Geodesic-based Manifold Learning for Parameterization of Triangular Meshes

Diego A. Acosta [2], Oscar E. Ruiz [1], Santiago Arroyave [1],
Roberto Ebratt [1], Carlos Cadavid [1], Juan J. Londono [1]
[1] Laboratory of CAD CAM CAE
[2] DDP Research Group
Universidad EAFIT
K. 49 7-sur-50, Medellin, 050022, Colombia
e-mail: ccadavid@eafit.edu.co

Contents

Abstract

Reverse Engineering (RE) requires representing with free forms (NURBS, Spline, Bezier) a real surface S_0 which has been point - sampled. To serve this purpose, we have implemented an algorithm that minimizes the accumulated distance between the free form and the (noisy) point sample. We use a dual-distance calculation point to / from surfaces, which discourages the forming of outliers and artifacts. This algorithm seeks a minimum in a function f that represents the fitting error, by using as tuning variable the control polyhedron for the free form. The topology (rows, colums) and geometry of the control polyhedron are determined by alternative geodesic - based dimensionality reduction methods: (a) graph - approximated geodesics (Isomap), or (b) PL orthogonal geodesic grids. We assume the existence of a triangular mesh of the point sample (a reasonable expectation in current RE). A bijective composition mapping $S_0 \subset \mathbb{R}^3 \longleftrightarrow \mathbb{R}^2$ allows to estimate a size of the control polyhedrons favorable to uniform-speed parameterizations. Our results show that orthogonal geodesic grids is a direct and intuitive parameterization method, which requires more exploration for irregular triangle meshes. Isomap gives a usable initial parameterization whenever the graph approximation of geodesics on S_0 be faithful. These initial guesses, in turn, produce efficient free form optimization processes with minimal errors. Future work is required in further exploiting the usual triangular mesh underlying the point sample for (a) enhancing the segmentation of the point set into faces, and (b) using a more accurate approximation of the geodesic distances within S_0, which would benefit its dimensionality reduction.

keywords

Computational Geometry, Parametric Surfaces, Surface Reconstruction, Reverse Engineering

Glossary

PL	Piecewise Linear
B	solid object in $\mathbb{R}^3$. $B \subset \mathbb{R}^3$ is the closure of a bounded and connected open set, whose border ∂B is a 2-dimensional manifold.
S_0	freeform parametric surface on which a Face of ∂B is mounted
$\mathbf{P}$	$\{p_0, p_1, ...\}$ unordered point sample of S_0
$S(u, v)$	parametric surface, which fits the set $\mathbf{P}$, so $S \approx S_0$
u, v	surface parameters
$N_{i,p}, N_{j,q}$	B-spline base functions $\mathbb{R} \rightarrow \mathbb{R}$,
n, m	number of control points of S in u, v directions respectively
$\mathbf{Cp}$	control polyhedron for S
k	norm degree. $\lvert(x_1, x_2, ..., x_n)\rvert_k = \sqrt[k]{\sum_{i=1}^{i=n} \lvert x_i\rvert^k}$
f	function minimized when fitting S to $\mathbf{P}$
d_i	minimum distance between the i-th point p_i of $\mathbf{P}$ and S
LM	Levenberg-Marquardt
RE	Reverse Engineering
Gr	regular, axis-aligned vertex grid in $\mathbb{R}^2$
G	graph $(\mathbf{P}, E)$ with vertex set $\mathbf{P}$ and edge set E, nearly embedded in S_0
D	square matrix in which $D(i, j) = dist(p_i, p_j)$, with *dist()* approaching the geodesic distance on S_0 between sample points p_i and p_j
T	$\{t_1, t_2, \ldots\}$ triangular mesh of triangles t_i with vertices in $\mathbf{P}$
B_{UV}	parametric rectangular connected subset of $\mathbb{R}^2$
c_G	PL geodesic curve on T.

1 Introduction

In this article, we address the following problem: **Given** a point set $\mathbf{P} = \{p_1, p_2, ..., p_n\}$, randomly sampled on a local surface S_0 of a physical part B, the **Goal** is to find a parametric free form $S(u, w)$ (approximating S_0) by minimizing the distance between S and $\mathbf{P}$.

The point sample $\mathbf{P}$ is supposed to be tight enough to capture the smallest geometrical detail in which the designer is interested. Such a point sample is called *Nyquist-compliant*. In it, the point sampling interval must be at most half times the smallest geometrical detail to be pursued [Nyq28],[Sha49].

Reverse Engineering (RE) is the process of converting (partitions of) $\mathbf{P}$ into a topologically and geometrically correct CAD model [VMC97]. RE is widely employed in many applications, such as CAD design, data visualization, virtual reality, medical imaging, movie industries, cultural heritage preservation, etc.

Curve and surface reconstruction are central to RE, in which a material piece B is available for digitization or scanning, while a suitable CAD model for the piece is yet to be found. The most common CAD solid modeling schemes are: (1) Boundary Representation, (2) Constructive Solid Geometry, (3) Enumerations, and (4) Constraint-based Models. It is not evident which scheme should be used for a particular point set. However, the mathematical difficulty of fitting CSG or Constraint-based models is prohibitive, which leaves only Boundary Representations and Enumerations for practical applications. Enumerations (voXels, octrees, etc.) are a natural choice when the sampled set is the interior of the object (e.g. a scalar field). In these cases, it is usual to sample in grid patterns, as occurs in Computer Tomograms, Magnetic Resonance, etc.

Given a body B, the representation of its boundary ∂B is called the Boundary Representation (B-Rep) of B. B-Rep is the obvious choice for surface sampling, since the surface sampled is precisely the boundary of the object, in the topological sense. However, even if the scheme chosen to model the solid is a B-Rep, there are plenty of modeling decisions which influence the goodness of the model. A central decision is the partition (segmentation) of the Shell (a 2-manifold without border), which bounds the solid, into Faces (connected 2-manifolds with possibly disconnected border). When the CAD model precedes the object, the CAD software usually avoids defining a Face which is mounted in more than one underlying parametric surface. That implies that, as an example, a Face will not contain a subset of a cylinder and other subset of a sphere. When the physical model precedes the CAD model (as in RE), there is no obvious manner to determine that a subset of the sampled points represents *one* Face of the solid model [CWQ+07] and [VS05]. This occurs because, precisely, the parametric surface carrying the Face is unknown and it needs, in turn, to be found based on the chosen point set. To break this circular argument, a human user first partitions the point set, with the support of statistical and graphical tools applied to the point sample. These tools, for example, diagnose whether an analytical form (e.g. a cylinder) fits well a particular subset of the point sample. In negative case, other analytical form, or other partition of the point sample are tried.

1.1 Interactivity in Mesh Parameterization and Segmentation

By principle, Computer Aided Geometric Design intends to reduce the need for human interaction. However, the current state of the art requires the human input to decide which computer-generated options are best suited for the application at hand. Manifold Parameterization is not an exception. However, the user interaction present when manifold parameterization fails (which is frequently the case) is invested in the complementary process of further segmenting or re-segmenting the Manifold and not directly in re-making of the parameterization. Manifold Segmentation (see survey in [Sha08]) partitions the main triangular mesh in submeshes. Partition criteria are: (1) tangent plane discontinuities (dihedral angle), (2) weakness in the triangulation graph, among others. In this article, a desirable segmentation goal is the quasi-developability of the sub-meshes, therefore facilitating their parameterization. The Spectrum of the mesh connectivity graph (Graph Laplacian methods [LZ07]) tends to split the mesh in weakly connected components, therefore isolating the limbs, head, tail, etc. (e.g. for an animal shape). However, the classification of which eigenvalues of the Graph Laplacian contain the information for the partition is still an open problem, and definitely requires human interaction. Several Refs. ([JLCW06, LLS+05, Sha08]) present mesh segmentation software, which allows an intuitive interaction from the user, for the purpose of Mesh Segmentation for easy parameterization (among other criteria). The material present in this article corresponds to Fig. 1. Our Reverse Eng.system receives a triangulation T to segment and parameterize ([Oro14]). Direct user interaction or heuristics (Box 1) are used for either Geometry-driven segmentation (e.g. Dihedral Angle, Box 2) or Topology-driven (e.g. Laplacian of Graph, Box 3) Segmentation to produce an initial hint of the set of sub-meshes T_i ($i = 1, 2, ..$). The sub-meshes T_i are parameterized (e.g. Geodesics, Box 4, this article). The user or numerical criteria approve or reject the Segmentation and Parameterization (Box 5). In the second case, user or automatic heuristics are added, and the cycle is repeated.

As a primary precondition for Manifold Parameterization, Manifold Segmentation requires an interactive set-up. Fig. 15 shows the segmentation of the Frog data set (i.e. input) fed to our geodesic - based Manifold Parameterization algorithms, as well as the Manifold Parameterizations (i.e. output).

1.2 Article Focus

In this article we concentrate on the optimized fitting of the surface to the sample point set, leaving aside the partition of large point samples into subsets which are specific for the faces of the B-Rep to construct. In this article, section 2 reviews the relevant existing literature. Section 3 discusses the theoretical background of the implemented methodology. Section 4 presents the results of the implementation. Section 5 concludes the article and suggests future work domains.

