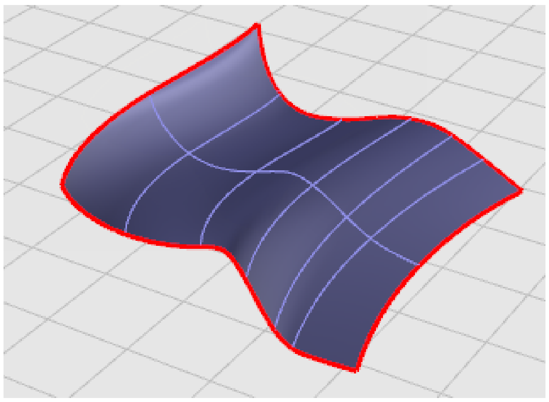
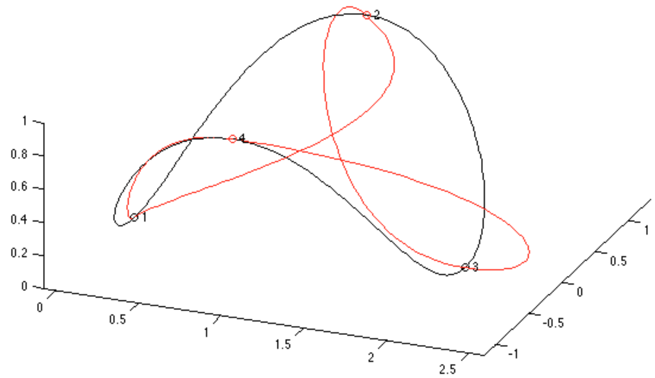
CSCE 645 Project

Shape fitting with curves and surfaces In 3D

Creator: Keigo Ma



(3D surface fitting)



(Curve fitting) [[image]]()

Summary:

A problem that often arises while fitting implicit polynomials to 2D and 3D data sets is the following: although the data set is simple, the fit exhibits undesired phenomena, such as loops, holes, extraneous components, etc. Previous work tackled these problems by optimizing heuristic cost functions, which penalize some of these topological problems in the fit[1]. Therefore, we are trying to make many three-dimensional geometrical shapes that can be fitted by the image and XYZ Fit Shape functions have common parameters.

Importance of Problem:

We often find out the high difficulty of imaging the 3D curve and surface when we have a relative topic. Therefore, we want to visualize the points that we have and turn them into a 3D model that is easier to understand.

Proposal:

I will be using the Point Cloud Library (PCL) to build a program that will accept a list of dataset, then the program will run a B-spline fitting algorithm on a point-cloud, to obtain a smooth, parametric surface representation. The algorithm consists of the following steps:

1. Initialization of the B-spline surface by using the Principal Component Analysis (PCA). This assumes that the point cloud has two main orientations, i.e. that it is roughly planar.
2. Refinement and fitting of the B-spline surface.
3. Circular initialization of the B-spline curve. Here we assume that the point cloud is compact, i.e. no separated clusters.
4. Fitting of the B-spline curve.
5. Triangulation of the trimmed B-spline surface.

Originality

By using Point Cloud Library, the program will allow users to create the shape fitting with more freedom and different user input.

List of Goals

Primary Goals:

>> Representation of 3D Bezier curve, B-spline curve, Bezier surface and B-spline surface in 3D space.

>> By computing appropriate first derivatives in the parametric space and specifying end conditions (by drawing arrows for showing the direction of derivative), to get a

piecewise smooth second order continuous curves passing through each of these points

>> By using cubic B-Splines and Beta-splines (with appropriate parameters as input) through these control points

>> By drawing the n-th order Bezier curve with (n+1) control points.

