Intelligent Visual Computing [Spring 2023]

Assignment 1: Basic surface reconstruction from point clouds

Overview

In this assignment you will implement two basic implicit surface reconstruction algorithms to approximate a surface represented by scattered point data. The problem can be stated as follows:

Given a set of points **P** = {$\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}$} in a point cloud, we will define an implicit function f(x,y,z) that measures the signed distance to the surface approximated by these points. The surface is extracted at f(x,y,z) = 0 using the marching cubes algorithm. All you need to implement are two implicit functions that measure distance in the following ways: (a) signed distance to tangent plane of the surface point nearest to each point (x,y,z) of the grid storing the implicit function (b) Moving Least Squares (MLS) distance to tangent planes of the K nearest surface points to each point (x,y,z) of the grid storing the implicit function. The scikit-image package already provides implementations of marching cubes. Thus, all you need to do is to fill the code in the script 'basicReconstruction.py' to implement the above implicit functions. The implicit functions rely on surface normal information per input surface point. In the provided test data files, surface normals are included (the format of the point cloud file is: 'point_x_coordinate' 'point_y_coordinate' 'point_z_coordinate' 'point_normal_x' 'point_normal_y' 'point_normal_z' [newline]). There are three test point cloud (bunny-500.pts, bunny-1000.pts, and sphere.pts) that you will experiment with. Download the starter code below. You need to install the packages 'open3d', 'skimage', and 'sklearn' to run the code.

This assignment counts for **10 points** towards your final grade if you are taking the 574 section. **If you are taking the 674 section, divide your total points by 2** (i.e., the assignment counts for 5 points).

What You Need To Do

[30%] One way to estimate the signed distance of any point $\boldsymbol{p}=\{x,y,z\}$ of the 3D grid to the sampled surface points $\boldsymbol{p_i}$ is to compute the distance of $\boldsymbol{p}$ to the tangent plane of the surface point $\boldsymbol{p_i}$ that is nearest to $\boldsymbol{p}$. In this case, your signed distance functon is:

$f(\mathbf{p})=\boldsymbol{n_j}\cdot(\boldsymbol{p}-\boldsymbol{p_j})$ *with* $j=\mathrm{argmin}_i\{\ \|\boldsymbol{p}-\boldsymbol{p_i}\|\ \}$

<u>Your task:</u> Implement this distance function in the naiveReconstruction function of the script basicReconstruction.py. Show screenshots of the reconstructed bunny (500 and 1000 points) and sphere in your report.

[70%] The above scheme results in a C^0 surface (i.e., the derivatives of the implicit function are not continuous). To get a smoother result, the Moving Least Squares (MLS) distance from tangent planes is more preferred. The MLS distance is defined as the weighted sum of the signed distance functions to all points $\boldsymbol{p_i}$:

$f(\mathbf{p})=\boldsymbol{\Sigma_i} d_i(\mathbf{p})\varphi(\|\boldsymbol{p}-\boldsymbol{p_i}\|) / \boldsymbol{\Sigma_i}\varphi(\|\boldsymbol{p}-\boldsymbol{p_i}\|)$

where:

$d_i(\mathbf{p})=\boldsymbol{n_i}\cdot(\boldsymbol{p}-\boldsymbol{p_i})$

$\varphi(\|\boldsymbol{p}-\boldsymbol{p_i}\|) = \exp(-\|\boldsymbol{p}-\boldsymbol{p_i}\|^2/\beta^2)$

Practically, computing the signed distance function to all points $\boldsymbol{p_i}$ is computationally expensive. Since the weights $\varphi(\|\boldsymbol{p}-\boldsymbol{p_i}\|)$ become very small for surface sample points that are distant to points **p** of the grid, in your implementation you will compute the MLS distance to the K=50 nearest surface points for each grid point.

Your task: Implement this distance function in the mlsReconstruction function of the script basicReconstruction.py. You will also need to compute an estimate of $(1/\beta^2)$. Set β to be twice the average of the distances between each surface point and its closest neighboring surface point. Show screenshots of the reconstructed bunny (500 and 1000 points) and sphere in your report.

Submission

Please follow the **Submission** instructions in the course policy to submit your assignment on Gradescope. The zip file should contain all your Python code **plus a short PDF report with the screenshots (6 in total - put a title for each of them describing which method is used for each of them).**

python_starter.zip ✚ February 16 2023, 12:14 AM

Submission status

Submission status	This assignment does not require you to submit anything online
Grading status	Not graded
Due date	Thursday, March 2, 2023, 5:00 PM
Time remaining	The due date for this assignment has now passed
Last modified	-
Submission comments	▸ Comments (0)

© University of Massachusetts Amherst • Site Policies • Site Contact

Moodle Help for Students • Moodle Help for Instructors

You are logged in as Prachi Jain (Log out)
COMPSCI574_COMPSCI674_147255_SP23

Get Help
Help for Students
Help for Instructors