# Geodesic Gaussian Processes for the 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

September 27, 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 based on 3-dimensional cloud point data. The surface is represented parameterically, with each of the three cartesian coordinates $(x, y, z)$ a function of surface coordinates $(u, v)$. This 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. In the proposed method, parameterization algorithms from the manifold learning and computer graphics literature are applied to find the $(u, v)$ surface coordinates. 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. The parametric surface representation is compatible with computer-aided-design (CAD) models and allows differential-geometry manipulations such as the computation of surface areas.

Keywords: Manifold data analysis, Anisotropic Covariance, Non-contact sensed data.

*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 $S$ 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 cloud point 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 cloud point 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.

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 has been reported to be gaussian-like and strongly spatially correlated (Sun et al., 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 models of cloud point surface data can be used for this purpose (Colosimo et al., 2012). 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 cloud point data, in order to create a Computer Aided Design (CAD) file.

Finally, while not our direct interest, a related problem occurs when modeling geostatistical data obtained from deposits whose directions follow non-euclidean paths (Boisvert et al., 2009).

The proposed Geodesic Gaussian Process (GGP) approach uses a *parametric representation* of the surface 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 $\widehat{z}(u, v)$. This is the preferred representation of surfaces in CAD (e.g see Patrikalakis and Maekawa, 2002) and CAD file standards (e.g., IGES). Our approach solves a dilemma one faces when modeling a surface using GP's or Kriging: it is not clear which of the 3 coordinates should be the 'response' and which ones the 'locations'.

The rest of this paper is organized as follows. Section 2 reviews related prior work on Gaussian Process modeling and cloud point 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 an example using a real data set obtained with a laser scanner. Section 7 discusses the problem of using noisy surface locations $(u, v)$ and presents an iterative version of the basic GGP approach to reduce the impact of error in the locations. The paper concludes with some general discussion and suggestions for further research. Appendices include details of the parameterization algorithms used, the software implementation of GGP, and an appendix on the differential geometry analysis of the surfaces fitted with GGP, including the problem of how to estimate the surface area of the underlying true surface.

# 2 Related prior work

Gaussian Processes have been used to model metrology data obtained via a coordinate measurement machine (CMM) by Xia, Ding et al. (2008 and 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). Pedone et al. (2009) have recently used universal Kriging for devising inspection sampling plans for one dimensional features of surfaces, such as circularity and straightness. Colosimo et al. (2012) use Gaussian Process models for process monitoring of manufactured surfaces.

There exists also considerable work on spatial models where the covariance is not only a function of the euclidean distance between points, the so-called anisotropic covariance case. This relates to our problem, since in surface data the non-euclidean 2-manifold where the data resides causes the anisotropy. 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 $f : \mathbb{E}^2 \rightarrow \mathbb{E}^2$ (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 to *extend* the mapping from the observed points to any other new point $\boldsymbol{w}$ at which it is desired to predict the response of interest. Boisvert and Deutsch (2011) use the Isomap algorithm (Tenenbaum et al., 2000) to find near geodesical distances in a mining application in order to allow Kriging predictions on a scalar spatial response. Their procedure requires considerable prior knowledge of the directions of the deposits in order to specify an anisotropic field (5 additional parameters per spatial cell are needed).

Other work that follows a space transformation strategy is by Schmidt and O'Hagan (2003), who present a bayesian approach to find a transformation $f : \mathbb{E}^2 \rightarrow \mathbb{E}^2$. They set the prior of $f$ as a Gaussian Process and use MCMC techniques for posterior inference. Kim et al. (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 $f : \mathbb{E}^2 \rightarrow \mathbb{E}^2$ 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 gaussian process. In contrast with these procedures, our geodesic Gaussian Process 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).

A different strategy for modeling anisotropy is to directly fit a non-stationary covariance model to the data, rather than transforming the 2D locations space and use stationary covariance models in the transformed space. Recent efforts along this line are Paciorek and Schervish (2008) who modify the Mahalanobis distance $|\boldsymbol{Ah}|$ by including the covariance matrices of a gaussian kernel centered at each point $i$. Jun and Stein (2008) propose non-stationary models for the specific case the process occurs on a 2-sphere.

Some early work on manifold learning, in particular, the Principal Surfaces method by Hastie (1984), a nonlinear generalization of principal components, is related to a part of our method. Similarly to the more general manifold learning methods which we use in section 4, this method provides a parameterization of a manifold, and hence dimensionality reduction, although it has some limitations with respect to dataset size and estimation (see Chang and Gosh, 1999). As it will be seen, finding geodesic distances on a surface is facilitated by a parameterization transformation (parameterization methods we tried and used are described in sections 4 and Appendix A).

We now present our proposed method, which falls in the first category of strategies mentioned above, that of space transformation methods.

