Point Cloud to Mesh using Poisson Surface Reconstruction

Ken Kobayashi
UC Berkeley
Berkeley, CA
takeshi_kobayashi@berkeley.edu

Zhaoyan Lin
UC Berkeley
Berkeley, CA
zhaoyan_lin@berkeley.edu

Yirong Zhen
UC Berkeley
Berkeley, CA
yirongzhen@berkeley.edu

ABSTRACT

In this project, we try to implement the Poisson algorithm and do a survey comparing different existing algorithms for surface reconstruction.

KEYWORDS

Poisson equation, surface reconstruction, point cloud, mesh, computer graphics

ACM Reference Format:
Ken Kobayashi, Zhaoyan Lin, and Yirong Zhen. 2019. Point Cloud to Mesh using Poisson Surface Reconstruction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Poisson's equation is a partial differential equation with broad applications. Surface reconstruction is an inverse problem. The goal is to reconstruct a smooth surface based on a large number of points, a point cloud, where each point contains information about their location and an estimate of the local surface normal. Poisson's equation can be utilized to solve this problem with a technique called Poisson Surface Reconstruction first published in (Kazhdan et al., 2006). The goal of this technique is to reconstruct an implicit function f whose value is zero at the points and whose gradient at the points equals to the normal vectors. The set of points and the normal vectors are thus modeled as a continuous vector field. The implicit function f can found by integrating the vector field. Since not every vector field is the gradient of a function, the problem may or may not have a solution: the necessary and sufficient condition for a smooth vector field V to be the gradient of a function f is that the curl of V must be identically zero. In case this condition is difficult to impose, it is still possible to perform a least-squares fit to minimize the difference between V and the gradient of f. In order to effectively apply Poisson's equation to the problem of surface reconstruction, it is necessary to find a good discretization of the vector field V. The basic approach is to bound the data with a finite difference grid. For a function valued at the nodes of such a grid, its gradient can be represented as valued on staggered grids, i.e. on grids whose nodes lie in between the nodes of the original grid. It is convenient to define three staggered grids, each shifted in one and only one direction corresponding to the components of the normal data. On each staggered grid we perform [trilinear interpolation] on the set of points. The interpolation weights are then used to distribute the magnitude of the associated component of ni onto the nodes of the particular staggered grid cell containing pi. In (Kazhdan et al., 2006)[3], the authors give a more accurate method of discretization using an adaptive finite difference grid, i.e. the cells of the grid are smaller (the grid is more finely divided) where there are more data points. They suggest implementing this technique with an adaptive octree.

2 CURRENT PROGRESS FOR MILESTONE

We have read and understood the paper. We downloaded some point cloud files(.ply file) from the Stanford repository. The first step we had to take was being able to parse the ply files. We did this using the RPly library suggested by the previous years point cloud to mesh group. But then point cloud files need to be merged first since there are different scan files from different angles. Additionally, the files don't have normal vectors. So far, we are able to merge different scan files and compute the normal vectors, which is required for Poisson surface reconstruction. We are currently implementing the Octree for the algorithm.Please refer to the video for more details.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn