UDBHAN CHUGH
170101081

CS461 Assignment 3

POISSON SURFACE RECONSTRUCTION

Question: Can you define an implicit function given a set of points? - concept of normal - Poisson surface reconstruction

- Intuition - Mathematical Background - Working

Before explaining the intuition behind Poisson surface reconstructions, here are a few notations that will be helpful in understanding the intuition and mathematical background.

→ For a single dimension function, $f: R \to R$, (df/dx) denotes the derivative and (d^2f/dx^2) denotes the double derivative. Here d/dx is the differential operator.

Extending the definition to 3 dimensions, the following notations are important:

- Gradient: $\nabla f(x,y,z) = (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z)$. Gradient represents the direction of the steepest slope at point (x,y,z). It is a mapping of scalar field f to a vector field.

- Divergence: $\nabla \vec{V} = \partial V_x/\partial x + \partial V_y/\partial y + \partial V_z/V_z$. It represents the amount of flux passing and is basically a mapping of a vector field $\vec{V}$ to a scalar field.

- Laplace: $\Delta f = \nabla . \nabla f = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2 + \partial^2 f / \partial z^2$. It is the second order differential and represents average bending of a field. It maps a scalar field f to a scalar field.

INTUITION BEHIND POISSON SURFACE RECONSTRUCTION

Firstly, I will introduce what an implicit representation of a surface means. An implicit representation is a relation of the form $f(x_1, x_2, \ldots) = 0$ [or specifically $f(x, y, z) = 0$ for 3D) that can be used to define a curve or a surface.

Using an implicit relation, surface rendering becomes easy as the implicit equation can very well define boundaries of a surface.

The authors of this method noticed a relation between the normals at the surface boundary and the gradient of the indicator function. Indicator function is simply notion of a function that takes positive values inside the surface boundary and negative values outside. The zero set of this function is the boundary surface of the objct we need. (Implicit equation)

Thus, the main observation (or intuition) is that the direction of the normal should be in the same direction as the direction of the gradient of the implicit function.

Thus, mathematically speaking, if $\vec{V}$ represents the vector field of the normals, the aim of the approach is the finding of the function $f(x,y,z)$ such that mean square error between gradient of f and vector field V is minimised. The solution to such problem is of form $\Delta f = \nabla$. , i.e; Laplacian of f = Gradient of V) which is basically the Poisson equation.

To better understand the implementation and w
it is important to first have a basic underst
of a few mathematical concepts.

MATHEMATICAL BACKGROUND

To understand the thought process on how the
Poisson surface reconstruction works, I will
introduce a few concepts from the perspective of
and then extend it for three dimensions.

For a 1D function $f: R \to R$,

$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Here d/dx is simply the differential operator notation defined before. It maps one function in $\mathbb{R}$ to another function in $\mathbb{R}$

Now, consider $\frac{df}{dx} = g$

Given g is known and integrable, f can be calculated using,

$$f = \int (df/dx)\, dx = \int g\, dx$$

But here we will have discrete points as input and hence we need to look at finding f from the perspective when g is not analytically integrable.

Then we can look for approximate solutions, drawn from some parameterized family on candidate functions.

Formally, consider a family of function F, the mean squared approximation error over some interval Ω and functions $f \in F$ is given by:

$$\int_\Omega |(df/dx) - g|^2\, dx.$$

To minimize the mean squared approximation error, we can take help of the Euler-Lagrange equation which states that,

Boundary minima and maxima (stationary points) for a functional form,

$$\int_\Omega L(x, f(x), f'(x))\, dx$$

is given by the solution to the equation:

$$\frac{\partial L}{\partial f} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial f'}\right) = 0 \quad \text{— } (1)$$

In this case, $L = (f'(x) - g(x))^2$

Take $y = f(x)$ and $z = f'(x)$,

$$L(x, f(x), f'(x)) = L(x, y, z) = (z - g(x))^2$$

$$\frac{\partial L}{\partial f} = \frac{\partial L}{\partial y} = 0 \quad \text{— } (2)$$

$$\frac{\partial L}{\partial f'} = \frac{\partial L}{\partial z} = 2(z' - g'(x)) = 2(f''(x) - g'(x)) \quad \text{— } (3)$$

