Technical Document: Poisson Surface Reconstruction in MATLAB

Maolin Tian
School of Mathematical Science
Shanghai Jiao Tong University
Email: tml10016@163.com

March 17, 2020

Abstract

The details of Poisson surface reconstruction for computing in MATLAB.

1 Poisson Reconstruction

The Poisson equation [1] is

$$\Delta\chi = \nabla \cdot \vec{V}. \tag{1}$$

Let the smoothing filter F be B-spline. That is,

$$F_p(q) = B(\frac{q-p}{w})\frac{1}{w^3}, \tag{2}$$

$$B(x,y,z) = (B(x)B(y)B(z))^{*n}, \tag{3}$$

and

$$B(t) = \begin{cases} 1, |t| < 0.5 \\ 0, \text{otherwise.} \end{cases} \tag{4}$$

The sampling density is estimated by

$$W(q) = \sum_{s \in S} F_{s.p}(q). \tag{5}$$

The vector field is

$$\vec{V}(q) = \sum_{s \in S} F_{s.p}(q)\ s.\vec{N}\ \frac{1}{W(s.p)}. \tag{6}$$

Assume the solution

$$\chi(p) = \sum_{i=1}^{N} x_i F_i(p). \tag{7}$$

By Galerkin method, the linear system from (1) is

$$-\langle \nabla \chi(p), \nabla F_i \rangle = \langle \nabla \cdot \vec{V}, F_i \rangle, \tag{8}$$

denoted by $-Ax = b$, where

$$A_{ij} = \langle \nabla F_i, \nabla F_j \rangle \tag{9}$$

and

$$b_i = \langle \vec{V}, \nabla F_i \rangle. \tag{10}$$

2 Computation in MATLAB

The equations (5), (9) and (10) in Section 1 are multivariate. Computationally, we need represent them in univariate form. Here we consider the case of 2D, where input is 2-D points and output is the curve. Let $b(x) = B(x)^{*n}$ be the univariate B-spline with degree n. Then $B(x, y)$ is separable since from (3)

$$\begin{aligned} B(x, y) &= (B(x)B(y))^{*n} \\ &= (B(x))^{*n}\ (B(y))^{*n} \\ &= b(x)\ b(y). \end{aligned} \tag{11}$$

By (9),

$$\begin{aligned} A_{ij} &= \langle \nabla F_i, \nabla F_j \rangle \\ &= \int \nabla F_i \cdot \nabla F_j \, dp \\ &= \int \frac{\partial}{\partial x} F_i \frac{\partial}{\partial x} F_j \, dp + \int \frac{\partial}{\partial y} F_i \frac{\partial}{\partial y} F_j \, dp. \end{aligned} \tag{12}$$

Let $i.c$ be the center of grid i, $i.x$ is the first component and $i.y$ is the second. By (2),

$$\begin{aligned} &\int \frac{\partial}{\partial x} F_i \frac{\partial}{\partial x} F_j \, dp \\ &= \int \frac{\partial}{\partial x} (B(\frac{p - i.c}{i.w}) \frac{1}{i.w^2}) \ \frac{\partial}{\partial x} (B(\frac{p - j.c}{j.w}) \frac{1}{j.w^2}) \, dp \\ &= \int \frac{\partial}{\partial x} B(\frac{p - i.c}{i.w}) \frac{1}{i.w^3} \ \frac{\partial}{\partial x} B(\frac{p - j.c}{j.w}) \frac{1}{j.w^3} \, dp \\ &= \int \int \frac{\partial}{\partial x} b(\frac{x - i.x}{i.w}) b(\frac{y - i.y}{i.w}) \frac{1}{i.w^3} \ \frac{\partial}{\partial x} b(\frac{x - j.x}{j.w}) b(\frac{y - j.y}{j.w}) \frac{1}{j.w^3} \, dx \, dy \\ &= -\frac{1}{i.w^3 \, j.w^3} \int b'(\frac{i.x - j.x - x}{i.w}) b'(\frac{x}{j.w}) \, dx \int b(\frac{i.y - j.y - y}{i.w}) b(\frac{y}{j.w}) \, dy \end{aligned} \tag{13}$$

The width is in $\{2^{-d} | \, d \leq D\}$, where D is max depth. D is small, therefore the number of $s.w$, $i.w$, $j.w$ is finite. Let

$$b_w(t) = b(\frac{t}{w}), \tag{14}$$

$$b'_w(t) = (b_w(t))'. \tag{15}$$

(13) is

$$\begin{aligned}&\int \frac{\partial}{\partial x}F_i \frac{\partial}{\partial x}F_j \, dp \\ &= -\frac{1}{i.w^2 \, j.w^2}(b'_{i.w} * b'_{j.w})(i.x - j.x) \; (b_{i.w} * b_{j.w})(i.y - j.y)\end{aligned} \tag{16}$$

Similarly, (5) can be computed by

$$\begin{aligned}W(p) &= \sum_{s \in S} F_{s.p}(p) \\ &= \sum \frac{1}{s.w^2} b(\frac{x - s.x}{s.w}) \; b(\frac{y - s.y}{s.w}).\end{aligned} \tag{17}$$

And (10) can be computed by (6) and

$$\begin{aligned}b_i &= \int \vec{V} \cdot \nabla F_i \, dp \\ &= \int \sum_{s \in S} F_{s.p}(q) \; s.\vec{N} \; \frac{1}{W(s.p)} \cdot \nabla F_i \, dp \\ &= \sum_{s \in S} \frac{1}{W(s.p)} \int F_{s.p} \; s.\vec{N} \cdot \nabla F_i \, dp \\ &= \sum \frac{1}{W(s.p)} \; s.\vec{N}.x \int F_{s.p} \; \frac{\partial}{\partial x} \; F_i \, dp \\ &\quad + \sum \frac{1}{W(s.p)} \; s.\vec{N}.y \int F_{s.p} \; \frac{\partial}{\partial y} \; F_i \, dp.\end{aligned} \tag{18}$$

Here $s.\vec{N}.x$ is the first component of $s.\vec{N}$. As in (13),

$$\int F_{s.p} \; \frac{\partial}{\partial x} F_i \, dp = \frac{1}{s.w^2 i.w^2}(b_{s.w} * b'_{i.w})(s.x - i.x) \; (b_{s.w} * b_{i.w})(s.y - i.y) \tag{19}$$

Hence, we can represent $A_{ij}, W(p), b_i$ by $b(x)$, and we need compute $b_w * b_w$, $b_w * b'_w$, $b'_w * b'_w$, b_w. It is easy to compute them, since the $4D$ univariate piecewise functions are simple. We first compute a value table of them to accelerate. Then get an approximate value from the table in computation.

References

[1] Kazhdan, M., Bolitho, M., and Hoppe, H. *Poisson surface reconstruction.* In Proceedings of the fourth Eurographics symposium on Geometry processing (Vol. 7).