Second Goals:

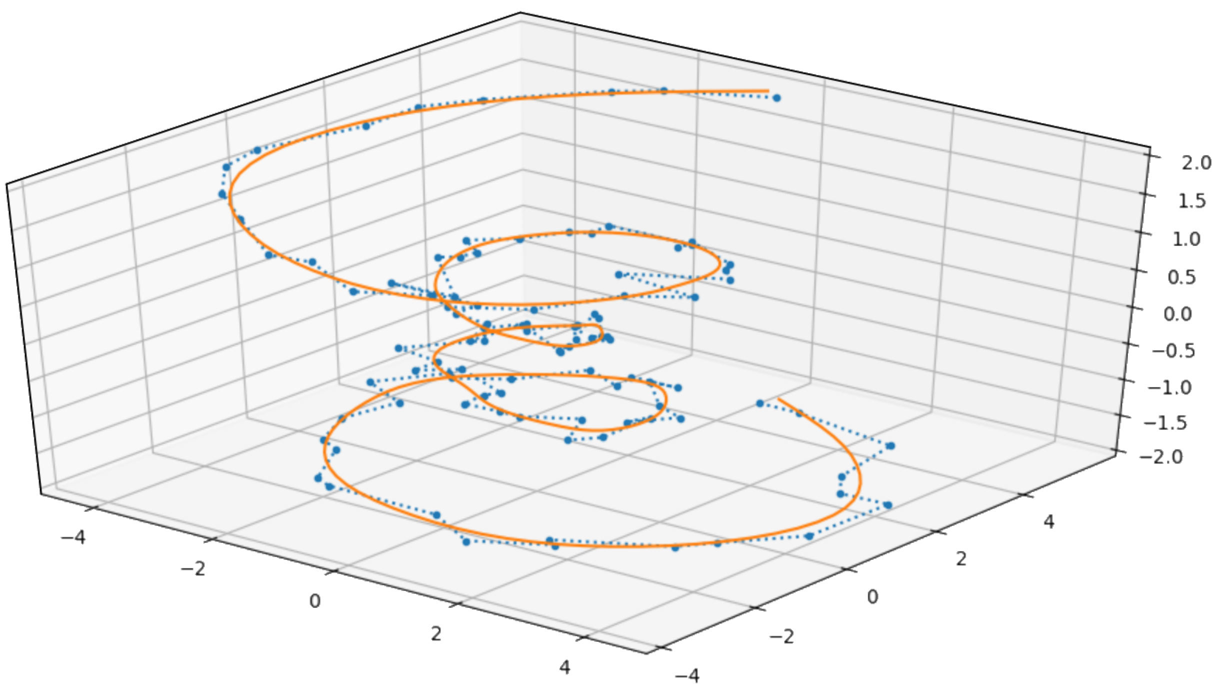
>> Representing shape fitting in different forms by using different reconstruction techniques.

>> The tool can simultaneously handle different sets or groups of control points.

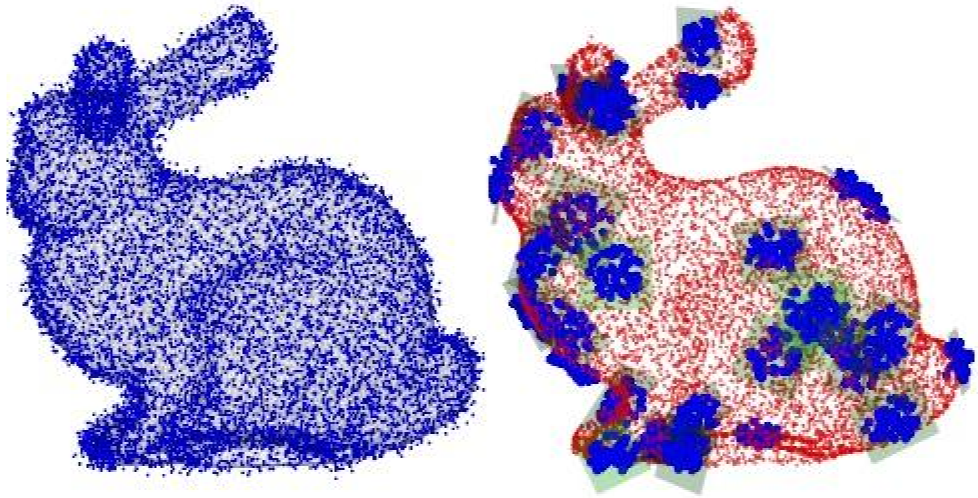
Literature Review

Curve Smoothing

In statistics and image processing, to smooth a data set is to create an approximating function that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures/rapid phenomena. In smoothing, the data points of a signal are modified so individual points higher than the adjacent points (presumably because of noise) are reduced, and points that are lower than the adjacent points are increased leading to a smoother signal. Smoothing may be used in two important ways that can aid in data analysis (1) by being able to extract more information from the data as long as the assumption of smoothing is reasonable and (2) by being able to provide analyses that are both flexible and robust. Many different algorithms are used in smoothing.[5]



