Surface reconstruction from a point cloud

Supervisor: Davoud Mirzaei

Division of Scientific Computing, Department of Information Technology, Uppsala University.

davoud.mirzaei@it.uu.se

September 30, 2022

1 Background

A point cloud is a set of unorganized points in $\mathbb{R}^3$ which is usually obtained by laser scanning of an object. Because of the limited sample density, partial information of the real object has lost in the discrete point cloud, so the object's geometry can not be completely described. In order to obtain the geometric shape of the measuring object, we must fit a proper continuous surface which preserves geometrical features of the physical entity to the point cloud. See Figure 1

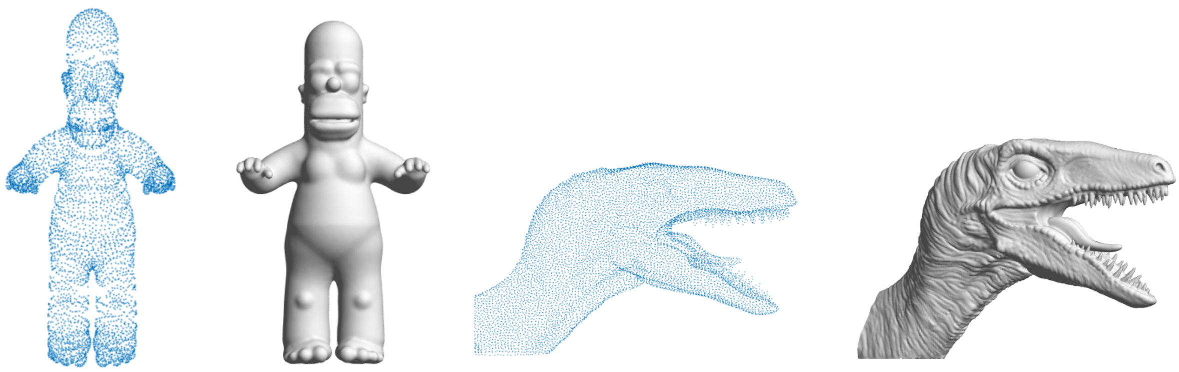

Figure 1: Point clouds and reconstructed surfaces using the method of [2].

The problem of reconstructing a surface from a point cloud has been used in a variety of applications, including computer graphics, computer aided design, medical imaging, image processing, manufacturing, and remote sensing.

2 Project description

Given a point cloud

$$X := \{\boldsymbol{x}_k = (x_k, y_k, z_k) \in \mathbb{R}^3,\ k = 1, 2, \dots, N\}$$

coming from an unknown surface Γ, the goal is to find another surface $\widetilde{\Gamma}$ which is a reconstruction (approximation) of Γ. In an implicit surface approach, Γ is defined as the surface of all points $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$ that satisfy the implicit equation

$$f(x, y, z) = 0,$$

for an unknown function f. In this case Γ is said to be the level set of f. A way to approximate f is to assume that f is zero at on-surface points X, and use additional off-surface points X^+ and X^- with positive and negative f values, respectively, and finally interpolate f on this $3N$ interpolation points. This approach leads to a surface reconstruction method which consists of three main steps: (1) generation of off-surface points, (2) interpolating f on the extended dataset, (3) computing the interpolation zero iso-surface. If the set of normals

$$\{\boldsymbol{n}_k = (n_k^x, n_k^y, n_k^z) : k = 1, \ldots, N\}$$

to Γ at points x_k are known (or can be approximated) the off-surface points can be generated by marching a small distance δ along the positive and negative directions of normals:

$$X^+ = \{\boldsymbol{x}_k + \delta \boldsymbol{n}_k : k = 1, 2, \ldots, N\}, \quad X^- = \{\boldsymbol{x}_k - \delta \boldsymbol{n}_k : k = 1, 2, \ldots, N\}.$$

Then we can assume that $f_X = 0$, $f_{X^+} = 1$ and $f_{X^-} = -1$ and interpolate f. The zero iso-surface of f approximates Γ, hopefully well.

But the technique described above is not the best one. A better technique is based on the fact that if $f : \mathbb{R}^3 \to \mathbb{R}$ defines a zero-level set Γ and $\boldsymbol{n}$ is a normal vector to Γ, then $\boldsymbol{n}$ is curl-free, i.e.,

$$\mathrm{curl}(\boldsymbol{n}) = 0.$$

The reason is clear: since $\boldsymbol{n}$ is proportional to ∇f (gradient of f) and the curl of a gradient field is zero. So, we can design an algorithm to recover (approximate) the potential f, such that $\nabla f \approx \boldsymbol{n}$ at every point $\boldsymbol{x}_k$.

In this project we focus on interpolation methods based on *radial basis functions* (RBF) for surface reconstruction. You are refereed to [3] to study the RBF approximations and their programming in MATLAB. A standard RBF interpolation with a proper localization will simply work for the first reconstruction approach but for the second approach we will focus on a curl-free RBF approximant to the normal vectors. For this part you need to follow [1] and [2]. Planed tasks are:

- learning and understanding the basics of the interpolation with RBFs,
- developing a numerical algorithm for surface reconstruction based on off-surface points,
- learning and understanding the interpolation with divergence-free and curl-free RBFs
- developing an algorithm for surface reconstruction based on the curl-free approach.

The suggested programming environment is Matlab, but other languages can be used as well.

References

[1] K. P. Drake, E. J. Fuselier, and G. B. Wright, A partition of unity method for divergence-free or curl-free radial basis function approximation, SIAM J. Sci. Comput., 43 (2021), pp. A1950–A1974.

[2] K. P. Drake, E. J. Fuselier, and G. B. Wright, Implicit surface reconstruction with a curl-free radial basis function partition of unity method, SIAM J. Sci. Comput., 44 (2022), pp. A3018–A3040.

[3] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdiscip. Math. Sci. 6, World Scientific, Singapore, 2007.