2 Literature Review

2.1 Data Input

The physical process that samples B has direct influence on the geometric reconstruction algorithms applied to the collected data: (1) Medical Imaging produces VoXel data, which can be either directly processed in 3D to produce Exhaustive Enumerations and Octrees, or sliced by cross cuts to produce 2D contours,

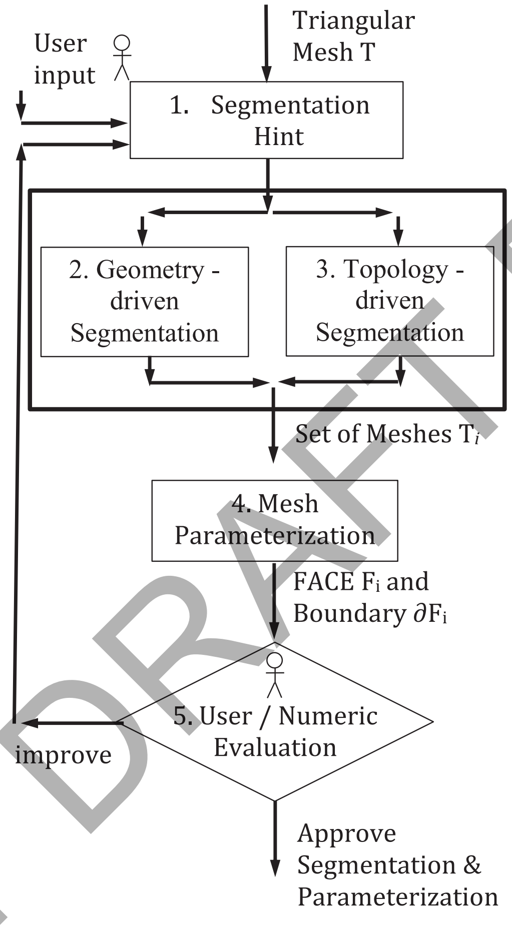
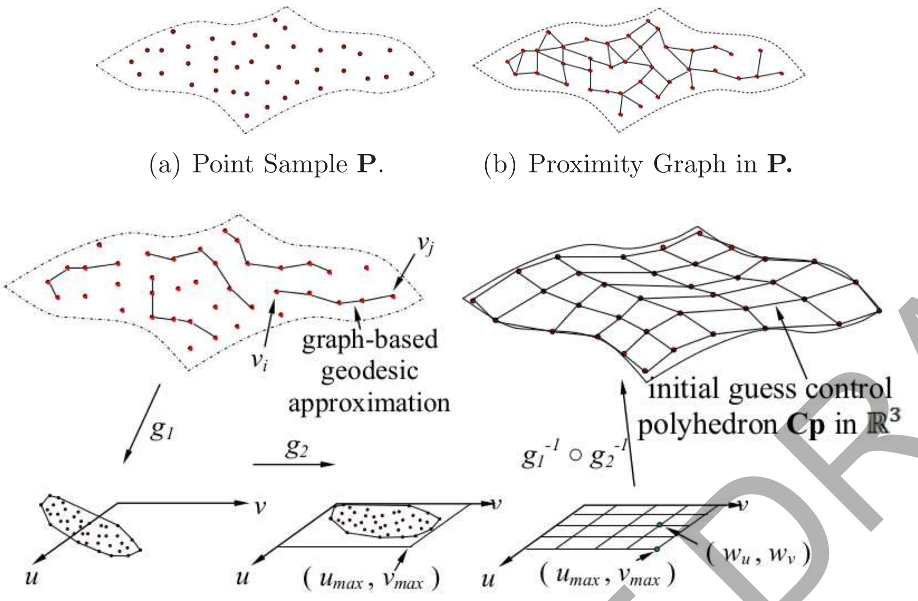


Figure 1: Interactive Workflow for Segmentation, Learning and Parameterization of Manifolds.


v_j
v_i
graph-based geodesic approximation
initial guess control polyhedron **Cp** in $\mathbb{R}^3$
g_1
g_2
$g_1^{-1} \circ g_2^{-1}$
v
u
(u_{max} , v_{max})
(w_u , w_v)

(a) Point Sample **P**. (b) Proximity Graph in **P.**

(c) Isometries to/from $\mathbb{R}^2$.

Figure 2: Manifold Learning applied to initially guessing the control polyhedron **Cp**.

which can be lofted to produce triangular meshes [KSS99, RCG+05]. This sampled reaches internal cavities in natural manner. (2) Optical Scanning produces point sets in $\mathbb{R}^3$, sampled on the visible part of the object surfaces [BK05]. The 3D points are usually equiped with triangle connectivity. (3) Coordinate Measuring Machines (CMMs) produce point sets or sequences sampled by direct contact on the reachable surfaces of B. This data is structured in polylines or contours in the case of point sequences. If randomly sampled, the natural structure added to the points is a triangular mesh.

Although there are hundreds of variations, the subdivision above covers the basic taxonomy of 3D scaning methods. It must be noticed that all methods imply a level of stochastic noise. Therefores, unless said otherwise, we asume the point sample as noisy and random. Likewise, we asume that the point sample is Nyquist/Shannon-compliant, so the Level of Detail to record is consistent with the sample interval and the level of noise admissible.

LSM (Least Squares) or non-iterative methods seek to solve the surface fitting problem in one iteration, by solving an over-sampled $Ax = B$ system. They need a base surface for projecting the data points and to assign for each data an initial parameter value (u, v). This process is also called parametrization [PT01].

Given a set of points Q sampled on the surface of an object, and a freeform weight system expressed by matrix R, the goal of Least Square Minimization methods (e.g [BAR11]) is to find the control polyhedron P, which satisfies the form $R.P = Q$ and minimizes f (Glossary). The Least Square solution P for $R.P = Q$ satisfies also $R^T.R.P = R^T.Q$, with the basis R matrix including terms related to the knot vector, weight degree, parameter values, etc. [BAR11] reports that the solution of this problem has time complexity $O(N^2.M)$ (N: number of control points, M: number of sampled points) and storage complexity $O(M.N + N^2)$. By exploiting sparsity in the $R^T.R$ matrix, the storage complexity may be lowered to $O(N)$. The LSM methods encompass a large set of individual tasks, such as: knot insertion, minimizes distance between control polyhedron and surface, penalize the displacement of control points and, in general, require considerable approximations to solve $R^T.R.P = R^T.Q$ (Gauss-Seidel, Cholesky, etc.). In general, LSM methods require computing effort larger than iterative methods. The later ones minimize the f functional by iteratively approaching the optimal tuning variables, and are faster than the LSM ones.

[ZLZY11] uses a recursive decomposition in which large patches are built based on a recursive construction of smaller ones. At the actual patch reconstruction, the algorithm assumes to have available curves, contained in the sought surface, which express the vectors which are normal to the four curve segments which limit the patch. With this information, the four curve segments are found by indirect integration of such normal vectors. In our work, we do not count with the normal curve information. Instead, we start with the basic point sample.

[LAH+09] presents CAD model modification based on the deformation results of FEA methods. In this context, it is necessary to fit parametric surfaces to triangular meshes, already segmented. The fitting is focused on surfaces which are slight deformations of planes. The bending - energy is used to calculate the parametric surfaces and small features are temporarily ignored to accelerate the FEA calculations. In our work, by contrast, we address surfaces which cannot be bijectively projected onto a plane. We use dimensional reduction (Manifold Learning) to get a rough approximation of such complex surfaces.

[ZSGS04] proposes the use of the IsoMap method (isometric feature mapping) to find a set of points in a low-dimensional space that conserves similar distances to the given set of points in a high-dimensional space and provide an initial parameterization of the surface. The strech from the paremeterization is minimized by an iterative method. If the user specified strech criterion is not fulfilled, a spectral clustering is performed in order to partition the surface into charts and meet the criterion. We use the IsoMap method to estimate the control polyhedrons for the parametrization but we do not focus on any mesh partitions as done in [ZSGS04]'s work.

[Aza04] discusses the fitting of a parametric surface to a surface point sample, whenever the four parametric curves which bound the surface are known. The algorithm also assumes that the point cloud can be always projected onto the Dynamic Base Surface DBS formed by the four mentioned curves. This basically is the same assumption of [LAH$^+$09]. [KWLM12] uses DBS enhanced with mirror and other symmetries to economically fit surfaces to symmetric point sets. Our algorithm requires that the surface S be quasi - developable, but it does not require S to allow a 1-1 projection onto some plane.

[PMK12] reports a spatial recursive octree subdivision in a 3D point set sampled on a smooth closed (manifold) surface. Each octant of the octree is subdivided if the surface fit to the points inside it does not fulfill a minimal value for a functional. This functional expresses the accumulated distance of the point set to the (radial basis) surface fit inside the octant. The method uses level sets methods and therefore, the synthesized function solves a differential equation in the given domain. This Ref. addresses the issue of point set segmentation and does not address the fitting of free form parametric surfaces.

In the domain of curve reconstruction, [RCAA12] reports the usage of PCA in local neighborhoods to construct an initial approximation of the sought curve. Notice that the point sample of a curve admits a *total* order, in which it is clear if a point precedes or follows another sample point. In a surface, a *total* order is not natural, augmenting the difficulty of surface parameterization. On the positive side, a triangulation T of the surface point sample provides the neighborhood information, which is a necessary step in parameterizing S. In our work, triangular information is central in finding a starting parameterization of S (i.e. an initial estimation of the control polyhedron $\mathbf{Cp}$).

The usage of whole of the data points as the control polygon of the initial curve is presented in [XLM12]. It could be useful only when the sample frequency is high and when the data points are noiseless.

