Stochastic Poisson Surface Reconstruction with One Solve using Geometric Gaussian Processes

Cornell University | Massachusetts Institute of Technology

Sidhanth Holalkere,[1] David S. Bindel,[1] Silvia Sellán,[2,3] Alexander Terenin[1]

[1]Cornell University [2]MIT [3]Columbia University

Columbia University in the City of New York

Abstract

Poisson Surface Reconstruction is a widely-used algorithm for reconstructing a surface from an oriented point cloud. To facilitate applications where only partial surface information is available, or scanning is performed sequentially, a recent line of work proposes to incorporate uncertainty into the reconstructed surface via Gaussian process models. The resulting algorithms first perform Gaussian process interpolation, then solve a set of volumetric partial differential equations globally in space, resulting in a computationally expensive two-stage procedure. In this work, we apply recently-developed techniques from geometric Gaussian processes to combine interpolation and surface reconstruction into a single stage, requiring only one linear solve per sample. The resulting reconstructed surface samples can be queried locally in space, without the use of problem-dependent volumetric meshes or grids. These capabilities enable one to (a) perform probabilistic collision detection locally around the region of interest, (b) perform ray casting without evaluating points not on the ray's trajectory, and (c) perform next-view planning on a per-ray basis. They also do not requiring one to approximate kernel matrix inverses with diagonal matrices as part of intermediate computations, unlike prior methods. Results show that our approach provides a cleaner, more-principled, and more-flexible stochastic surface reconstruction pipeline.

Poisson Surface Reconstruction

Input: point cloud with surface normals

Steps:
1. Interpolate normals onto 3D finite element mesh to get v
2. Solve $\Delta v = \nabla \cdot f$ for f

Result: finite element representation of inside-outside function f

Stochastic Poisson Surface Reconstruction

Previous work: replaces mesh interp. with Gaussian process

+ Quantifies uncertainty − Expensive global PDE solve
− Not output-sensitive − Inaccurate covariance approximations

Our work: computes cross-domain posterior directly

+ Faster output-sensitive runtime + Accurate covariance
+ Analytic description of process + High-resolution sampling

Computing the Inter-domain Posterior

Posterior over Poisson equation's solution:

$$\underbrace{(f \mid \boldsymbol{v})(\cdot)}_{\substack{\text{posterior}\\ \text{over solution}}} = \underbrace{f(\cdot)}_{\substack{\text{prior over}\\ \text{solution}}} + \underbrace{\mathbf{K}_{f(\cdot)\boldsymbol{v}}}_{\substack{\text{cross-covariance}\\ \text{from Poisson eqn.}}} \underbrace{(\mathbf{K}_{\boldsymbol{vv}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{v} - v(\boldsymbol{x}) - \boldsymbol{\varepsilon})}_{\text{standard GP solve}}$$

Unknown quantities for posterior mean and covariance:
1. Cross-covariance $k_{f,v}(\cdot,\cdot') = \mathrm{Cov}(f(\cdot), v(\cdot'))$
2. Scalar field covariance $k_f(\cdot,\cdot') = \mathrm{Cov}(f(\cdot), f(\cdot'))$

Posterior samples: also need to jointly sample of f and v

Geometric Gaussian Processes

Key idea:
1. Assume periodic boundary conditions: lift GP to *torus* $\mathbb{T}^3$
2. Derive Fourier expansions of f and v to obtain covariance

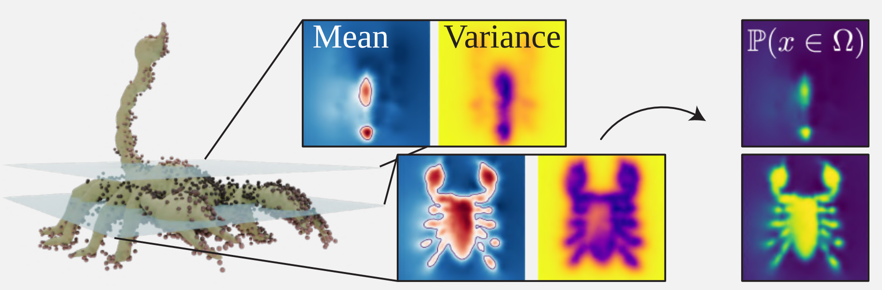
Proposition. *We have*

$$k_{f,v_i}(x,x') = \sum_{\substack{n\in\mathbb{Z}^d\\ n\neq 0}} \frac{n_i\sqrt{\rho_{v_i}(n)}}{\|n\|^2}\sin(n\cdot(x-x'))$$

$$k_f(x,x') = \sum_{\substack{n\in\mathbb{Z}^d\\ n\neq 0}} \frac{\sum_{i=1}^d n_i^2\rho_{v_i}(n)}{\|n\|^4}(\sin(n\cdot x)\sin(n\cdot x')+\cos(n\cdot x)\cos(n\cdot x')).$$

See paper for Karhunen–Loève decomposition for sampling f and v

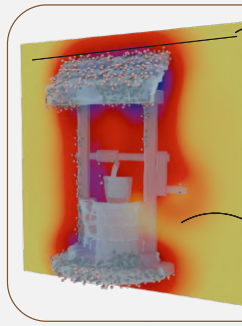
Statistical Queries



Input point cloud | **This work:** stochastic reconstruction | Statistical query

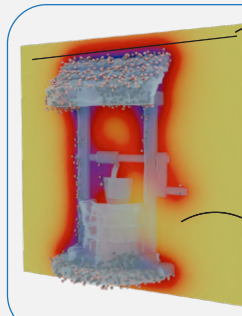
Runtime Improvement

SPSR



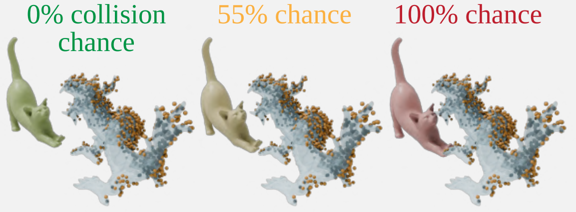
Querying GP on a single line: **10 min**
Querying GP on a plane: **10 min**
Querying GP on entire volume: **10 min**

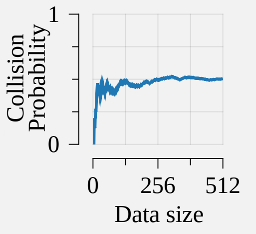
This work



Querying GP on a single line: **4 s** (in GPU: 0.8 s)
Querying GP on a plane: **33 s** (in GPU: 1.0 s)
Querying GP on entire volume: **7.5 min** (in GPU: 25 s)

Application: Collision Detection





Application: Next View Planning