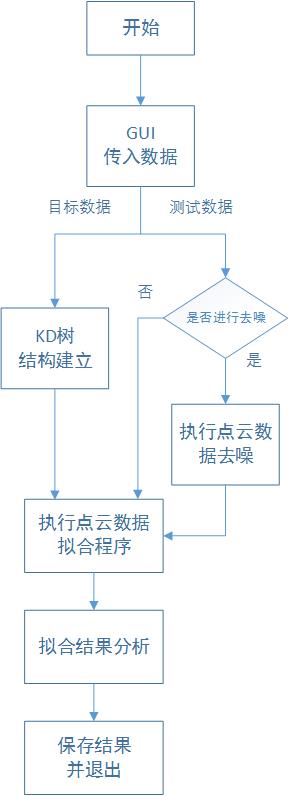
点云算法说明书

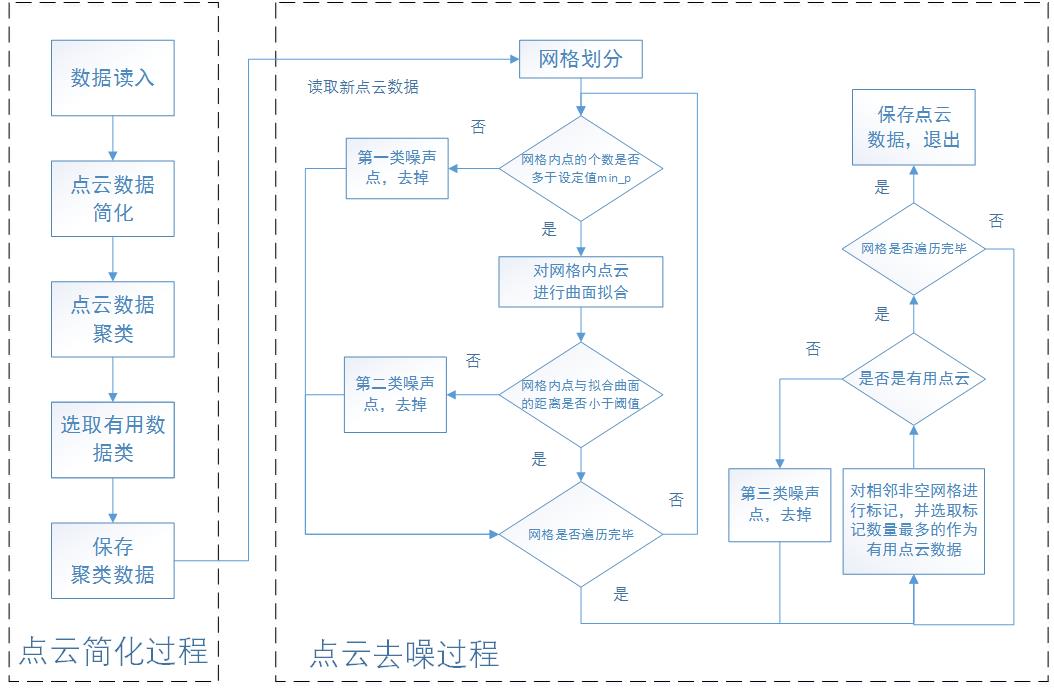
本次点云处理过程基本流程图如下：



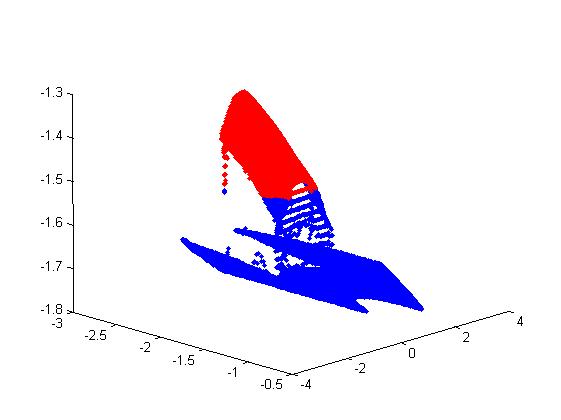
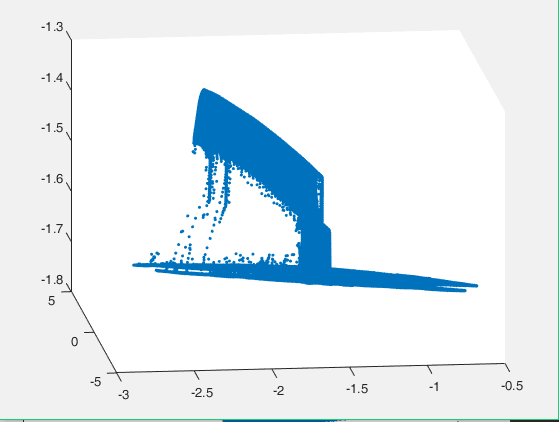
GUI从电脑中读取目标数据和测试数据，首先是显示原始数据，然后点击去噪按钮，程序执行点云去噪算法，并返回去噪点数，另一方面，点击拟合按钮时，程序执行点云拟合程序，若没有对点云进行去噪时，默认则是用原始测试数据去与目标数据进行拟合。

程序包含两部分算法模块：点云去噪算法和点云拟合算法。

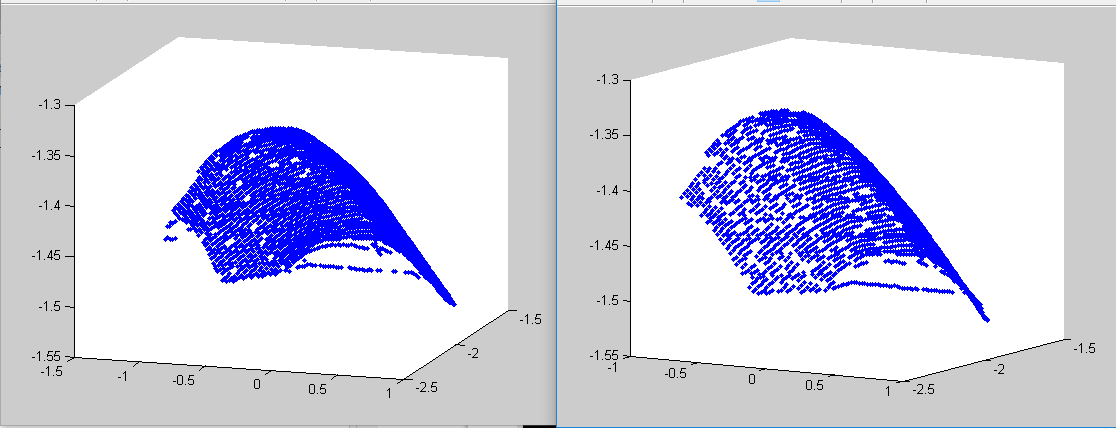
其中点云去噪算法的实现流程图如下：



点云去噪算法包含两个过程，点云简化和点云去噪。首先是点云简化的过程，因为由于激光扫描仪器得到的数据比较庞大，而且有很大一部分是背景数据，是对我们拟合毫无用处的，因此需要我们首先在进行点云处理前对其做预处理将背景去掉以及将数据量缩小。采用的方法是用网格法，将网格内所包围的点云判定为同一类点云，然后用这一类点云的中心来代替原始点云数据（通过设置cubsize参数来设定网格的尺寸以保证点云得到一定的简化，并且基本不失真）。