Shape fitting with Point Cloud Library(PCL)

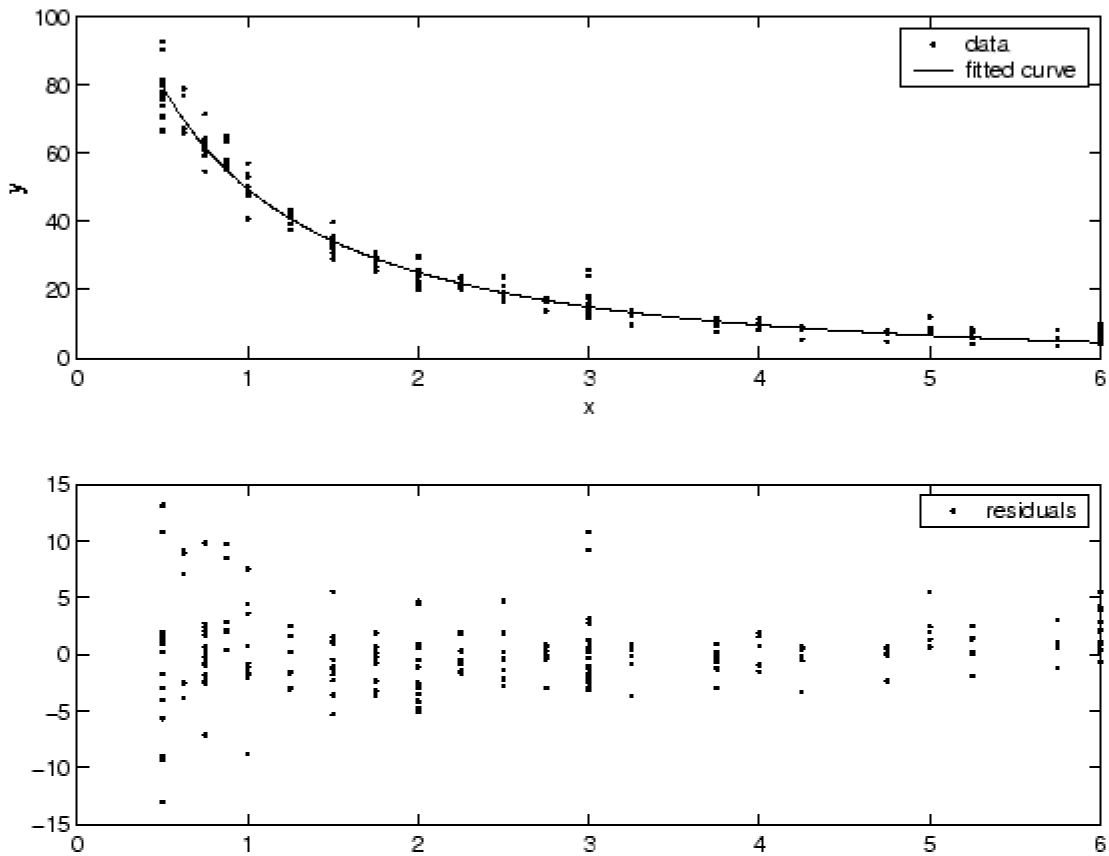


Fitting a curve to a set of data points is a fundamental problem in graphics (e.g., Pavlidis [1983]; Plass and Stone [1983]; Pratt [1985]; Walton and Xu [1991]; Goshtasby [2000]) and many other application areas. Fitting geometric primitives to 3D point cloud data bridges a gap between low-level digitized 3D data and high-level structural information on the underlying 3D shapes. Therefore, I am using the PCL library for this project. The Point Cloud Library (PCL) is an open-source library of algorithms for point cloud processing tasks and 3D geometry processing, such as occur in three-dimensional computer vision. The library contains algorithms for filtering, feature estimation, surface reconstruction, 3D registration, model fitting, object recognition, and segmentation. Each module is implemented as a smaller library that can be compiled separately (for example, libpcl_filters, libpcl_features, libpcl_surface, ...). PCL has its own data format for storing point clouds - PCD (Point Cloud Data) but also allows datasets to be loaded and saved in many other formats[6].

