Geodesic Gaussian Processes for the Parametric Reconstruction of a Free-Form Surface

Enrique del Castillo*
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University, University Park, PA 16802, USA

Bianca M. Colosimo†
Dipartimento di Meccanica - Politecnico di Milano, 20133 Milano, Italy

and

Sam Tajbakhsh‡
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University, University Park, PA 16802, USA

December 11, 2013

Abstract

Reconstructing a free-form surface from 3-dimensional noisy measurements is a central problem in inspection, statistical quality control, and reverse engineering. We present a new method for the statistical reconstruction of a free-form surface patch based on 3-dimensional point cloud data. The surface is represented parametrically, with each of the three Cartesian coordinates (x, y, z) a function of surface coordinates (u, v), a model form compatible with computer-aided-design (CAD) models. This model form also avoids having to choose one Euclidean coordinate (say, z) as a "response" function of the other 2 coordinate "locations" (say, x and y), as commonly used in previous Euclidean kriging models of manufacturing data. The (u, v) surface coordinates are computed using parameterization algorithms from the manifold learning and computer graphics literature. These are then used as locations in a spatial Gaussian process model that considers correlations between two points on the surface a function of their *geodesic* distance on the surface, rather than a function of their Euclidean distances over the xy plane. It is shown how the proposed Geodesic Gaussian Process (GGP) approach better reconstructs the true surface, filtering the measurement noise, than when using a standard Euclidean kriging model of the 'heights', i.e., $z(x, y)$. The methodology is applied to simulated surface data and to a real dataset obtained with a non-contact laser scanner.

Keywords: Non-contact sensors, Parametric Surface model, manifold data analysis, CAD.

[*] Corresponding author. Dr. Castillo is Distinguished Professor of Industrial & Manufacturing Engineering and Professor of Statistics. e-mail: exd13@psu.edu

[†] Dr. Colosimo is an Associate Professor in the Production Technology Group.

[‡] Mr. Tajbakhsh is a Ph.D. student in the Industrial & Manufacturing Engineering department.

1 Introduction

We consider the statistical reconstruction of a surface patch S embedded in 3-dimensional (3D) Euclidean space from noisy measurements. In applications in engineering and geostatistics, kriging and Gaussian processes have been used for modeling spatially distributed data of some scalar field, e.g., temperature, under the assumption that observations $z(x, y)$ that occur on nearby locations $(x, y) \in \mathbb{E}^2$ (Euclidean 2D space) will tend to be alike, where "closeness" is defined by the standard Euclidean distance on $\mathbb{E}^2$. Our focus is instead on those situations where there is no such scalar field of interest: the (x, y, z) data occurs on a non-Euclidean surface and the object of interest is the true 3 dimensional underlying surface, which can only be inferred –or reconstructed–from noisy measurements in the form of a point cloud dataset of Euclidean coordinates (x, y, z). This is an increasingly common situation in industry given the wide availability of non-contact measuring sensors which provide 3D point cloud data. In this paper, we adopt a *geodesic hypothesis*: due to the physics involved in generating and measuring the surface, correlations between the measured coordinates may exist, but the spatial correlation will depend on the geodesic distance between the points located on the surface, rather than depending on the inter-point Euclidean distances on the space the surface is embedded in. By *geodesic distance* between two points on a surface we mean the minimum arc length among all possible such arcs *on the surface* that join the two points, where a *geodesic* curve on an arbitrary surface is a generalization of a straight line in Euclidean space (O'Neill, 2006, p. 346). We focus on reconstructing a surface *patch*, formally defined below, where a 3D object may be composed of a collection of such patches.

Our geodesic hypothesis is motivated on engineering/manufacturing knowledge grounds: a machined part would tend to have correlated point coordinates *not* depending on the Euclidean distance between the points (since there might be "empty space" between two points on a curved surface) but along distances as measured *on* the surfaces (geodesic distances), since typically the manufacturing process will have an effect on the shape –the exterior surface– of the object, which is what a sensor measures. Specific instances of manufacturing processes where the geodesic hypothesis is plausible include free-form sheet-metal forming, where local similarity (e.g., mechanical properties) are maintained along geodesics, i.e, the path on the surface, and phenomena such as shrinkage and springback are observed depending on the local curvature; free-form surfaces obtained by milling (e.g., metal dies), where nearby points on the final surface correspond to points machined in similar conditions, and free-form surfaces obtained by casting where points close on the final surface follow a similar solidification and cooling history. Some *empirical* evidence in favor of the geodesical assumption is given in later sections of this paper.

There are two main applications that motivated our work. First, in industrial quality control, measurements (x, y, z) on the surface of a free-form manufactured part are taken by definition on a non-Euclidean 2-manifold with the purpose of inspecting the part by comparing it to some ideal geometry. Here it is relevant to model and filter –as much as possible– the measurement error, which occurs in all 3 spatial coordinates. Furthermore, correlations will likely occur as a function of distance on the surface. Data obtained with non-contact sensors (laser scanners) from machined surfaces have been reported to be Gaussian-like and strongly spatially correlated (Sun, Rosin, Martin, and Langbein, 2008), although empirical investigations have only considered planar surfaces. Besides inspection, engineers may wish to perform statistical process control on surface data, and Gaussian Process (GP) models of point cloud surface data can be used for this purpose (Colosimo, Pacella, Vlaco and Cicorella, 2013b). A second motivation for the present work is in the area of "reverse engineering" in manufacturing, where one measures some complex surface of a product in order to build a model of it, usually with the final purpose of copying it. It is then necessary to reconstruct the surface from unorganized point cloud data, in order to create a Computer Aided Design (CAD) file.

The proposed Geodesic Gaussian Process (GGP) approach uses a *parametric representation* of a surface patch where each of the three coordinates is modeled via a Gaussian process on the parametric space defined by surface coordinates (u, v), i.e., GGP produces models $\widehat{x}(u, v), \widehat{y}(u, v)$ and $\widehat{z}(u, v)$, where the (u, v) coordinates need to be computed first. We choose a parametric surface model form as it is the preferred representation of surfaces in CAD (e.g see Patrikalakis and Maekawa, 2002) and CAD file standards (e.g., IGES) and this facilitates tolerancing and reverse engineering applications of the GGP model. Our approach solves a dilemma faced by prior authors who used GP's for manufacturing metrology data: it is not clear why one should consider one of the 3 coordinates the 'response' and the other two the 'locations' when working with point cloud data obtained by a non-contact scanner.

The type of applications we focus on can be better seen by considering a point cloud data set acquired with a structured light scanner first studied by Cavallaro, Moroni and Petro (2010) and further analyzed by Colosimo and Pacella (2011) and Colosimo, Pacella, and Senin (2013a), displayed in Figure 8 below. A structured light scanner yields a large set of points arranged in a regular grid which are characterized by their high density and low precision. The low precision (relative to a contact sensor) implies that a method to reconstruct the true underlying surface by "filtering" the measurement noise as much as possible would be desirable. We return to the analysis of this data set in section 6.2.

The rest of this paper is organized as follows. Section 2 reviews related prior work on Gaussian process modeling and point cloud data. Section 3 introduces the main GGP model assumptions as well as the differential geometry notions that will be used later on. Section 4 discusses the computation of a near isometric parameterization of a 3D surface (and therefore, computation of geodesic distances), a problem intensively studied in recent years in the fields of computer graphics and manifold learning. Section 5 describes how to fit the GPP model. Section 6 presents examples of surface reconstruction using the GGP model, including simulated examples and the aforementioned real data set obtained with a laser scanner. The paper concludes with some general discussion and suggestions for further research. Supplementary materials includes an additional parameterization example of a surface, discussion about how to reduce the impact of noisy observations, differential-geometric analysis details of a GGP-fitted surface and computational and software implementation details of our method.

2 Related prior work

Gaussian processes have been used to model metrology data obtained via a coordinate measurement machine (CMM) by Xia, Ding, and Wang (2008) and by Xia, Ding and Mallick (2011). Rather than using a GP model for each measured coordinate in $\boldsymbol{m} = (m_x, m_y, m_x)'$ as we do here, they consider modeling the projection of $\boldsymbol{m}$ on the direction of approach of the CMM probe, which results in a scalar that is then modeled with a GP. While this approach is useful for CMM data, it cannot be applied for modeling surface data obtained by other means (e.g., non-contact sensors). Colosimo et al. (2013b) use GP models for process monitoring of manufactured surfaces.

There exists also considerable related work on non-isotropic covariance spatial models. A standard approach in the earth sciences to model anisotropic spatial covariance whose contours are elliptical is to use Mahalanobis distances between two points $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$, rather than their Euclidean distance, i.e., the covariance function is $C(\boldsymbol{A}\boldsymbol{h})$ instead of $C(\boldsymbol{h})$, where $\boldsymbol{h} = \boldsymbol{w}_i - \boldsymbol{w}_j$ and $\boldsymbol{A}$ is some invertible $n \times n$ matrix (Schabenberger and Gotway, 2005). This, however, will not be adequate when there is *local* anisotropy, a term used in geostatistics to describe the changing direction behavior of deposits on a region subdivided in cells, a situation that can be due to the data originating from deposits forming a non-Euclidean manifold (Boisvert and Deutsch, 2011). As discussed by Curriero (2007), covariance functions that are known to be valid (positive definite) in Euclidean space are not necessarily valid on non-Euclidean space.

Using a Mahalanobis distance is an instance of so-called *space deformation* methods. These suggest transforming the non-Euclidean space into an Euclidean space, a line of work that originated with Sampson and Guttorp (1992). Their procedure requires repeated measurements at a set of 2-dimensional space locations $\{\boldsymbol{w}_i = (x, y)_i\}$ (the process is assumed time stationary), and applies multidimensional scaling (MDS) on the variances computed from the replicates to obtain locations $\{\boldsymbol{w}_i^* = (x^*, y^*)_i\}$ on a transformed, Euclidean space, where a standard variogram or covariance model can then be estimated (and its validity be assured). Finally, using thin plane splines, they fit a function $f : \mathbb{E}^2 \to \mathbb{E}^2$ (note this is a function from 2D to 2D) such that an isometry is found, i.e., $f(\boldsymbol{w}) = \boldsymbol{w}^*$. This function then allows the extension of the mapping from the observed points to any other new point $\boldsymbol{w}$ at which it is desired to predict the response of interest.

Other work that follows a space transformation strategy is by Schmidt and O'Hagan (2003), who present a Bayesian approach to find a transformation $f : \mathbb{E}^2 \to \mathbb{E}^2$. They set the prior of f as a GP and use MCMC techniques for posterior inference. Kim, Mallick and Holmes (2005) focus on modeling sharp transitions in the covariance function, which they argue cannot be modeled with the type of smooth splines used by Sampson-Guttorp approaches. They also consider finding a transformation $f : \mathbb{E}^2 \to \mathbb{E}^2$ but their approach is based on partitioning the domain D, assuming each subregion is homogeneous and hence adequately modeled by a standard stationary kriging or GP. In contrast with these procedures, our GGP model finds a 2D to 3D parameterization since it models all 3 measured coordinates and does not require replicated observations at the same locations or extra surface parameters. Furthermore, it allows inferences in the true underlying surface in the presence of noise, a modeling aspect recently emphasized by Cressie and Wikle (2011, p. 136).

Some recent work in machine learning on graphs is also related to our approach. Sollich, Urry and Coti (2009, see also Jakab, 2011) use a GP to approximate a function f defined on the nodes i of a given graph. The covariance kernel of the GP is a function of the shortest distances between nodes on the graph. For point cloud data, such a graph can easily be constructed (e.g., with a triangulation) and then one would fit GP models to $x(i)$, $y(i)$ and $z(i)$. Although such approach would model correlations among approximate geodesics, we do not pursue this approach as this does *not* provide a parametric surface model. Advantages of a parametric surface model are its compatibility with CAD representations, the possibility of providing continuous interpolations on the surface and the easiness of performing Differential Geometry computations on the fitted model (e.g., computations of curves and areas on the surface, see supplementary material section E).

3 Model assumptions

The spatial statistical modeling of data obtained on a surface requires appropriate definition of the 'locations' at which the data are observed. In the same way that a curve C embedded in a 2D space can be described by a single (scalar) parameter t, i.e., by points $\boldsymbol{p}(t) = (x(t), y(t)) \in C \subset \mathbb{E}^2$ such that $t \in D \subset \mathbb{E}$, a surface S embedded in 3D space can be described by two parameters, i.e., by points $\boldsymbol{p}(u, v)$ such that

$$\boldsymbol{p}(u,v) \equiv \boldsymbol{p}(\boldsymbol{w}) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad \boldsymbol{w} = (u,v) \in D \subset \mathbb{E}^2, \quad \boldsymbol{p}(\boldsymbol{w}) \in S \subset \mathbb{E}^3 \tag{1}$$

thus $\boldsymbol{p} : D \subset \mathbb{E}^2 \rightarrow S \subset \mathbb{E}^3$ is said to be a parameterization (see, e.g., O'Neill, 2006) from the space D of surface coordinates or *parameters* (u, v) to a 3-dimensional point $\boldsymbol{p}(u, v)$ on the surface S (see Figure 1). Since we wish to model the uncertainty on all 3 coordinates, we decompose $\boldsymbol{p}(u, v)$ in its three parametric component surfaces (Figure 2).

We assume points $\boldsymbol{p}(\boldsymbol{w})$ lie on a 2-dimensional manifold that forms a *surface patch* embedded in $\mathbb{E}^3$. This means that $\boldsymbol{p}$ is a one-to-one differentiable function (so its inverse exists, see Figure 1) and its Jacobian $J = (\partial \boldsymbol{p}/\partial \boldsymbol{w})$ has rank 2 (see O'Neill, 2006). This regularity condition guarantees any 2 of the 3 inverse functions can be solved to "extend" the mapping (see Kreyszig, 1991) to a new location (u_0, v_0) on D (e.g., once the models are fit, we can solve, e.g., $\hat{p}_x(u_0, v_0) = x_0$ and $\hat{p}_y(u_0, v_0) = y_0$ for u_0 and v_0). In practice this implies a patch does not bend or curve on itself. The parametric surface representation (1) is the preferred approach to model a surface in CAD as it is used by Non-Uniform-Rational B-spline Surface models (NURBS) (Patrikalakis and Maekawa, 2002). We assume points $\boldsymbol{p}(\boldsymbol{w})$ on the true underlying surface are not directly observable, but are observed only in the presence of measurement error (Cressie and Wikle, 2011, p.136), thus we observe:

$$\boldsymbol{m}(\boldsymbol{w}) = \begin{pmatrix} m_x(\boldsymbol{w}) \\ m_y(\boldsymbol{w}) \\ m_z(\boldsymbol{w}) \end{pmatrix} = \boldsymbol{p}(\boldsymbol{w}) + \boldsymbol{\varepsilon}(\boldsymbol{w}), \quad \boldsymbol{w} \in D \tag{2}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{w}) \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\varepsilon)$ denotes a non-smooth i.i.d. measurement error process defined on D with $\boldsymbol{\Sigma}_\varepsilon$ containing the "nuggets" $\tau_i^2, i \in \{x, y, z\}$. It is further assumed the true underlying surface is a smooth, non-stationary spatial GP, which makes up the "state" equation

$$\boldsymbol{p}(\boldsymbol{w}) = \boldsymbol{\mu}(\boldsymbol{w}) + \boldsymbol{\delta}(\boldsymbol{w}), \quad \boldsymbol{w} \in D \tag{3}$$

where

$$\boldsymbol{\mu}(\boldsymbol{w}) = \begin{pmatrix} \boldsymbol{\beta}'_x \boldsymbol{f}_x(\boldsymbol{w}) \\ \boldsymbol{\beta}'_y \boldsymbol{f}_y(\boldsymbol{w}) \\ \boldsymbol{\beta}'_z \boldsymbol{f}_z(\boldsymbol{w}) \end{pmatrix}$$

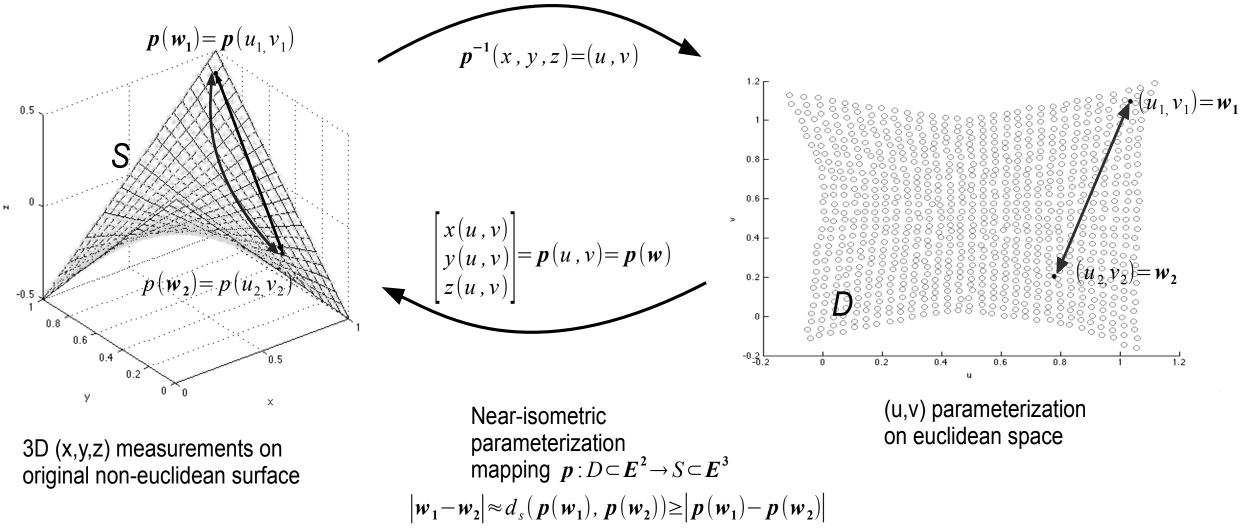
$p(w_1)=p(u_1,v_1)$
S
$p(w_2)=p(u_2,v_2)$
x
y
z
$p^{-1}(x,y,z)=(u,v)$
$\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} = p(u,v) = p(w)$
$(u_1,v_1)=w_1$
$(u_2,v_2)=w_2$
D
u
v
3D (x,y,z) measurements on original non-euclidean surface
Near-isometric parameterization mapping $p: D \subset E^2 \rightarrow S \subset E^3$
$|w_1 - w_2| \approx d_s(p(w_1), p(w_2)) \geq |p(w_1) - p(w_2)|$
(u,v) parameterization on euclidean space

Figure 1: An isometric parameterization is a mapping $p : D \subset \mathbb{E}^2 \rightarrow S \subset \mathbb{E}^3$ such that distances on the non-Euclidean surface $d_s(\mathbf{p}(\mathbf{w_1}), \mathbf{p}(\mathbf{w_1}))$ equal the Euclidean distances between the corresponding points $\mathbf{w}_1 = (u_1, v_1)$ and $\mathbf{w}_2 = (u_2, v_2)$ in the parameterized space, obtained by "flattening" the surface S.