将简化后的数据作为新的点云数据进行聚类，以达到将有用点云和背景分开得而效果，这里写了两种方法：第一种是基于密度和距离的聚类，即从一点开始，若以它为中心半径为r范围内的点的个数大于最小密度值min\_p，则认为是同一个类中的点，并从原点云数据中去掉该点，然后以这些相邻点再去做相同的密度聚类，直到没有满足条件得而点为止，然后继续去遍历点云数据中还没被去掉的点，直到原始点云数据中所有的点都已被聚类，这种聚类方法比较精确，但是很依赖于半径和密度值。第二种是采用kmeans聚类的想法，首先人为定义类的个数（这里因为是要将点云和背景分开，所以设定为两个类），然后随机选取两个点作为这两个类的初始中心点（为了精确，进行多次选取，以方差最大的点作为初始点），然后去遍历所有的点，将距离某一个类中心点较近的点划为与其同一类的点，并同时更新中心点坐标，直到所有点都被聚类（这里从数据上看，因为x、y轴坐标相对z轴坐标差值大得多，所以聚类时，只是用z轴来做聚类），聚类效果如下图所示，可以看到，基本的形状是可以分开的，但是因为是基于z轴坐标来分类的，所以很依赖于图像初始的角度以及初始类中心点的选择，优化可以从加入x、y轴坐标权重到分类标准中或者是用其他更好的方法来做聚类来着手。