[LLB11] presents an evolutionary strategy that uses IsoMap for mapping $\mathbf{P}$ onto $\mathbb{R}^2$, and adds vertices to the point sample wherever the point sample is sparse. In our strategy, we do not affect the original point sample, since we take advantage of the triangular mesh T, which frequently underlies the point sample $\mathbf{P}$.

[GIPP12] reports the implementation of a genetic algorithm to fit b-spline surfaces to noisy point samples of surfaces. The algorithm first constructs a parameterization of the point sample. Then, other parameters such as knots and control points are tuned, to obtain an accurate approximation (low error) of S for the point set $\mathbf{P}$. Although the authors claim that the method handles all type of surface topologies, it must be remembered that, simply, not every surface permits a connected parameterization. The examples shown indeed correspond to surfaces which are mappings of a rectangular connected parameterization in $\mathbb{R}^2$. [GI12] presents a similar strategy to the one in [GIPP12], but replacing the genetic algorithms with Particle Swarm Optimization (PSO). In PSO, each particle is equipped with memory, close and far communications with other particles, velocity, and any other required attribute needed to evolve to an optima state, according to a social measure of goodness. Likewise, the surfaces fit by using PSO in ([GI12] are mappings of a rectangular connected subset $B_{UV} \subset \mathbb{R}^2$. Because these surfaces are self-intersecting, they give the appearance of higher complexity. Should the non-manifold condition be corrected, it would be impossible to express them as the mapping of such B_{UV} set. On the other hand, it must be remarked that swarm strategies considerably increase the overhead of computational expense, since large amounts of additional data are required per particle.

[RCK11] reports surface fitting to a point sample in the context of ultra-precision engineering. To solve the fundamental issue of an initial parameterization, a bidirectional sampling extracts a rough control polyhedron from the point sample. Then, an optimization algorithms proceeds, to balance the accuracy following the point samples in the strictest possible manner vs. the smoothness of the surface. This requirement makes sense, obviously, when the point sample itself has an extraordinary precision quality. Our work does not address such samples, and therefore they do not imposse a condition of local point vs. surface adherence, but instead a global one.

2.2 Literature Review Conclusions and Contribution of this Article

The reviewed literature indicates that the control polyhedron **Cp** is central in optimizing the fit of a freeform parametric surface to a triangle mesh or point cloud. We assume throughout that the order and the knot vector of the B-spline surfaces are fixed so they are not subject to optimization. The impact of **Cp** overweights the impact of knots, freeform degree, norm degree, etc. In particular, finding optimal **Cp** crucially depends on the quality of the initial guess for the control polyhedron **Cp**.

Because of this reason, this manuscript will explore two methods for determination of initial guesses for **Cp**: (1) Orthogonal Geodesic Grids, and (2) Geodesic based manifold Learning (Iso-map). These two approaches differ in that Isomap assumes that the geodesic distances on S_0 can be approximated by the proximity graph of the point sample $\mathbf{P}$. Orthogonal Geodesic Grids explicitly seeks orthogonal PL geodesics on the triangular mesh T and uses them for dimensional reduction.

Dimensional Reduction with Isomap and with Orthogonal Geodesic Grids are facilitated in developable manifolds. Because of this reason, it is advisable to avoid large extents of triangular surfaces to parameterize. Smaller meshes are more appropriate for dimensionality reduction and parameterization. Likewise, an excessive number of control points (large **Cp**) tends to create instability and artifacts in the surface.

3 Methodology

Consider a Nyquist-compliant [Nyq28, Sha49] noisy unordered point set $\mathbf{P} = \{p_1, p_2, ..., p_n\}$, sampled on the surface ∂B of a solid B embedded in 3D.

3.1 Minimization Problem

Let $S(u, v)$ be the best (in the statistical sense) surface fitting the point set $\mathbf{P}$. Given $p_i \in \mathbf{P}$, the point $S(u_i, v_i)$ is the closest one to p_i, belonging to S. We define $d_i = dist(p_i, S(u_i, v_i)) = d(p_i, S)$ and $f = \sum_{r=1}^{t} d_i^w$ as the accumulated distances between $\mathbf{P}$ and S. We call f as the residual and w as the order of the residual. [WPL06] uses $w = 2$. Notice that d_i must satisfy the properties of a distance function. A possible estimation for d_i would be: $d_i = ||S(u_i, v_i) - p_i||_k$, the *k-norm* with $k \in \mathbb{R}^+$ and k being the *norm degree*). This article minimizes f by tuning the $n \times m$ points in $\mathbf{Cp}$, the control polyhedron of the Parametric Freeform.

3.2 Parametric Freeform

This article addresses B-spline as the parametric freeform surface S to be fit to the point set $\mathbf{P}$. The parametric freeform has the form $S : [0, 1] \times [0, 1] \longrightarrow \mathbb{R}^3$ [WMS07] in Eq. 1. The tuning variables are the vertices $P_{i,j}$ of the control polyhedron.

$$S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{1}$$

Where:$N_{i,p}(u), N_{j,q}(v)$: b-splines basis functions defined by the knot vectors $\boldsymbol{U}, \boldsymbol{V}$ in Eqs. 2 and 3.
$P_{i,j}$: control point in i-th row and j-th column.
n, m: number of control points in u, v directions respectively.
p, q: surface degree in u and v direction respectively. Typically, degrees are $p = q = 3$.

$$\boldsymbol{U} = (\underbrace{0, ..., 0}_{p+1}, u_{p+1}, ..., u_{r-p-1}, \underbrace{1, ..., 1}_{p+1}) \tag{2}$$

$$\boldsymbol{V} = (\underbrace{0, ..., 0}_{q+1}, v_{q+1}, ..., v_{s-q-1}, \underbrace{1, ..., 1}_{q+1}) \tag{3}$$

With: $r = n + p + 1$, and $s = m + q + 1$

In this article we will consider the control polyhedron **Cp** as the set of variables to minimize f. The degrees for the base functions (p, q), the knots vectors (U, V) and the norm degree (k) will be considered as parameters of the minimization, and therefore fixed. For more information on the sensitivity of f to those parameters, see [RCAA12].

3.3 Levenberg - Marquardt (LM) Minimization

We use here the Gauss-Newton iterative method for solving non-linear optimization problems, which approximates the Hessian H by using the Jacobian J (i.e., $H = J * J^T$). Let us define $\boldsymbol{x}$ as the decision (or tuning) variables vector. $\boldsymbol{x}$ in iteration $k+1$ is a function of $\boldsymbol{x}$ in iteration k, as per Eqs. 6 where $\boldsymbol{r_k}$ is the residuals vector at iteration k.

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - H^{-1} * J \tag{4}$$

$$H \approx J * J^T \tag{5}$$

$$\Rightarrow \boldsymbol{x_{k+1}} \approx \boldsymbol{x_k} - (J(\boldsymbol{x_k})^T * J(\boldsymbol{x_k}))^{-1} * J(\boldsymbol{x_k})^T * \boldsymbol{r_k} \tag{6}$$

If f is not strictly convex, J might be singular at some iterations, causing the algorithm to diverge. This problem can be solved by using the Levenberg-Marquardt (LM) Method [Lev44, Mar63]:

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} - (J(\boldsymbol{x_k})^T * J(\boldsymbol{x_k}) + \mu(k) * I)^{-1} * J(\boldsymbol{x_k})^T * \boldsymbol{r_k} \tag{7}$$

where $\mu(k) \geq 0$ is the LM (damping) parameter and I is the identity matrix.

3.4 Initial Estimation of Control Polyhedron

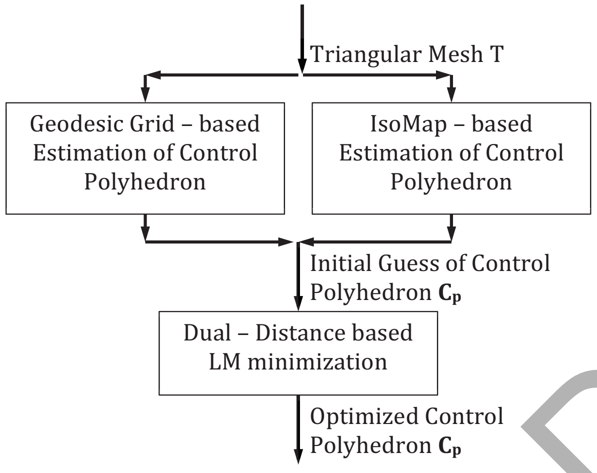

Triangular Mesh T
Geodesic Grid – based Estimation of Control Polyhedron
IsoMap – based Estimation of Control Polyhedron
Initial Guess of Control Polyhedron **Cp**
Dual – Distance based LM minimization
Optimized Control Polyhedron **Cp**

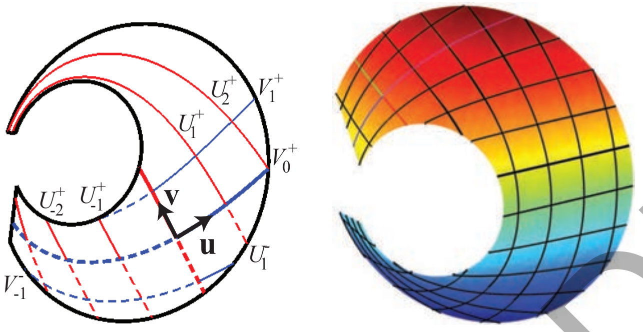
Figure 3: Geodesic Grid vs. IsoMap - based Initial Guess for Control Polyhedron **Cp**.

