实验数据采集：

准备：相机（及配套软件）， 3D汽车模型，棋盘格标定板， 6自由度云台，角度测量仪

实验步骤：

1. 设置相机分辨率为1296\*972，帧率为7 fps
2. 拍摄一定数量（>15）任意不同位置的棋盘格图像，用于标定内参（内参已有，此步可省略）。标定程序见calibration\_V3.py
3. 外参（双目）标定，拍摄要求同步骤2， 注意棋盘格要在两个相机的视野中。为保证外参精度，每次实验前均需进行标定。
4. 将汽车模型固定在云台上，用角度测量仪测量云台平面的倾斜角并调节至水平（可借助千斤顶）
5. 读数：记录云台的x, y, z三轴的读数，对应的角度alpha(x轴旋转)，beta(y轴旋转)，gamma（z轴旋转）
6. 拍摄此基准位置的照片。
7. 调节云台位置，包括平移、旋转，拍摄一组照片并记录每次调节后的6自由度读数。平移量x, y, z和旋转角gamma直接从云台刻度读取，旋转角alpha, beta用角度测量仪测量。

项目目标：通过单个或多个相机求解物体位姿，得到物体在基准位置与偏移位置之间的旋转矩阵R与平移向量t。

开发环境：

Python 3.6.4

MATLAB >=2017b

OpenCV 3.4

方法介绍：

方法1（单目）：

相机成像的过程可以用透视几何来描述，根据小孔相机模型，空间中任意一点P与其在图像平面的投影p之间的变换可以用两次线性变换描述，共计11个参数。因此，当这样的3D-2D点对足够多时，即可通过求解线性方程组得到这11个参数，进而可以恢复出物体位姿R, t。

对于该项目，汽车模型的三维坐标是事先已知的，因此可以通过定义一些特征点对来求解参数。比如在基准图像上可以找到100个特征和对应的三维坐标，就可以求得物体所在的坐标系（世界坐标系）到相机所在的坐标系之间的转换（R, t）。

更进一步，如果基准图像上的这些特征在其它图像中也可以被检测到，那么就可以通过特征之间的匹配关系，得到一组新的3D-2D点对，进而求得移动后的物体与相机之间的变换(R’, t’)。由于相机是固定不变的，所以可以解出物体在不同位置的变换关系。

