ŠVOČ 2015
Bratislava

Reconstruction of 3D objects using direct surface evolution

Rekonštrukcia 3D objektov z mračna bodov pomocou priamej evolúcie plôch

Meno a priezvisko študenta:	Bc. Patrik Daniel
Škola:	Slovenská technická univerzita
Fakulta:	Stavebná fakulta
Ročník a program/odbor štúdia:	2. ročník, 2. stupeň, matematicko-počítaćové modelovanie
Vedúci práce:	doc. Mgr. Mariana Remešíková, PhD.
Katedra:	KMaDG

Bratislava, 27.-29.5.2015

Contents

Abstract

We present a method for reconstruction of a single 3D object's surface from representative point cloud. Given the set of points, we construct a triangular mesh approximation of a surface they represent. The process of reconstruction is based on appropriately designed evolution of initial condition which is a simple triangulated surface containing the set of points in its inside. We establish a parabolic PDE describing the Lagrangian surface evolution and explain numerical approximation of our model. In order to control the quality of the mesh during the Lagrangian evolution and to obtain an appropriate representation of the desired surface, we perform tangential redistribution of mesh points during the surface evolution. We applied the proposed method to the biological data representing an early developmental stage of a zebrafish embryo. The obtained results can help in quantitative analysis of the embryogenesis.

Keywords: surface reconstruction, point cloud, surface evolution, tangential redistribution, finite volume method

Abstrakt

V tejto práci navrhujeme metódu na rekonštrukciu 3D objektu z mračna bodov. Pre zadané mračno bodov konštruujeme triangulovanú aproximáciu povrchu, ktorý dané mračno bodov reprezentuje. Algoritmus rekonštrukcie je založený na vhodne navrhnutom vývoji počiatočnej podmienky, ktorou je jednoduchý triangulovaný povrch obsahujúci dané mračno bodov vo svojom vnútri. Navrhujeme parabolickú parciálnu diferenciálnu rovnicu popisujúcu Lagrangeovský vývoj plochy a následne popisujeme numerickú aproximáciu takéhoto modelu. Za účelom riadenia kvality diskretizaćnej mriežky počas Lagrangeovského vývoja a dosiahnutia primerane kvalitnej reprezentácie vykonávame počas celého vývoja tengenciálnu redistribúciu bodov priestorovej diskretizačnej mriežky. Navrhovanú metódu sme aplikovali na biologické dáta predstavujúce skoré vývojové štádium embrya rybky zebrafish. Dosiahnuté výsledky môžu pomôcť v kvantitatívnej analýze embryogenézy.

Kľúčové slová: rekonštrukcia povrchu, mračno bodov, vývoj povrchu, tangenciálna redistribúcia, metóda konečných objemov

1 Introduction

It is one of the main tasks in the field of computer graphics and computer vision to obtain a proper digital representation of real world objects. A common way of representing a 3D object is to capture a set of points lying on the object's surface – point cloud representation of an object. A point cloud is basically a set of points in some coordinate system, usually the three-dimensional Cartesian coordinate system. Each point of the 3D point cloud defined by its x, y and z coordinate can also contain some other information e.g. colour or normal. This data structure is often used to represent not only the external surface of single 3D objects but also more general scenes. Point cloud data can be acquired by 3D laser scanners, RGB-D cameras (Microsoft Kinect), stereo cameras or can be artificially created by a software.

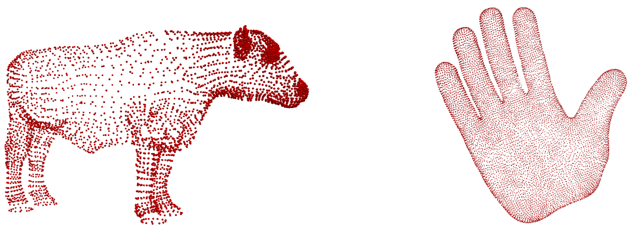


Figure 1: Point cloud representations of simple objects.

However, such a basic representation is not directly usable in most applications. Therefore a substantial amount of effort has been dedicated to the problem of obtaining a different surface representation of an object from point cloud data [1]. The process of converting a point cloud into more convenient representation is commonly referred to as surface reconstruction. The need for the new surface reconstruction techniques increases even more with the development of 3D scanning and 3D printing.

In this paper we present a method for constructing triangular mesh representation of a single static object's surface given a representative point cloud. As an input we need a set of points representing the object and an initial approximation of the desired surface. The initial approximation is a closed triangulated surface containing the given set of points in its inside. The process of surface reconstruction is based on appropriately designed evolution of this initial condition.

In general, two approaches are used for solving surface evolution problems. While the Eulerian approach considers a surface to be an isosurface (level-set) of a function whose domain is a subset of $\mathbb{R}^3$, the Lagrangian approach solves a surface evolution problem by evolving the surface directly.

We establish a parabolic PDE describing the Lagrangian surface evolution and explain numerical approximation of our model. In numerical approximation of any Lagrangian evolution model it is important to control the quality of the mesh during the process of evolution. Regarding this hardship of this kind of methods we explain the technique for required tangential redistribution of the discrete points as the surface evolves.

In this work we explain the surface evolution problem in a general intrinsic setting based on ideas from the general case of an m-dimensional manifold evolving in an n-dimensional manifold [2]. The Riemannian manifolds can be viewed simply as topological spaces with an extra structure defined on it. The basic knowledge about topology, topological spaces and Riemannian geometry is expected.

2 Surface evolution

We will call the general case of manifold evolution [2] any time dependant embedding $F : M \times [0, T_f] \to N$ of m-dimensional Riemannian manifold (M, g_M) in an n-dimensional Riemannian manifold (N, g_N).

In this paper we are concerned with the particular case of manifold evolution – surface evolution. We define a surface evolution as any smooth map $F : X \times [0, T_f] \to Y$, where (X, g_X) is a 2D Riemmanian manifold and (Y, g_Y) is $\mathbb{R}^3$, or an open subset of $\mathbb{R}^3$, accompanied with the standard Euclidean metric, here denoted by g_Y. We will assume that each time slice F^t is a smooth embedding of X in $Y \subseteq \mathbb{R}^3$. For our purposes it is sufficient to consider X to be a manifold without boundary.

We always assume the map $F : X \times [0, T_f] \to Y$ to be smooth, hence it can be defined as a solution of the evolution equation

$$\partial_t F = v. \tag{2.1}$$

The vector field $v(\cdot, t)$ serves as the velocity field of the evolution.

It proved to be useful to see the velocity v as a composition of two parts – the tangential and the normal component of the velocity vector

$$v = v_N + v_T. \tag{2.2}$$

Consequently the evolution equation (2.1) can be rewritten as

$$\partial_t F = v_N + v_T. \tag{2.3}$$

Notice that while the movement in the normal direction directly determines the shape of the evolving manifold, the motion in the tangential direction has no effect on it. However, the tangential movement of discrete grid points plays an important role in numerical realizations of problem (2.3). Inappropriate choice of tangential velocity v_T may lead to unacceptable numerical errors or even crash of computation process.

In many applications the uniform or asymptotically uniform redistribution of grid points is required. This type of redistribution can significantly stabilize and speed up the numerical computation. Furthermore, besides this type of redistribution, it makes sense to sample the grid points in regions with higher curvature more densely than in nearly flat ones. This type of curvature based redistribution can lead to better representation of a particular object.

2.1 Tangential redistribution of points on an evolving surface

We can assume without any loss of generality the surface evolution equation (2.3) to be in the following form

$$\partial_t F = v_N. \tag{2.4}$$

We will enrich equation (2.4) with tangential term v_T purely in order to control the mesh quality. Our aim is to prescribe a condition which will allow us to determine the tangential velocity field.

Given the Riemannian manifold (X, g_X), the metric g_X induces the measure on Borel sets of X, denoted by μ_X. We can induce another metric on X as the pull-back of the metric g_Y along the map F^t. This new metric $g_{F^t} = (F^t)^*(g_Y)$ also induces a measure on

Borel sets of X. We will denote this induced measure by μ_{F^t}. The measure of an arbitrary measurable set $U \subseteq X$ with respect to the measure μ_{F^t} can be expressed as

$$\mu_{F^t}(U) = \int_U \mathrm{d}\mu_{F^t}. \tag{2.5}$$

Notice that while the measure μ_X remains the same during the whole evolution, the induced measure μ_{F^t} evolves according to the changes of shape of evolving embedding of the manifold $\{\mathrm{Im}\left(F^t\right)\}_{t\in(0,T_f)}$.

