Estimating Surface Normals in Noisy Point Cloud Data

Niloy J. Mitra, An Nguyen, Leonidas Guibas

Abstract

We analyze a method based on local least square fitting for estimating the normals at all the sample points of a point cloud data (PCD) set, in the presence of random noise. We study the effects of neighborhood size, curvature, sampling density, and noise on the normal estimation when the PCD is sampled from a smooth curve in 2D or a smooth surface in 3D and noise is added.

The Normal Estimation Problem

Given

A noisy point cloud data (PCD) sampled from a surface (or curve) without any connectivity information

Goal

Compute surface normals at each point

Related work

Voronoi-diagram based approach

Do a surface reconstruction (Crust, Cocone)

Compute normals from the reconstructed surface

Works in absence of noise

Surface fitting

As found in works of Hoppe, Pauly, Alexa

Pointshop3D, Progressive Point Set

Use least square fit to a neighborhood of radius r around point p

Works in presence of noise

Neighborhood size picked manually

Algorithm for Normal Estimation in 3D

Choose c1, c2

Initialize k=15

Iterate and refine (10 steps or less)

Compute r, r+ locally

r = distance to k-th nearest neighbor

r+ = k+1

x computed using Gumhold’s method

Use r1, r to estimate rnew

Re-estimate knew = m2(rnew)

Stop if

k+ threshold or

k saturates

Use least square fit on the k-nearest neighbors to get the normal estimate to the surface at p

Contributions

Study the effects of curvature, noise, sampling density on the choice of neighborhood size

Use this insight to choose an optimal neighborhood size

Compute bound on the estimation error

Analysis

Assume, k points are uniformly distributed with density n within a ball of radius r

Noise independent of measurement:

zero mean, known variance \( \sigma_n \)

From the evenly distributed sample assumption

Bound all entries of the covariance matrix M

Important fact: Summation tends to expectation as the number of samples increases

Get a bound on the estimation error

Result in 2D

\[ \text{error} \leq c_1 \sqrt{\frac{\sigma_n}{r^2}} + c_2 \frac{\sigma_n^2}{r^4} \]

Result holds with probability 1 - \( \varepsilon \)

This suggests, to minimize error when density is high, choose optimal r as \( c_1 \sqrt{\frac{\sigma_n}{r^2}} \)

low, choose optimal r as \( c_2 \frac{\sigma_n^2}{r^4} \)

For flat surface, \( c_1=0 \), can make error arbitrarily small

For no noise, \( n=0 \), pick as small a neighborhood as possible

Result in 3D

A more involved but similar analysis in 3D gives the following bound

\[ \text{error} \leq c_1 \sqrt{\frac{\sigma_n}{r^2}} + c_2 \frac{\sigma_n^2}{r^4} + c_3 \epsilon r^2 \]

Se a good choice of r is given by

\[ r = \left( \frac{1}{c_1} \frac{\sigma_n}{\sqrt{\sigma_n}} + c_2 \frac{\sigma_n^2}{r^4} + c_3 \epsilon \right)^{1/3} \]