models long-range (systematic) variation and $\boldsymbol{\delta}(\boldsymbol{w})$ is a zero-mean, smooth (no-nugget), 3-dimensional vector stationary GP with covariance functions $C_x(\boldsymbol{h})$, $C_y(\boldsymbol{h})$, and $C_z(\boldsymbol{h})$, respectively, where $\boldsymbol{h} = \boldsymbol{w}_i - \boldsymbol{w}_j$. Reconstructing S implies making inferences about the state (the underlying surface) $\boldsymbol{p}(\boldsymbol{w})$, not about $\boldsymbol{m}(\boldsymbol{w})$ (the observed surface). The functions $\boldsymbol{f}_\bullet(\boldsymbol{w})$ are vector functions of the $\boldsymbol{w} = (u, v)$ surface coordinates and the vectors $\boldsymbol{\beta}_\bullet$ are the corresponding regression parameters. In most cases, a linear or an interaction model in (u, v) suffices for $x(u, v)$ and $y(u, v)$, as can be seen in Figure 2. In applications in manufacturing metrology, the state equation (3) can represent the deviation surface from a nominal geometry $\boldsymbol{T}(\boldsymbol{w})$, usually specified by a NURBS patch in CAD systems. In such an application, our model allows the deviation surface to have systematic $(\boldsymbol{\beta}'_\bullet \boldsymbol{f}_\bullet(\boldsymbol{w}))$ and random $(\delta_\bullet(\boldsymbol{w}))$ components, which would vary depending on the state of the manufacturing process, a matter that has implications for process monitoring, a topic we do not discuss herein. If no CAD model is available (e.g., in a reverse engineering situation) then (3) models directly the manufactured surface S. Hereafter, we refer to (1-2) as the GGP model. The main steps of this modeling methodology are shown diagrammatically in Figure 3.

Our GGP method will be contrasted to the most common alternative used in manufacturing practice for modeling a surface using a GP. This consists in using what is called a *Monge patch* (Kreyszig, 1991) in Differential Geometry, resulting in the Euclidean GP

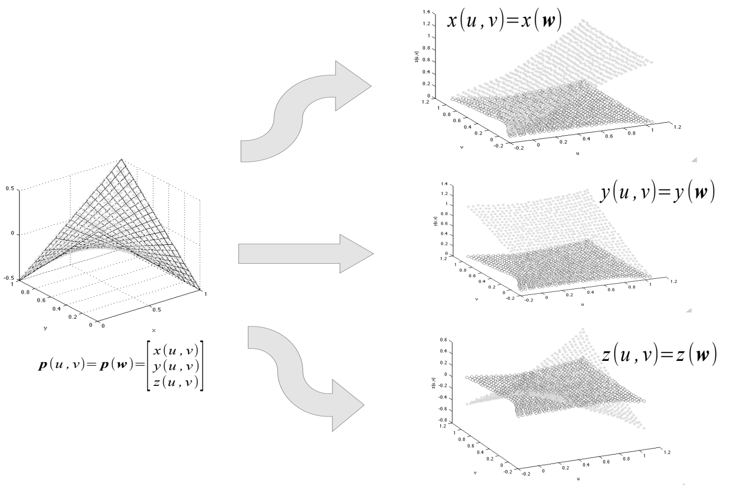

Figure 2: With a parametric representation, a surface in $\mathbb{E}^3$ is decomposed into its three Euclidean coordinate functions $x(u, v)$, $y(u, v)$ and $z(u, v)$ (right 3 graphs), each described over the same space of coordinates (u, v), shown as a plane of darker points in the graphs on the right.

model:

$$\boldsymbol{m}(x, y) = \begin{pmatrix} x \\ y \\ p_z(x, y) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \varepsilon_z(x, y) \end{pmatrix} \quad \text{and} \quad \boldsymbol{p}(x, y) = \begin{pmatrix} 0 \\ 0 \\ \mu_z(x, y) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \delta_z(x, y) \end{pmatrix}, \tag{4}$$

where $(x, y) \in \mathbb{E}^2$. In simpler words, (4) models only the ‘heights’ (z) of the surface patch as a function of the other 2 *Euclidean* coordinates, i.e., $z(x, y)$. In such a model, spatial correlation is a function of Euclidean distances in the xy space, and not a function of distances on the surface space S as model (1) assumes. In this case, $\boldsymbol{p}(x, y)$ is a trivial parameterization of the surface. Depending on the application, selecting one of the three coordinates to be the (univariate) ‘response’ and to assume the remaining two coordinates to be noise-free ‘locations’ may be arbitrary and not justifiable in general. If spatial correlation is a function of geodesic distances on S, this model will result in biased predictions. We discuss further the issue of considering the errors in the locations in the supplementary material.

4 Finding an $\mathbb{E}^2 \rightarrow \mathbb{E}^3$ surface parameterization

A key step in the proposed surface reconstruction method is finding a parameterization $\boldsymbol{p}(u, v)$, for $(u, v) \in D$ (Figure 1). Since the parametric coordinate space D is Euclidean, once a parameterization is available we can use any standard valid spatial covariance models on this space (Curreiro, 2007). Given observed coordinates $(m_{x_i}, m_{y_i}, m_{z_i})_{i=1}^n$, we wish to find the corresponding surface coordinates $(u_i, v_i)_{i=1}^n$. There exist several techniques to do

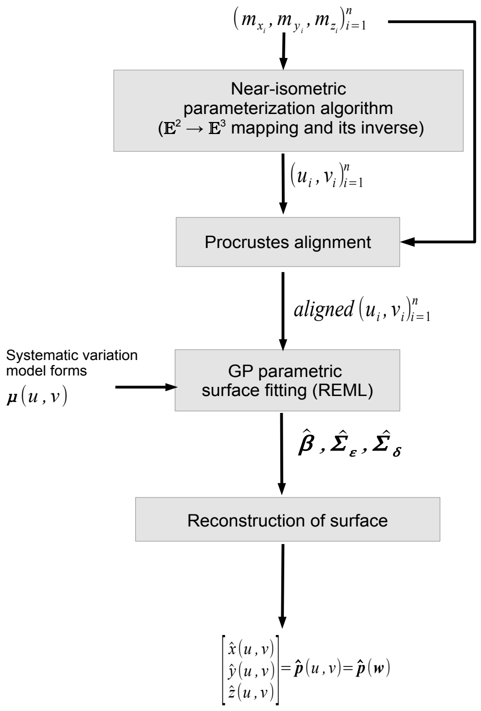
$(m_{x_i}, m_{y_i}, m_{z_i})_{i=1}^{n}$
Near-isometric parameterization algorithm ($\mathbf{E}^2 \to \mathbf{E}^3$ mapping and its inverse)
$(u_i, v_i)_{i=1}^{n}$
Procrustes alignment
aligned $(u_i, v_i)_{i=1}^{n}$
Systematic variation model forms
$\boldsymbol{\mu}(u, v)$
GP parametric surface fitting (REML)
$\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\delta}}$
Reconstruction of surface
$\begin{bmatrix} \hat{x}(u,v) \\ \hat{y}(u,v) \\ \hat{z}(u,v) \end{bmatrix} = \hat{\boldsymbol{p}}(u,v) = \hat{\boldsymbol{p}}(\boldsymbol{w})$

Figure 3: Main steps of the proposed GGP surface reconstruction method.

this task. These include algorithms from the area of manifold learning, such as the Isomap method (Tenenbaum, de Silva, and Langford 2000) and the LLE method (Roweis and Saul, 2000). Manifold learning methods, extensively studied in the last decade, attempt to solve the more general problem of dimensionality reduction from $\mathbb{E}^{l_2}$ to $\mathbb{E}^{l_1}$, where $l_2 \gg l_1$. In the field of CAD and computer graphics there exists another very large thread of literature on methods to solve the more specific $l_1 = 2, l_2 = 3$ parameterization problem. The CAD literature is naturally concerned with the surface parameterization problem, given the use of NURBS models in CAD software systems. Some of the earlier parameterization methods in CAD were described by Ma and Kruth (1996). Weiss, Andor, Renner and Varady (2002) review other parameterization techniques used by CAD systems, and suggested using algorithms from the computer graphics literature for this task.

There are different ways to define what a good parameterization is. The ideal case is to find an *isometry*, a mapping that preserves distances between corresponding points. Formally (O'Neill, 2006, p. 265), if $\boldsymbol{p} : D \subset \mathbb{E}^2 \to S \subset \mathbb{E}^3$ is an isometry, then

$$d_D(\boldsymbol{w}_1, \boldsymbol{w}_2) = d_S(\boldsymbol{p}(\boldsymbol{w}_1), \boldsymbol{p}(\boldsymbol{w}_2)) \quad \forall\ \boldsymbol{w}_1, \boldsymbol{w}_2 \in D \tag{5}$$

where in our case $d_D(\boldsymbol{w}_1, \boldsymbol{w}_2) = |\boldsymbol{w}_1 - \boldsymbol{w}_2|$, the Euclidean distance on $D \subset \mathbb{E}^2$. An isometric mapping can be thought of as a transformation that bends the surface S into a different