We are particularly interested in the relation between the measures μ_X and μ_{F^t}. For this purpose we again express the measure of an arbitrary measurable set $U \subseteq X$ with respect to the measure μ_{F^t}

$$\mu_{F^t}(U) = \int_U \mathrm{d}\mu_{F^t} = \int_U G^t \,\mathrm{d}\mu_X \,. \tag{2.6}$$

The quantity G^t in (2.6) is the Radon-Nikodým derivative of measure μ_{F^t} with respect to measure μ_X. We will call it area density. We can observe from (2.6) that this quantity expresses how the mapping F locally shrinks or expands areas of evolving surface.

From work [3] we know that G evolves in a manner described by following evolution equation

$$\partial_t G = \left(-v_N \cdot h + \mathrm{div}_{g_F} w_T\right) G. \tag{2.7}$$

where h^t is the mean curvature vector of F^t. The operator div_{g_F} stands for divergence on manifold X with respect to metric g_{F^t} and w_T^t is the pull-back of v_N^t along the map F^t.

Combining the formula (2.7) and relation (2.6) we get the evolution equation for the global area of X measured by the induced measure μ_{F^t} denoted by A^t. After applying the divergence theorem and assuming $\partial X = \varnothing$ or $g_{F^t}(w_T, \nu)|_{\partial X} = 0$, where ν is the outward unit normal to ∂X with respect to g_{F^t}, we obtain

$$\partial_t A = \int_X (-v_N \cdot h) \,\mathrm{d}\mu_{F^t} \,. \tag{2.8}$$

2.2 Area-oriented asymptotically uniform tangential redistribution

The area-oriented redistribution means that we want to control the area density G of the evolution $F : X \times [0, T_f] \to Y$. In the discrete setting it will allow us to control the area of our mesh elements.

We will call an embedding F^t area uniform with respect to g_X if the area density G^t is constant on its domain. In practical applications, it is reasonable to start the evolution with a particular embedding $\{\mathrm{Im}\left(F^t\right)\}_{t=0}$ (the initial condition) with corresponding varying G and as the evolution process proceeds, we let the embedding approach the area uniform embedding with constant G.

We can compute the global area of X measured with respect to the measure μ_X

$$A_X = \mu_X(X) = \int_X \mathrm{d}\mu_X \tag{2.9}$$

and using (2.6) the global area of X measured according to μ_{F^t}

$$A = \mu_{F^t}(X) = \int_X G \,\mathrm{d}\mu_X \,. \tag{2.10}$$

With constant G we can rewrite (2.10) as $A = GA_X$. Hence, for a constant G we get $G = \frac{A}{A_X}$ and the corresponding dimensionless condition is

$$\frac{G}{A} = \frac{1}{A_X}\,. \tag{2.11}$$

An asymptotically area-uniform evolution $F : X \times [0, T_f] \to Y$ satisfies the condition

$$\frac{G}{A} \xrightarrow[t\to\infty]{} \frac{1}{A_X}\,. \tag{2.12}$$

For instance, (2.12) is satisfied if $\frac{G}{A}$ is a solution of the following differential equation

$$\partial_t \left(\frac{G}{A}\right) = \omega \left(\frac{1}{A_X} - \frac{G}{A}\right), \tag{2.13}$$

where parameter $\omega : [0, T_f] \to \mathbb{R}_+$ acts as a speeding factor.

We can use the product rule on the left hand side of (2.13) and combine it with (2.7) and (2.8). We obtain the following criterion for the vector field w_T on X defined as the pull-back of v_T alongside F

$$\mathrm{div}_{g_F} w_T = v_N \cdot h - \frac{1}{A}\int_X v_N \cdot h \,\mathrm{d}\mu_F + \omega\left(\frac{A}{GA_X} - 1\right). \tag{2.14}$$

2.3 Tangential redistribution of points on evolving surface based on the curvature

In some applications, the uniform redistribution of points might not always meet the requirements on the surface's representation. Our aim is to find a tangential velocity field which will lead to higher density of discrete points in regions with higher curvature and lower density in nearly flat ones.

Considering G constant over a domain $U \subseteq X$ we have

$$\mu_F(U) = G\,\mu_X(U). \tag{2.15}$$

It can be inferred from (2.15), that higher value of G causes the local expansion of the area of U and lower value leads to the local shrinking. In the discrete setting the density of grid points will be higher in regions with lower value of G. On the contrary, in the regions with higher value of G the areas are expanded and as a result the density of grid points is lower. We want to redistribute the points according to the value of the mean curvature H. One possible way to do this is requiring

$$G(x)\, f(H(x)) = const \qquad \forall x \in X, \tag{2.16}$$

where f is a positive increasing function of mean curvature H at point x. Since $Gf(H)$ is constant over X, the value of G lowers with rising value of $f(H)$. Possible choices of function $f : \mathbb{R} \to \mathbb{R}$ are for example [4]

$$\begin{aligned} f(H) &= H^{1/3} \\ f(H) &= H^{2/3} \\ f(H) &= \exp(\alpha H), \quad \alpha \in \mathbb{R}^+. \end{aligned}$$

Similarly like in the case of the asymptotically uniform redistribution, we can obtain the criterion for the vector field w_T on X defined as the pull-back of v_T alongside F

$$\mathrm{div}_{g_F} w_T = v_N \cdot h - \frac{1}{A}\int_X v_N \cdot h \, \mathrm{d}\mu_F + \omega\left(\frac{A}{G}\frac{\frac{1}{f(H)}}{\int_X \frac{1}{f(H)}\mathrm{d}\mu_X} - 1\right), \tag{2.17}$$

where parameter $\omega : [0, T_f] \to \mathbb{R}_+$ acts as a speeding factor.

Unfortunately neither (2.14) nor (2.17) do not determine the vector field w_T uniquely. There is infinite number of solutions for equations (2.14) and (2.17). In order to uniquely determine a single solution, we will assume w_T to be a gradient field

$$w_T^t = \nabla_{g_{F^t}} \psi^t \,, \tag{2.18}$$

where ψ is a scalar function defined on X, $\psi : X \times [0, T_f] \to \mathbb{R}$.

Using assumption (2.18), we obtain the following equation for the Laplace-Beltrami operator of ψ^t for the asymptotically area-uniform redistribution

$$\Delta_{g_{F^t}} \psi^t = \mathrm{div}_{g_{F^t}}(\nabla_{g_{F^t}} \psi^t) = v_N \cdot h - \frac{1}{A}\int_X v_N \cdot h \, \mathrm{d}\mu_F + \omega\left(\frac{A}{GA_X} - 1\right). \tag{2.19}$$

and in case of redistribution of points based on curvature

$$\Delta_{g_F} \psi^t = v_N \cdot h - \frac{1}{A}\int_X v_N \cdot h \, \mathrm{d}\mu_F + \omega\left(\frac{A}{G}\frac{\frac{1}{f(H)}}{\int_X \frac{1}{f(H)}\mathrm{d}\mu_X} - 1\right). \tag{2.20}$$

After adding an appropriate boundary condition to equation (2.19) or (2.20) we are able to uniquely determine ψ^t. Given an evolving surface without boundary, we have to prescribe the value of ψ^t in one selected point.

3 Mathematical model of surface reconstruction from point clouds

3.1 Distance function

Let $\Omega \subseteq \mathbb{R}^n$, given the set of points $P \subset \Omega$ with its elements p_i, $i = 1 \ldots n_P$, the distance function $d : \Omega \to \mathbb{R}$ for the points in P is defined at a point $x \in \Omega$ as the smallest Euclidean distance of x to the points in P

$$d(x) = \min_{p \in P} ||x - p||.$$

It can also be defined as the solution of Eikonal equation, i.e. the distance function d to the set of points P (Figure 2) satisfies

$$|\nabla d(x)| = 1\,, \quad x \in \Omega, \tag{3.1}$$

with boundary condition $d(x) = 0, x \in P \subset \Omega$.