The optimization process described in section 3.1 requires an initial guess of the control polyhedron **Cp**. As with any optimization process, the quality of the initial guess is critical for the convergence to an optimal point. For the estimation of **Cp**, we require a bijective mapping $S_0 \leftrightarrow \mathbb{R}^3$. For this purpose, we will present two alternatives (Fig. 3): (a) creation of an 2D geodesic grid on T, (b) dimensionality reduction from $T \subset \mathbb{R}^3$ to $\mathbb{R}^2$.

3.4.1 Orthogonal Geodesic Grids for Initial Guess of Control Polyhedron Cp.

Orthogonal PL Geodesics

In a fully developable surface S, parallel geodesic curves $[c_1(u), c_2(u), ...]$ remain always at the same geodesic distance (on S) from each other (Fig. 4). Likewise, parallel geodesic curves $[c_1(w), c_2(w), ...]$ perpendicular to the $c_i(u)$ ones will remain perpendicular to them in all the surface S. Fig. 5 shows that the intersections between the $c_i(u)$ and the $c_j(w)$ geodesic curve families form a vertex grid. For developable surfaces (Fig. 4), this geodesic grid provides an exact procedure for an iso-metric flattening of the surface and it is a high quality Control Polyhedron for the free form smoothing of T. In this article, we consider quasi - developable surfaces. They are not strictly developable, but they allow us to build a geodesic grid in which the $c_i(u)$ (and the $c_j(w)$) geodesics will not cross among themselves. Fig. 4(a) shows the $c_i(u)$ (blue) and $c_j(w)$ (red) geodesics on a cone.



(a) Partial Grid. (b) Full Grid.

Figure 4: Grid of Orthogonal Geodesic Curves on a Developable Surface (cone).

To parameterize T, we seek a flattening function $f : T \rightarrow \mathbb{R}^2$ as follows: $f(p) = (u, w)$ with (u, w) being the coordinates of $p \in T$ under a grid of geodesic curves $c_i(u)$ and $c_j(w)$ on T (Fig. 5). We seek that the geodesic curves $c_i(u)$ and $c_j(w)$ be orthogonal to each other on T, and the curves $c_i(u)$ be parallel to each other on T (likewise for $c_j(w)$ curves). The $c_i(u)$ should not cross each other, and each $c_i(u)$ should intersect each $c_j(w)$ in perpendicular manner. T being developable is a precondition for this ideal situation. However, we aim to quasi - developable T. This expectation (also present in ML) is a reasonable one in Reverse Engineering by applying Manifold Segmentation. By using this mapping f, a family of (u, w) parameterizations of T is reachable, which allows us to develop a trimmed surface representing T (an important landmark in Reverse Engineering).

Geodesics on Triangular Meshes

Fig. 6 displays our approximation of a PL geodesic curve c_G embedded in a triangular mesh T. The World Coordinate System is $S_w = [X_w, Y_w, Z_w, O_w]$. A geodesic curve c_G on a surface S_0 has a second derivative with respect to its curve length s (acceleration $\partial^2 c_G / \partial s^2$) which is always normal to the surface S_0. Eqs. 8 describe the transition of a PL geodesic curve c_G whose velocity is v_1 at triangle

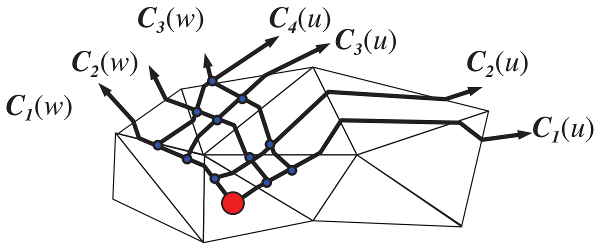
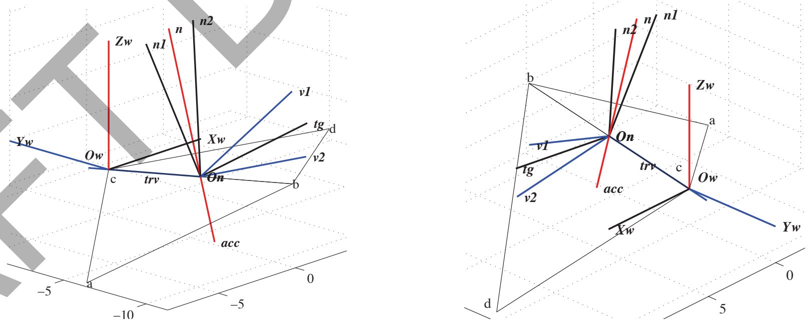


Figure 5: Origin of a Geodesic Curve Grid on a Triangular 2-Manifold T.

$t_1 = [a, b, c]$, to a neighboring triangle $t_2 = [b, d, c]$ that hosts the next geodesic segment of c_G, with a new velocity v_2. The geodesic c_G intersects edge $\overline{bc}$ (common to triangles t_1 and t_2) at O_n. Triangles $t_1 = [a, b, c]$ and $t_2 = [b, d, c]$ have normal vectors n_1 and n_2, respectively. The normal vector $\hat{\mathbf{n}}$ at the point of transition O_n is approximated as the average of n_1 and n_2. The acceleration vector of the geodesic c_G at $O_n \in \overline{bc}$ is $v_{acc} = v_2 - v_1$ and it must be parallel to n for a geodesic curve. An orthogonal, right handed (i.e. Special Orthogonal $SO(3)$) coordinate system is defined at O_n as $S_n = [v_{tg}, v_{trv}, \hat{\mathbf{n}}, O_n]$, with v_{tg} and $\hat{\mathbf{n}}$ being tangent and perpendicular to the geodesic c_G at O_n, respectively. The transverse vector v_{trv} is coincident with the edge $\overline{bc}$ and $v_{tg} \times v_{trv} = \hat{\mathbf{n}}$. The entry vector v_1 solved in the S_n system has coordinates $[\alpha, \beta, \gamma]$, while the exit vector v_2 will have coordinates $[\alpha, -\beta, \gamma]$ in S_n. Finding v_2 enables to track the geodesic curve to the next triangle t_3 (not shown), incident in this example to edge $\overline{bd}$.



(a) Upstream View of Geodesic Flow.

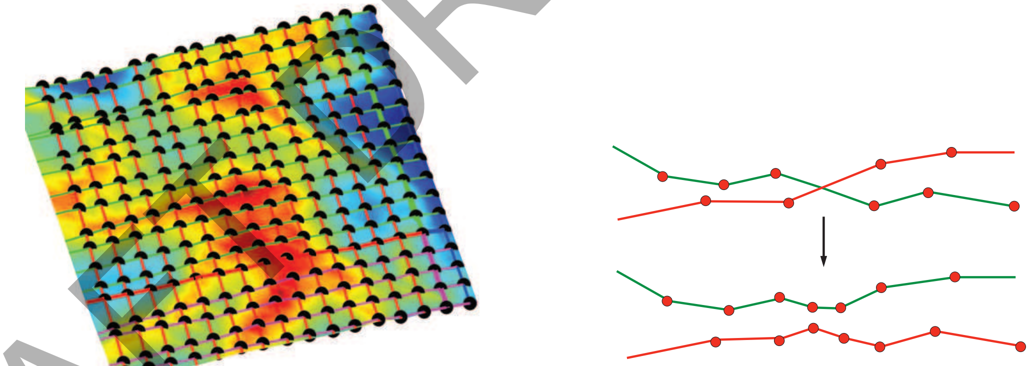
(b) Downstream View of Geodesic Flow

Figure 6: Geodesic Continuation between neighboring triangles.

$$
\begin{aligned}
\hat{\mathbf{n}} &= (n_1 + n_2)/||n_1 + n_2|| \\
v_{tg} &= (n_2 - n_1)/||n_2 - n_1|| \\
v_{trv} &= \hat{\mathbf{n}} \times v_{tg} \\
\alpha &= v_1 \bullet v_{tg} \\
\beta &= v_1 \bullet \hat{\mathbf{n}} \\
\gamma &= v_1 \bullet v_{trv} \\
v_2 &= \alpha * v_{tg} - \beta * \hat{\mathbf{n}} + \gamma * v_{trv} \\
v_{acc} &= (v_2 - v_1) = k * \hat{\mathbf{n}} \\
\hat{\mathbf{n}} \bullet v_{tg} &= 0
\end{aligned}
\qquad (8)
$$

Converging Geodesics in Non-Developable Surfaces

When surface S_0 (or in the discrete case, triangular mesh T) is not developable, initially parallel geodesics $c_i(u)$ will eventually intersect each other (Fig. 7(a)). In mild cases (quasi - developable surfaces) we break the intersecting curves (at the intersection, Fig. 7(b)) and swap the pieces, therefore producing non-intersecting curves that lose the geodesic property only at a finite number of points.



(a) Defective Geodesic Grid.

(b) Correction for defective (crossing) Geodesics.

Figure 7: Defect and Repair of Geodesic Grid on Quasi-developable T.