# 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), 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 Gaussian Process, which makes up the "state" equation

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

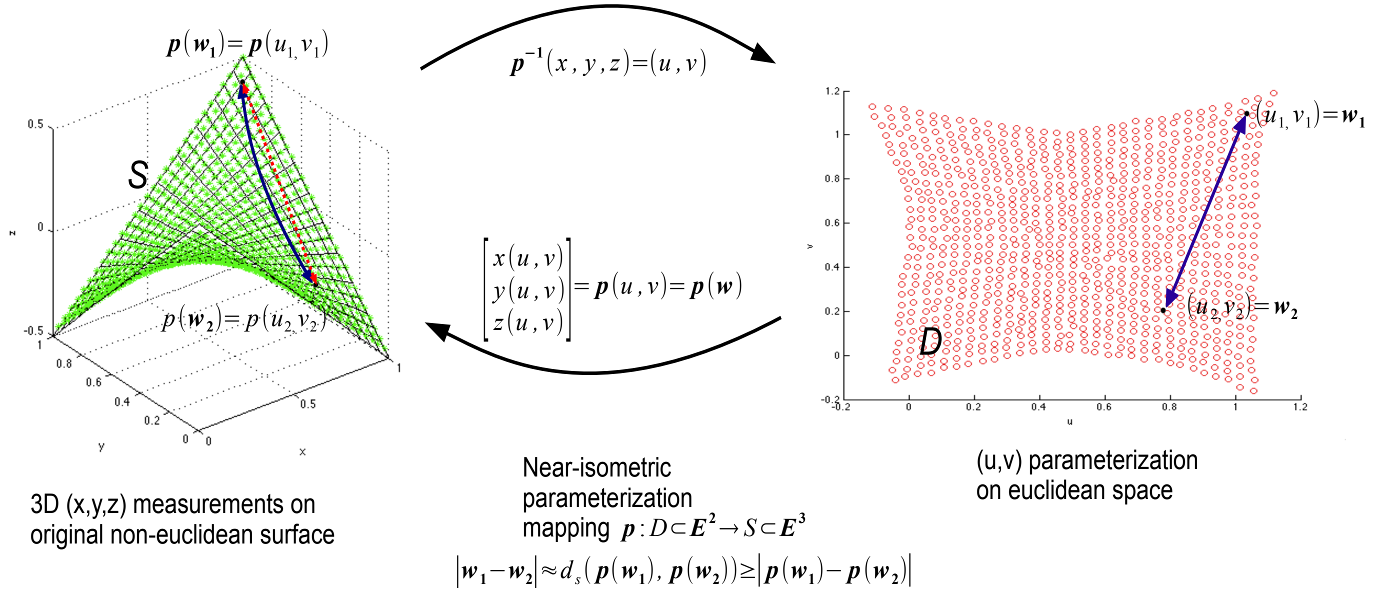
$p(w_1) = p(u_1, v_1)$
S
$p(w_2) = p(u_2, v_2)$
z
y
x
0.5
0
-0.5
1
0.8
0.6
0.4
0.2
0
0
0.5
1
3D (x,y,z) measurements on original non-euclidean surface
$p^{-1}(x, y, z) = (u, v)$
$\begin{bmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{bmatrix} = p(u,v) = p(w)$
Near-isometric parameterization mapping $p : D \subset E^2 \rightarrow S \subset E^3$
$|w_1 - w_2| \approx d_s(p(w_1), p(w_2)) \geq |p(w_1) - p(w_2)|$
$(u_1, v_1) = w_1$
$(u_2, v_2) = w_2$
D
v
u
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
(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 $S$. As it is well-known in cartography, an exact parameterization is not possible except for the case of developable surfaces. There exist several algorithms for finding near isometric mappings.

where

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

models long-range (systematic) variation and $\boldsymbol{\delta}(\boldsymbol{w})$ is a zero-mean, smooth (no-nugget), 3-dimensional vector stationary Gaussian Process 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 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 equation (3) models directly the manufactured surface $S$. Hereafter, we refer to equations (1-2) as the *Geodesic Gaussian Process* (or GGP) model. The main steps of this modeling methodology are shown diagrammatically in Figure 3.

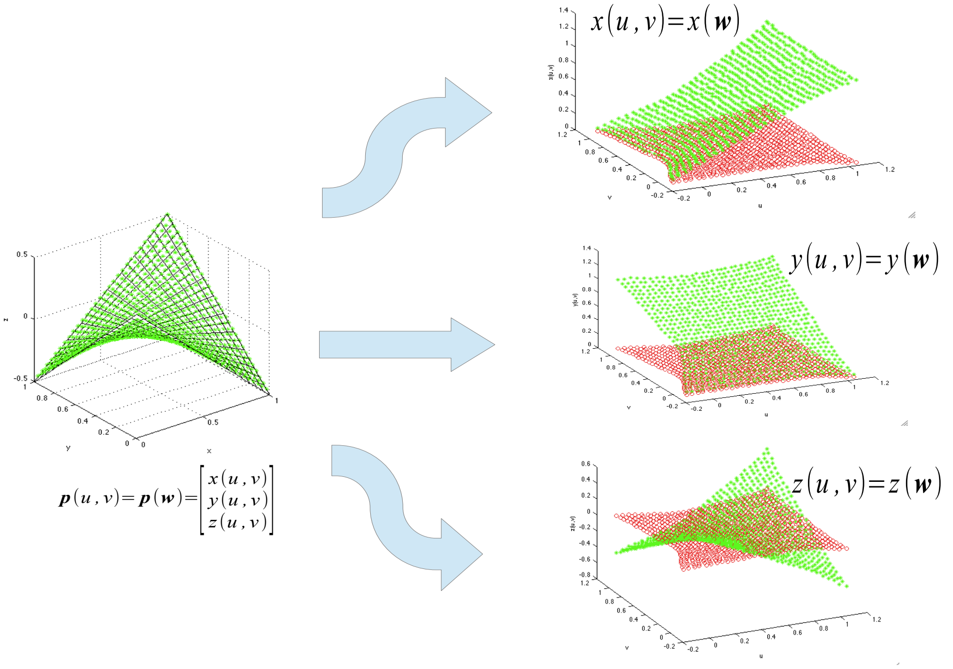

Figure 2: Points on a surface embedded in 3D euclidean space can be described in parametric form as a set of points $\mathbf{p}(u,v)$ where $(u,v)$ are coordinates described on the surface. With this parametric representation, the surface is decomposed into its three euclidean coordinate functions $x(u,v)$, $y(u,v)$ and $z(u,v)$, each described over the same space of coordinates $(u,v)$, shown as a plane of points in the right 3 graphs.