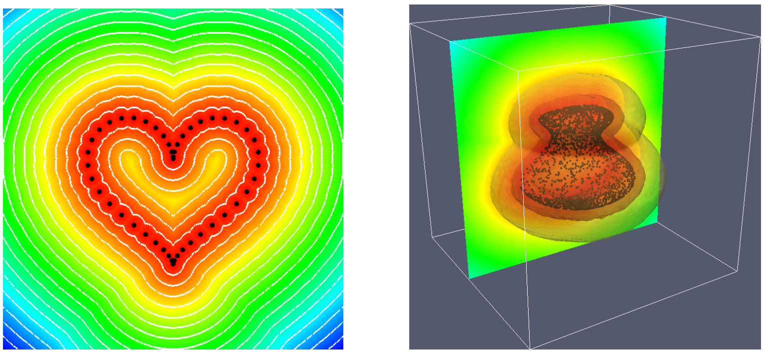


Figure 2: Examples of distance functions to the given point clouds. The 2D graph and contours of the distance function to the points of the heart curve on the left and the slice of the graph and two contours of the distance function to the 3D point cloud representing the vase.

3.2 Mathematical model

Our aim is to construct a triangular mesh representation of an object's surface $\Sigma \subset \mathbb{R}^3$ given only a finite sampling of it. As an input we need the corresponding point cloud without any additional information, such as normals or scanner information, and a closed triangulated surface S^0 containing the given set in its inside – an approximation of the desired surface, which will serve as an initial condition for our model. The evolution of the initial approximation is appropriately designed in order to obtain the desired representation of point cloud P.

The suggested model consists of an advection term with the velocity proportional to ∇d and a diffusion term with the velocity proportional to d. These two terms represent the normal component of velocity v in general evolution equation (2.1). Moreover, we enrich the model with the appropriate tangential movement term which will allow us to control the mesh quality during the whole process of the evolution.

Let Ω be a subset of $\mathbb{R}^3$ and $P = \{p_1, \ldots, p_{n_P}\} \subset \Omega$ be a set of points sampling the surface Σ. Furthermore, let $d_0 : \Omega \to \mathbb{R}$ be the distance function to the point cloud P, which is the solution of hyperbolic partial differential equation (3.1). The existence and uniqueness of the solution d_0 are shown in [5]. Even though the solution is continuous, it may not be everywhere differentiable. Therefore, we will consider regularized distance function d defined as the convolution $G_\sigma * d_0$, where G_σ is a Gauss kernel. From any point $x \in \Omega$ we can approach P moving towards the steepest distance descent, e.g. following the direction of $-\nabla d$. Given a smooth closed initial condition surrounding P, the basic idea is to let each point of it move in the direction of $-\nabla d$ and consequently approach the point cloud P.

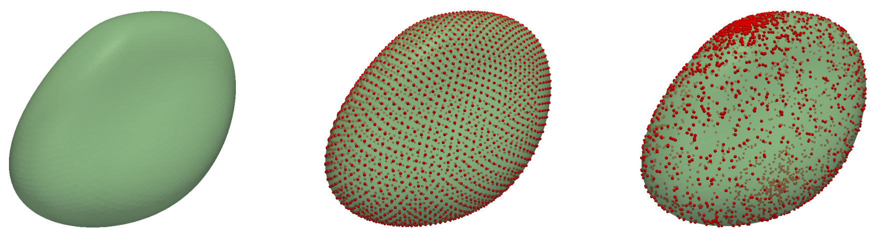
Next, let (X, g_X) be a two-dimensional Riemannian manifold without boundary (e.g. 2D Riemannian sphere) and $F : X \times [0, T_f] \to \Omega$ be an evolution of X in Ω. The metric g_X again induces the Borel measure on X denoted by μ_X . In addition, let g_F and μ_F represent respectively the metric and the Borel measure induced by F. Recalling the definition of surface evolution from the previous chapter, each time slice F^t is a smooth embedding of X in Ω. We will denote the image of $F^t = F(\cdot, t)$ by S^t. The map F is a solution of the evolution equation

$$\partial_t F = v_N + v_T = w_a \left(-\nabla d \cdot N\right) N + w_d \, d \, \Delta_{g_F} F + v_T, \quad F(\cdot, 0) = S^0, \tag{3.2}$$

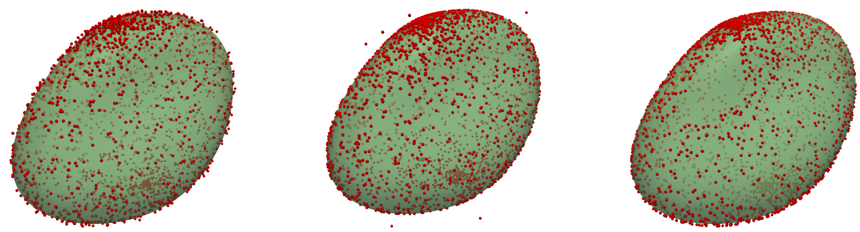
where N is the outward unit normal to S and Δ_{g_F} stands for the Laplace-Beltrami operator with respect to the metric g_F. The parameters $w_a : [0, T_f] \to \mathbb{R}_+$ and $w_d : [0, T_f] \to \mathbb{R}_+$ act as speeding factors for the advection and the diffusion respectively.

Let us take a closer look at the three terms on the right hand side of equation (3.2). The advection term $w_a\,(-\nabla d \cdot N)\,N$ is the main driving force of the evolution process and it makes the image S approach the point cloud P. The projection of $-\nabla d$ onto the surface outward normal N eliminates the tangential movement which does not affect the actual shape of surface S and it also makes the movement of a point dependent on its neighbourhood – the surface is moving as a whole rather than a set of independent points. If $-\nabla d$ is perpendicular to the normal N, the advective evolution stops and thus a surface patch is formed in the empty space between the points of P. Since $\Delta_{g_F} F = HN$, where H is mean curvature of surface S, the diffusion term $w_d\,d\,\Delta_{g_F} F$ represents the evolution driven by mean curvature and in the model it guarantees the sufficient smoothness of the evolving surface S. In addition, it enables our method to effectively handle substantial imperfections of the point cloud data like non-uniform sampling density, noise or outlying points (Figure 3). Moreover, it also speeds up the evolution process proportionally to the distance to the point cloud P.

Hence, only these two terms affect the actual shape of the evolving surface S. The third term is added in order to control the area density G, which in the discrete setting will allow us to control the area of mesh elements and therefore the redistribution of grid points on the evolving surface.



(a) Original surface (b) Uniform sampling. (c) Nonuniform sampling.



(d) Noisy data (e) Data with outliers (f) Missing data

Figure 3: Different forms of point cloud imperfections.

We introduced two techniques for tangential redistribution of points on the evolving surface in case of general surface evolution. Now we can apply the general results for our model.

In order to determine the tangential term v_T leading to asymptotically uniform redistribution we have to solve (2.19). In case we require the points to be asymptotically redistributed according to the mean curvature of the evolving surface S, we have the equation (2.20). In both equations (2.19) and (2.20) we use the normal velocity term from our model (3.2)

$$v_N = w_a\,(-\nabla d \cdot N)\,N + w_c\,d\,\Delta_{g_F} F. \tag{3.3}$$

Since we deal with a Riemannian manifold without boundary, we have to prescribe the value of ψ just in one selected point.

4 Discretization of surface reconstruction model

Computation of the distance function is essential for our model. Thus, we need a fast and accurate algorithm for approximating the distance function to the point cloud P. From the variety of available methods for solving problem (3.1) we chose the fast sweeping method introduced by Zhao [6].

The first step of the discretization of our model is the discretization of the computational domain Ω which is considered to be a box. It is discretized into a finite number of non-intersecting volume elements – voxels. Each voxel $v_{i,j,k}$, $i, j, k = 1 \dots N$, of the grid is a cube of side length h. Let $x_{i,j,k}$ denote the centre of voxel $v_{i,j,k}$. We approximate the solution of problem (3.1) in each voxel with the value of numerical solution $d_{i,j,k}$ given by the fast sweeping method in $x_{i,j,k}$.

4.1 Discretization of the initial condition

The initial condition for the evolution process is usually a simple surface, e.g. a sphere or an ellipsoid. The advantage of using these simple surfaces is that they can be easily triangulated in many different ways. We employ the method based on subdividing an icosahedron (Figure 4). In this method, in order to get finer approximation of the unit sphere, each triangle is subdivided into four smaller triangles, whose points are then projected on the sphere.