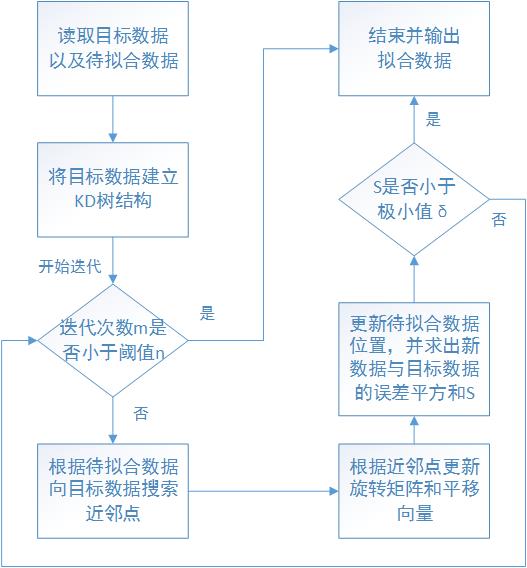


然后就是利用上述聚类后得到的有用数据来进行点云去噪，采用三种方式进行去噪，第一步，首先对数据进行网格划分，判断网格内点的个数是否满足最小值条件，与上面简化时做的工作差不多，然后对网格内的数据进行曲面拟合，采用的是最小二乘法来求解的，然后判断点到曲面的距离是否小于所设定的阈值，如果大于的话，就当作是噪声点处理，最后是通过标记法来对点云做最后的聚类，采用的是迭代法，将相邻的元素记相同标签，类似于上面简化中的密度法聚类，但是这里做分类是必不可少的，因为上面做简化时并不会考虑噪声的情况，而这里分类后选取点数最多的类作为有用点云，其他不在这个类中的点都会被当作噪声点去掉。最后去噪后的点云图像为（图为两种参数下的去噪效果，网格尺寸cubsize = (a) 0.03 (b) 0.05）：



1. (b)

接着是拟合图像，基本流程图为：



这里采用的最经典的icp（迭代搜索就近点）算法（这个网上的资料比较多，这里简单介绍一下icp算法，顾名思义，它是一种迭代算法，每次根据当前点云数据向目标点云数据搜索就近点，用最小二乘法求解出旋转矩阵和平移矩阵，然后根据旋转矩阵和平移矩阵来更新点云数据，并计算它与目标点云数据的误差平方和，如果平方和小于某个最小误差时停止迭代，否则不断重复上述工作，根据不同的搜索就近点的方法，icp算法分为点到点、点到面和点到物三种算法，本文用到了点到点和点到面算法实现），当然，在进行拟合前，需要对数据进行预处理，对目标数据是KD树构造（KD树的实现：KD树就是k维的二叉搜索树，根据KD树结构，我们可以很快的搜索到与它相邻的点，实现也比较简单，首先是计算每个维度的方差，选取方差最小的为当前分割维度，然后按照分割维度的坐标进行排序，选取中点作为分割点，然后对两边的数据重复上述过程，直到所有点云数据都在KD树结构中），而对待拟合点云数据则是进行坐标变换，这里所做是x、y轴按目标点云数据的100\*100来划分，z轴将最大最小值按比例扩到到与目标数据的z轴范围内，这里就有一个严重的问题，因为两个点云的形状、角度都不一样，那么人为去定义范围没有多少意义，如果能将目标点云数据图像和拟合点云数据图像旋转至相同角度，再去做坐标变换，这样就精确的多了。