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

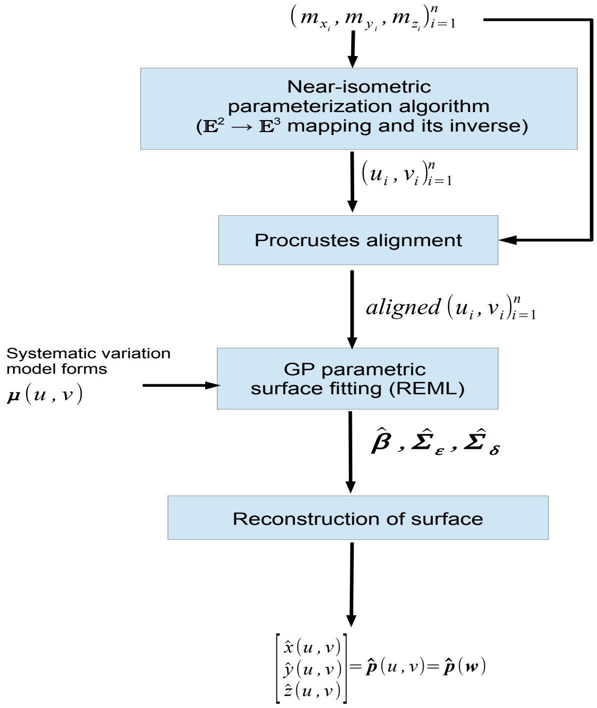
$(m_{x_i}, m_{y_i}, m_{z_i})_{i=1}^n$
Near-isometric parameterization algorithm ($\mathbf{E}^2 \rightarrow \mathbf{E}^3$ mapping and its inverse)
$(u_i, v_i)_{i=1}^n$
Procrustes alignment
*aligned* $(u_i, v_i)_{i=1}^n$
Systematic variation model forms
$\boldsymbol{\mu}(u, v)$
GP parametric surface fitting (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 Geodesic Gaussian Process (GGP) surface reconstruction method.

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 section 7.

# 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 this task. These include algorithms from the area of manifold learning, such as the Isomap method (Tenenbaum et al., 2000) and the LLE method (Roweis et al., 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 exist 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). These authors propose a "base surface" method, which has been influential in this literature. Their idea is to use a simple (base) surface that approximately resembles the surface one wishes to parameterize. The base surface is created interactively by selecting 4 corner points and fitting a NURBS surface. The parameterization is then obtained by closest-distance projection of the points on the surface of interest onto the base surface. Weiss et al. (2002) review other parameterization techniques used by CAD systems, and suggested using algorithms from the computer graphics literature for this task.

The main interest in the computer graphics literature is *texture mapping*, that is, to set a "texture" value (usually, RGB color values) at a particular location $(u_i, v_i)$ on a 3D surface $\boldsymbol{p}(u, v)$ in a scene in order to make it appear in the most realistic way.

There are many 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 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).

Most popular parameterization algorithms in the computer graphics literature find a conformal mapping, which has nice mathematical properties (Floater and Hormann, 2005);

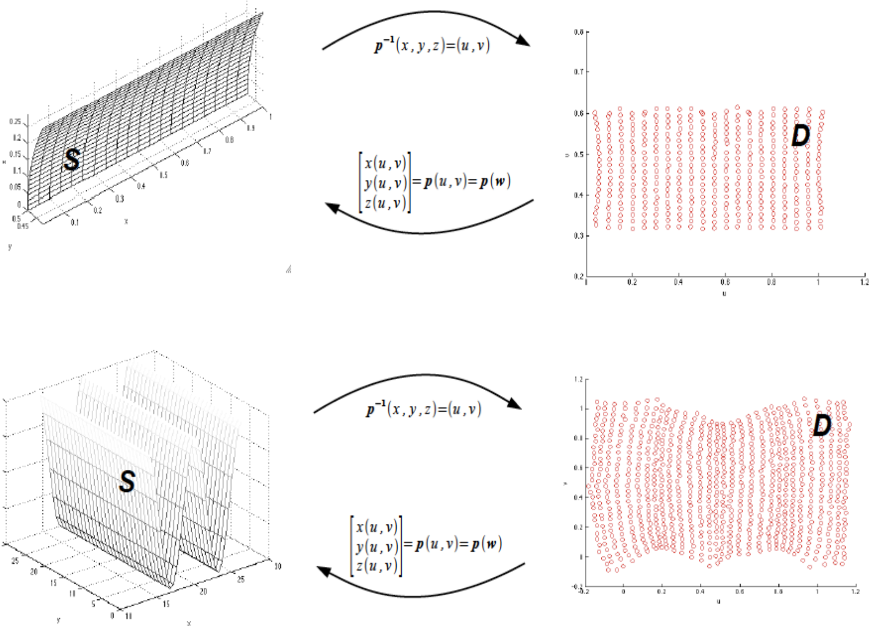

Figure 4: Near isometric parameterizations from noisy data obtained from two sample surfaces, a cylindrical patch (400 points) and a sinusoidal patch (900 points). The distances between pairs of points, $d_D(\boldsymbol{w}_i, \boldsymbol{w}_j)$ on the $uv$ plane $D$ on the right are approximately equal 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. These parameterizations were obtained using the ARAP algorithm (Liu et al., 2008), see Appendix A.