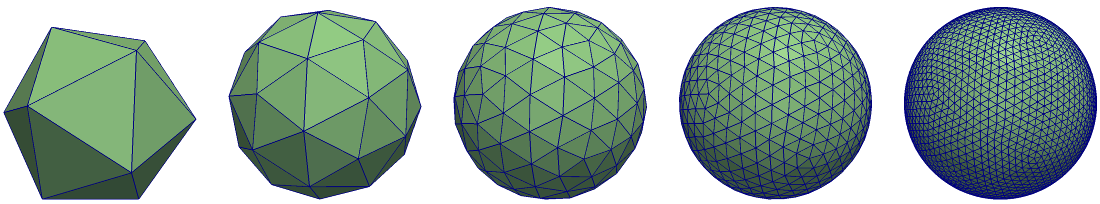


Figure 4: Various triangulations of unit sphere obtained by subdividing the icosahedron. Number of points used in triangulation from left to right: 12, 42, 162, 642, 2562.

After obtaining the triangulation of the unit sphere, we can simply use standard geometric transformations to obtain desired triangulated surface containing the given point cloud in its inside. We assume the topological equivalence between the initial condition and the desired surface. Hence, the topology of the initial condition is preserved during the whole process of evolution.

4.2 The time discretization

We now discuss the discretization of (3.2) in time. We consider a uniform partition of the time interval $[0, T_f]$ with the time step τ, $t^n = n\tau$. For the sake of clarity, we will use the notation $F^n = F(\cdot, t^n)$. After applying the semi-implicit approach, we obtain

$$\frac{F^n - F^{n-1}}{\tau} = w_a \left(-\nabla d \cdot N^{n-1}\right) N^{n-1} + w_d \, d \, \Delta_{g_{F^{n-1}}} F^n + v_T^{n-1}. \tag{4.1}$$

4.3 The space discretization

The space discretization is performed by the finite volume approach. We suppose to have the triangular representation of the abstract two-dimensional manifold X consisting of vertices X_i, $i = 1 \dots n_v$, edges e_j, $j = 1 \dots n_e$ and triangles $\mathcal{T}_k$, $k = 1 \dots n_t$.

Now we present a detailed description of the technique for constructing the co-volume mesh by barycentric subdivision of triangles $\mathcal{T}_k$. For the sake of clarity we describe the procedure for one particular co-volume V_i constructed around the vertex X_i. We suppose it is the common vertex for m triangles $\mathcal{T}_1 \dots \mathcal{T}_m$, thus it is also the common vertex for m edges $e_1 \dots e_m$, where e_p connects X_i with X_{i_p}. Moreover, let us denote the barycentre of $\mathcal{T}_p$ by B_p and the center of the edge e_p by C_p for $p = 1 \dots m$. The polygonal co-volume V_i with boundary edges $\sigma_{p,1} = C_p B_p$ and $\sigma_{p,2} = B_p C_{p+1}$ is constructed as the union of the triangles $\mathcal{V}_{p,1} = X_i C_p B_p$ and $\mathcal{V}_{p,2} = X_i B_p C_{p+1}$ for $p = 1 \dots m$, where we set $C_{m+1} = C_1$. The mesh constructed in this manner is a simplicial partition of the manifold X.

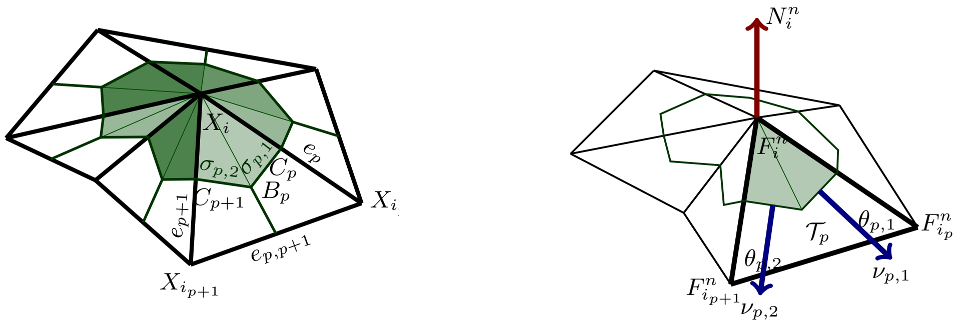


Figure 5: The space discretization mesh. The triangulation of the abstract manifold X on the left and the approximation of its embedding in $\mathbb{R}^3$ on the right.

We construct a piecewise linear approximation of embedding F^n denoted by $\bar{F}^n$. This approximation satisfies $\bar{F}^n(X_i) = F^n(X_i)$. The points F_i^n will be the unknowns in the fully discretized model. The embedding $\bar{F}^n$ induces a metric g_{F^n} on X which induces a measure μ_{F^n} on Borel sets of X. In the numerical scheme we will use the angles $\theta^n_{p,1}$, $\theta_{p,2}$ of $\bar{F}^n(\mathcal{T}_p)$ adjacent to $F(X_i)$ and $F(X_{i_{p+1}})$ respectively. In addition, we will use the outward unit normals $\nu_{p,1}$ and $\nu_{p,2}$ to the embedded co-volume boundary edges $\bar{F}^n(\sigma_{p,1})$, $\bar{F}^n(\sigma_{p,2})$.

The basic idea of the finite volume method is that we integrate the differential equation over each co-volume and then approximate the integrals using different interpolation techniques from the discrete data in the grid points. Integrating (3.2) over a co-volume V_i yields

$$\int_{V_i} \frac{F^n - F^{n-1}}{\tau} \, \mathrm{d}\mu_{F^{n-1}} = \int_{V_i} w_a \left(-\nabla d \cdot N^{n-1}\right) N^{n-1} \, \mathrm{d}\mu_{F^{n-1}} + \int_{V_i} w_d \, d \, \Delta_{g_{F^{n-1}}} F^n \, \mathrm{d}\mu_{F^{n-1}} + \int_{V_i} v_T^{n-1} \, \mathrm{d}\mu_{F^{n-1}}. \tag{4.2}$$

The approximation of the left hand side of (4.2) reads

$$\int_{V_i} \frac{F^n - F^{n-1}}{\tau} \, \mathrm{d}\mu_{F^{n-1}} \approx \mu_{F^{n-1}}(V_i) \, \frac{F^n - F^{n-1}}{\tau}. \tag{4.3}$$

4.3.1 Discretization of the advection term

First, we compute the unit outward normal N_i^n at F_i^n simply as the average of the outward unit normals to all triangles $\bar{F}(\mathcal{T}_p)$, $p = 1 \dots m$, containing the vertex F_i^n. Secondly ∇d_i in the corresponding voxel is computed by central differences

$$\nabla d_i = \nabla d_{i,j,k} = \left(\frac{d_{i+1,j,k} - d_{i-1,j,k}}{2h}, \frac{d_{i,j+1,k} - d_{i,j-1,k}}{2h}, \frac{d_{i,j,k+1} - d_{i,j,k-1}}{2h} \right), \tag{4.4}$$

where the triplet (i, j, k) represents indices of voxel containing the point F_i^{n-1} of the evolving surface. Finally we are able to approximate the advection term

$$\int_{V_i} w_a \left(-\nabla d \cdot N^{n-1}\right) N^{n-1} \, \mathrm{d}\mu_{F^{n-1}} \approx w_a \left(-\nabla d_i \cdot N_i^{n-1}\right) N_i^{n-1} \, \mu_{F^{n-1}}(V_i), \tag{4.5}$$

where $\mu_{F^{n-1}}(V_i)$ is the measure of the co-volume V_i with respect to the measure $\mu_{F^{n-1}}$.

4.3.2 Discretization of the Laplace-Beltrami operator term

For approximating the Laplace-Beltrami operator term in (4.2) we use the cotangent scheme [7]

$$\int_{V_i} w_d \, d \, \Delta_{g_{F^{n-1}}} F^n \, \mathrm{d}\mu_{F^{n-1}} \approx w_d \, d_i \, \frac{1}{2} \sum_{p=1}^{m} \left(\cot\left(\theta_{i,p-1,1}^{n-1}\right) + \cot\left(\theta_{i,p,2}^{n-1}\right) \right) \left(F_i^n - F_{i_p}^n \right), \tag{4.6}$$

where d_i is the value of the distance function in voxel containing the point F_i^{n-1} and $\theta_{i,0,1} = \theta_{i,m,1}$.