Hence for (1) to hold,
substituting (2) and (3) in (1),

$$0 - 2(f''(x) - g'(x)) = 0$$

$\Rightarrow$ $\boxed{f''(x) = g'(x)}$ must hold to minimize $\int_\Omega \left|\frac{df}{dx} - g\right|^2 dx$

Now,
$$f''(x) = g'(x) \Rightarrow \frac{d}{dx}\frac{d}{dx} f = \frac{d}{dx} g$$

Since the two sides are equal for all points x, we can sample any n consecutive points from Ω (for discrete analysis)
$x_1, \dots x_n$ and assume $x_{i+1} - x_i = h$

For any i, the derivative at x_i can be approximated as:
$$f'(x_i) = \frac{f(i+1) - f(i)}{h} = \frac{1}{h} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} f_i \\ f_{i+1} \end{bmatrix}$$

Using this, derivative for all n points can be listed as $A\bar{f} = \bar{g}$ — (4)
where
$$\bar{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \quad A = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \\ 0 & 0 & \cdots & & -1 & 1 \\ 0 & & & & 0 & -1 \end{bmatrix} \times \frac{1}{h}$$

Hence, now we have $\bar{f}$ and $\bar{g}$ which are discrete approximations for functions f and g.
A is a discrete approximation for continuous derivative operator d/dx.

Now, we have converted our problem to a discrete problem, we can represent the earlier minimisation as:
$$\|r\|^2 = \|g - Af\|^2$$ using least square approach. — (5)

Note, (4) can also be written as

$B\bar{f} = \bar{g}$ where $B = (-A^T)$ since $f'(x_i) \approx \frac{f_i - f_{i-1}}{h} = \frac{1}{h}[-1 \; 1]\begin{bmatrix} f_i \\ f_{i-1} \end{bmatrix}$

Hence,

for minimizing (5),

Directional derivative in direction δf is,

$$\nabla \|r\|^2 \cdot \delta f = 2\delta \bar{f}^T (A^T g - A^T A f)$$

To minimize the value, directional derivatives must be 0.

$$\Rightarrow A^T \bar{g} = A^T A \bar{f}$$
$$\Rightarrow A^T A \bar{f} = A^T \bar{g}$$
$$\Rightarrow (-A^T) A\bar{f} = (-A^T)\bar{g} \Rightarrow BA\bar{f} = B\bar{g}$$

Here $-A^T A$ is simply discretization of $d^2/dx^2 = \frac{d}{dx}\frac{d}{dx}$

From (4), we see A is an invertible matrix, so there will be unique solution $\bar{f} = A^{-1}\bar{g}$ in discrete domain.

Extending this approach to n Dimensions

From earlier analysis, we saw how to construct an implicit function from a set of points in 1 dimension.

We saw the Euler-Lagrange method for minimizing mean square error and also demonstrated the discretised version of the problem where we approximated differential operator as a matrix A for n set of discrete points. And then we saw since A^{-1} is invertible, it allowed us to have a unique implicit solution in 1 dimension.

Extending this to higher (3) dimensions, the notion of derivative is replaced by the notion of gradient explained in the beginning. This means $f'(x)$ is replaced with $\nabla f(x,y,z)$. Similarly, $g(x)$ is replaced by vector field $\vec{V}$.

As mentioned before, V represents the vector field of the normals. One approach following the extension of 1D would be to integrate vector field $\vec{V}$. But, here, since not every vector field $\vec{V}$ is the gradient of a function, implying $\vec{V}$ doesnt have a curl 0, and wont have a solution always.
Hence, like explained earlier, we try to minimize mean square error of V wrt ∇f, i.e., $\|\nabla f - V\|^2$ is minimised.
The solution to such problem is given by the Poisson function which we get after applying divergence operator

$$\nabla(\nabla f) = \nabla V$$
$$\Delta f = \nabla V \quad (\text{Laplacian of } f = \text{Divergenge of } V)$$

We now look at the working and implementation of the Poisson Surface reconstruction.

WORKING AND IMPLEMENTATION

I will now briefly describe the steps in the working of Poisson Surface Reconstruction. The mathematics part is discussed in detail before So I will try to highlight the steps with only needed mathematical notions.