3.4.2 Manifold Learning for Initial Guess of Control Polyhedron Cp.

Manifold Learning, also called *nonlinear dimensional reduction* [TDSL00, CC01], pursuits the goal of mapping data originally lying on (or nearly on) an unknown manifold embedded in a high dimensional euclidean space, into a low dimensional euclidean space, while preserving some desired characteristics. The Manifold Learning community has developed a variety of methods that suit different situations. IsoMap, Locally Linear Embedding, Laplacian Eigenmaps, Semidefinite Embedding are among the more prominent ones. IsoMap, the oldest of all, assumes that the data lie on (or nearly on) the image under an unknown isometric embedding of an unknown region in d-dimensional euclidean space. Roughly speaking, the idea of the method is to first compute the distance matrix D between any two data points as the length of a shortest polygonal path, within S_0, connecting them. This path has only data points in $\mathbf{P}$ as vertices. Using the matrix D one produces a set of points in d-dimensional euclidean space whose mutual euclidean distances are also given by D.

Here is a very broad view of how the method discussed in the present article proceeds:

1. A point set $\mathbf{P}$ sampled on a manifold of dimension 2 (i.e. a surface) S_0 embedded in $\mathbb{R}^3$ is given.

2. S_0 is assumed to be *developable*, i.e. there exists a bounded, connected open set $A \subset \mathbb{R}^2$ and a one to one C^1 isometric function $g_1 : A \to \mathbb{R}^3$ such that $S_0 = g_1(A)$. The fact that g_1 is isometric means that the length of $g_1(C)$ is the same as that of C for each curve C in A.

3. A and a g_1 as in the previous step are found [TDSL00, CC01].

4. A is transformed by an appropriate rigid motion $g_2 : \mathbb{R}^2 \to \mathbb{R}^2$ which achieves a minimal 2D bounding box for $A' = g_2(A)$.

5. An $n \times m$ regular grid Gr in $\mathbb{R}^2$ approximately enclosing A', is created.

6. $g_1^{-1}(g_2^{-1}(Gr))$ is taken as an initial guess for the control polyhedron Cp of S (the approximation of S_0).

Fig. 2 presents our approach of using of Manifold Learning for the synthesis of an initial guess for $\mathbf{Cp}$, the control polyhedron for the sough surface S.

The process is as follows:

1. Start with the point set $\mathbf{P}$ sampled on $S_0 \subset \mathbb{R}^3$ (Fig. 2(a)).
2. Build a graph $G = (\mathbf{P}, E)$ representing neighborhood information within $\mathbf{P}$ (Fig. 2(b)). There are two alternatives:
   (a) Build the edge set E with the nodes satisfying the k-nearest criterium within $\mathbf{P}$: the edge $\overline{p_i, p_j} \in E$ if p_i is one of the k points of $\mathbf{P}$ nearest to p_j.
   (b) Use $T = \{t_1, t_2, \ldots\}$ the available triangular mesh of $\mathbf{P}$. The edge $\overline{p_i, p_j} \in E$ if a triangle in T contains p_i and p_j.
3. Interpret each $e \in E$ as a chord of S_0 (which is true if the sample $\mathbf{P}$ is Nyquist - compliant for S_0).
4. Approximate the geodesic curve between any two vertices p_i and p_j in $\mathbf{P}$ as the shortest path formed with edges e of G, joining p_i and p_j (Fig. 2(c)).
5. Form the square matrix D whose entry $D(i, j)$ is the geodesic distance on S_0 between vertices p_i and p_j in $\mathbf{P}$.
6. Calculate, by using D, a bijective isometric map $g_1 : \mathbf{P} \to \mathbb{R}^2$ (Fig. 2(c)).
7. Rotate and translate (within $\mathbb{R}^2$) the set $g_1(\mathbf{P}) \subset \mathbb{R}^2$ by using a rigid transformation g_2 which minimizes the 2D bounding box B_{UV} containing the set $g_2(g_1(\mathbf{P}))$.
8. Sample the bounding box B_{UV} with a regular, equispaced grid Gr, of size $n \times m$.
9. Map $Gr \subset \mathbb{R}^2$ back to $\mathbb{R}^3$ by using $h = g_1^{-1} \circ g_2^{-1}$.
10. $h(Gr)$ is an initial guess for the control polyhedron Cp of S.

As we devised this application of Manifold Learning to our problem, we initially tried alternative 2-a above. However, this alternative produced a poor initial guess for Cp, which in turn produced a poor S approximating S_0 (Fig. 13(c)). Because of this reason, we replaced the k-nearest citerium for forming E by the connectivity of the triangle mesh T (alternative 2-b above) which is reasonably available in RE). We have not found such alternative formulated in the relevant literature. Fig. 13(e) displays the improved result for the initial guess CP by using alternative 2-b.

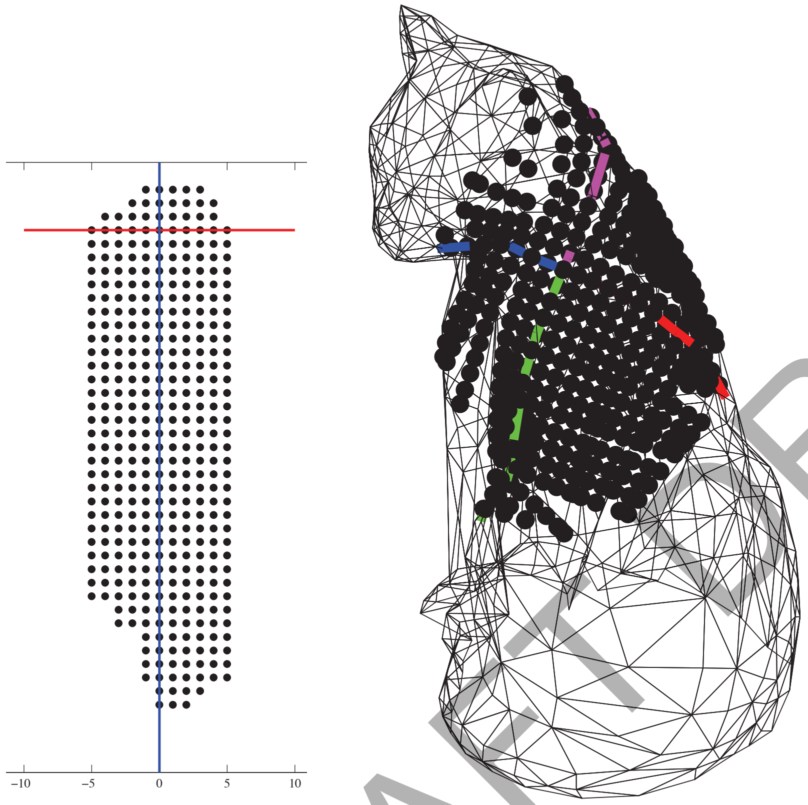
The $g_2()\mathbb{R}^2 \to \mathbb{R}^2$ rigid rotation function helps to define a U-V parameterization such that the initial guess $h(Gr)$ for the control polyhedron Cp is approximately aligned with the surface S. Figs. 11(a) and 11(c) show that the iso-parametric lines are slanted when only $g_1()$ is used. When function $g_2()$ is used, the iso-parametrics are aligned as in Figs. 11(b) and 11(d). Figs. 12(a) and 12(c) display the initial guesses of the control polyhedron **Cp**, calculated with the help of $g_2()$. Figs. 12(b) and 12(d) show the final control polyhedra and surfaces found with the optimization process.

Notice that almost all surfaces are not developable. However, ∂B can be segmented into small parts (Faces, mounted on surfaces S_0) which are close to be developable, because ∂B is 2-manifold.

4 Results

4.1 Initial Guess with Geodesic Grids

Fig. 4 shows early and finished status of the geodesic grid for a developable surface (e.g. cone), where an orthogonal geodesic mesh is possible. Fig. 7(a) shows a defective grid, resulting from a non-developable manifold. In this case, geodesic curves $c_i(u)$ (or curves $c_j(w)$) intersect. In such a case, we use a heuristic remedy to force the separation of the $c_i(u)$ and $c_j(w)$ by re-defining them as per Fig. 7(b).

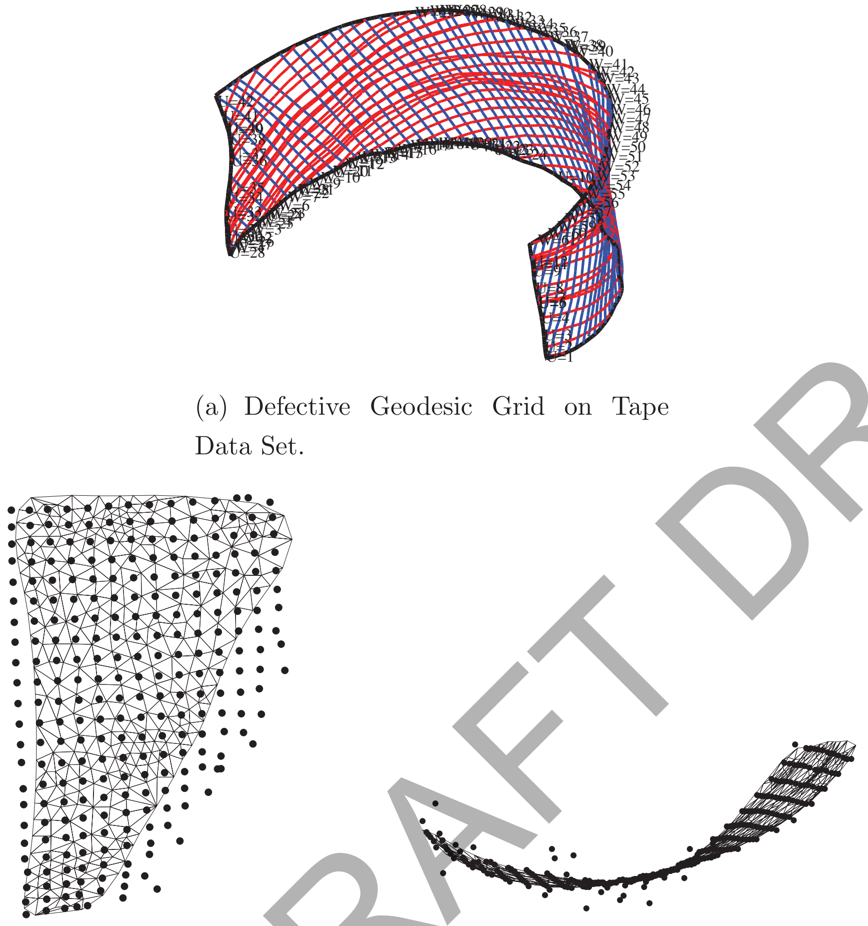


(a) Grid on Parametric Space.