however, they 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 et al., 2008, Sorkine and Alexa, 2007, Deneger et al. 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 Appendix A. As it can be seen, both methods are near isometries, since the scatters are close

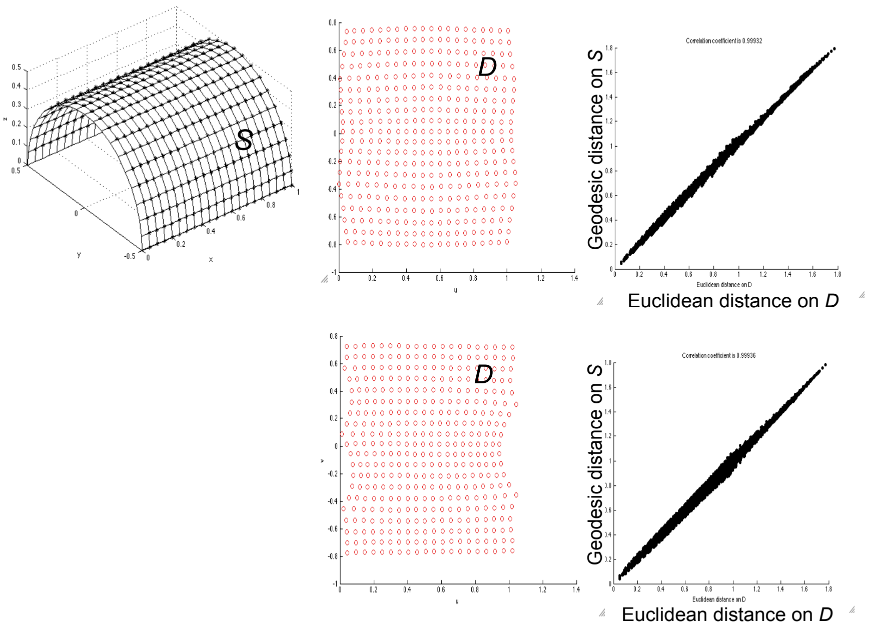

Figure 5: Parameterizations of noisy measurements from a cylinder based on 400 points using the Isomap algorithm (top) and ARAP algorithm (bottom). The scatter plots show the exact geodesic distance on the true underlying surface between all 79800 pairs of points plotted against the euclidean distance between the corresponding estimated $(u, v)$ points provided by each method. The estimated correlation coefficients of the scatter plots equal 0.9984 (Isomap) and 0.9953 (ARAP), indicating both mappings are near-isometric.

to a $45^{\text{o}}$ line (in view of equation (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 et al., 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 der Marteen et al. (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 et al. (2000), an assumption that is false if the surface has holes). ARAP scales 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 3 below.

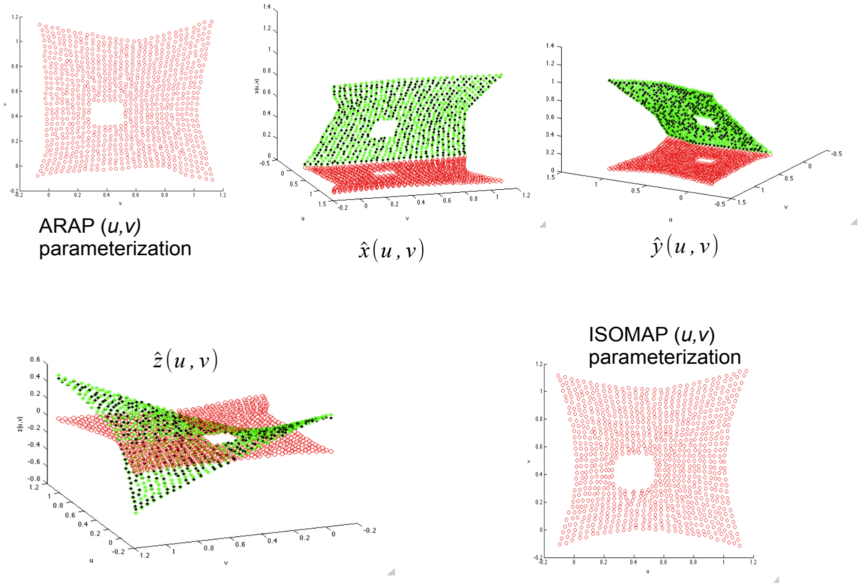

Figure 6: Parameterization of a bilinear NURBS surface patch with a rectangular hole. Whereas the ARAP parameterization preserves the geometry of the hole, the ISOMAP algorithm does not retain the rectangular features of the surface hole.

# 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 a Gaussian Process model 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 gaussian process. 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 to:

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

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 $C(\boldsymbol{w}, \boldsymbol{w}) = \operatorname{Cov}(\boldsymbol{p}(\boldsymbol{w}), \boldsymbol{p}(\boldsymbol{w})) = \operatorname{Var}(\boldsymbol{p}(\boldsymbol{w})) \in \mathbb{E}^{3\times 3}$ 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 et al., 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 eq. (2) and $\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{w}') = \mathrm{diag}(C_x(\boldsymbol{h}), C_y(\boldsymbol{h}), C_z(\boldsymbol{h}))$ in eq. (3) where $\boldsymbol{h} = \boldsymbol{w} - \boldsymbol{w}'$ (see conclusions section for more on this).

