Point Cloud Segmentation via Constrained Nonlinear Least Squares Surface Normal Estimates

Edward Castillo
Radiation Oncology Department
University of Texas MD Anderson Cancer Center, Houston TX
ecastillo3@mdanderson.org

Hongkai Zhao
Department of Mathematics
University of California, Irvine CA
http://www.math.uci.edu./~zhao

Abstract

We present a point cloud segmentation scheme based on estimated surface normals and local point connectivity, that operates on unstructured point cloud data. We can segment a point cloud into disconnected components as well as piecewise smooth components as needed. Given that the performance of the segmentation routine depends on the quality of the surface normal approximation, we also propose an improved surface normal approximation method based on recasting the popular principal component analysis formulation as a constrained least squares problem. The new approach is robust to singularities in the data, such as corners and edges, and also incorporates data denoising in a manner similar to planar moving least squares.

1. Introduction

Two key processing tools for unstructured point cloud data are surface normal estimation and segmentation. Surface normal approximation is utilized by several applications such as reconstruction, estimating local feature size, computer aided design (CAD), and inside-outside queries, while segmentation is useful for efficient point cloud representation.

We present a point cloud segmentation scheme that utilizes estimated surface normals and operates on raw point cloud data. The procedure is based on constructing an adjacency matrix representing the point connectivity of the full data set from unoriented surface normal estimates. The connected components present in the adjacency matrix constitute the initial segmentation, while the point connectivity information provides a means for orientating the surface normal estimates consistently, even for data sets containing one or multiple closed surfaces. A finer segmentation can then be derived from the initial connected component segmentation by classifying the oriented estimated normals according to their direction or their singularity and the local geometry information contained in the adjacency matrix.

Naturally, the performance of the segmentation routine depends on the quality of the surface normal approximation. As such, we also propose an improved surface normal approximation method based on recasting the popular principal component analysis formulation as a constrained least squares problem. The new approach is robust to singularities in the data, such as sudden change in normals or edges, and also incorporates data denoising in a manner similar to planar moving least squares.

2. Least Squares Normal Estimation and PCA

Principal component analysis (PCA) is a popular method for computing surface normal approximations from point cloud data [3]. Given a point cloud data set $D = \{x_i\}_{i=1}^{n}$, the PCA surface normal approximation for a given data point $p \in D$ is typically computed by first determining the *K-nearest neighbors*, $x_k \in D$, of p. Given the *K-neighbors*, the approximate surface normal is then the eigenvector associated with the smallest eigenvalue of the symmetric positive semi-definite matrix

$$P = \sum_{j=1}^{K} (x_k - \bar{p})^T (x_k - \bar{p}), \tag{1}$$

where $\bar{p}$ is the local data centroid, $\bar{p} = (\frac{1}{K}) \sum_{j=1}^{K} x_j$.

The PCA normal approximation, also referred to as total least squares [5], is accurate when the underlying surface is smooth, but tends to smear across singularities, such as corners or intersecting planes. Figure 1 illustrates this phenomenon for a cube. The smearing is caused by the con-

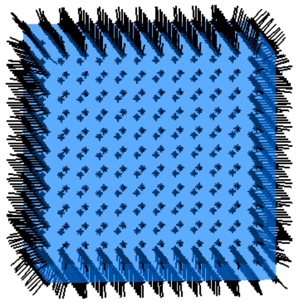

Figure 1. PCA normal estimation for a cube. Notice the smeared normals along the edges of the cube.

tribution of data points from across the singularity to the PCA covariance matrix P. Simply reducing the weight of such contributions will therefore improve the PCA estimate. However, equation (1) alone does not provide much insight into determining which data point contributions should be dampened, and which should be considered valid.

Fortunately, the PCA normal approximation can also be described as the solution to the following constrained least squares problem:

$$\begin{aligned} \min_{\eta} \quad & \tfrac{1}{2}\|V\eta\|_2^2 \\ \text{s.t.} \quad & \|\eta\| = 1, \end{aligned} \tag{2}$$

where the rows of the matrix $V \in \mathbb{R}^{K\times 3}$ are the difference vectors $v_k = x_k - \bar{p}$. The equality constraint is necessary to ensure a nontrivial solution.