(b) Geodesic Grid on Cat Triangulation.

Figure 8: Highly non-developable (Cat) Data Set.

Fig. 8 shows an attempt for parameterization of mesh T (Cat data set) with orhtogonal geodesic grids. It is evident that Orthogonal Geodesics cannot attack the problem of non-developable data sets.



(a) Defective Geodesic Grid on Tape Data Set.

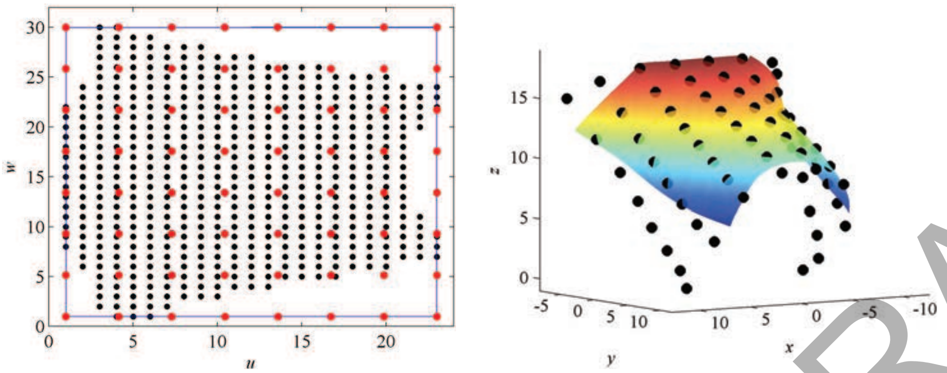
(b) Defective Geodesic Grid on Trimmed Tape Data Set.

(c) Defective Geodesic Grid on Trimmed Tape Data Set.

Figure 9: Defective Geodesic Grid.

Fig. 9(a) illustrates that the Tape data set causes the $c_i(u)$ (or the $c_j(w)$) geodesic curves to intersect each other. This local inconvenience can be corrected by disentangling the curves as per Fig. 7(b). The run for Tape data set also presents the need to rotate either (a) the geodesic grid on $T \subset \mathbb{R}^3$ or (b) the lower dimension set on $\mathbb{R}^2$ by using a rigid transformation $g_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ (Fig. 2(c)).

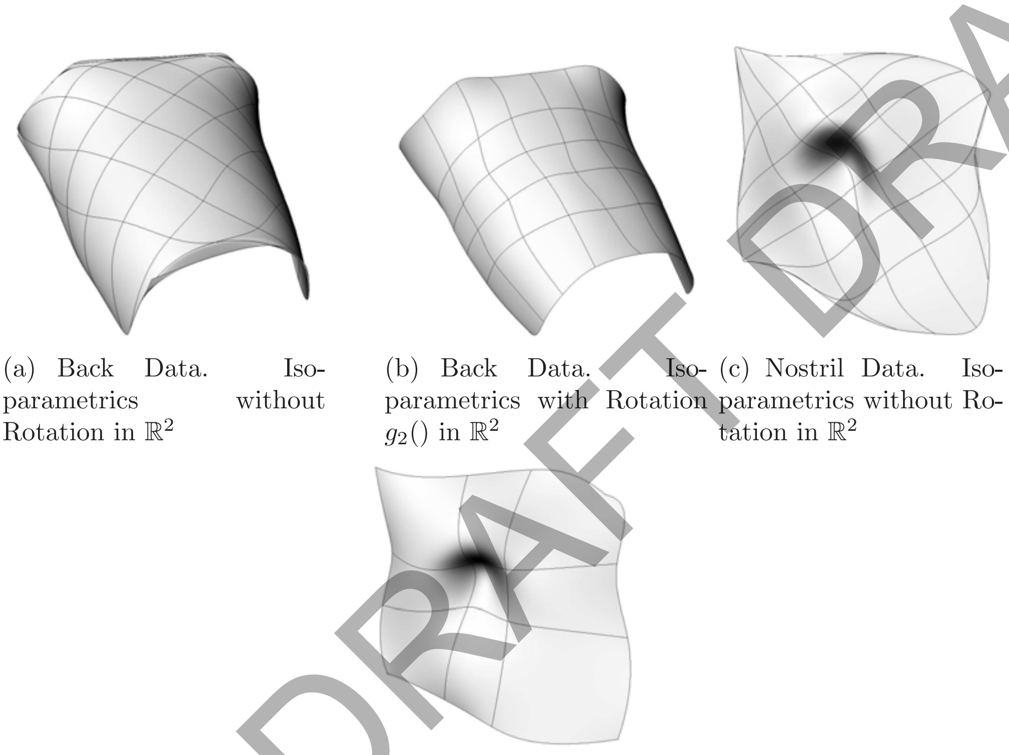
In contrast, when the T Tape data set is trimmed (Figs. 9(b) and 9(c)) the geodesic grid becomes insufficient to ensure a consistent back mapping $g_1^{-1} \circ g_2^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ (Fig. 2(c)) , therefore producing erratic portions of the control polyhedron **Cp**. This behavior is evident also in Fig. 10 applied on the *Frog Back* data set.



(a) 2D Grid from Geodesic Grid.

(b) 3D Back Map from 2D Grid.

Figure 10: *Frog Back* Data Set. Parameterization and Outliers.



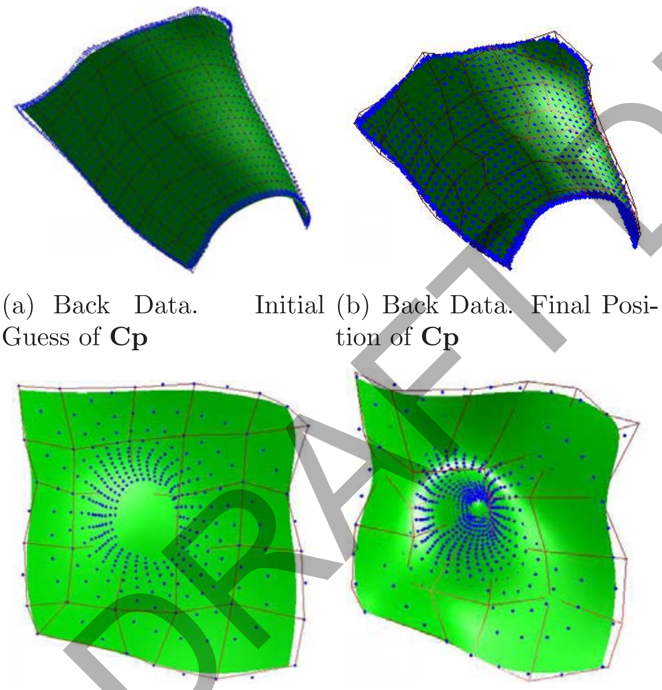
(a) Back Data. Iso-parametrics without Rotation in $\mathbb{R}^2$

(b) Back Data. Iso-parametrics with Rotation $g_2()$ in $\mathbb{R}^2$

(c) Nostril Data. Iso-parametrics without Rotation in $\mathbb{R}^2$

(d) Nostril Data. Iso-parametrics with Rotation $g_2()$ in $\mathbb{R}^2$

Figure 11: Iso-parametric curves without and with the use of a minimized bounding box in $\mathbb{R}^2$.