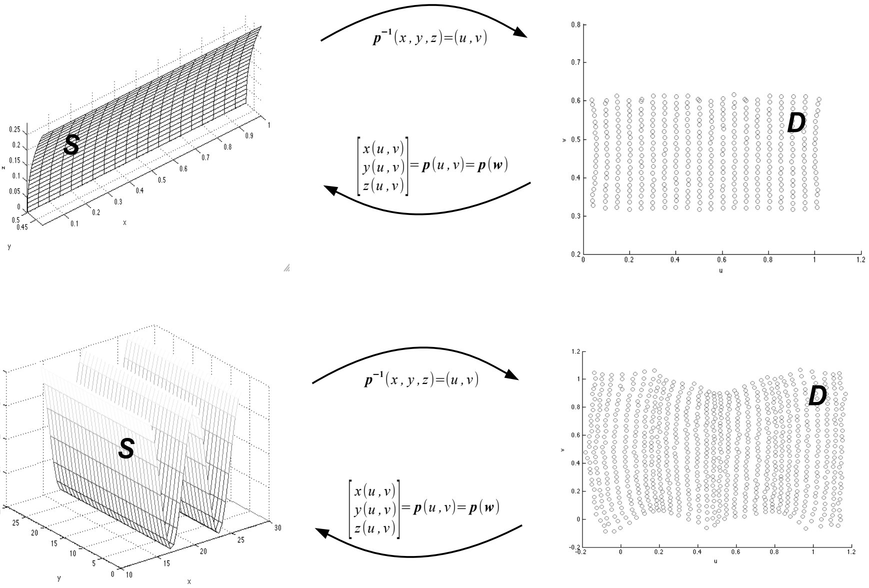

Figure 4: Parameterization examples obtained using the ARAP algorithm (Liu et al., 2008). Cylindrical (400 points) and sinusoidal (900 points) patches. The distances between pairs of points on the uv plane D on the right are approximately equal to the geodesic distances between the corresponding points $\boldsymbol{p}(\boldsymbol{w}_i)$ and $\boldsymbol{p}(\boldsymbol{w}_2)$ on the patches S on the left.

shape without changing the intrinsic distances between points on S. Hence, it can be shown that an isometry also preserves areas on S and angles between curves on S (i.e., it is a *conformal* mapping). An isometric mapping is also a *geodesic* mapping, in which geodesic distances between points in one space (d_D) map into geodesic distances d_S on the image space (Kreyszig, 1991, Theorem 94.2). But as it is well-known in cartography, finding a perfectly isometric mapping is possible only if the surface is *developable*, i.e., if the surface has a Gaussian curvature of zero everywhere (Kreyszig, 1991, p. 181).

Some popular parameterization algorithms in the computer graphics literature find a conformal mapping, which has nice mathematical properties (Floater and Hormann, 2005) but result in pronounced area deformations. Extensive work on the surface parameterization problem over the past decade has resulted in algorithms that instead attempt to preserve areas, or that minimize a weighted sum of distortions due to differences in angles and due to differences in areas, achieving in this way an “as isometric as possible” mapping (e.g., Liu, Zhang, Gotsman and Gortler, 2008, Sorkine and Alexa, 2007, Deneger, Meseth and Klein, 2003). This type of parameterization methods are particularly useful for our approach, since we assume correlations are a function of the geodesic distances on the surface, and these are provided by an isometric mapping. Figure 4 shows two instances of surface patches, observed with noise, and their near-isometric parameterization.

Figure 5 shows scatter plots of the exact geodesic distances between points $\boldsymbol{p}(u,v)_i$ and $\boldsymbol{p}(u,v)_j$ on a cylindrical patch plotted against the Euclidean distance between the corresponding (u_i, v_i) and (u_j, v_j) points (for 400 points there are 79800 such pairs) obtained with two parameterization algorithms, Isomap (Tenenbaum et al., 2000) and the "As-Rigid-As-Possible" (ARAP) method (Liu et al., 2008) that we describe more fully below and in the supplementary materials. As it can be seen, both methods are near isometries, since the scatters are close to a 45° line (in view of (5), the correlation coefficient of the scatters is a measure of near-isometry) with the estimated correlations exceeding 0.995 for each method. Table 1 shows the estimated correlation coefficients of similar scatter plots (not depicted) obtained with other algorithms used for the parameterization step, applied to 400 noisy observations taken from a half cylinder (here we added noise generated with a geodesic Gaussian process with an exponential correlated function with parameters $\phi_\bullet = 1, \sigma^2 = \tau^2 = 0.0001$ to the true points on the cylinder, see next section for a description of the covariance model used). Note that if noise is added, the measured observations no longer form a developable surface, so one should not expect a *perfect* rectangle on the uv plane. The first 2 algorithms are from the computer graphics literature, while the bottom 4 are from the manifold learning literature. Although the correlations shown are only point estimates, the overall conclusion is clear: among the tested parameterization algorithms, only Isomap and ARAP are able to find a near isometry in the case of a cylinder. If an algorithm is unable to "unfold" this particularly simple, developable surface, it will typically be unable to unfold near isometrically more complicated, non-developable surfaces. In particular, the first algorithm on the table (Least Squares Conformal Map or LSCM, Levy, Petitjean, Ray and Maillot, 2002) shows how conformal parameterization algorithms from the computer graphics field are not useful for our purposes, since they severely distort distances. A complete survey of parameterization methods from the manifold learning literature up to 2009 is given by van der Mateen, Postma, and van der Herik (2009). These authors also provide a very useful library of Matlab programs some of which were used to prepare Table 1. For our purposes, all that is necessary is to find a reliable *near-isometric* parameterization method, perhaps one that is fast to compute for large point clouds, and both Isomap and ARAP have these properties. Although we suggest using either method, it is important to point out their weaknesses: as it can be seen in Figure 5, ARAP typically distorts the boundaries of the object (this is also a problem, but of lesser magnitude, for Isomap). Likewise, (see Figure 6) Isomap distorts a surface near a "hole" (ISOMAP proof of asymptotic convergence to a near isometry rests on the assumption observations lie on a geodesically convex manifold, see main theorem in Bernstein, da Silva, Langford, and Tenenbaum (2000), an assumption that is false if the surface has holes). ARAP scales

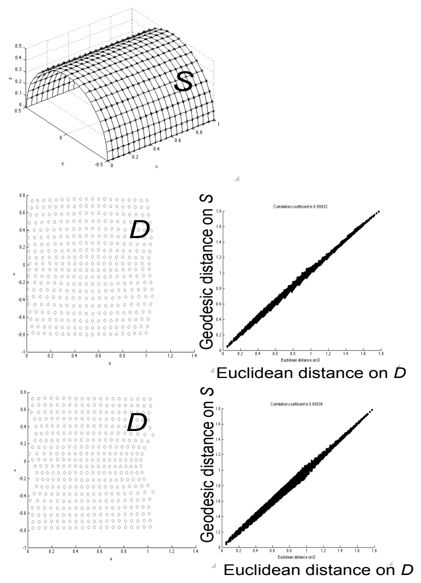

Figure 5: Parameterization of a cylinder patch (400 points, top) using the Isomap (middle) and ARAP (bottom) algorithms. The scatter plots show the exact geodesic distance on the true underlying surface between all 79800 pairs of points plotted against the Euclidean distance between the corresponding estimated (u, v) points provided by each method.

better with the number of points than Isomap, which needs to be modified for large data sets (see Appendices A.1 and A.2).

		Estimated correlation ($\hat{\rho}$)	
Algorithm	Reference(s)	No measurement error	With measurement error
LSCM	Levy et al., 2008	0.9291	0.8784
ARAP	Liu et al., 2008	**0.9976**	**0.9953**
LLE	Roweis et al., 2000	0.9420	0.8998
HLLE	Donoho et al., 2005	0.9442	0.9434
KPCA	Shölkopf et al., 1998	0.9557	0.9557
Isomap	Tenenbaum et al., 2000	**0.9995**	**0.9984**

Table 1: Correlation coefficients between Euclidean and geodesic distances obtained with different parameterization algorithms applied to the 79800 pairs of points from a grid of 400 noise-free observations generated on a half cylindrical patch.

Surfaces with holes might be a common situation in a metrology situation: some regions of the object might have no measurements due to the pose of the object relative to a non-contact scanner, and this results in gaps in the measured surface. Therefore, we look at this issue in more detail in example 4 (supplementary material).

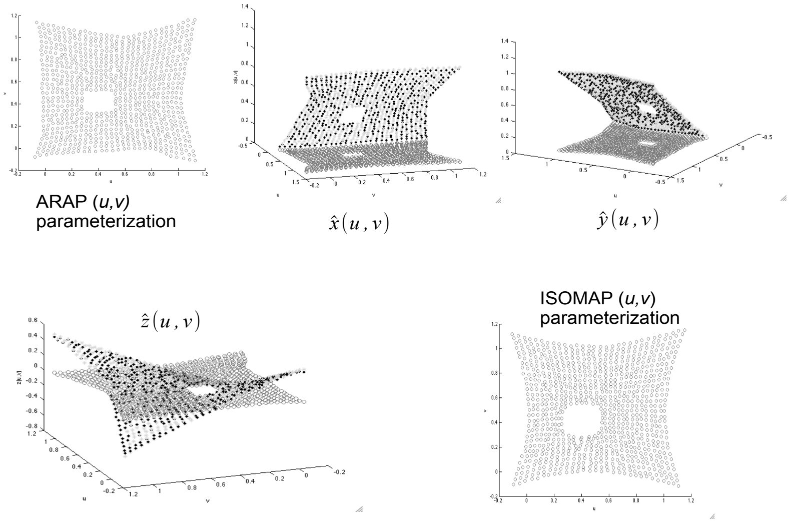

Figure 6: Parameterization of the bilinear NURBS surface patch of figures 1–2 with a rectangular hole. Whereas the ARAP parameterization preserves the geometry of the hole, the ISOMAP algorithm does not retain the rectangular features of the surface hole.