For each component $\delta_\bullet$ in equation (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 $\boldsymbol{\beta}_\bullet$ (eq. 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 $C_\bullet(\mathbf{0})$ 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 $f_\bullet(\boldsymbol{w})$ in eq. (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 et al. (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 Snag and Huang (2012) used in the laser scanner example of section 6.2

and further explained in Appendix B. Given the minimizing parameters $\boldsymbol{\theta}_\bullet$, we estimate $\boldsymbol{\beta}_\bullet$ from its generalized least squares estimator (Santner et al., 2002). 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 Gaussian Processes (Santner et al., 2003). The prediction equation for each true underlying surface component in $\boldsymbol{p}(u_0, v_0)$ is given by:

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

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 equation (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 ($\boldsymbol{m}(u_0, v_0)$).

Finding prediction error estimates $\hat{\sigma}_p^2$ can be done in the usual way by using the "plug in" standard error estimate, or a bootstrapping estimate as suggested by Der Hertog et al., (2006). This will provide a 3D prediction ellipsoid around a point prediction. A bootstrapping estimate would have to incorporate the imprecision in the near-isometric parameterization algorithm, which is not perfect. Some ideas on how to do this via cross-validation are given in the last section of the present paper.

# 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 (section 8 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 generated spatially correlated normal noise using a powered exponential correlation function, where euclidian distances (in the $D$ space) were used, as these correspond closely to the geodesic distances on $S$. The model fitting and prediction methods shown below were then applied in the GGP approach 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 gaussian process 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 $\phi_{\bullet} = 1$, $\sigma^2_{\bullet} = \tau^2_{\bullet} = 0.00001$. 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 Gaussian process 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 Appendix B 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 to make some general observations. As it can be seen from the table, in some cases, the parametric 3D predictions get closer to the true unknown surface than the simulated data, since the mean square error of the simulated points is higher than that of the parametric 3D predictions (thus, the proposed approach is effectively filtering the observational noise in the state-space model 2-3). The euclidean GP approach ($z(x, y)$) incurs in considerable higher prediction errors (around 50% compared to the parametric GGP predictions).

| $n$ | MSE$_{3D}$ | MSP$_{GGP}$ | 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 example, 30 simulations, mean and standard deviations of performance statistics (standard deviations inside parenthesis). MSP$_{GGP}$ is the per observation mean square 3D prediction error using the proposed GGP model (eq. 8), MPS$_{z(x,y)}$ is the corresponding error if an euclidean gaussian process is used (eq. 9). MSE$_{3D}$ is the simulated mean square error of the 3-dimensional points. If a surface reconstruction model gives a MSP smaller than MSE$_{3D}$, it is effectively "filtering" the measurement noise.

**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 parameters 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 $z(x,y)$.

The euclidean GP approach ($z(x,y)$) incurs in worse prediction errors compared to the parametric 3D predictions (Table 3). 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. This is probably due to the shorter inter-point distances which makes the effect of curvature less severe. 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.

| $n$ | $\text{MSE}_{3D}$ | $\text{MSP}_{GGP}$ | $\text{MSP}_{z(x,y)}$ |
|---|---|---|---|
| 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 example, 30 simulations, mean and standard deviations of performance statistics (standard deviations inside parenthesis). $\text{MSP}_{GGP}$ is the per observation mean square 3D prediction error using the proposed GGP model (eq. 8), $\text{MPS}_{z(x,y)}$ is the corresponding error if an euclidean GP is used (eq. 9). $\text{MSE}_{3D}$ is the simulated mean square error of the 3-dimensional points.

**Example 3.- a surface with a hole.** To study the effect of a hole on the GGP approach, consider the simulated NURBS depicted in Figures 1, 2, and 6. This is a NURBS bilinear

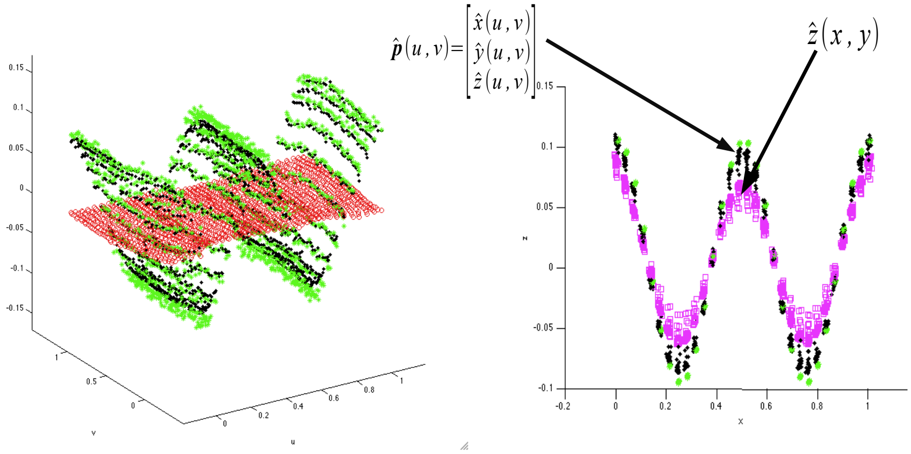

Figure 7: Left: Sinusoidal function observed points (light dots), GGP parametric surface predictions $\hat{\boldsymbol{p}}(u, v)$ (dark dots), $n = 400$ points. The plane of dots are the parameters $(u, v)$. Right: side view, showing also the predictions based on the Euclidean Gaussian Process, $\hat{z}(x, y)$, which badly fail to reconstruct the full depth of peaks and troughs.