After substituting $\eta^T\eta = 1$ for the original equality constraint, the first order KKT conditions for problem 2 describe a standard eigenvalue problem:

$$V^T V\eta - \mu\eta = 0, \tag{3}$$

Hence, the stationary points for problem 2 are the eigenpairs of $P = V^T V$, where the matrix P is exactly the PCA covariance matrix in 1. The global minimum is achieved at the smallest eigenvalue, and thus the solution to problem 2 is equivalent to the PCA approximation, contrary to the assertion in [2]. From the perspective of moving least squares, the PCA approximation is the normal associated with the plane, passing through the centroid, that best fits the local data in the least squares sense []. The advantage of formulation 2 over the standard PCA formulation is that the optimization problem gives insight into how to improve normal estimation near singularities, and also in the presence of noise.

The objective function describes variance from orthogonality. In essence, problem 2, and consequently the PCA method, determine the normal approximation as the vector which is most orthogonal to each difference vector v_k, by minimizing the sum of squares of the inner products. Thus, in the simplest case where all v_k live in the same plane, the value of the objective function $V^T\eta$ at the solution η^* is zero, and η^* is orthogonal to the plane. From this perspective, it is not necessary to compute the local centroid, since utilizing the difference vectors $a_k = x_k - p$, as opposed to the vectors $v_k = x_k - \bar{p}$, yields the same result. Furthermore, the magnitude of the inner product between η^* and each difference vector quantifies the "planar deviation" of each local neighbor. Monitoring this quantity thus provides a means for determining which data points, if any, are contributing from across a singularity.

3. Normal Estimation via Nonlinear Least Squares

The solution to problem (2) may be suspect near surface edges, due to erroneous contributions from data points across the singularity. Erroneous data contributions can be detected by examining the magnitude of the orthogonality mismatch, $|a_k^T\eta|$, where $a_k = x_k - p$. Assuming noise free data, a large orthogonality mismatches corresponds to neighbors that deviate from the local least squares fitted plane.

In order to reduce the contribution of points from across the singularity, we propose modifying formulation (2) by incorporating automatic weights, adaptive to orthogonality mismatch, to deflate such contributions. The result is the following constrained nonlinear least squares formulation:

$$\begin{aligned} \min_{\eta} \quad & \tfrac{1}{2}\textstyle\sum_{k=1}^{K}\left[e^{-\lambda(a_k^T\eta)^2}(a_k^T\eta)\right]^2 \\ \text{s.t.} \quad & \|\eta\| = 1. \end{aligned} \tag{4}$$

Traditional weighting terms place emphasis on proximity, and also serve as a window function which isolates the local neighbors. The weighting term $e^{-\lambda(a_k^T\eta)^2}$ adaptively deflates the contribution of terms with high orthogonality mismatch at a rate defined by the parameter λ. Naturally, setting λ to zero results in the original PCA linear least squares problem 2, with the exception that the difference vectors are taken from the data point p and not the centroid $\bar{p}$.

At first glance, formulation 4 appears to be significantly more difficult to solve than the small eigenvalue problem associated with the PCA approach. However, the equality constraint is easily absorbed into the objective function by representing the unknown surface normal η in spherical coordinates with magnitude set to unity:

$$\eta(\phi,\theta) = \begin{bmatrix} \cos(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta) \\ \sin(\phi) \end{bmatrix}.$$

In addition, the spherical coordinate representation reduces the number of unknowns in the formulation to two. Thus, the simplified, unconstrained formulation is now:

$$\min_{\phi,\theta} \frac{1}{2} \sum_{k=1}^{K} \left[e^{-\lambda(a_k^T \eta(\phi,\theta))^2} (a_k^T \eta(\phi,\theta)) \right]^2. \quad (5)$$

Applying formulation 4 to the box data results in significant improvement in the surface normal estimates, as illustrated by figure 2.

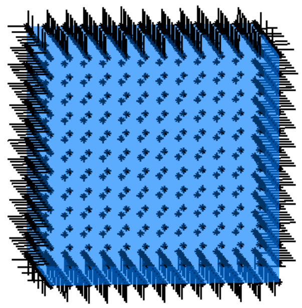

Figure 2. Nonlinear least squares normal estimation for a cube. Notice the improvement in the normal estimation along the edges of the cube compared to standard PCA (figure 1).

4. Incorporating Noise Robustness into Normal Estimation

Noise in point cloud data is not uncommon and is a significant source of error for surface normal approximation. In addition, point cloud denoising is helpful to other applications, such as surface reconstruction. For a given noise polluted point $\tilde{p}$, the task of denoising can be viewed as finding the projection of the point $\tilde{p}$ onto the unknown surface, a process often referred to as point cloud thinning. Developing methods for projecting points onto curves and surfaces is an active area of research with applications in computer graphics and vision (see [4]). Work has also been done for the problem of projecting points onto point cloud represented surfaces [1, 6], based on first determining a "projection direction." The goal here is to incorporate point cloud denoising into the surface normal computation, an approach which can be thought off as point projection utilizing the estimated surface normal as the projection direction.

