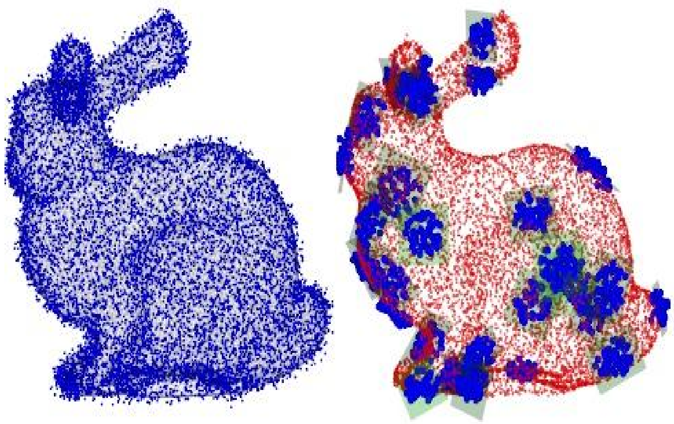
Final Report of CSCE 645

Problem Summary



Computing a curve or surface to approximate data points. This is a problem encountered frequently in many applications in computer graphics. Many three-dimensional geometrical shapes that can be fitted by the image and XYZ Fit Shape functions have common parameters.

Issues with shape fitting in 3D

The point cloud may have non-uniform distribution with considerable noise; this assumption makes it difficult or impossible to order data points along the target curve or surface.

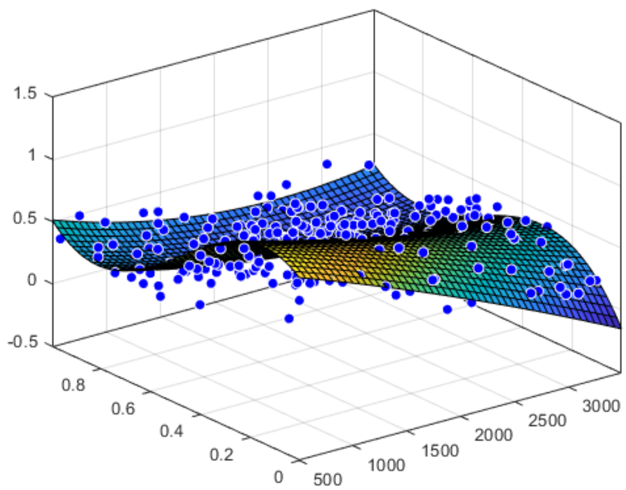
The curvature of the curve or surface might involve some really complicated calculations to the program.

Squared Distance Minimization(SDM)

This method is from one of Dr. Wang's papers which were published a few years ago. In SDM, the curve starts from some properly specified initial shape and converges towards the target shape through iterative quadratic minimization of the fitting error. The SD error term faithfully measures the geometric distance between a fitting curve and a target shape, thus leading to faster and more stable convergence than other methods that were commonly used in computer graphics.

$$e_{SD,k}(\mathcal{D}) = \begin{cases} \frac{d}{d-\rho}[(P_+(t_k) - X_k)^T T_k]^2 + [(P_+(t_k) - X_k)^T N_k]^2, & \text{if } d < 0, \\ [(P_+(t_k) - X_k)^T N_k]^2, & \text{if } 0 \le d < \rho, \end{cases}$$