surface with corner control points (0.0, 0.0, 0.5), (1.0, 0.0, -0.5), (0.0, 1.0, -0.5), and (1.0, 1.0, 0.5). Geodesically correlated Gaussian noise was simulated as described above with parameters $\phi_\bullet = 1$, $\tau^2_\bullet = \sigma^2_\bullet = 0.0001$ and $p_\bullet = 1$ (powered exponential correlation function) and added to the 400 3D points representing the surface. An interaction mean model was specified for $x(u, v)$ and $y(u, v)$ and a quadratic mean model was set for $z(u, v)$. We created a rectangular hole on this surface and applied the proposed GGP methodology to it. Figure 6 shows the 2D surface parameterization obtained with the ARAP algorithm and the 3 fitted parametric surfaces using GGP. In this particular case, the performance of predicting the three coordinates using GGP result in a mean square error of the predictions of 0.0073 that compares favorably with the prediction error from fitting the euclidean GP model $z(x, y)$ (which yields a mean squared prediction error of 0.0162). The simulated error MSE was 0.0072, so the GGP-fitted surface is quite precise, showing the procedure is unaffected by holes or gaps. As a comparison, the figure also shows the parameterization obtained with the ISOMAP algorithm (reducing the dimensionality from $\mathbb{E}^3$ to $\mathbb{E}^2$). As it can be seen, the rectangular feature of the hole is lost with this parameterization. Appendix A describes briefly the Isomap and the ARAP algorithms.

## 6.2 Reconstruction from real surface data: laser scanner data

In this section we reconstruct a surface from a cloud point dataset acquired with a structured light sensor first considered by Cavallaro et al. (2010) and further studied by

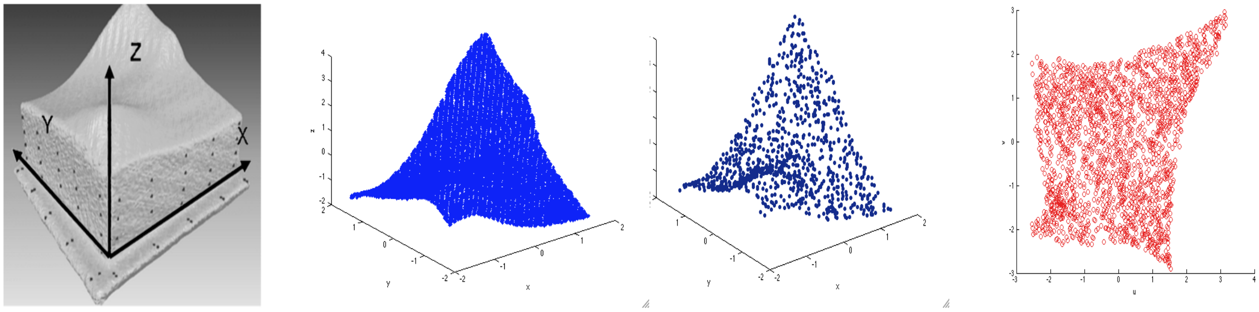

Figure 8: Laser scan data. From left to right: original artifact, full dataset ($n = 9635$), a decimated data set ($n = 964$), and on the right the $(u, v)$ parameterization of the decimated 3-dimensional data using the ARAP algorithm.

Colosimo and Pacella (2011). 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. 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 laser 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 24 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 Appendix B for computational details). The ARAP algorithm was used for the parameterization.

Table 4 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. This was done for each parametric surface in our model. Finally, we compute the mean square error of the predictions generated by GGP ($\text{MSP}_{GGP}$) and also by the euclidean GP method ($\text{MSP}_{z(x,y)}$, see section 2). As mentioned in section 4, since the underlying surface is unknown, the mean square error statistics (8-9) were computed by substituting the true surface points ($\boldsymbol{p}(u_i, v_i)$) with the observed measurements ($\boldsymbol{m}(u_i, v_i)$).

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 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 $\text{MSP}_{GGP}$ *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 $z(x, y)$.

Table 5 shows the parameter estimates when using the "interaction, interaction, quadratic" mean models for both $n_{fit} = 402$ and $n_{fit} = 964$. Here, $\hat{\tau}^2 + \hat{\sigma}^2$ represents a mix of model error and measurement error (or random noise due to other causes). If the true model for $\boldsymbol{p}(\boldsymbol{w})$ were known as in the previous simulated cases, then we could clearly identify these two variances with measurement (small scale) and spatial (larger scale) variability, but this type of knowledge is not practically available in the majority of applications. The estimated parameters do not change much as a function of $n_{fit}$. Since for $n_{fit} = 9000$ a different, approximated covariance function was used (see Appendix B), we do not include

| $n_{fit}$ | $n_{predict}$ | Mean models for $(x, y, z)$ | MSP$_{GGP}$ (MSP$_{z(u,v)}$) | 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 4: Cross-validation results for Laser scanner data, mean square errors performance statistics. Different mean models were used. 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. The prediction errors given by the GGP method are consistently about *half* of what an euclidean GP model provides for $z(x, y)$, regardless of the number of points used to fit the model.

the parameter estimates for this case in the table.

| $n_{fit}$ | $n_{predict}$ | $(\hat{\phi}, \hat{\sigma}^2, \hat{\tau}^2)_{x(u,v)}$ | $(\hat{\phi}, \hat{\sigma}^2, \hat{\tau}^2)_{y(u,v)}$ | $(\hat{\phi}, \hat{\sigma}^2, \hat{\tau}^2)_{z(u,v)}$ | $(\hat{\phi}, \hat{\sigma}^2, \hat{\tau}^2)_{z(x,y)}$ |
|---|---|---|---|---|---|
| 402 | 401 | 0.34, 1e-5, 0.7e-15 | 0.34, 0.0019, 0.13e-14 | 0.50, 0.0007, 0.4e-15 | 0.67, 0.0006, 0.5e-15 |
| 964 | 963 | 0.34, 0.0003, 0.3e-16 | 0.34, 0.0131, 0.7e-14 | 0.50, 0.0002, 0.5e-14 | 0.67, 0.5e-5, 0.7e-17 |

Table 5: Cross-validation parameter estimates for the GGP model, laser scanner data, for mean models: interaction for $x(u, v)$ and $y(u, v)$ and a quadratic model for $z(u, v)$ and $z(x, y)$.