5 Model fitting

Given the coordinates on the surface $\{u_i, v_i\}_{i=1}^n$ that correspond to the n measurements $\{\boldsymbol{m}(u_i, v_i)\}_{i=1}^n$ (collected in the $n \times 3$ design matrix $\boldsymbol{M}$), the next step (see Figure 3) is to fit GP models to the $x(u,v), y(u,v)$, and $z(u,v)$ surfaces. In principle, one could model the three parametric surface components with a multivariate GP. Such model would require specification of the spatial cross-covariance matrix $\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{w}') \in \mathbb{E}^{3\times 3}$, (where recall $\boldsymbol{w} = (u,v)$) which equals:

$$\mathrm{Cov}(\boldsymbol{p}(\boldsymbol{w}), \boldsymbol{p}(\boldsymbol{w}')) = \begin{pmatrix} \mathrm{cov}(x(\boldsymbol{w}), x(\boldsymbol{w}')) & \mathrm{cov}(x(\boldsymbol{w}), y(\boldsymbol{w}')) & \mathrm{cov}(x(\boldsymbol{w}), z(\boldsymbol{w}')) \\ \mathrm{cov}(y(\boldsymbol{w}), x(\boldsymbol{w}')) & \mathrm{cov}(y(\boldsymbol{w}), y(\boldsymbol{w}')) & \mathrm{cov}(y(\boldsymbol{w}), z(\boldsymbol{w}')) \\ \mathrm{cov}(z(\boldsymbol{w}), x(\boldsymbol{w}')) & \mathrm{cov}(z(\boldsymbol{w}), y(\boldsymbol{w}')) & \mathrm{cov}(z(\boldsymbol{w}), z(\boldsymbol{w}')) \end{pmatrix}$$

for $\boldsymbol{w} \neq \boldsymbol{w}'$, which as emphasized by Cressie and Wikle (2011) needs *not* be symmetric (note that the within-location variance-covariance matrix $C(\boldsymbol{w}, \boldsymbol{w}) = \mathrm{Cov}(\boldsymbol{p}(\boldsymbol{w}), \boldsymbol{p}(\boldsymbol{w})) = \mathrm{Var}(\boldsymbol{p}(\boldsymbol{w})) \in \mathbb{E}^{3\times 3}$ is symmetric). Specifying a non-symmetric cross-covariance has proved difficult (Gneiting et al., 2010) because of the positive definitiveness constraint. Simplifying assumptions are usually made, such as adopting a "separable" correlation matrix (Banerjee, Carlin, and Gelfand, 2004), $C(\boldsymbol{w}, \boldsymbol{w}') = \rho(\boldsymbol{w}, \boldsymbol{w}') \cdot \boldsymbol{T}$, where $\boldsymbol{T}$ models within-location correlations and $\rho(\boldsymbol{w}, \boldsymbol{w}')$ models spatial correlation between locations, assumed the same for all responses (clearly inadequate for our case). Furthermore, this results in a symmetric cross-covariance. Other methods that require symmetry are a multivariate Matern model by Gneiting et al. (2010) and Corregionalization (see Banerjee et al., 2004), although Kleijnen and Mehdad (2012) indicate that Corregionalization usually does not outperform separate kriging predictions of each response. As discussed by Cressie and Wikle (2011), the symmetry assumption is very strong, and this is particularly true for our surface modeling application.

For these reasons, we proceed to fit each parametric surface model independently, assuming $\boldsymbol{\Sigma}_\varepsilon = \mathrm{diag}(\tau_x^2, \tau_y^2, \tau_z^2)$ in (2) and $\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{w}') = \mathrm{diag}(C_x(\boldsymbol{h}), C_y(\boldsymbol{h}), C_z(\boldsymbol{h}))$ in (3) where $\boldsymbol{h} = \boldsymbol{w} - \boldsymbol{w}'$ (see conclusions section for more on this).

For each component $\delta_\bullet$ in (3), we use a powered exponential spatial covariance model (Banerjee et al., 2004) such that the $n \times n$ covariance matrix of each surface component $x(\boldsymbol{w}_i)$, $y(\boldsymbol{w}_i)$ and $z(\boldsymbol{w}_i)$ over all measurements can be written as

$$\boldsymbol{\Sigma}_\bullet = \sigma_\bullet^2 \exp(-\phi_\bullet \boldsymbol{D}_{\boldsymbol{w}})^{p_\bullet} + \tau_\bullet^2 \boldsymbol{I}_n, \quad \bullet \in \{x, y, z\} \tag{6}$$

where $\boldsymbol{D}_{\boldsymbol{w}}$ is an $n \times n$ Euclidean distance matrix on the D space. Therefore, the covariance parameters for each surface component model are $\boldsymbol{\theta}_\bullet = \{\phi_\bullet, \sigma_\bullet^2, \tau_\bullet^2, p_\bullet\}$. These parameters and $\boldsymbol{\beta}_\bullet$ (equation 2) are estimated using restricted maximum likelihood (REML, see Santner et al., 2003). For each parametric surface model, the REML estimator minimizes

$$(n - k_\bullet)\log(\sigma_m^2(\theta_\bullet)) + \log(|\boldsymbol{R}(\theta_\bullet)|) + \log(|\boldsymbol{F}_\bullet' \boldsymbol{R}(\boldsymbol{\theta}_\bullet)^{-1} \boldsymbol{F}_\bullet|)$$

where $\sigma_m^2(\boldsymbol{\theta}_\bullet)$ is the variance $C_\bullet(\boldsymbol{0})$ expressed as a function of its covariance parameters, $\boldsymbol{F}_\bullet$ is the $n \times p$ matrix which expands the set of uv locations $\boldsymbol{M}$ according to the terms in the mean model form $\boldsymbol{f}_\bullet(\boldsymbol{w})$ in (3), $\boldsymbol{R}(\boldsymbol{\theta})$ is the $n \times n$ correlation matrix between the n points computed from (6) and $k_\bullet$ is the number of parameters estimated in each parametric surface model. The REML objective has several minima, and therefore we use a simulated annealing (SA) global optimization routine (MATLAB, 2011) started from a set of well-dispersed initial points for its minimization. At each point returned by the SA routine, we ran the `fmincon` interior point nonlinear minimization routine in MATLAB. In this paper, when $n \leq 1600$ we used the full $n \times n$ matrix and followed the recommendations in Lophaven, Nielsen and Sondergaard (2002) for dealing with numerical issues related to the computations of the inverses and determinants in the likelihood function. For larger n, we use a sparsification approach due to Sang and Huang (2012) used in the laser scanner example of section 6.2 and further explained in the supplementary materials. Given the minimizing parameters $\boldsymbol{\theta}_\bullet$, we estimate $\boldsymbol{\beta}_\bullet$ from its generalized least squares estimator (Santner, Williams and Notz, 2003). This procedure is then repeated for each parametric surface, giving the parameter estimates $\hat{\boldsymbol{\theta}}_\bullet$ and $\hat{\boldsymbol{\beta}}_\bullet$, for $\bullet = \{x, y, z\}$.

Given the surface coordinates (u_0, v_0) where a prediction is desired, minimum mean square (MSE) prediction follows the usual approach in GP's (Santner et al., 2003). The prediction equation for each true underlying surface component in $\boldsymbol{p}(u_0, v_0)$ is given by:

$$\hat{p}_\bullet(u_0, v_0) = \boldsymbol{f}(u_0, v_0)'\hat{\boldsymbol{\beta}}_\bullet + \boldsymbol{c}_{p_\bullet}'\boldsymbol{\Sigma}_\bullet^{-1}(\mathcal{M}_\bullet - \boldsymbol{F}_\bullet\hat{\boldsymbol{\beta}}_\bullet), \quad \bullet = \{x, y, z\} \tag{7}$$

where $\mathcal{M}_\bullet$ are all the measurements of each coordinate $\bullet = \{x, y, z\}$ across the n observations in (2). To predict all three surface components $\hat{\boldsymbol{p}}(u_0, v_0)$ we now only need to evaluate *all three* of them at (u_0, v_0).

An important detail in expression (7) is that the vector $\boldsymbol{c}_{p_\bullet}$ is equal to

$$\boldsymbol{c}_{p_\bullet} \equiv \mathrm{Cov}(p_\bullet(u_0, v_0), \mathcal{M}_\bullet) = \begin{pmatrix} c_{p_\bullet}(\boldsymbol{w}_0 - \boldsymbol{w}_1) \\ c_{p_\bullet}(\boldsymbol{w}_0 - \boldsymbol{w}_2) \\ \vdots \\ c_{p_\bullet}(\boldsymbol{w}_0 - \boldsymbol{w}_n) \end{pmatrix}$$

which are the covariances between the true underlying surface component $p_\bullet$ in (3) and the *observed* coordinate $\mathcal{M}_\bullet$ in (2), where $\bullet = \{x, y, z\}$. Cressie and Winkle (2011) emphasize how these covariances should *not* contain the nuggets ($\tau_\bullet^2$), since we are predicting the true underlying surface ($\boldsymbol{p}(u_0, v_0)$), not the observed one ($\boldsymbol{m}(u_0, v_0)$).

