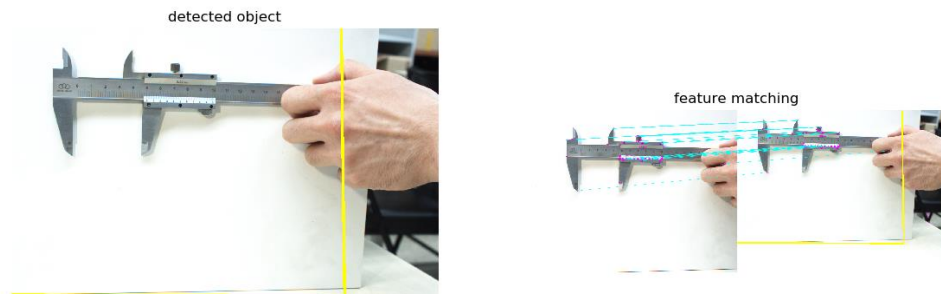


3.3.3 目标检测

目标检测。一旦匹配特征，下一步就是使用 `cv2.findhomography()` 函数在两个图像中匹配特征点的位置之间找到透视转换关系。`opencv` 提供了几种计算单应性矩阵的方法：`RANSAC`、`LMEDS`、`PROSAC(RHO)`的方法。在这个实验中，我们使用的是 `RANSAC`。如果重投影误差大于 5.0，则相应的匹配对被视为异常值。函数计算并返回由特征点位置定义的源平面和目标平面之间的透视转换矩阵。

最后，基于透视变换矩阵，我们将计算图像中对象的四个角。为此，我们将根据原始图像的形状计算其四个角，并使用 `cv2.perspectiveTransform()` 函数将其转换为目标角。

Feature matching and homography computation for object recognition



可以看到特征匹配和单应性矩阵计算，是目标识别的两个关键步骤。