# 7 Reducing the impact of measurement noise on the estimated $(u, v)$ surface locations

The GGP approach thus far presented takes 3D cloud point data $\{m_{x_i}, m_{y_i}, m_{z_i}\}_{i=1}^n$ generated by some sensor to estimate surface coordinates $\{\boldsymbol{w}_i = (u_i, v_i)\}_{i=1}^n$ from which gaussian process models are then fit to each spatial coordinate. Hence this approach is based on initial "noisy", or unreliable locations $\boldsymbol{w}_i$. If the surface location errors can be reduced or filtered, the reconstruction of the surface could in principle be improved over what was shown in early sections. This problem is related to work by Gabrosek and Cressie (2002) and Cressie and Kornak (2003), who proposed spatial statistical models in the presence of location errors. These authors concentrate on the case when there is a set of intended, known locations which differ from the actual locations where the process is observed. The assumption of known locations does not hold in the type of applications that motivated the present study, particularly non-contact sensed data. Fanshawe and Diggle (2011) propose a model where no intended set of locations is assumed. However, they point out how the location error variance is confounded with the nugget variance and the range parameters, and hence, they make the simplifying assumption that the location error variance is known from past history, again an assumption untenable for the type of applications we desire to

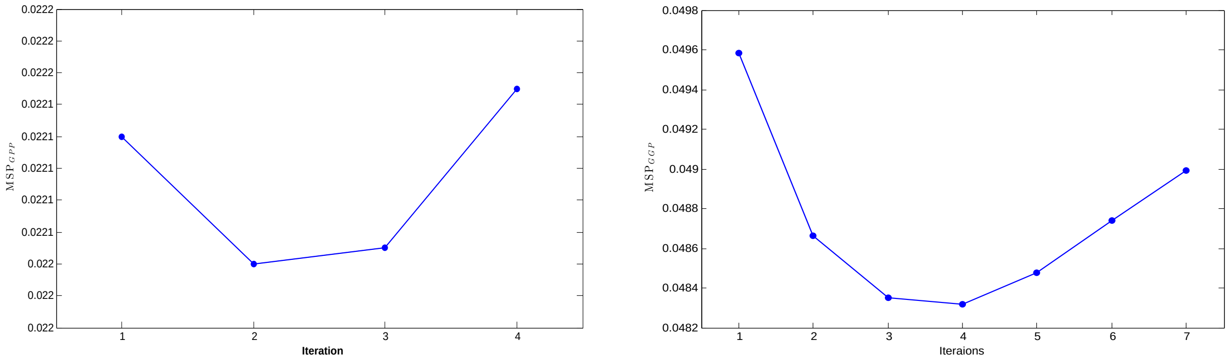

Figure 9: Mean squared prediction errors per observation for the iterative GGP. Left: for a simulated cylindrical surface patch with parameters $\phi = 1$, $\tau^2 = 0.0001$, $\sigma^2 = 0.0001$, $p = 1$ (same for all 3 coordinates). The simulated surface noise was $\mathrm{MSE}_{3D} = 0.02251 > \min \mathrm{MSP}_{GGP} = 0.02204$. The prediction error given by the euclidean GP approach was $\mathrm{MSP}_{z(x,y)} = 0.02647$. Right: for a simulated sinusoidal surface patch with parameters $\phi = 1$, $\tau^2 = 0.001$ , $\sigma^2 = 0$ (*no* spatial correlation, hence only uncorrelated measurement noise is present), $p = 1$ (same for all 3 coordinates). The simulated surface noise was $\mathrm{MSE}_{3D} = 0.05486 > \min \mathrm{MSP}_{GGP} = 0.048317$, while the prediction error given by the euclidean GP approach was $\mathrm{MSP}_{z(x,y)} = 0.050734$.

work in.

As an alternative solution to reduce the location error we tried the following iterative procedure: starting from the raw cloud point measurements, estimate initial surface coordinates as in the basic GGP approach of Figure 1. We then use the *reconstructed surface* $\{\widehat{x}(\boldsymbol{w}_i), \widehat{y}(\boldsymbol{w}_i), \widehat{z}(\boldsymbol{w}_i)\}$ as the data in a parameterization algorithm to provide revised locations $\{\boldsymbol{w}_i^{(\mathrm{rev})} = (u_i^{(\mathrm{rev})}, v_i^{(\mathrm{rev})})\}$. The GGP models for each coordinate are then fitted again at these revised locations (using the original data $x_i, y_i, z_i$ as responses), from where revised predictions are obtained to provide a new surface reconstruction, in a process that can be iterated. Given that the GGP predictions are less noisy than the original observations (provided the surface patch is developable), as was demonstrated above, the revised surface coordinates will be more precise, and will yield a more precise surface reconstruction. Figure 9 shows the sequence of MSP iterations for two examples using the iterative GGP approach.