4.3.3 Discretization of the tangential term

Using the assumption that w_T is a gradient field and applying the special version of Stokes theorem [8] we obtain

$$\int_{V_i} v_T^{n-1} \, \mathrm{d}\mu_{F^{n-1}} = \int_{\partial V_i} \psi^{n-1} \, \nu_i^{n-1} \, \mathrm{d}H_{\mu_{F^{n-1}}} - \int_{V_i} \psi^{n-1} h^{n-1} \, \mathrm{d}\mu_{F^{n-1}}. \tag{4.7}$$

The approximation of (4.7) reads

$$\begin{aligned} \int_{V_i} v_T^{n-1} \, \mathrm{d}\mu_{F^{n-1}} \approx & \sum_{p=1}^{m} \left(||\sigma_{i,p,1}||_{n-1} \, \psi_{i,p,1}^{n-1} \, \nu_{i,p,1}^{n-1} + ||\sigma_{i,p,2}||_{n-1} \, \psi_{i,p,2}^{n-1} \, \nu_{i,p,2}^{n-1} \right) \\ & - \mu_{F^{n-1}}(V_i) \, \psi_i^{n-1} \, h_i^{n-1}, \end{aligned} \tag{4.8}$$

where $|| \cdot ||_{n-1}$ stands for length measured with respect to the metric induced by F^{n-1} and $\psi_{i,p,1}$, $\psi_{i,p,2}$ are the values of function ψ^{n-1} in the midpoints of the co-volume boundary edges $\sigma_{i,p,1}$ and $\sigma_{i,p,1}$ respectively. These midpoint values of ψ^{n-1} can be approximated by linear interpolation from the known values of ψ^{n-1} in the vertices X_i

$$\psi_{i,p,1} = \frac{5\psi_i^{n-1} + 5\psi_{i_p}^{n-1} + 2\psi_{i_{p+1}}^{n-1}}{12}, \quad \psi_{i,p,2} = \frac{5\psi_i^{n-1} + 2\psi_{i_p}^{n-1} + 5\psi_{i_{p+1}}^{n-1}}{12}. \tag{4.9}$$

In the following text we will examine the computation of function ψ more closely. In order to solve equation (2.19) or (2.20) we need to approximate the area density G. We can approximate the area of 2-manifold X measured with respect to the measure $\mu_{F^{n-1}}$ in two ways

$$A^{n-1} = \int_X G^{n-1}\,\mathrm{d}\mu_X \approx \sum_{i=1}^{n_v} G_i^{n-1}\,\mu_X(V_i)\,. \tag{4.10}$$

$$A^{n-1} \approx \sum_{i=1}^{n_v} \mu_{F^{n-1}}(V_i) \tag{4.11}$$

The choice of measure μ_X is up to us, since we do not have any conditions imposed on it. Assuming $\mu_X(X) = A_X$ we can set $\mu_X(V_i) = A_X/n_v$ for $i = 1 \ldots n_v$. When we use the following approximation of area density G

$$G_i^{n-1} = \frac{n_v}{A_X}\,\mu_{F^{n-1}}(V_i), \tag{4.12}$$

the equality between right hand sides of (4.10) and (4.11) holds true.

We can again make use of the finite volume approach. We integrate (2.19) and (2.20) over each co-volume V_i and then approximate the obtained integrals. For approximation of the integral from Laplace-Beltrami operator of the unknown function ψ^{n-1} we can employ the cotangent scheme presented earlier

$$\int_{V_i} \Delta_{g_{F^{n-1}}} \psi^{n-1}\,\mathrm{d}\mu_{F^{n-1}} \approx \frac{1}{2}\sum_{p=1}^{m}\left(\cot\left(\theta_{i,p-1,1}^{n-1}\right) + \cot\left(\theta_{i,p,2}^{n-1}\right)\right)\left(\psi_i^{n-1} - \psi_{i_p}^{n-1}\right). \tag{4.13}$$

In the approximation of $\int_{V_i} v_N \cdot h\,\mathrm{d}\mu_F$ we will use the equality

$$\Delta_{g_F} F = h = HN, \tag{4.14}$$

where h is the mean curvature normal of F, H is the the mean curvature of F and N is the outward unit normal to the embedding of X. For the approximation of h we can once again make use of the cotangent scheme (4.6). We can just divide the approximation of the integral (4.6) by $w_d\, d_i\, \mu_{F^{n-1}}(V_i)$ to approximate h_i^{n-1}

$$h_i^{n-1} \approx \frac{1}{2\,\mu_{F^{n-1}}(V_i)}\sum_{p=1}^{m}\left(\cot\left(\theta_{i,p-1,1}^{n-1}\right) + \cot\left(\theta_{i,p,2}^{n-1}\right)\right)\left(F_i^{n-1} - F_{i_p}^{n-1}\right). \tag{4.15}$$

The approximation of mean curvature H in the vertex X_i is then computed as the Euclidean norm $||h_i^{n-1}||$. Now we are able to discretize the integrals of all terms on the right hand side of (2.19) or (2.20). For approximation of the first term we use

$$\begin{aligned}\int_{V_i} v_N^{n-1} \cdot h^{n-1}\,\mathrm{d}\mu_{F^{n-1}} &\approx \mu_{F^{n-1}}(V_i)\,({v_N}_i^{n-1} \cdot h_i^{n-1})\\ &= \mu_{F^{n-1}}(V_i)(w_a(-\nabla d_i \cdot N_i^{n-1})N_i^{n-1} + w_d\, d_i\, h_i^{n-1}) \cdot h_i^{n-1}\\ &= \mu_{F^{n-1}}(V_i)(w_a(-\nabla d_i \cdot N_i^{n-1}) + w_d\, d_i\, H_i^{n-1})\, H_i^{n-1}.\end{aligned} \tag{4.16}$$

Similarly, for the second term we have

$$\begin{aligned}-\int_{V_i} \frac{1}{A}\int_X {v_N}^{n-1} \cdot h^{n-1}\,\mathrm{d}\mu_{F^{n-1}}\,\mathrm{d}\mu_{F^{n-1}} &\approx -\mu_{F^{n-1}}(V_i)\frac{1}{A}\sum_{i=1}^{n_v}\mu_{F^{n-1}}(V_i)({v_N}_i^{n-1} \cdot h_i^{n-1})\\ &= -\mu_{F^{n-1}}(V_i)\frac{1}{A}\sum_{i=1}^{n_v}\mu_{F^{n-1}}(V_i)(w_a(-\nabla d_i \cdot N_i^{n-1}) + w_d\, d_i\, H_i^{n-1})\, H_i^{n-1}\end{aligned} \tag{4.17}$$

For asymptotically uniform redistribution the last term reads

$$\int_{V_i} \omega \left(\frac{A}{GA_X} - 1 \right) \mathrm{d}\mu_{F^{n-1}} \approx \mu_{F^{n-1}}(V_i)\, \omega \left(\frac{A}{n_v\, \mu_{F^{n-1}}(V_i)} - 1 \right) \tag{4.18}$$

and for the curvature based redistribution we have

$$\begin{aligned} &\int_{V_i} \omega \left(\frac{A}{G} \frac{\frac{1}{f(H)}}{\int_X \frac{1}{f(H)} \mathrm{d}\mu_X} - 1 \right) \mathrm{d}\mu_{F^{n-1}} \\ &\approx \mu_{F^{n-1}}(V_i)\, \omega \left(\frac{A}{\frac{n_v}{A_X}\, \mu_{F^{n-1}}(V_i)} \frac{\frac{1}{f(H_i^{n-1})}}{\sum_{i=1}^{n_v} \left(\mu_X(V_i) \frac{1}{f(H_i^{n-1})} \right)} - 1 \right), \end{aligned} \tag{4.19}$$

where we define $A_X = \mu_X(A) = 1$ and $\mu_X(V_i) = 1/n_v$.