Given an estimate η for the surface normal at $\tilde{p}$, we wish to determine the position along η, emanating from $\tilde{p}$, that minimizes the orthogonality mismatch energy (formulation 5 with $\lambda = 0$):

$$\min_{t} \sum_{k=1}^{K} (x_k - (\tilde{p} + t\eta))^T \eta.$$

Rearranging terms and taking advantage of the fact that $\|\eta\| = 1$, the problem reduces to a simple least squares problem for the scalar value t:

$$\min_{t} \|A\eta - t\|^2, \quad (6)$$

where the rows of A are the difference vectors $a_k = x_k - \tilde{p}$, and the denoised location is given by the solution t^* as $p^* = \tilde{p} + (t^*)\eta$.

As is the case for most denoising procedures, naively applying 6 to clean data will result in over smoothing and the loss of sharp surface features. However, given a bound T for the distance between $\tilde{p}$ and the underlying surface, data fidelity can easily be incorporated into the formulation by way of linear inequality constraints. Specifically,

$$\begin{aligned} \min_{t} \quad & \|A\eta - t\|^2, \\ \text{s.t.} \quad & |t| < T. \end{aligned} \quad (7)$$

Single variable calculus dictates that the solution to problem 7 is achieved at either the unconstrained minimum $t_{unc} = (\frac{1}{K}) \sum_{i=1}^{K} a_i^T \eta$, or at one of the bounds $-T, T$.

For the typical case where η is not known, the noise in the data can lead to inaccurate PCA or NLSQ surface normal approximations. Noise robustness can be achieved within our least squares framework by coupling the nonlinear least squares formulation 5 with the denoising formulation 7:

$$\begin{aligned} \min_{\phi,\theta,t} \quad & \tfrac{1}{2} \textstyle\sum_{k=1}^{K} \left[e^{-\lambda(a_k^T \eta(\phi,\theta) - t)^2} (a_k^T \eta(\phi,\theta) - t) \right]^2 \\ \text{s.t.} \quad & |t| <= T. \end{aligned} \quad (8)$$

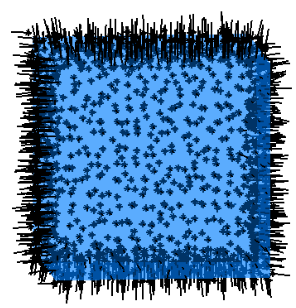

Figure 3. Nonlinear least squares normal estimation with noise robustness for a noisy cube. three percent random noise was added to the original cube. The estimated normals are still accurate.

Given the simplicity of the objective function and linear constraints, problem 8 can be solved via any standard nonlinear least squares solver package, such as MINPACK or MATLAB's built in lsqnonlin routine. Our current implementation includes the two linear constraints into a global formulation via log barrier functions:

$$L_1(t) = \ln(\frac{h-t}{h})$$

$$L_2(t) = \ln(\frac{h+t}{h})$$

Thus, the nonlinear least squares problem contains $K+2$ equations and three unknowns. Problem 8 coupled with the log-barrier functions, comprise a standard, unconstrained, nonlinear least squares problem for which several high performance solver packages exist. Our current implementation utilizes the MINPACK software package, which is based on the Levenberg-Marquardt Quasi-Newton trust region algorithm, using exact values for the gradient of the objective function. The initial guess used for the optimization routine is computed by solving problems 5 and 7 for λ, $h = 0$, respectively.

5. Segmentation Based on Point Connectivity

The overall segmentation strategy is two-fold. First, an initial segmentation is obtained from an adjacency matrix, constructed from unoriented surface normal estimates, describing the point connectivity for the entire data set. The final, finer segmentation is determined from the initial segmentation, the point connectivity information contained in the adjacency matrix, and the oriented surface normals estimates.

Our strategy for the initial segmentation is built upon the assumption that one can obtain a reasonable approximation to the underlying unoriented surface normals (presumably by applying the methodology described in the previous section, but this is not necessary) at each point in the cloud. Based on the approximated normals and the k-nearest neighbor information, a symmetric adjacency matrix A is built such that $A_{i,j} = 1$ if the data points i, j are "connected", or in terms of graph theory, there exists an edge between the two points. $A_{i,j} = 0$ otherwise. The initial segmentation then is given by the connected components of the undirected graph represented by A.