6 Examples of surface reconstruction using a GGP

6.1 Examples with simulated surface data

In the simulations shown in this section, we first generated a grid of points over the true underlying surface (in most cases below, a NURBS surface) to which we added *geodesically correlated* Gaussian errors (in the supplementary material we also considers the case of *no* spatial correlation present). To do this, we computed the (u, v) parameterization of the noise-free points using either the ARAP or Isomap method and then added to these spatially correlated normal noise generated using a powered exponential correlation function, where euclidian distances (in the D space) were used, as these correspond closely to the geodesic distances on S (this provided the $\boldsymbol{p}(u_i, v_i)$ points shown in the formulae below). The GGP model fitting and prediction methods shown in the previous section were then applied to these simulated data sets. To evaluate the surface reconstruction performance of the GGP and the more common Euclidean Gaussian process (Section 2), we simply predict the surface at the simulated points with each method and compute the Euclidean distance between the predicted 3D points and the corresponding true surface points, since these are available. We then report the mean squared prediction error *per point*. For the GGP model this is:

$$\mathrm{MSP}_{GGP} = \sqrt{\frac{\sum_{i=1}^{n} |\hat{\boldsymbol{p}}(u_i, v_i) - \boldsymbol{p}(u_i, v_i)|^2}{n}} \tag{8}$$

where $\hat{\boldsymbol{p}}(u_i, v_i) = (\hat{x}(u, v), \hat{y}(u, v), \hat{z}(u, v))'$ and $|\cdot|$ denotes Euclidean distance. For the Euclidean GP model the mean prediction error is:

$$\mathrm{MSP}_{z(x,y)} = \sqrt{\frac{\sum_{i=1}^{n} \left| \begin{pmatrix} m_{x_i} \\ m_{y_i} \\ \hat{z}(x_i, y_i) \end{pmatrix} - \boldsymbol{p}(u_i, v_i) \right|^2}{n}} \tag{9}$$

where the first two coordinates are not predicted and the z coordinate is predicted as a function of the Euclidean coordinates (x, y) instead. Since we are simulating data from known surfaces to which we add noise, in all of these expressions above we are comparing the predicted 3D points against the true underlying 3D Cartesian coordinates at each point (u_i, v_i) on the surface $(\boldsymbol{p}(u_i, v_i))$. In simulated cases (where the true surface points are available), the mean square prediction error statistics above can be compared to the simulated mean square error:

$$\mathrm{MSE}_{3D} = \sqrt{\frac{\sum_{i=1}^{n} |\boldsymbol{m}(u_i, v_i) - \boldsymbol{p}(u_i, v_i)|^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} |\boldsymbol{\varepsilon}(u_i, v_i)|^2}{n}}$$

which is a measure of the mean "noise" added to all 3D points on the surface. If in a simulation it turns out that $\mathrm{MSP}_{GGP} < \mathrm{MSE}_{3D}$ this means the GGP model is able to filter the measurement error enough to get predictions that on average are closer to the true surface than what the observed measurements are.

Example 1.- a cylindrical surface patch.- Table 2 shows the performance metrics of a series of simulations taking the cylindrical patch of Figure 5 as the true underlying surface. Geodesically correlated Gaussian noise was added to a grid of points on the surface, as described before, with correlation function parameters $\phi_\bullet = 1$, $\sigma_\bullet^2 = \tau_\bullet^2 = 0.00001$. An interaction model (in (u, v)) was fit to the mean of $x(u, v)$, while a quadratic model was fit to the mean of $y(u, v)$, $z(u, v)$, and $z(x, y)$. We studied the performance of the GGP methodology compared to the alternative Euclidean GP predictions (see section 2) for different number of points. The statistics are averages and standard deviations from 30 independent simulations and model fits (same data used across methods). Fitting both the GGP and the Euclidean GP models required inversions of $n \times n$ matrices, with the total computing time of 30 model fits exceeding 12 hr., hence the apparently small sample size (see supplementary materials section D for more details about how to fit the GGP model for large n). Despite this, the standard errors of the mean squared prediction errors are relatively small, and the results allow us to make some general observations. As it can be seen from the table, in some cases, the mean square error of the simulated points MSE_{3D} is higher than that of the GGP predictions (MSP_{GGP}). This means that in such cases, the GGP predictions $\hat{\boldsymbol{p}}(\boldsymbol{w})$ are closer to the true unknown surface $\boldsymbol{p}(\boldsymbol{w})$ than what the true surface is with respect to the simulated (noisy) surface points $\boldsymbol{m}(\boldsymbol{w})$. This happens because the kriging predictor is *smoothing* the data, i.e., the reconstruction is not an exact interpolation, and because in these cases the prediction bias is low. When $\text{MSE}_{3D} > \text{MSP}_{GGP}$, the proposed approach is effectively filtering the observational noise in the state-space model (2-3). In contrast, the Euclidean GP approach ($z(x, y)$) incurs in considerable higher prediction errors (around 50% compared to the GGP predictions; all differences in MSP between the GGP and the $z(x, y)$ models have p-values for the t-test of equality < 0.0001).

n	MSE_{3D}	MSP_{GGP}	$\text{MSP}_{z(x,y)}$
400	0.00756	0.00773	0.01178
	(0.00006)	(0.00108)	(0.00091)
900	0.00798	0.00796	0.01188
	(0.00109)	(0.00108)	(0.00099)
1600	0.00771	0.00764	0.01192
	(0.00098)	(0.00099)	(0.00094)

Table 2: Prediction results for cylindrical patch, 30 simulations, mean and standard deviations (in parenthesis) of performance statistics. MSP_{GGP} is the per observation mean square 3D prediction error using the proposed GGP model (8), $\text{MPS}_{z(x,y)}$ is the corresponding error if the Euclidean GP (9) is used . MSE_{3D} is the simulated mean square error of the 3D points.

Example 2.- a sinusoidal surface patch.- In this case the true surface is $z(u, v) = 0.1 \sin(u)$, depicted in Figure 4. This is a type of surface patch reported to be useful as a model in high precision micro machining (Zhang et al., 2009). The same noise values as in the cylindrical patch case were added to a grid of points generated on this surface. An interaction model (in (u, v)) was fit to the mean of $x(u, v)$, and to the mean of $y(u, v)$, while a constant (intercept only) model mean was used for $z(u, v)$, and $z(x, y)$.

n	MSE_{3D}	MSP_{GGP}	$\text{MSP}_{z(x,y)}$
400	0.00761	0.02435	0.04552
	(0.00089)	(0.00693)	(0.00187)
900	0.00785	0.01284	0.02946
	(0.00132)	(0.00284)	(0.00148)
1600	0.00752	0.00954	0.01957
	(0.00099)	(0.00123)	(0.00112)

Table 3: Results for sinusoidal patch, 30 simulations, mean and standard deviations (in parenthesis) of performance statistics. MSP_{GGP} is the per observation mean square 3D prediction error using the proposed GGP model (8), $\text{MPS}_{z(x,y)}$ is the corresponding error if the Euclidean GP (9) is used . MSE_{3D} is the simulated mean square error of the 3D points.

The Euclidean GP approach ($z(x, y)$) incurs in worse prediction errors compared to the

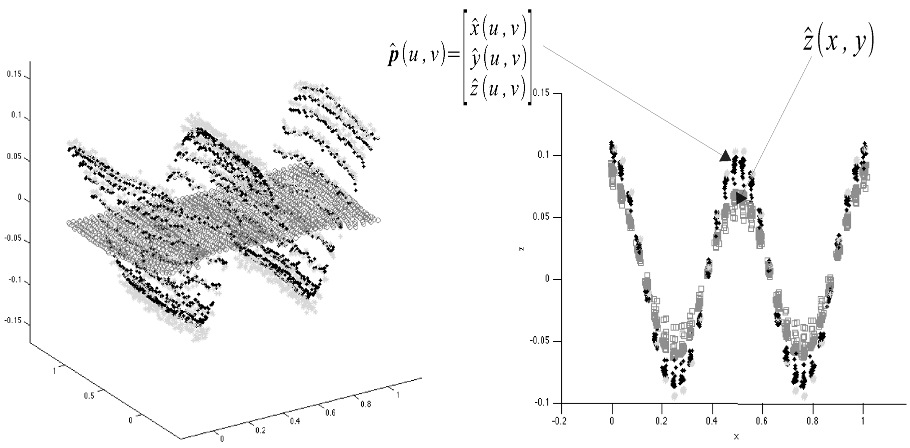

Figure 7: Left: Sinusoidal patch observed points (light dots) and GGP predictions $\hat{\boldsymbol{p}}(u, v)$ (darkest dots), $n = 400$ points. The plane of dots are the parameters (u, v). Right: side view, showing the Euclidean GP predictions $\hat{z}(x, y)$ (lighter squares), which fail to reconstruct the full depth of peaks and troughs.

parametric 3D predictions (Table 3; all MSP differences between the GGP and the $z(x, y)$ have p-values < 0.0001 for an equality of means test). While the underlying surface is developable (Gaussian curvature is zero) the measured points are not, so the uv surface will not be a perfect rectangle even if a perfect isometry were to be found. This surface has strong curvature, so modeling the heights as a function of 2D Euclidean spaces badly estimates the distances, and hence the correlations, between points on the surface, resulting in an underestimation of the peaks and troughs of the function (see Figure 8). This curvature also makes the GP predictions worse relative to those in the cylindrical patch example. By strong curvature we are referring to large values of the *principal curvatures* at some points, not to the Gaussian curvature values, which for a near-developable surface will be near zero. As the density of points increases, all models fit better. Still, for $n = 1600$ the GGP achieves a MSP error of less than half that of the Euclidean GP model, approaching the level of the simulated noise.

Example 3. Non-smooth surfaces. The simulated surfaces in previous examples are considerably smooth, typical of many manufacturing process (e.g., manufacturing of "free form" surfaces by milling, forming or casting). There might be other manufacturing applications, especially in micro-manufacturing, where the surfaces may be less smooth. "Non-smooth" does not refer to surfaces that have sharp edges, which would require a segmentation procedure to model the surface into smaller patches, a problem not discussed in this paper (see conclusions). By smoothness we mean the degree or order of mean square differentiability of the GP surface realizations. If the surfaces one wishes to model are smooth, a Gaussian spatial covariance function ($p = 2$ in the power exponential function) could suffice

$\sigma^2_\bullet$	$\tau^2_\bullet$	MSE_{3D}	MSP_{GGP}	$\text{MSP}_{z(x,y)}$
0.01	0.0001	0.17423	0.17094	0.17107
		(0.01181)	(0.01208)	(0.01170)
0.0001	0.01	0.17424	0.16522	0.16361
		(0.00291)	(0.00274)	(0.00348)
0.0001	0.001	0.05746	0.05505	0.05922
		(0.00133)	(0.00134)	(0.00189)
0.001	0.0001	0.05761	0.05699	0.06061
		(0.00336)	(0.00342)	(0.00323)

Table 4: Results for non-smooth cylindrical surface patches. $\phi = 5.0$ for all 3 coordinates, $n = 400$.

for modeling $\boldsymbol{\delta}(\boldsymbol{w})$. However, as it is well-known (e.g., Santner et al., 2003) the Gaussian covariance function represents an extreme case of smoothness since it is "infinitely smooth" (i.e., it is mean square differentiable of any order). Instead of using a Gaussian covariance function, we have taken a more conservative approach by using $p = 1$ (exponential spatial covariance function) which, strictly speaking, results in non-differentiable surface realizations. If the noise levels were high, this model *could* result in very non-smooth surfaces $\boldsymbol{p}(\boldsymbol{w})$. In addition, we included a "nugget" term in (2) which by definition is non-smooth. Furthermore, if the ϕ parameter increases, the range of the spatial correlation decreases and this will make surface realizations more "wiggled". Hence for all these reasons the observed surface $\boldsymbol{m}(\boldsymbol{w})$ could be quite non-smooth under the assumed model. The observed smoothness depends on the noise levels; even when a non-smooth covariance function is used, if the noise levels are very low the sample realizations will appear smooth due to the existing trend.