(a) Back Data. Initial Guess of **Cp**

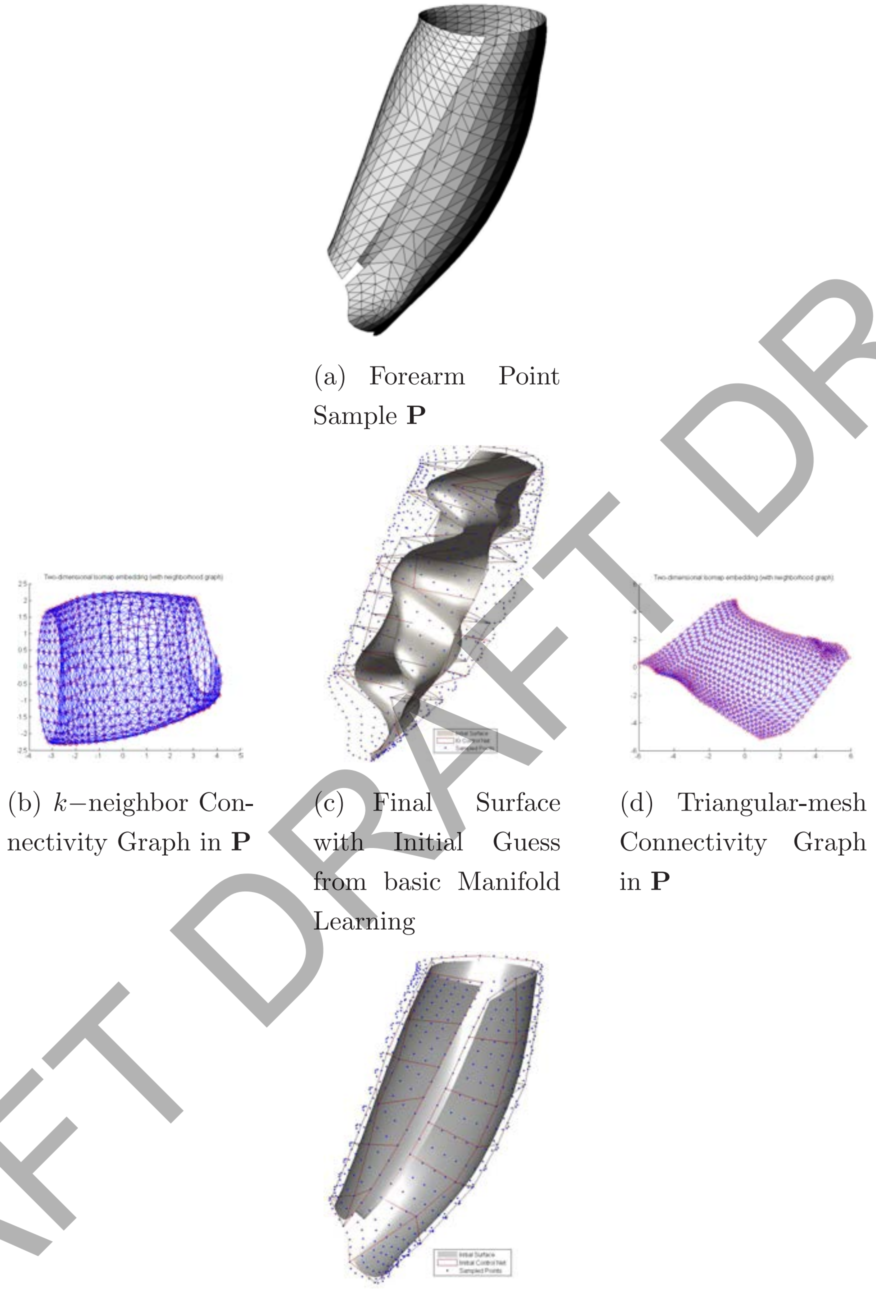
(b) Back Data. Final Position of **Cp**

(c) Nostril Data. Initial Guess of **Cp**

(d) Nostril Data. Final Position of **Cp**

Figure 12: Initial Guess and Final Status of Control Polyhedron **Cp**

4.2 Initial Guess with Manifold Learning


Two-dimensional Isomap embedding (with neighborhood graph)
Initial Surface
Initial Control Net
Sampled Points
Two-dimensional Isomap embedding (with neighborhood graph)
Initial Surface
Initial Control Net
Sampled Points

(a) Forearm Point Sample **P**

(b) $k-$neighbor Connectivity Graph in **P**

(c) Final Surface with Initial Guess from basic Manifold Learning

(d) Triangular-mesh Connectivity Graph in **P**

(e) Final Surface with Initial Guess from Triangle mesh - enhanced Manifold Learning

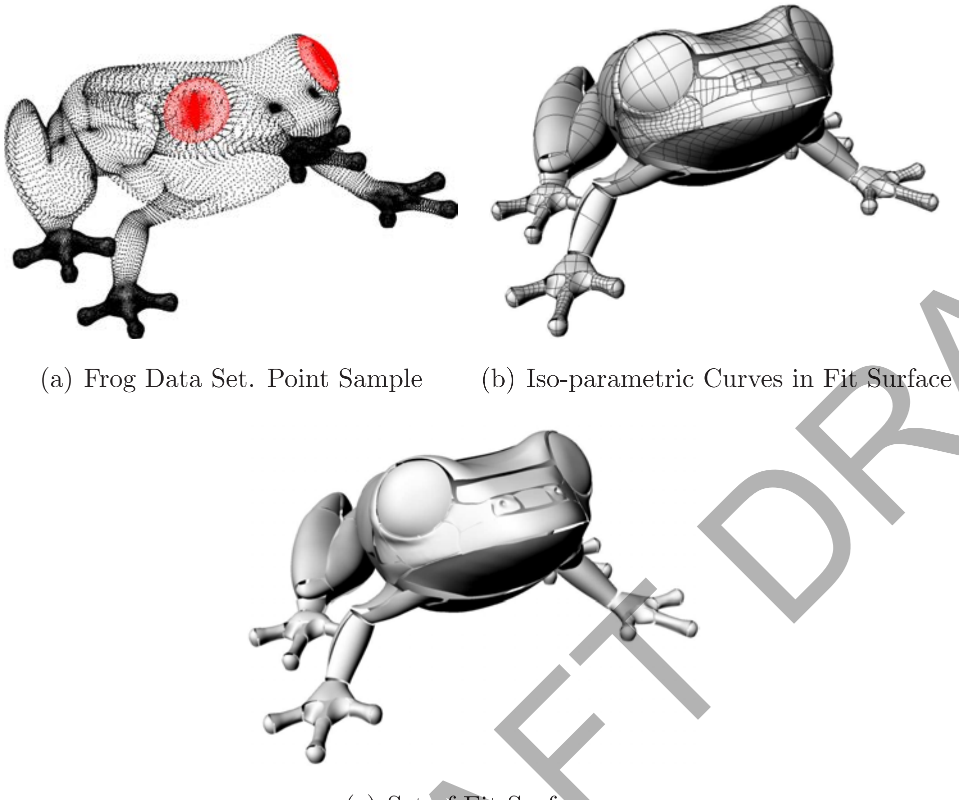
Figure 13: Optimized surface fitting by using initial guess of the control polyhedron **Cp** found with triangle-mesh based Manifold Learning.

Fig. 12 presents two examples (back and nostril) of surface S fit to the point sample $\mathbf{P}$ by using and initial guess of the control polyhedron $\mathbf{Cp}$ which is found using our variation Manifold Learning (k-nearest graph replaced by the graph derived from the triangular mesh underlying the point set $\mathbf{P}$), plus an additional rotation of $\mathbf{Cp}$ (via $g_2()$ in Figs. 11(b) and 11(d)) to avoid the problems shown in Figs. 11(a) and 11(c).

Fig. 13(a) shows an example in which the point data is sampled on a surface S_0 which is similar to a cylinder. In this case, an initial guess for its control polyhedron $\mathbf{Cp}$ obviously cannot be produced with PCA. We use instead Manifold Learning as discussed in section 3.5 to find an initial approximation for $\mathbf{Cp}$. The k-nearest graph based on $\mathbf{P}$ is displayed in Fig. 13(b) as per alternative 2-a in section 3.5. In such a case, the initial guess for $h(Gr) \approx \mathbf{Cp}$ in step 10 appears in Fig. 13(c). In contrast, alternative 2-b in section 3.5 produces the connectivity graph G in Fig. 13(d). In this alternative, the triangular mesh connectivity replaces the k-nearest one (step 2-b in section 3.5), with evident advantage. The initial guess for the control polyhedron $\mathbf{Cp}$ is approximated by $h(Gr)$ in step 10. Because in the optimization the quality of the starting point is critical in non-convex scenarios, this approximation for $\mathbf{Cp}$ enables the minimization process in such cases.

Fig. 14(a) presents the *Frog* data point sample. The segmentation of this general set in smaller point samples $\mathbf{P}$ was carried out by a human user, as the scope of our article does not include point set segmentation. Fig. 14(b) shows the several results S of the optimized surface fitting for those $\mathbf{P}$ sets, emphasizing the iso-parametric lines. Fig. 14(c) displays the same free surface partition without the iso-parametric lines. The eye data sets were treated with another surface optimization algorithm, specific for analytic surfaces (in this case, ellipsoids [RAA13]).

Figs. 15(a) and 15(b) show the results of the surface fitting algorithm in upper and bottom views, respectively. We do not attempt, at this time, the blending among the surfaces.

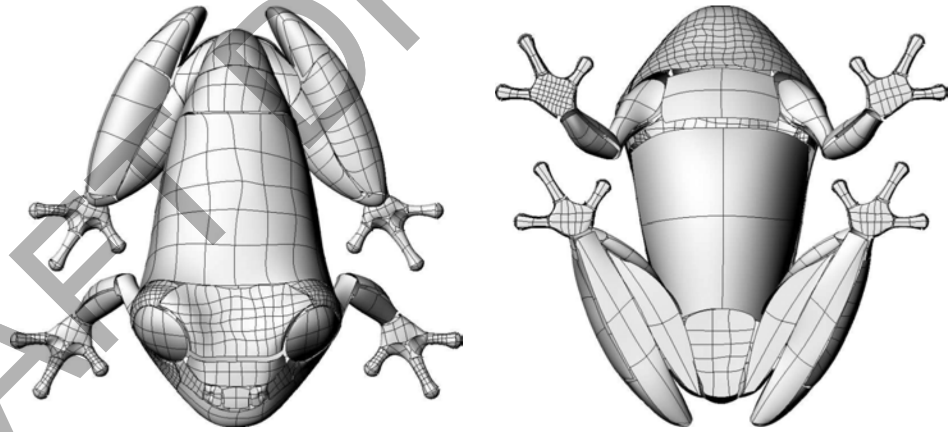


(a) Frog Data Set. Point Sample

(b) Iso-parametric Curves in Fit Surface

(c) Set of Fit Surfaces

Figure 14: Results in the 'Frog' Data Set.