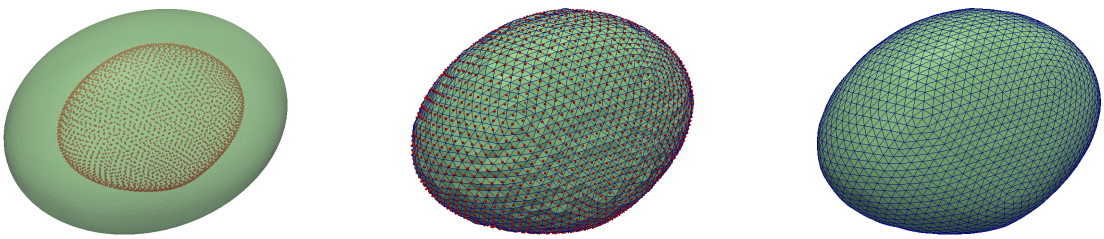
Combining (4.13) with (4.16), (4.17) and (4.18) or (4.19), depending on the type of required redistribution, we obtain the linear system for the values of ψ_i^{n-1}. We have to prescribe the value of ψ^{n-1} in one particular vertex, e.g. $\psi_1^{n-1} = 0$.

After solving the linear system for $\psi_i^{n-1}, i = 1 \ldots n_v$, the appropriate combination of (4.3), (4.5), (4.6) and (4.8) leads to the fully discretized surface reconstruction model. For solving linear systems of equations we use the iterative Bi-CGStab (BiConjugate Gradient Stabilized) method [9].

5 Numerical experiments

We implemented the proposed method in C++. Some computations were parallelized using OpenMP (Open Multi-Processing) API [10]. In order to demonstrate the robustness of the proposed method we performed the following numerical experiments on testing data shown in Figure 3.

For the first set of testing data, the distance function was computed in a voxel grid with $200 \times 200 \times 200$ voxels. The initial condition is an ellipsoid containing the point cloud. We used the triangular mesh of the ellipsoid consisting of 2562 vertices and 5120 triangles. In the first experiment, shown in Figure 6, we show the use of proposed method for reconstruction of the surface given the uniformly sampled point cloud. The parameters of the model were set to $w_a = 300$, $w_d = 100$, $\tau = 0.001$. We used asymptotically uniform redistribution of points with $\omega = 100$. The result of the evolution process can be easily smoothed by mean curvature flow (MCF), we just set $w_a = 0$ and the value of distance function in each voxel to 1. We performed 95 time steps of the evolution and afterwards we smoothed the obtained result with 25 additional steps of MCF with $\tau = 0.0005$ and $w_d = 450$.

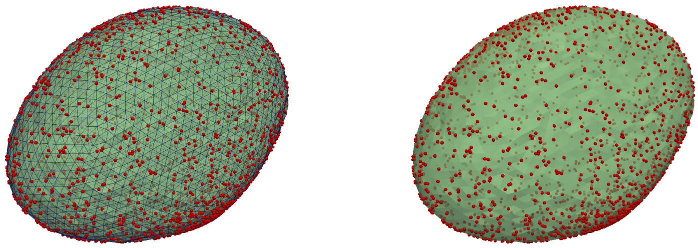


(a) Initial surface together with the point cloud

(b) Evolved surface after 95 steps of evolution

(c) Reconstructed surface after smoothing.

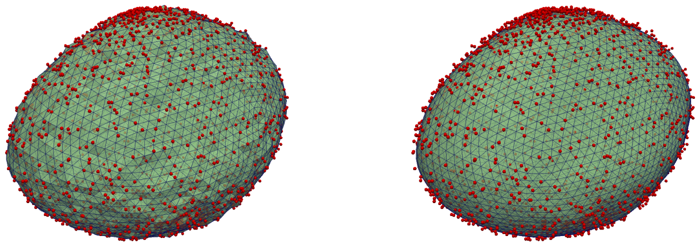
Figure 6: Reconstruction of the testing surface from a uniform point cloud.



(a) Evolved surface after 95 steps of evolution

(b) Evolved surface after 95 steps of evolution displayed without its triangulation

Figure 7: Reconstruction of the testing surface from a non-uniform point cloud.



(a) Evolved surface after 100 steps of evolution

(b) Evolved surface after additional 30 steps of mean curvature flow

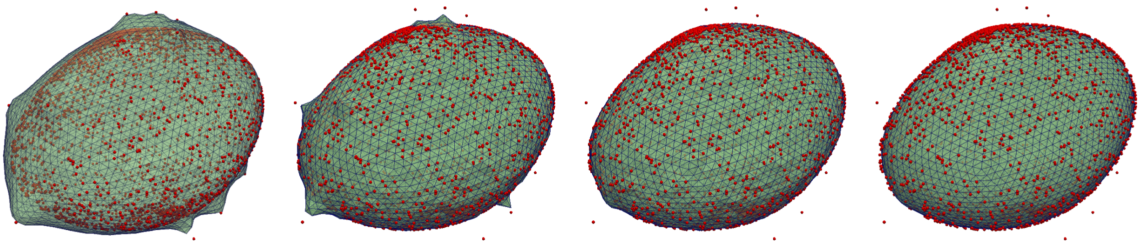
Figure 8: Reconstruction of the testing surface from a noisy point cloud.

In the second experiment we used the same setting of parameters, but the point cloud was non uniformly sampled in this test case. We can observe from Figure 7 that the suggested method is robust enough to non uniform sampling.

Next test case (Figure 8) represents the very common imperfection occurring in the point cloud data – noise. We performed 100 steps of evolution with the same setting of parameters as in the previous two test cases. In order to suppress the level of remaining noise we smoothed the resulting surface with 30 additional MCF steps with magnitude $w_d = 500$.

Furthermore, Figure 9 shows how our method can handle the outlying points. Due to the use of the regularized distance function the diffusion term does not vanish and hence the evolving surface overcomes the outlying points. The parameters were set to $w_a = 300$, $w_d = 120$, $\tau = 0.001$, $\omega = 100$. We again performed additional 25 steps of MCF with $w_d = 450$. The diffusion term also incorporates a minimal surface area regularization to ensure the smoothness in regions of missing data as demonstrated in Figure 10. In this test case the parameters were set to the same values as in the test case with outlying points.

We performed all experiments on Intel®Core™ i7-4702MQ 2.20 GHz processor. The amount of needed CPU time was very similar for each test case from the first set of testing data. For instance, the whole process of surface reconstruction (130 time steps) from the noisy point cloud took 7.38 seconds of CPU time.



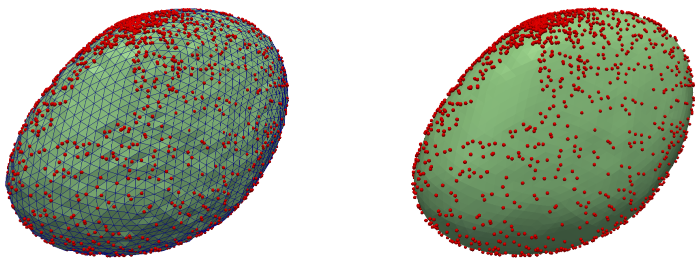
(a) Evolved surface after 80 steps of evolution

(b) Evolved surface after 95 steps of evolution

(c) Evolved surface after 105 steps of evolution

(d) Reconstructed surface after additional 25 steps of mean curvature flow.

Figure 9: Reconstruction of the testing surface from a point cloud with outlying points.



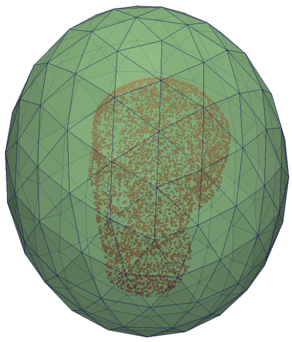
(a) Evolved surface after 105 steps of evolution displayed with its triangulation

(b) Evolved surface after additional 25 steps of smoothing

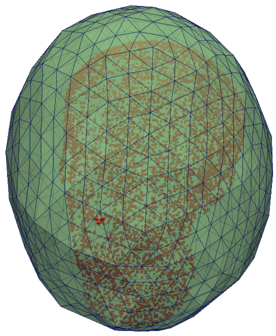
Figure 10: Reconstruction of the testing surface from a point cloud with a missing part.