It may be recalled, however, that the *parameterization* procedure assumes the 2-manifold $\boldsymbol{p}(\boldsymbol{w})$ is differentiable, so when using a non-smooth covariance function, such as the power exponential with $p = 1$, if the noise level $\sigma^2 + \tau^2$ is large compared to the trend, the parameterization algorithms may work badly, and this will be reflected in poor predictions. We illustrate this effect in Table 4 which shows additional simulations for the cylindrical surface patch shown earlier under higher noise levels (and larger value of ϕ). The noise levels in the table result in dramatically non-cylindrical surfaces, not typical of free-form manufactured parts we focus on in this paper but that may be more typical in other types of manufacturing (e.g., micro-manufacturing).

Sample realizations of this "cylinder" for the *lowest* set of noise levels indicate these surfaces are extremely non-smooth. It is therefore somewhat reassuring to see that for the last 2 cases in the table the GGP method results in significantly better predictions than a regular Euclidean GP model (p-value of t-test comparison of the MSP's is < 0.0001); for the first case in Table 4 there is no significant difference (p-value $= 0.9675$) and only for the second case in the table the Euclidean GP model predicts slightly better (p-value$=0.0513$). As the smoothness or the curvature increases, the GGP predictions will reconstruct a surface patch increasingly better than the Euclidean GP model.

6.2 Reconstruction from real surface data: laser scanner data

In this section we reconstruct the surface briefly discussed in the introduction and displayed in Figure 8. We contrast the performance of the proposed approach in modeling this surface via cross-validation, fitting the different models in a subset of data and predicting a different subset of points, given that contrary to the previous simulated examples, there is no "known underlying surface" available. Hence, we estimate the mean square prediction errors substituting the true surface points ($\boldsymbol{p}(u_i, v_i)$) with the observed measurements ($\boldsymbol{m}(u_i, v_i)$) at which we are predicting (different to the points at which we fit the model), so n is substituted in (8-9) by $n_{predict}$, the number of points at which we are predicting, not the points used to fit the model as in the previous section.

The original dataset consists of 9635 points from a free form surface (see Figure 8) of base size 100 mm. × 100 mm., acquired with a scanner system. We fit the GGP and Euclidean GP models to 9000 ($= n_{fit}$) randomly sampled observations, using the remaining 635 ($= n_{predict}$) for cross-validation. We also fit preliminary models for smaller number of points ($n_{fit} = 402$, obtained by selecting every 24th point and $n_{fit} = 964$, obtained from selecting every 10th point). The preliminary fits from the smaller datasets are useful to select mean models to use in the full dataset, given the computational effort to fit the models when n_{fit} is large. Just as for a standard Euclidean GP model, when n_{fit} is large, fitting the GGP model requires sparse matrix techniques for handling the inverse and determinant operations needed in the REML routine (see supplementary materials section D for computational details). The ARAP algorithm was used for the parameterization.

Table 5 shows mean square prediction errors obtained by cross-validation. The cross validation was done at $n_{predict}$ different points than the original ones, for which the (u_i, v_i) parameters were computed first (i.e., the parameterization mapping was extended) and then the predictions for these points were computed using the GGP models fitted with the original data. Finally, we compute the mean square error of the predictions generated by GGP (MSP$_{GGP}$) and also by the Euclidean GP method (MSP$_{z(x,y)}$, see Section 2).

We tried different mean models for $x(u, v), y(u, v)$, $z(u, v)$ and $z(x, y)$. From the mean square errors per point of the cross-validated predictions, the best mean models are an interaction model for $x(u, v)$ and $y(u, v)$ and either a constant (intercept) or a quadratic model

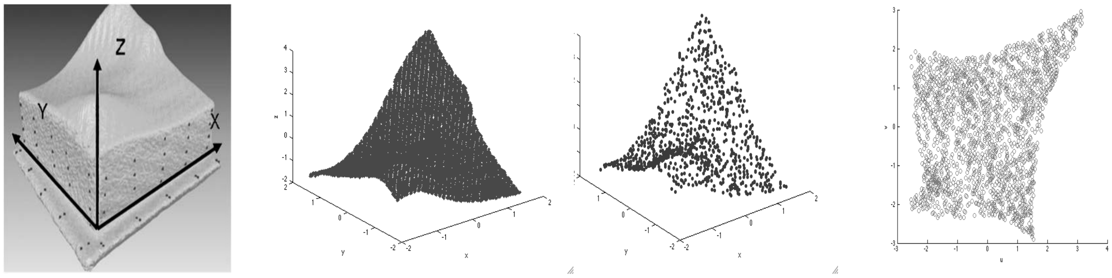

Figure 8: Laser scan data. From left to right: original artifact, full dataset ($n = 9635$), a decimated data set ($n = 964$), and on the right the (u, v) parameterization of the decimated 3-dimensional data using the ARAP algorithm.

for $z(u, v)$ and $z(x, y)$, which were about the same. We compared the GGP predictions to the Euclidean GP method (i.e., only predicting the heights z as function of Euclidean coordinates (x, y)). The mean square errors of doing this in the best fitting models are considerably higher than assuming correlations along geodesic distances. Note how $\text{MSP}_{z(x,y)}$ is computing squared errors *only on the heights* z, whereas MSP_{GGP} *computes squared errors on all 3 coordinates.* Thus, it is notable how in every case $\text{MSP}_{GGP} < \text{MSP}_{z(x,y)}$. If we consider only the GGP prediction errors along the $z(u, v)$ coordinate response (column labeled $(\text{MSP}_{z(u,v)})$), the table shows how these are about *half* of what a standard Euclidean GP model fit to $z(x, y)$ would provide, regardless of the number of points used to fit the model n_{fit}. Hence, these statistics provide evidence that the data set contains correlations that are better modeled along the surface rather than in Euclidean space, and that the GGP model is predicting this surface substantially better than a standard universal kriging model fitted in Euclidean space to $z(x, y)$. In practice, differences in prediction errors of the magnitudes shown in Table 5 may likely result in a part, whose surface has been modeled in the two different ways described in this paper, be accepted or not when performing tolerancing. Alternatively, in a reverse engineering situation, the differences may imply the possibility to successfully develop (or not) a part with precision levels similar than those from a part produced by a competing manufacturer.

7 Discussion and conclusions

A new parametric approach for the statistical reconstruction of a surface patch embedded in 3-dimensional space based on point cloud data. The Geodesic Gaussian Process (GGP) method first finds a parameterization on the surface patch under study and then fits spatial GP models on each of the three Cartesian coordinates as a function of the two surface coordinates. This avoids the problem of having to select one coordinate as the "response" (usually z is chosen) and using the other two coordinates as the (noise free) "locations"

n_{fit}	$n_{predict}$	Mean models for (x, y, z)	MSP_{GGP} ($\text{MSP}_{z(u,v)}$)	$\text{MSP}_{z(x,y)}$
402	401	interaction, interaction, intercept	0.0160 (0.0103)	0.0230
964	963	interaction, interaction, intercept	0.0140 (0.0104)	0.0195
402	401	interaction, interaction, quadratic	0.0165 (0.0110)	0.0257
964	963	interaction, interaction, quadratic	0.0129 (0.0088)	0.0181
9000	635	interaction, interaction, quadratic	0.0112 (0.0077)	0.0157

Table 5: Cross-validation results for Laser scanner data. The estimated mean square prediction errors per observation are shown. The error of the GGP in predicting only coordinate $z(u, v)$ is shown in parenthesis. $\text{MSP}_{z(u,v)}$ is consistently about *half* of $\text{MSP}_{z(x,y)}$, regardless of the number of points used to fit the model.

(usually, (x, y)) that one faces when using a standard kriging model for surface data. The parametric surface form of the model is compatible with CAD models, and this facilitates its application in tolerancing, quality control, and reverse engineering. It was shown how the GGP approach reconstructs surfaces better than the usual kriging/GP modeling approach found in the literature which assumes correlations occur over an Euclidean space and only the "heights" $z(x, y)$ are modeled. If the correlation occurs as a function of geodesic distance between points on the surface or when there is no spatial correlation, Euclidean spatial models resulted in considerable inferior predictions, giving mean square prediction errors that on average were around *twice* those given by the GGP model for the laser scanner data set in section 6.2. We have confirmed in a real laser scanner data set how the assumed "geodesic hypothesis" holds, i.e., we determined how the spatial correlation can be better modeled geodesically than over Euclidean distances. This hypothesis was found also true in a CMM dataset of the artifact depicted in Figure 8, and we have confirmed this with another real scanner dataset (Colosimo et al., 2013c). However, for a full application in industry of the GGP model, it is of interest to determine if for more real-life free-form scanned surfaces the "geodesic hypothesis" holds.