- Problem Discretization (Using Octet tree)

The method uses an octree approach for discretization. Octree is a data structure with each internal node having exactly eight children. In the method, they define an octet - tree O wherein the original cube is successively divided into octants till a depth D. It is ensured that the points S lie in a node at depth D. Hence division of those nodes still containing the point extends till max depth D.

(Function space definition)

Then for the base function $F: R^3 \rightarrow R$ and for every node $o \in O$,

$$F_o(q) = F\left(\frac{q - o.c}{o.w}\right)\frac{1}{o.w^3}$$

o.c represents center
o.w represents width of node o.

q is smoothened out point of sample space

(Base function selection)
The n^{th} convulation of a box filter with itself is set as the base function F.

$$F(x,y,z) = (B(x)\, B(y) B(z))^{*n} \quad \text{with}$$

$$B(t) = \begin{cases} 1 & |t| < 0.5 \\ 0 & 0 \text{ otherwise} \end{cases}$$

With increasing n, F closely approximates a Gaussian.

- $\boxed{\text{Defining Vector Field}}$

The vector field is defined as

$$\vec{V}(q) = \sum_{s \in S} \sum_{o \in Ngbr_D(s)} \alpha_{o,s} F_o(q)\, s.\vec{N}$$

where, $Ngbr_D(s)$ are eight depth-D nodes closest to s.p and $\{\alpha_{o,s}\}$ are the trilinear interpolation weights, q is the smoothened point of sample (S), o is the node, s.p is the point and $s.\vec{N}$ is the inward facing normal $\forall s \in S$ where S is input data.

Here we see, a sample's position is distributed across 8 nearest nodes instead of clamping to the centre of the containing leaf node.
It is assumed that S (Sample space) is uniformly distributed and hence $\vec{V}$ is a good approximation of the gradient of the smoothened indicator function.

- ## Poisson Solution

Having defined vector field V, the method next solves for the function $f \in F_{O,F}$. f has same notation as explained in mathematical background. O is the octree node and F is the base function explained before.

We need to ensure gradient of f is closest to $\vec{V}$, i.e., a solution to the Poisson equation $\Delta f = \nabla \vec{V}$ as described in mathematical background section.
Thus solving for f amounts to finding
$\min_{x \in R^{|O|}} \|Lx - v\|$
where L is an $|O| \times |O|$ matrix such that Lx returns the dot product of Laplacian with each F_O and v is an $|O|$ dimension vector with O^{th} coordinate as $v_O = \langle \nabla \cdot \vec{V}, F_O \rangle$

Additionally, it can be noted that L is a sparse matrix since F_O are compactly supported. L is also symmetric since $\int f''g = -\int f'g'$. Also, there is an inherent multiresolution structure on $F_{O,F}$. Hence, a multigrid approach is used to solve the above equation for f.

- ## Isosurface extraction

In order to obtain a reconstructed surface ∂M,

, an isovalue is chosen such that the extracted surface closely approximates the positions of the input samples. It is chosen by evaluating f at the sample positions and use the average of the values for isosurface extraction.

$$\partial \tilde{M} \equiv \{ q \in \mathbb{R}^3 \mid f(q) = \gamma \} \text{ with } \frac{1}{|S|} \sum_{s \in S} f(s.p) = \gamma$$

where S is the input data, $s.p$ is position of each sample and q is smoothened point of sample S

To extract isosurface from the indicator function (defined before), a method of adaptations of Marching cubes to octree representations is used with the modification of defining the positions of zero-crossings along an edge in terms of the zero-crossings computed by the finest level nodes adjacent to the edge.

In order of extend this approach to non-uniform surfaces, instead of having a magnitude of a fixed-width kernel associated with each point, an additional kernel width is adapted.

PTO

APPLICATION OF POISSON SURFACE RECONSTRUCTION:

- We often have lasers to create point cloud of 3D objectsThis method can provide efficient rendering of these models.

LIMITATION OF POISSON SURFACE RECONSTRUCTION

- It cannot handle incremental point arrival for surface construction since it is an offline algorithm.
- It is computationally and, space (memory) intensive

REFERENCES USED TO UNDERSTAND THE METHOD

- Lectures and slides from the course
- Poisson Surface Reconstruction Paper by Michael Kazhdam, Matthew Bolitho and Hugues Hoppe
- IIT Bombay CS 749 2016 lecture slides
- Poisson's equation - Wikipedia
- Octree - Wikipedia