(a) Iso-parametric Curves in Fit Surface. Upper view.

(b) Iso-parametric Curves in Fit Surface. Bottom View.

Figure 15: Upper and Bottom Views. Results in the 'Frog' Data Set.

5 Conclusions

This article presented the implementation of an algorithm for optimized fitting of a free form surface S to a point sample $\mathbf{P}$ of the original material surface S_0, so $S \approx S_0$. The surface S minimizes the functional $f = \sum_{r=1}^{t} d_i^w$, which is the summation of distances d_i between $p_i \in \mathbf{P}$ and S. The implemented algorithm uses both, the distance from p_i to S and the distance from S to p_i (they differ since S is finite).

In this work, the functional f is minimized by choosing the control polyhedron $\mathbf{Cp}$ of the free form S. The size $m \times n$ of $\mathbf{Cp}$ and the initial guess of $\mathbf{Cp}$ are determined by using a bijective map $g_2 \circ g_1 : \mathbf{P} \rightarrow \mathbb{R}^2$, which is quasi-isometrical. g_1 is implemented using Dimensionality Reduction by either: (1) generating a geodesic grid on T and using the intersections among $c_i(u)$ and $c_j(w)$ geodesic curve families as tentative vertices of the control polyhedron $\mathbf{Cp}$, or (2) approximating the geodesics on S_0 by using the graph implicit in triangulation T and using Isomap to find a parameterization in $\mathbb{R}^2$. The function g_2 is a rotation within $\mathbb{R}^2$, which minimizes the bounding box of $g_2(g_1(\mathbf{P})) \subset \mathbb{R}^2$.

Although the direct calculation of a geodesic grid (curve families $c_i(u)$ vs. $c_j(w)$) on T is intuitively appealing, it must be perfected for the cases of an irregular or pierced triangular mesh T. Because of this reason, we presently prefer Isomap for the calculation of an initial guess for the control polyhedron $\mathbf{Cp}$.

Isomap normally calculates an approximation (based on the k-neighbor graph) of the geodesic distance $d_{S_0}(p_i, p_j)$, within S_0, between points p_i and p_j of the sample $\mathbf{P}$. In our algorithm, the geodesic distance is based on the triangular-mesh graph among points $p_i \in \mathbf{P}$, typical from a reverse engineering 3D point scanning process. In our algorithm, these two Dimensionality Reduction methods are alternatively used to find an initial guess for the control polyhedron $\mathbf{Cp}$ (minimization tuning variables).Our algorithm chooses a dimensionality reduction function which builds an initial parameterization (hence $\mathbf{Cp}$) that agrees with the aspect ratio of the point sample. Therefore, we correct slanted parameterizations such as the ones shown in Fig. 11(a) and 11(c).

6 Future Work

Additional work is needed in these aspects: (1) With Geodesic Grids, it is necessary to handle their interruptions when there are holes or concavities in the mesh T. (2) To use the quality of the Dimensionality Reduction to guide the segmentation of the point set, by clustering in one face those triangles which represent quasi-developable portions of T.

Acknowledgments

The authors wish to thank undergraduate U. EAFIT students Juan Pablo Velasquez for the testing of discrete geodesics MATLAB (TM) code and Daniel Burgos for the end - user segmentation of data sets for this article.

References

[Aza04] P. N. Azariadis. Parameterization of clouds of unorganized points using dynamic base surfaces. *Computer-Aided Design*, 36(7):607623, 2004.

[BAR11] D. Brujic, I. Ainsworth, and M. Ristic. Fast and accurate nurbs fitting for reverse engineering. *The International Journal of Advanced Manufacturing Technology*, 54(5):691–700, 2011.

[BK05] A. Braquelaire and B. Kerautret. Reconstruction of lambertian surfaces by discrete equal height contours and regions propagation. *Image and Vision Computing*, 23(2):177–189, 2005.

[CC01] Trevor F Cox and Michael AA Cox. *Multidimensional scaling.* 2001.

[CWQ$^+$07] K.S.D. Cheng, W. Wang, H. Qin, K.Y.K. Wong, H. Yang, and Y. Liu. Design and analysis of optimization methods for subdivision surface fitting. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5):878–890, 2007.

[GI12] A. Galvez and A. Iglesias. Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points. *Information Sciences*, 192(0):174–192, 2012. ISSN 0020-0255.

[GIPP12] A. Galvez, A. Iglesias, and J. Puig-Pey. Iterative two-step genetic-algorithm-based method for efficient polynomial b-spline surface reconstruction. *Information Sciences*, 182(1):56–76, 2012. ISSN 0020-0255.

[JLCW06] Zhongping Ji, Ligang Liu, Zhonggui Chen, and Guojin Wang. Easy mesh cutting. *Computer Graphics Forum*, 25(3):283–291, 2006.

[KSS99] R. Klein, A. Schilling, and W. Straßer. Reconstruction and simplification of surfaces from contours. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 198–207. IEEE, 1999.

[KWLM12] Y. Kineri, M. Wang, H. Lin, and T. Maekawa. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. *Computer-Aided Design*, 2012.

[LAH$^+$09] B. Louhichi, N. Aifaouli, M. Hamdi, A. BenAmara, and V. Francois. An optimization-based computational method for surface fitting to update the geometric information of an existing b-rep cad model. *International Journal of CAD/CAM*, 9(1), 2009.

[Lev44] K. Levenberg. A method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[LLB11] N. Leal, E. Leal, and J. Branch. Automatic construction of nurbs surfaces from unorganized points. *Dyna*, 78(166):133–141, 2011. ISSN 0012-7353.

[LLS$^+$05] Yunjin Lee, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, and Hans-Peter Seidel. Mesh scissoring with minima rule and part salience. *Comput. Aided Geom. Des.*, 22(5):444–465, July 2005.

[LZ07] Rong Liu and Hao Zhang. Mesh segmentation via spectral embedding and contour analysis. In *EuroGraphics*, number 3, 2007.

[Mar63] D.W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[Nyq28] H. Nyquist. Certain Topics in Telegraph Transmission Theory. *American Institute of Electrical Engineers, Transactions of the*, 47(2):617–644, 1928.

[Oro14] Maria Orozco. Geometry and topology-based segmentation of 2-manifold triangular meshes in r^3. Master's thesis, Universidade de Vigo, Spain, July 2014. "Arno Formella (Supervisor), O. Ruiz and C. Osorno and C. Cadavid (advisors)".

[PMK12] C. Park, C. Min, and M. Kang. Surface reconstruction from scattered point data on octree. *Journal of the Korean Society for Industrial and Applied Mathematics*, 16(1):31–49, 2012. ISSN / CODEN - 1226-9433 / KSIAM.

[PT01] LA Piegl and W. Tiller. Parametrization for surface fitting in reverse engineering. *Computer-Aided Design*, 33(8):593–603, 2001.

[RAA13] O. Ruiz, S. Arroyave, and D. Acosta. Fitting of analytic surfaces to noisy point clouds. Technical report, Laboratorio de CAD/CAM/CAE, Universidad EAFIT, Medellin, Colombia, 2013.

[RCAA12] Oscar E. Ruiz, C. Cortes, Diego A. Acosta, and M. Aristizabal. Sensitivity analysis of optimized curve fitting to uniform-noise point samples. In *Proceedings of the 9th International Symposium on Tools and Methods of Competitive Engineering, TMCE 2012*, Karlsruhe, Germany, May 7-11 2012.

[RCG$^+$05] O.E. Ruiz, C.A. Cadavid, M. Granados, S. Pena, and E. Vásquez. 2d shape similarity as a complement for voronoi-delone methods in shape reconstruction. *Computers & Graphics*, 29(1):81–94, 2005.

[RCK11] M.J. Ren, C.F. Cheung, and L.B. Kong. A robust surface fitting and reconstruction algorithm for form characterization of ultra-precision freeform surfaces. *Measurement*, 44(10):2068 – 2077, 2011.

[Sha49] C.E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[Sha08] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.

[TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. DOI: 10.1126/science.290.5500.2319.

[VMC97] T. Varady, R.R. Martin, and J. Cox. Reverse engineering of geometric modeling. an introduction. *Computer-Aided Design*, 29(4):255–268, 1997.

[VS05] M. Vieira and K. Shimada. Surface mesh segmentation and smooth surface extraction through region growing. *Computer aided geometric design*, 22(8):771–792, 2005.

[WMS07] T. Wagner, T. Michelitsch, and A. Sacharow. On the design of optimisers for surface reconstruction. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 2195–2202. ACM, 2007.

[WPL06] W. Wang, H. Pottmann, and Y. Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics (TOG)*, 25(2):214–238, 2006.

[XLM12] Y. Xiong, G. Li, and A. Mao. Convergence analysis for b-spline geometric interpolation. *Computers & Graphics*, 2012.

[ZLZY11] S. Zhang, Z. Li, H. Zhang, and J. Yong. Multi-resolution mesh fitting by b-spline surfaces for reverse engineering. In *Computer-Aided Design and Computer Graphics (CAD/Graphics), 2011 12th International Conference on*, pages 251–257. IEEE, 2011.

[ZSGS04] Kun Zhou, John Synder, Baining Guo, and Heung-Yeung Shum. Isocharts: stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 45–54. ACM, 2004.