The parameterization approaches used (ARAP and Isomap) may find difficulties if the surface has severe curvature or sharp edges, inevitable problems common to all parameterization algorithms. A way to handle severe curvature is to *segment* a complicated 3D object that perhaps closes into itself (so it is not a surface patch) and partition it in such a way that we get a series of patches each easier to parameterize. There is a number of segmentation algorithms in the computer graphics literature, and we plan to study the problem of how to fit an overall GGP model to the collection of patches.

Supplementary material

Additional results: A further parameterization example, reducing the measurement noise, parameterization methods used, software implementation and differential-geometrical analysis of fitted surfaces (pdf file). **Code and data:** Matlab code and scanner dataset (zip file).

Acknowledgements

The authors wish to thank two anonymous referees, an associate editor and the editor for their comments and suggestions which have resulted in an improved presentation. This research was supported by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285075 - MuProD.

References

Banerjee, S., Carlin, B.P., and Gelfand, A.E., (2004), *Hierarchical modeling and analysis for spatial data*, Boca Raton, FL: Chapman & Hall/CRC Press.

Bernstein M., de Silva, V., Langford, J.C., and Tenenbaum, J.B., (2000), "Graph approximations to geodesics on embedded manifolds", unpublished manuscript, `http://isomap.stanford.edu`

Boisvert, J.B., Manchuck, J.G., and Deutsch, C.V., (2009), "Kriging in the presence of locally varying anisotropy using non-Euclidean distances", *Math. Geology*, 41, pp. 585-601.

Boisvert, J.B. and Deutsch, C.V., (2011), "Programs for kriging and sequential Gaussian simulation with locally varying anisotropy using non-Euclidean distances", *Computers and Geosciences*, 37, pp. 495-510.

Cavallaro, M., Moroni, G., Petrò, S. (2010). "Performance evaluation of non-contact systems considering bias" in *Innovative Developments in Design and Manufacturing*, P. Bartolo et al. (eds.), Taylor and Francis, London, pp. 167-173.

Colosimo, B.M., and Pacella, M., (2011), "On Integrating Multisensor Data for Quality", X Convegno A.I.Te.M., Naples, Italy, September 2011.

Colosimo, B.M., Pacella, M. and Senin, N., (2013a), "Multisensor Data Fusion for Quality Inspection", Technical paper, Politecnico di Milano, Italy (submitted for publication).

Colosimo, B.M., Pacella, M., Vlaco, M. and Cicorella, P., (2013b), "From profile to surface monitoring: SPC for cylindrical surfaces via Gaussian processes", Technical Paper, Politecnico di Milano, Italy.

Colosimo, B.M., Pagani, L., Semeraro, Q., and Del Castillo, E., (2013c), "Geodesic Gaussian Process for the Reconstruction of Micro-Pins Surfaces", *SCO 2013 Conference*, Milano, Italy.

Cressie, N., and Kornak, J., (2003), "Spatial statistics in the presence of location error with an application to remote sensing of the environment", *Statistical Science*, 18(4), pp. 436-456.

Cressie, N., and Wikle, C.K., (2011), *Statistics for Spatio-Temporal Data*, NY: Wiley.

Curriero, F.C., (2007), "On the use of non-Euclidean distance measures in Geostatistics", *Mathematical Geology*, 38(8), pp. 907-926.

Davies, T., (2011). Sparseinv: sparse inverse subset. `http://www.mathworks.com/matlabcentral/fileexchange/33966-sparseinv-sparse-inverse-subset/content/sparseinv/sparseinv.m`.

Degener, P., Meseth, J., and Klein, R., (2003), "An Adaptable Surface Parameterization Method" Proceedings of the 12th International Meshing Roundtable, pp. 201-213.

Donoho, D.L., and Grimes, C., (2005), “Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data”. *Proc. of the Nat. Acad. of Sciences*, 102(21) pp. 7426-7431.

Fanshawe T.R. and Diggle, P.J. (2011), “Spatial prediction in the presence of positional error”, *Environmetrics*, 22, pp. 109-122.

Floater, M.S., and Hormann, K., (2005), “Surface parameterization: a tutorial and survey”, in *Advances in Multiresolution for Geometric Modeling*, Springer, pp. 157–186.

Gabrosek, J., and Cressie, N. (2002). “The effect on attribute prediction of location uncertainty in spatial data”, *Geographical Analysis*, 34(3), pp. 262-285.

Gneiting, T., Klieber, W., Schlather, M., (2010), “Matern Cross-Covariance Functions for Multivariate Random Fields”, *J. of the American Statistical Association*, 105 (491), pp. 1167-1177.

Jakab, H. (2011), ”Geodesic Distance-Based Kernel Construction for Gaussian Process Value Function Approximation.” *Studia Universitatis Babes-Bolyai S. Informatica*, 61(3), pp. 51-57.

Kim, H-M., Mallick, B.K., and Holmes, C.C., (2005), “Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes”, *JASA*, 100:470, pp. 653-668.

Kliejnen, J., and Mehdad, E., (2012), “Kriging in multi-response simulation including a Monte Carlo Laboratory”, CentER Discussion Paper, Tilburg University, Netherlands, `http://ssrn.com/abstract=2060891`.

Kreyszig, E., (1991), *Differential Geometry.* NY: Dover Publications, NY.

Levy, B., Petitjean, S., Ray, N., and Maillot, J., (2002), “Least Squares Conformal Maps for Automatic Texture Atlas Generation”, *Journal ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2002*, 21(3), pp. 362 - 371.

Liu, L., Zhang, L., Gotsman, C., and Gortler, S.J., (2008), “A Local/Global Approach to Mesh Parameterization”, in Alliez and Rusinkiewics, S., (eds.), *Eurographics Symposium on Geometry Processing*, 27(5).

Lophaven, S.N., Nielsen, H.B., and Sondergaard, J., (2002), “Aspects of the Matlab toolbox DACE”. IMM Technical University of Denmark, Lyngby, Denmark.

Ma, W., and Kruth, J.P., (1995),“Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces”, *Comp. Aided Design*, 27(9), pp. 663-675.

Matlab, (2011), The MathWorks Inc., Natick, MA.

O’Neill, B., (2006), *Elementary Differential Geometry.* 2nd, edition, NY: Academic Press.

Patrikalakis, N.M. and Maekawa, T., (2002), *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer.

Roweis, S.T., and Saul, L.K., (2000), “Nonlinear dimensionality reduction by Locally Linear Embedding”. *Science*, 290(5500), pp. 2323-2326.

Sampson, P.D., and Guttorp, P., (1992), “Nonparametric Estimation of Nonstationary Spatial Covariance Structure”, *J. of the American Statistical Assoc.*, 87(417), pp. 108-119.

Sang, H, and Huang, J.Z., (2012). “A full scale approximation of covariance functions for large spatial data sets”, *Journal of the Royal Statistical Society, B*, 74, Part 1, pp. 111-132.

Santner, T. J., Williams B., and Notz W., (2003). *The Design and Analysis of Computer Experiments*, NY, Springer-Verlag.

Schabenberg, O., and Gotway, C.A., (2005), *Statistical Mehods for Spatial Data Analysis*, Chapman & hall CRC Press.

Schmidt, A.M. and O’Hagan, A., (2003), “Bayesian inference for non-stationary spatial covariance structure via spatial deformations”, *J. R. Statist. Soc. B*, 65, Part 3, pp. 743758.

Schmidt, R., (2009), *MatlabMesh Toolkit*, `http://www.dgp.toronto.edu/rms/software/matlabmesh/`

Schölkopf, B., Smola, A.J., and Muller, K.-R. (1998), “Nonlinear component analysis as a kernel eigenvalue problem”. *Neural Computation*, 10(5):1299-1319.

Sollich, P., Urry, M., and Coti, C. (2009). “Kernels and learning curves for Gaussian processes on random graphs”. In *Advances in Neural Information Systems.* pp. 1723-1731.

Sorkine, O., and Alexa, M, (2007), “As-rigid-as-possible surface modeling”, *Proc. Eurographics/ACM Symposium on Geometry Processing*, Belyaev, A., and Garland, M, eds., pp. 109-116.

Spink, D.M., (2010), NURBS Toolbox, a Matlab file. `http://www.mathworks.com/matlabcentral/fileexchange/26390-nurbs-toolbox-by-d-m-spink`.

Sun, X., Rosin, P.L., Martin, R.R., and Langbein, F.C., (2009), “Noise Analysis and Synthesis for 3D laser Depth Scanners”, *Graphical Models*, 71(2), pp. 34-48.

Tenenbaum, J.B. , de Silva, V., and Langford, J.C. , (2000), “A global geometric framework for nonlinear dimensionality reduction”. *Science*, 290(5500), pp. 2319-2323.

Van Der Maaten, L., Postma, M.E., and van der Herik, J., (2009), “Dimensionality Reduction: A Comparative Review”, Tilburg Centre for Creative Computing, Tilburg University, `http://www.uvt.nl/ticc`

Weiss, V., Andor, L, Renner, G., and Varady, T., (2002), “Advanced surface fitting techniques”, *Computer Aided Geometric Design*, 19, pp. 19-42.

Xia, H., Ding, Y., and Wang, J., (2008), “Gaussian process method for form error assessment using coordinate measurements, *IIE Transactions*, 40, pp. 931-946.

Xia, H., Ding, Y., and Mallick, B.K., (2011), “Bayesian hierarchical model for combining misaligned two- resolution metrology data, *IIE Transactions*, 43, pp. 242-258.

Zhang, X.D., Fan,F.Z., Wang,H.B., and Hu,X.T., (2009).“Ultra-precision machining of sinusoidal surfaces using cylindrical coordinates”, *J. of Micromechanics and Microengineering*, 19, pp. 1-7.