Of course, the nature of this "connected component segmentation" is dictated by the criteria governing the placement of edges between points. Specifically, given points x_i and x_j, the corresponding adjacency matrix entry $A_{i,j} = 1$ if both points are a k-nearest neighbor of each other, and if the line $x_j - x_i$ lies within a user defined angle θ (see figure 4) of the tangent plane associated with unoriented surface normal approximation for x_i, and vice versa (see figure 5). In this way, point proximity, as well as local geometry are taken into account when determining point connectivity. Or put another way, neighboring points are connected if the local geometry estimate, given by the associated surface normal approximations, are compatible with location of x_i and x_j in $\mathbb{R}^3$.

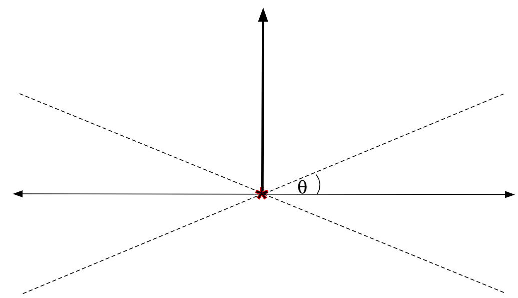

Figure 4. Two dimensional representation of a data point and its tangent plane.

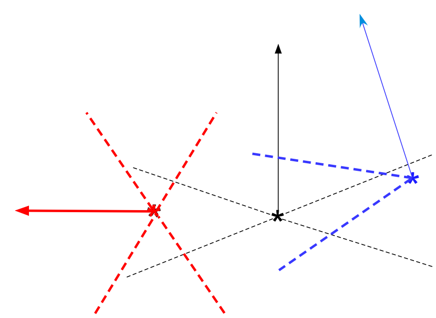

Figure 5. The black point and blue point are connected to one another. The red point is not connected to either point.

Based on this criteria, segmenting the point cloud according to the connected components of A can be considered an edge detection based scheme. However, a serendipitous characteristic of this methodology is its innate ability to account for the possibility of multiple closed surfaces, or *bodies*, within the data set. For a smoothly varying surface, such as a sphere, the corresponding adjacency matrix is comprised of one connected component. However, for the cylinder and city data depicted in figures 6 and 7, this is not the case. Consequently, many connected components (point cloud segments) are present, and the resulting segmentation breaks each data set into piecewise smooth segments.

A finer segmentation can be obtained from the connected component segmentation by first traversing the connectivity information contained in A to orient the approximated surface normals in the same manner as the minimum spanning tree method described in [3]. However, the benefit of employing the graph associated with A for the orientation is the ability to orient data sets containing more than one body. The orientation procedure is carried out on one individual connected component segment at a time, and is kept consis-

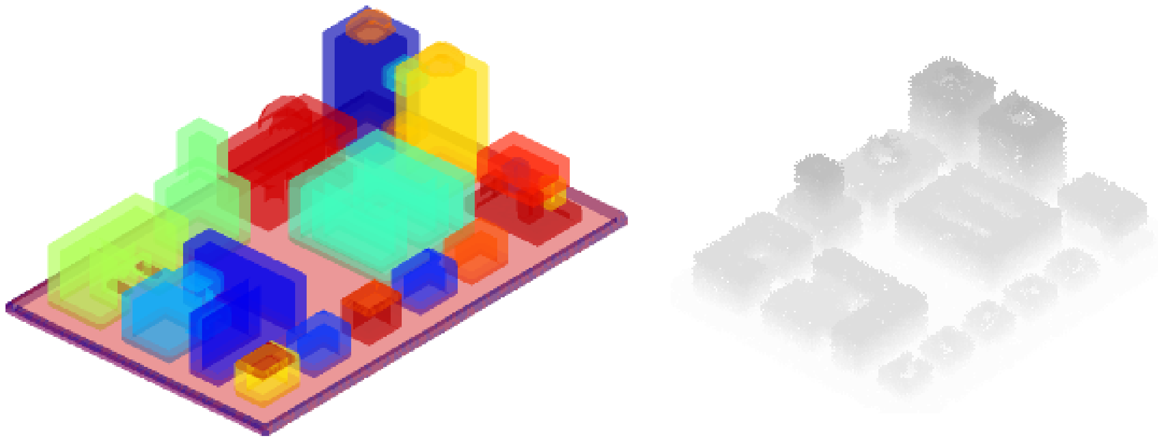

Figure 7. (Right) Point cloud representation of a city.(Left) Connected component segmentation.