算法具体可分为三个阶段：

1. 特征提取：实际拍摄图片如图1所示，图像中包含大量圆斑（Blob），边缘及角点。比较适合处理的特征是Blob。方法可参考OpenCV文档：

<https://www.docs.opencv.org/3.4.1/d0/d7a/classcv_1_1SimpleBlobDetector.html>

详见代码feature\_detection.py。

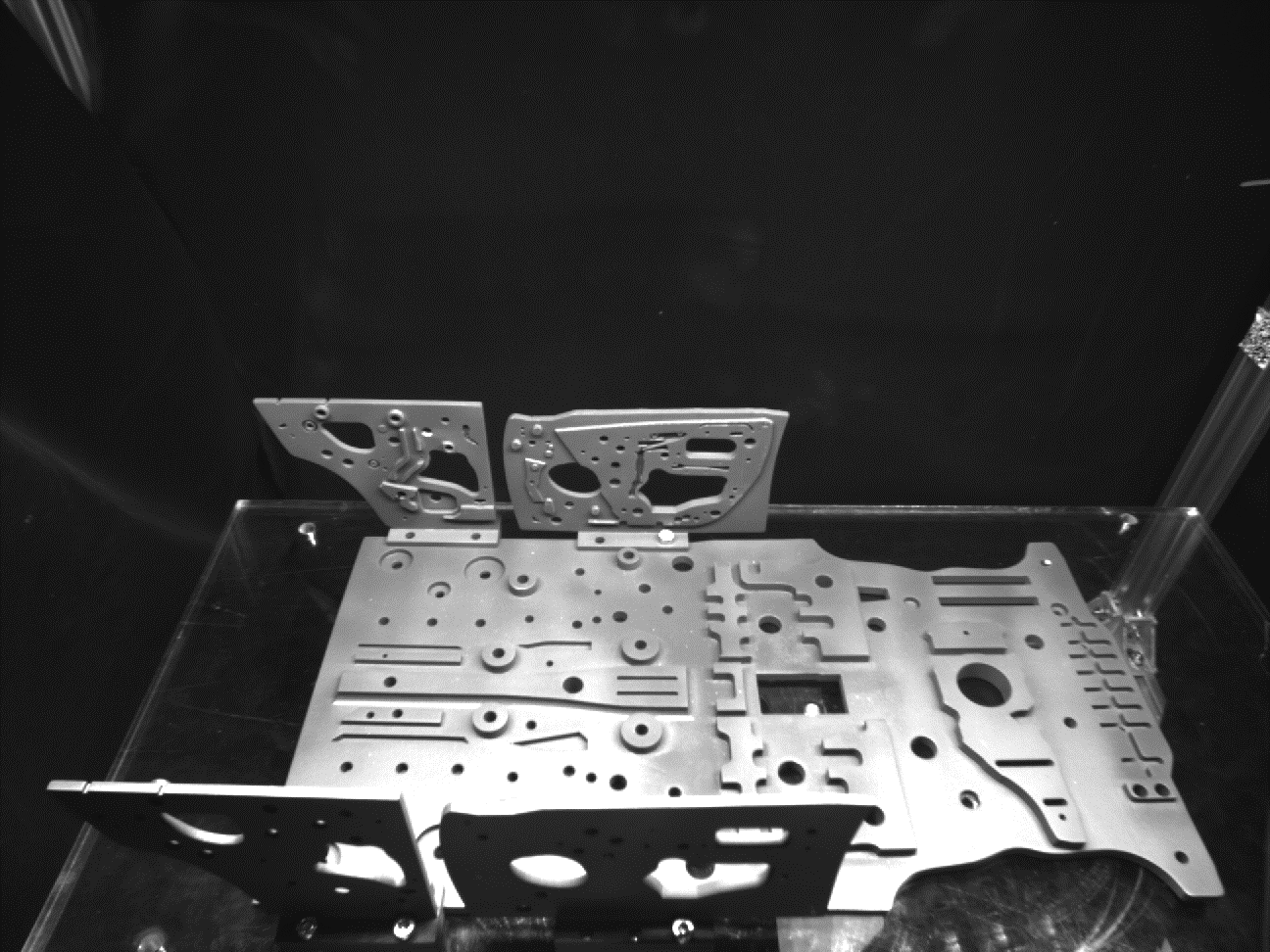


图1 汽车模型图

1. 特征匹配：2D-3D的特征匹配只对基准图像进行，目前采用人工标注的方法，标注完成后用Solidworks导出。考虑到特征之间相似性较高，2D-2D的匹配未采用常用的ORB一类的单点匹配方法，而是将所有点用混合模型表示，将一个点集看作另一个点集的抽样，再通过最大似然估计求解参数。参考论文：*Myronenko A, Song X. Point set registration: coherent point drift[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 32(12):2262-2275.*

代码用MATLAB和C编写，可通过Python调用。详见feature\_matching.py.

1. 位姿求解：通过3D-2D的匹配点对求解位姿采用的是Perspective-n-Points方法（见OpenCV文档），输入包括一组三维点坐标，一组对应的二维点坐标以及相机内参。输出即为位姿R, t。参与计算的坐标系包括：相机坐标系（原点在光心）、世界坐标系（原点自定义）、云台坐标系（原点在云台中心）。详见solvePoseFromOneCamera.py

Note: 利用2D特征点找对应的3D点需使用solid works软件。首先应打开带标号的基准图像（标号可通过feature\_matching.py模块中的showlabel函数获得），然后按顺序在solid works中选择三维点，自动导出为txt文件。随后可用extract3Dcoordinates.py读取并保存为python的数据结构，以供算法调用。txt文件中的文本格式有一定要求，详见extract3Dcoordinates.py.

方法2（双目）：

利用双目求解位姿的方法与单目不同，对基准图像和偏移之后的图像分别进行三维重建，然后解出重建的两个点集的刚体变换即可。