These algorithms have been used, for example, for perception in robotics to filter outliers from noisy data, stitch 3D point clouds together, segment relevant parts of a scene, extract key points, and compute descriptors to recognize objects in the world based on their geometric appearance, and create surfaces from point clouds and visualize them.

Squared Distance Minimization

Based on the research paper from Dr. Wang, We present a novel and efficient method, called squared distance minimization (SDM), for computing a planar B-spline curve, closed or open, to approximate a target shape defined by a point cloud, that is, a set of unorganized, possibly noisy data points. We show that SDM significantly outperforms other optimization methods used currently in the common practice of curve fitting. In SDM, a B-spline curve starts from some properly specified initial shape and converges towards the target shape through iterative quadratic minimization of the fitting error. Our contribution is the introduction of a new fitting error term, called the squared distance (SD) error term, defined by a curvature-based quadratic approximant of squared distances from data points to a fitting curve. 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 the point distance (PD) error term, which is commonly used in computer graphics and CAGD, and the tangent distance (TD) error term, which is often adopted in the computer vision community.



SDM can also be used to fit an open curve to a point cloud that represents an open target curve with some necessary modifications to ensure that the endpoints of the fitting curve are properly determined. We assume that the target curve is not self-intersecting and that a proper initial shape of an open fitting curve is provided.

References:

[1] S. Flöry, “Fitting curves and surfaces to point clouds in the presence of obstacles,” *Computer Aided Geometric Design*, 30-Apr-2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167839608000289. [Accessed: 20-Oct-2021].

[2] Garcia, J. A., and J. Fdez-Valdivia. “Boundary simplification in cartography preserving the characteristics of the shape features.” Computers & Geosciences 20.3 (1994): 349-368.

[3]B. Luca, A. Mark, M. Nikos, E. A. M. Ghalamzan, and C. Marco, “Primitive shape fitting in point clouds using the Bees Algorithm: Article Information: J-Global,” *Applied Sciences (Web)*, 01-Jan-1970. [Online]. Available: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=202102284810052483. [Accessed: 21-Oct-2021].

[4] Jiang, Peng, et al. "Rellis-3d dataset: Data, benchmarks and analysis." *arXiv preprint arXiv:2011.12954* (2020).

[5] J. S. Simonoff, “Smoothing methods in statistics | springerlink,” *SpringerLink*, 1996. [Online]. Available: https://link.springer.com/book/10.1007%2F978-1-4612-4026-6. [Accessed: 20-Oct-2021].

[6] M. M., “Pointcloud,” *pointcloud - vedo documentation*, 2019. [Online]. Available: https://vedo.embl.es/autodocs/content/vedo/pointcloud.html. [Accessed: 10-Oct-2021].

[7] W. Wang, H. Pottmann, and Y. Liu, “Fitting B-spline curves to point clouds by curvature-based squared distance minimization,” *ACM Transactions on Graphics (TOG)*, 01-Apr-2006. [Online]. Available: https://dl.acm.org/doi/10.1145/1138450.1138453. [Accessed: 23-Oct-2021].

[8] GOSHTASBY, A. A. 2000. Grouping and parameterizing irregularly spaced points for curve fitting. ACM Trans. Graph. 19

[9] Tek, Hüseyin, and Benjamin B. Kimia. “Boundary smoothing via symmetry transforms.” *Journal of Mathematical Imaging and Vision* 14.3 (2001): 211-223.

[10] Dyer, Charles R., and Azriel Rosenfeld. “Thinning algorithms for gray-scale pictures.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1 (1979): 88-89.

[11] Leymarie, Frédéric, and Martin D. Levine. “Simulating the grassfire transform using an active contour model.” *IEEE Transactions on Pattern Analysis & Machine Intelligence* 14.01 (1992): 56-75.

[12] Jerripothula, Koteswar Rao, et al. “Object co-skeletonization with co-segmentation.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[13] Kolesnikov, Alexander, and Pasi Fränti. “Polygonal approximation of closed discrete curves.” Pattern Recognition 40.4 (2007): 1282-1293.

[14] MAEKAWA, I. AND KO, K. 2002. Surface construction by fitting unorganized curves. Graphical Models 64, 316–332.