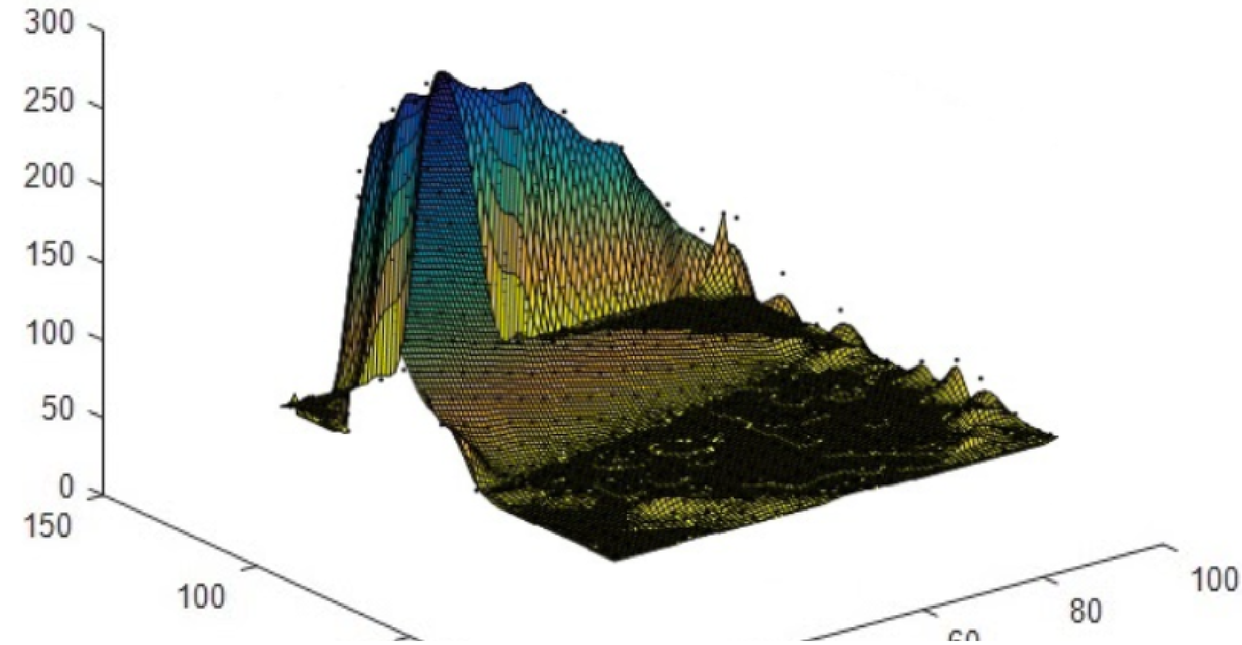
All the three methods we have discussed so far, PDM, TDM, and SDM, are local minimization schemes; that is to say, their convergence depends on the initial value, that is, the initial fitting curve. We would like to point out several possibilities of specification of the initial fitting curve though this is not a focus of the present article. The first obvious option is to let the user specify an initial B-spline fitting curve that is sufficiently close to the target shape and has an appropriate number of control points. For a target shape defined by a set of dense points, an alternative is to compute a quadtree partition of the data points and then extract a sequence of points approximating the target shape from nonempty cells, that is, those cells containing at least one data point. These extracted points can then be used as the control points of an initial B-spline fitting curve. Our experience shows that this method tends to produce too many control points at the beginning so control point deletion is normally required during the fitting process in order to obtain a fitting curve with a minimal number of control points while still meeting a prescribed error threshold.



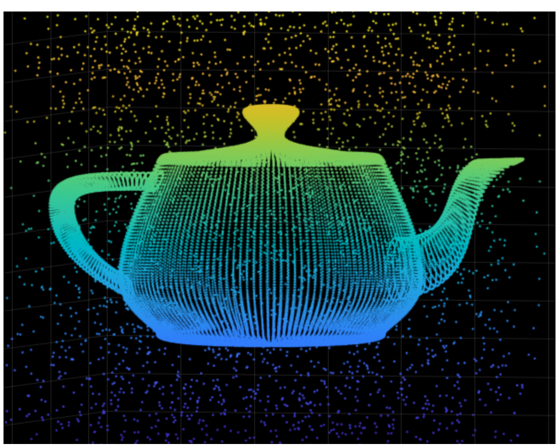
Point Cloud Library (PCL)

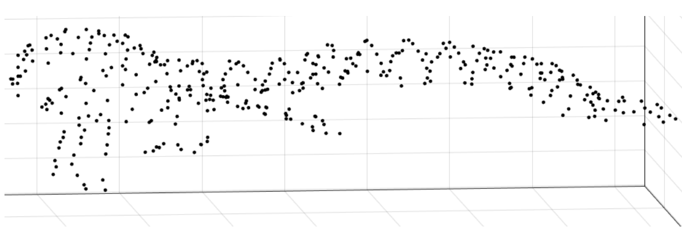
In order to fix the noise point problems, I used the PCL to reduce the noise points around the target surface and curve. PCL is an open source library that has a specific design for point cloud programs that can optimize the problem much faster, so the program will not take too many iterations to get the final result.
Github Link: https://github.com/PointCloudLibrary/pcl/wiki

Result:



This point cloud contains over 5000 points so that it is extremely difficult to do shape fitting without the PCL. By using the PCL, this point cloud only take ⅕ of time(2 mins) compare to the one (10 mins) without PCL. If the program does not even use the SDM method, then it can take over 20 mins for the result.


Teapot Shape Fitting demo (link for demo)


Bunny shape fitting demo

Conclusion:

In this project, I introduce a new method for point cloud shape fitting SDM and compare it to other commonly used methods in computer graphics, which proves that SDM is a better and faster method. My approach performs well on multiple types of shape fitting. However, this current version of the program is not 100% accurate for shape fitting, especially for the point cloud that has close noise points to target shape. So in the future, I will try to improve the accuracy of the SDM method on an open shape surface and also optimize the running time of the program.