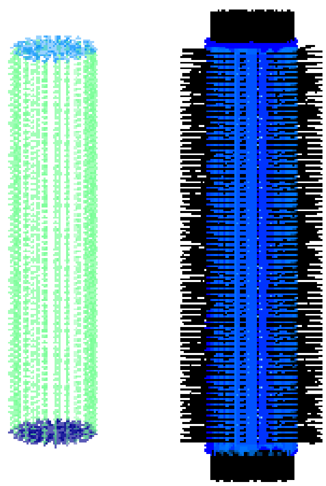

Figure 6. (Right) Point cloud representation of a cylinder, with calculated normals. (Left) Connected component segmentation.

tent within a body by identifying points with nearest neighbors belonging to two connected component segments, and then propagating the information from one connected component segment to another.

Given the oriented surface normal approximations, the data set is further segmented by classifying points with similar surface normals. Specifically, given a set of N classification vectors, each data point is categorized into one of N groups according to the similarity between the surface normal and the classification vectors. Consider the example of a sphere, with the classification vectors chosen to be the 6 outward normals for each face of a cube. Since the sphere is a smooth surface, there is only one connected component present in A (see 8). Assuming properly orientated surface normals, each point is categorized according to their closeness (in terms of angular difference) to one of the 6 classification vectors and then grouped together based on the connectivity described by A.

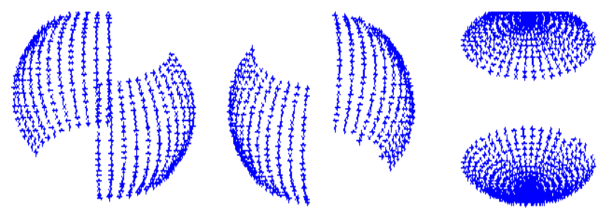

Figure 8. Segmentation results for a sphere using the six outward normals for the faces of a cube as classification vectors.

6. Results

The full algorithm is as follows:

Algorithm

1. Find the k-nearest neighbors of each data point.
2. Calculate the surface normal approximations.
3. Construct A by determining the edges between points.
4. Determine the initial segmentation by finding the connected components of A.
5. Orient the surface normal approximations.
6. Refine the segmentation according to N classification vectors.

The main computational workload is represented by the calculation of the k-nearest neighbors. For the city examples shown in figure 7 containing 1.99 million points, the k-nearest neighbor calculation (tree based sorting method with $O(n\ log\ n)$ complexity) required 66 seconds on a Dell Precision M6400 laptop, while the normal calculations utilizing our nonlinear least squares required 39 seconds. The PCA normals required 24 seconds.

7. Conclusions and Future Work

This paper presents a strategy for producing surface normals estimates for unstructured point cloud data based on a constrained nonlinear least squares formulation. The formulation is designed to be robust with respect to singularities in the data, such as surface edges, and incorporates data denoising in a fashion similar to planar moving least squares. Utilizing the surface normals, we also present a simple method for segmenting the point cloud data based on point connectivity.

Refinement of the initial connected component segmentation, according to classification vectors and the oriented surface normals, allows for each new sub-segment to easily be approximated by a polynomial, the immediate benefit being a potentially simple procedure for point cloud compression. Rather than storing each data point, one need only store the coefficients of the polynomial, and perhaps a list of the points which lie more than some tolerance away from the approximating polynomial.Furthermore, given the embarrassingly parallel nature of the procedure, such an algorithm could be implemented on a GPU (graphic processing unit) for high performance on large data sets.

8. Acknowledgments

This material is based upon work supported by, or in part by, the U.S. Army Research Office under contract/grant number W911NF-07-1-0185. Edward Castillo was a part of the UC Irvine Department of Mathematics when this work was accomplished.

References

[1] P. Azariadis. Parameterization of clouds of unorganized points using dynamic base surfaces. *Computer-Aided Design*, 36:607–623, 2004. 3

[2] M. Gopi, S. Krishnan, and C. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Proceedings of Eurographics*, volume 19, 2000. 2

[3] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of ACM Siggraph*, pages 71–78, 1992. 1, 4

[4] S. min Hu and J. Wallner. A second order algorithm for orhtogonal projection onto curves and surfaces. *Computer Aided Graphic Design*, 22(3):251–260, 2005. 3

[5] N. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. In *Special issue of International Journal of Computational Geometry and Applications*, volume 14, pages 261–276, 2004. 1

[6] Y. Shen Liu, J. Claude Paul, J. Hai Yong, P. Qiang Yu, H. Zhang, J. Guang Sun, and K. Ramani. Automatic least-squares projection of points onto point clouds with applications in reverse engineering. *Computer-Aided Design*, 38:1251–1263, 2006. 3