In the next phase of testing we applied our model to a more complicated test object representing a bust. The used distance function was computed on voxel grid of $300 \times 300 \times 300$ voxels. In this experiment we also used the additional mesh refinement of the evolving surface. We started the evolution precess with the coarse triangulation of an ellipsoid with only 162 vertices and parameters set to $w_a = 300$, $w_d = 30$, $\tau = 0.01$ and $\omega = 100$. During the evolution we refined the mesh four times using the subdivision of each triangle into four smaller triangles. The time step τ is set to be four times smaller after each mesh refinement due to the coupling $\tau \sim h^2$, where h characterizes the length of the triangle sides. Here, we again used the asymptotically uniform redistribution of points.

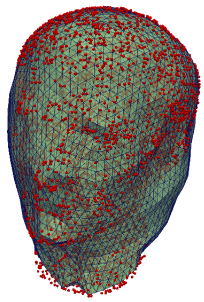
We tested the redistribution of points driven by the curvature on the point cloud sampling of a bumpy spherical surface (Figure 12). During the first 90 steps of evolution with the parameters set to $w_a = 20$, $w_d = 2$, $\tau = 0.001$ we used the asymptotically uniform redistribution in order to optimally capture the velocity field of the evolution. Then we performed the mesh refinement and smoothed the obtained surface with 5 steps of MCF. Afterwards, we continued the smoothing process with much lower magnitude to avoid undesired shrinking. At this stage we used the curvature based redistribution of points. We used function $f(\tilde{H}) = exp(1.5\,\tilde{H})$ in (2.20), where $\tilde{H}$ is the regularized curvature. In order to obtain the value of regularized curvature, we always perform 3 steps of MCF of the current surface, then we use the curvature of this smoothed surface as regularized curvature $\tilde{H}$ in the process of the tangential redistribution of points of the current surface. We use this regularization of curvature in order to stabilize the computation process.



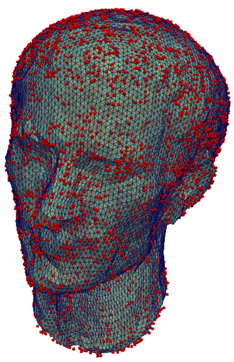
(a) Coarse initial condition with the point cloud



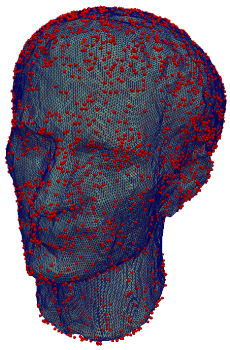
(b) Evolved surface after 10 steps of evolution and first mesh refinement



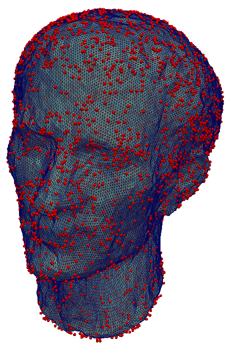
(c) Evolved surface after 45 steps of evolution and second mesh refinement



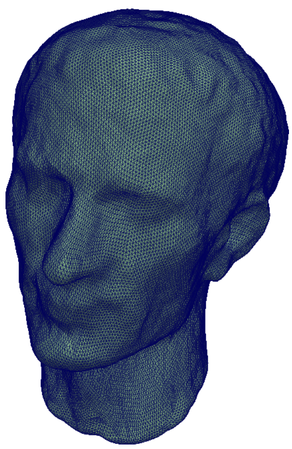
(d) Evolved surface after 220 steps of evolution and third mesh refinement



(e) Evolved surface after 240 steps of evolution and final mesh refinement

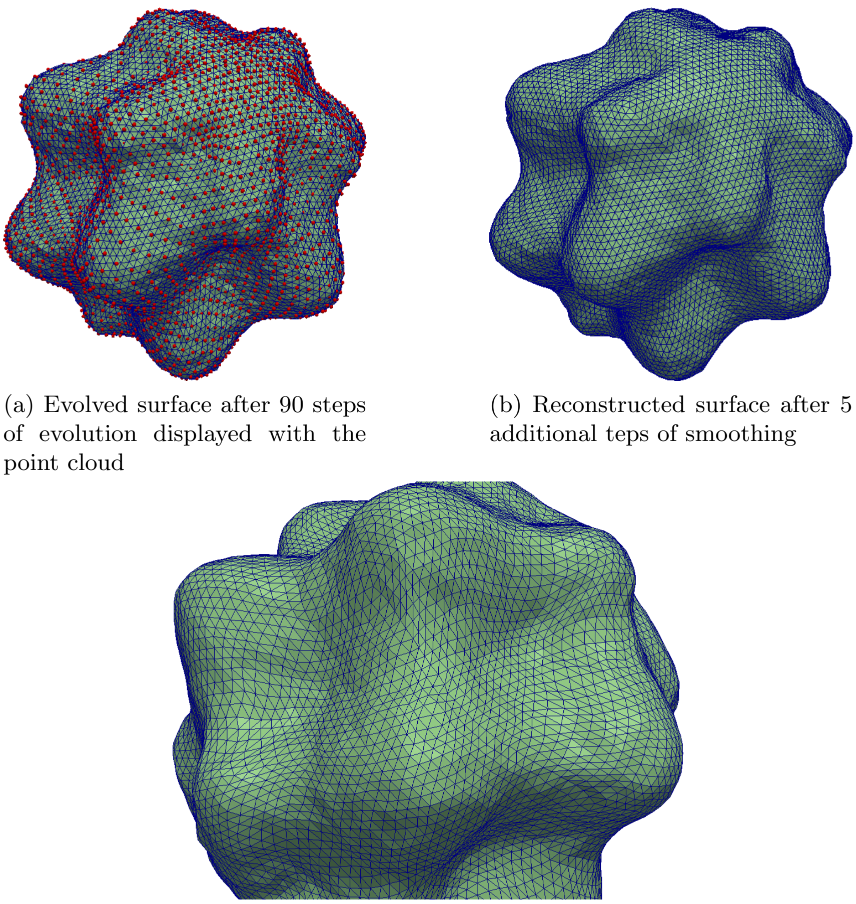


(f) Evolved surface after 245 steps of evolution



(g) Reconstruction obtained after additional 3 steps of MCF smoothing.

Figure 11: Reconstruction of a more complicated testing surface. We started the reconstruction process with coarse triangular mesh of initial ellipsoid consisting of 162 vertices. After performing 248 time steps of surface evolution with four mesh refinements we ended with the triangular approximation of desired surface consisting of 40962 vertices.



(a) Evolved surface after 90 steps of evolution displayed with the point cloud

(b) Reconstructed surface after 5 additional teps of smoothing

(c) Detail of surface after performing 10 steps of curvature based redistribution of points.

Figure 12: Reconstruction of a bumpy spherical surface with the use of curvature based redistribution of points.

6 Application to biological data

Nowadays, the advanced microscope technology allows us to obtain good quality 3D image data of early stages of embryogenesis of the model developmental organisms. Hence, various image processing algorithms are now frequently used by developmental biologists in order to explore and understand the mechanisms and dynamics of fundamental developmental events. Current challenges in biological image processing include algorithms for cell membrane shape identification, multidimensional image registration, cell nucleus tracking or segmentation.

An essential question in developmental biology is the mechanism of cell differentiation by which the cells with distinct fates are specified. We applied the proposed surface reconstruction method to the biological data representing an early developmental stage of a zebrafish (*danio rerio*) embryo. Using our method, we are able to reconstruct surfaces from the point clouds representing forming otic vesicle (*vesicula otica*) of the embryo during the process of cell differentiation. The point clouds were constructed based on the 3D image data obtained by a laser microscope (Figure 13). The reconstructed surfaces can help in quantitative analysis of the embryo, e.g. determining the volume of formed structures and measuring the differences between individuals developing in different conditions.