下图是目前根据上述方法拟合的结果：

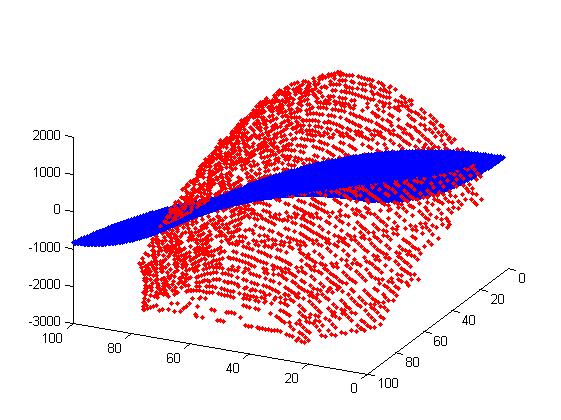


图 待拟合图像

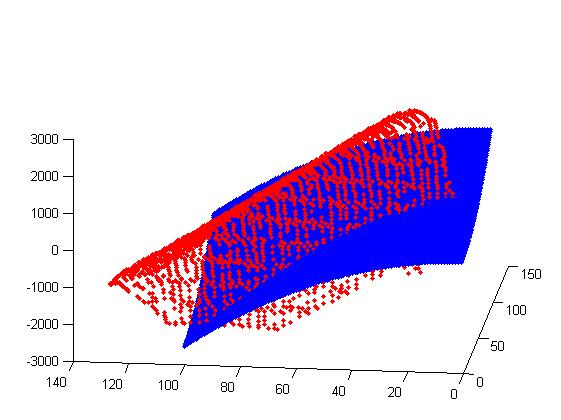
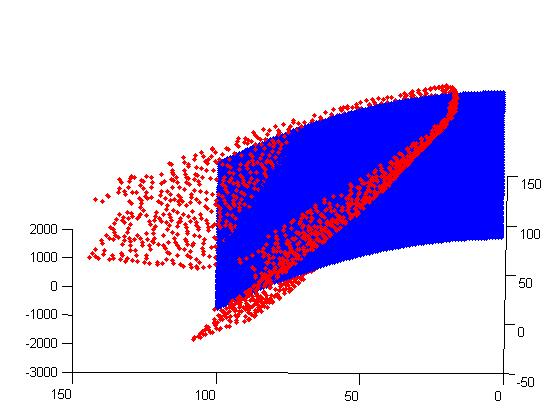
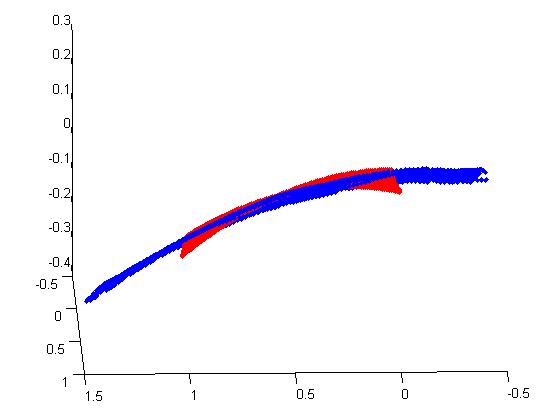
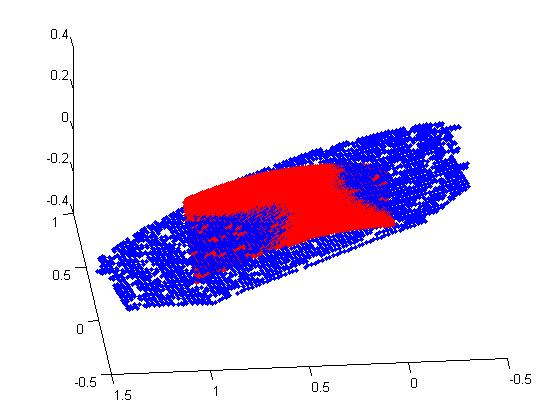


图 点对点算法拟合 图 点对面算法拟合

补充：因为上面的点云由于目标数据不太合理，所以人为的去重新定义了测试点云的范围，这也是上述点云拟合效果不佳的原因，针对最后一次提供的数据，没有人为去改变范围，直接按照点云去噪拟合的步骤得到了拟合图像：

（这里之用了点对点拟合的算法，（a）、（b）是同一组数据在两个方向上的截图）



1. (b)

从图中看到，虽然范围依然没有完全匹配，但拟合效果要好的多，可见初始点云数据的范围选定极其重要。