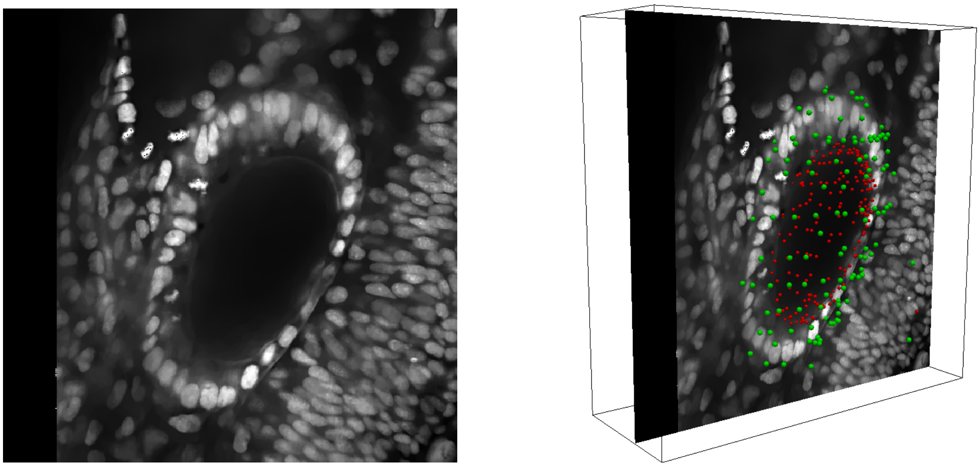


Figure 13: A slice of the 3D image data of zebrafish embryo obtained by a laser microscope on the left. On the right, displayed in 3D view together with the point cloud data used as an input for our method.

Using our method, we were able to construct a watertight approximation of the desired surfaces (Figure 14 and 15), even though the sampling density of the point cloud data was rather low and the data set contained several outlying points. After obtaining the triangulations of the surfaces we can compute the volume $\mathscr{V}$ of their interiors using the divergence theorem

$$\mathscr{V} = \iiint_V 1 \,\mathrm{d}V = \iiint_V \operatorname{div} \mathbf{v} \,\mathrm{d}V = \iint_S \mathbf{v} \cdot \mathbf{n} \,\mathrm{d}S, \tag{6.1}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the outward unit normal to surface S and $\mathbf{v}$ is a vector field with divergence equal to 1, we can use for example $\mathbf{v} = (x, 0, 0)$. Since the obtained triangulation of surface S is piecewise linear and also the vector field we chose is linear, the surface integral in (6.1) can be evaluated exactly using following formula

$$\mathscr{V} = \sum_{i=1}^{n_t} \int_{\mathcal{T}_i} x\, n_x \,\mathrm{d}S = \sum_{i=1}^{n_t} n_x \frac{A_{\mathcal{T}_i}}{3} \left(x_1 + x_2 + x_3\right), \tag{6.2}$$

where $A_{\mathcal{T}_i}$ is the area of triangle $\mathcal{T}_i$ and x_1, x_2, x_3 are the x–coordinates of its vertices.

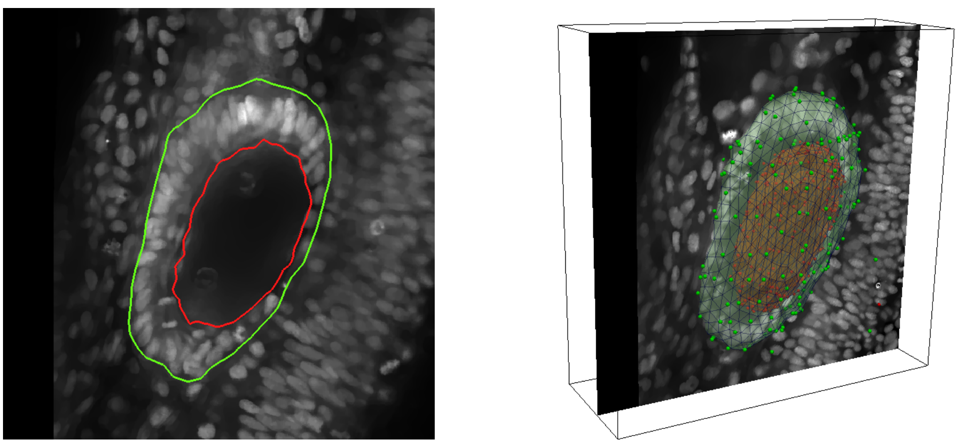


Figure 14: A slice of the reconstructed surfaces displayed together with the slice of the 3D image data on the left. 3D view of both surfaces together with corresponding point clouds on the right.

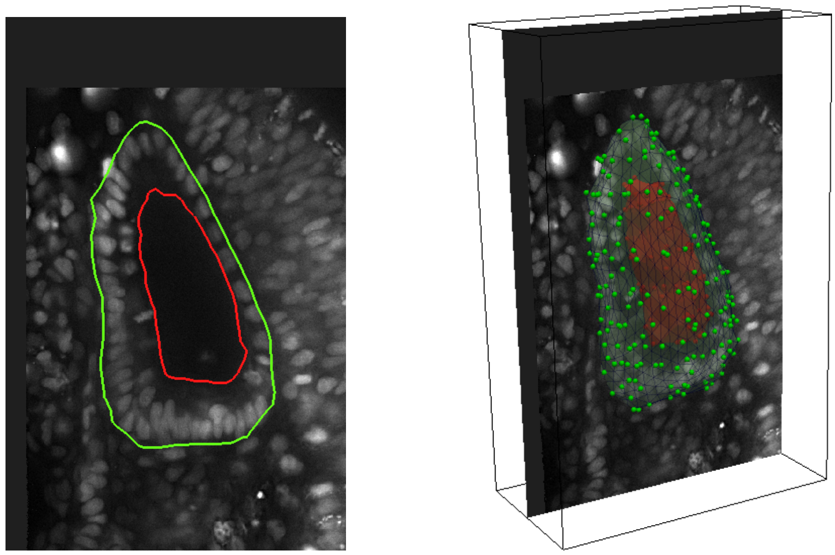


Figure 15: Slices of the reconstructed surfaces for different embryo displayed together with the slices of the 3D image data on the left. 3D view of reconstructed surfaces together with corresponding point clouds on the right.

7 Conclusion

In this paper, we have presented a method for surface reconstruction of 3D objects from representative point clouds based on Lagrangian surface evolution. After establishing the mathematical model we performed various numerical experiments in order to demonstrate the potential of using the proposed method for reconstruction of surfaces from challenging point clouds containing various types of imperfections.

However, there are still some issues left for improvement. It would be interesting to conduct further investigation on the tangential redistribution of points concerning the magnitude of the angles in individual triangles. Meshes containing triangles having an angle less than $30°$ or more than $120°$ should not be present in actual practice. We also did not consider the issue of self-intersections and topological changes during the process of evolution, but this hardship can be overcome using the ideas presented in [11].

References

[1] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, C. T. Silva: *State of the Art in Surface Reconstruction from Point Clouds,* Proceedings of Eurographics (2014).

[2] K.Mikula, M.Remešíkova, P.Sarkoci, D.Ševčovič: *Manifold evolution with tangential redistribution of points,* SIAM J. Scientific Computing, Vol. 36, No.4 (2014), pp. A1384-A1414.

[3] M. Bauer, P. Harms, P. W. Michor: *Sobolev metrics on shape space of surfaces,* J. Geom. Mech., 3 (2011), pp. 389–438.

[4] D. Sevcovic and S.Yazaki: *Evolution of plane curves with a curvature adjusted tangential velocity,* Japan J. Indust. Appl. Math., 28(3) (2011), 413-442.

[5] M.G. Crandall and P.L. Lions: *Viscosity solutions of Hamilton-Jacobi equations,* Transactions of the American Mathematical Society 277 (1983) no. 1, 1–42. MR 85g:35029

[6] Zhao, H.:*A Fast Sweeping Method for Eikonal Equations,* Mathematics of Computation 74 (250), pp. 603 627 (2004)

[7] M. Meyer, M. Desbrun, P. Schroeder, and A. H. Barr,*Discrete differential geometry operators for triangulated 2-manifolds,* in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., Springer-Verlag, Berlin, 2003, pp. 35–57.

[8] G. Dziuk and C. M. Elliott,*Finite elements on evolving surfaces,* IMA J. Numer. Anal., 27, pp. 262–292 (2007)

[9] H. A. van der Vorst, *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems,* SIAM Journal on Scientific and Statistical Computing, 13(2): pp.631–644 (1992)

[10] L. Dagum, M. Ramesh, *OpenMP: an industry standard API for shared-memory programming,* Computational Science & Engineering, IEEE, (**5**)1, pp. 46–55 (1998)

[11] A. Zaharescu, E. Boyer, R. Horaud, *Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing, and Multi-View Reconstruction.* [Research Report] RR-7136, 2009, pp.34. <inria-00438358v2>