As it can be seen from Figure 9, the iterative GGP reduces the mean squared prediction error after a few iterations compared to the standard (non-iterative) GGP, i.e., that given in iteration 1, and seems to be convex in the number of iterations. Figure 9 shows an example for the cylindrical patch used earlier, where there is significant spatial correlation on the surface, followed by the behavior for a case where there is *no* spatial correlation at all in the sinusoidal patch used earlier, and hence the cloud point data simply contains iid measurement noise on all coordinates. Remarkably, in this case not only the standard GGP provides a better reconstruction of the surface than the euclidean GP approach ($\text{MSP}_{z(x,y)} = 0.050734 > \text{MSP}_{GGP} = 0.0495$ at iteration 1) but the iterative procedure is able to further improve the reconstruction by approaching more closely the true surface $S$, with a minimum MSP reached after 4 iterations. Here the better reconstruction of the GGP approach can be explained by the superior parametric surface model approach it adopts.

# 8 Discussion and conclusions

A new parametric surface approach for the statistical reconstruction of surfaces embedded in 3-dimensional space based on cloud point data was presented. The Geodesic Gaussian Process (GGP) method first finds a parameterization on the surface patch under study and then fits spatial Gaussian Process models on each of the three cartesian coordinate 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" (usually, $(x, y)$) that one faces when using a standard kriging model for surface data. This parametric surface modeling approach is compatible with CAD models. 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 worst 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. An iterative GGP method was presented where the noise in the surface parameters is considered and this further improves the predictions.

From an industrial application point of view, it is of interest to determine if for a variety of real-life machined surfaces the "geodesic hypothesis" holds, i.e., determine whether correlation can be better modeled geodesically or over euclidean distances. A study similar to the laser data of section 6.2 could be conducted with this aim (there it was found that our GGP model which assumes correlations over geodesic distances fits better). We have found this hypothesis true also in a CMM dataset of the artifact depicted in Figure 8, but more empirical testing seems necessary.

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 with less curvature. There is a great number of algorithms to do this 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. We also plan to integrate GGP and CAD models in the near future.

**Acknowledgements**. The authors wish to thank an anonymous referee, 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.

## Appendix A.1. The Isomap algorithm

Assume we wish to find a function $\boldsymbol{p}(\boldsymbol{w})$ from $\mathbb{E}^{l_1}$ to $\mathbb{E}^{l_2}$ where $l_1 < l_2$. The Isomap algorithm (Tenenbaum et al., 2000) first approximates the geodesic distances $d_g(\boldsymbol{p}(\boldsymbol{w}_i), \boldsymbol{p}(\boldsymbol{w}_j))$ by connecting the points $\boldsymbol{p}(\boldsymbol{w}_i)$ with its closest $q$ neighbors, where $q$ is a parameter that can be tuned depending on the noise level of the data. These neighbors are represented by a weighted graph over the points, with edges equal to the *euclidean* distances $|\boldsymbol{p}(\boldsymbol{w}_i) - \boldsymbol{p}(\boldsymbol{w}_j)|$. The basic idea exploited in this and many similar "manifold learning" algorithms is precisely the definition of a manifold: an abstract space (perhaps of high dimension) which in small neighborhoods can be approximated well with an euclidean space. To compute the geodesic distance matrix $\boldsymbol{D}_S$ over all points, Isomap finds shortest paths between all points in the graph made up of all neighborhoods, a problem usually solved with Dijkstra's algorithm. These shortest distances are the approximated geodesic distances. Isomap then minimizes the Frobenius norm of $\boldsymbol{D} - \boldsymbol{D}_S$ with respect to the vectors $\boldsymbol{w}_i$, where $\boldsymbol{D}$ is the matrix of euclidean distances. This multidimensional scaling problem requires the eigendecomposition of $\boldsymbol{D}_S$, which is an $O(l_2^3)$ operation. Thus, for a large number of points $l_2$, Tannebaum et al. (2000) proposed an alternative "Landmark Isomap" algorithm which computes distances over a subset of $l_3 < l_2$ points (landmarks) only. Provided the domain $D$ of $\boldsymbol{p}(\boldsymbol{w})$ is convex and developable and the points come from a uniform distribution on the manifold, the authors prove the asymptotic convergence of Isomap to a perfect isometry (Bernstein et al., 2000). The algorithm is very useful for dimensionality reduction ($l_2 >> l_1$), since the objective can be minimized for different dimensions $l_1$ and the "optimal" reduced dimension can be obtained by choosing $l_1$ such that $|\boldsymbol{D} - \boldsymbol{D}_S|_F$ is smallest (where $|\cdot|_F$ denotes Frobenius norm). In our parameterization application, $l_1 = 2$ and $l_2 = 3$ are fixed instead.

## Appendix A.2. The "As-Rigid-As-Possible" (ARAP) algorithm

Suppose we have a triangulation mesh of 3D points where triangles are numbered 1 to $T$ and the areas of each triangle denoted by $A_t$. The goal is to find a parameterization from the 2D plane $uv$ to the 3D mesh. For each triangle, denote by $\boldsymbol{w}_t = \{\boldsymbol{w}_t^0, \boldsymbol{w}_t^1, \boldsymbol{w}_t^2\}$ its 2D vertices on $D$ (to be determined) and by $\boldsymbol{x}_t = \{\boldsymbol{x}_t^0, \boldsymbol{x}_t^1, \boldsymbol{x}_t^2\}$ its 2D vertices defined on the plane *local* to each triangle embedded in $\mathbb{E}^3$. The mapping between $\boldsymbol{x}_t$ and $\boldsymbol{w}_t$ has a $2 \times 2$ Jacobian $J_t(\boldsymbol{w})$ for each triangle which models the distortion it goes through. The ARAP algorithm (Liu et al., 2008) tries to minimize the area distortions:

$$\sum_{t=1}^{T} A_t |J_t(\boldsymbol{w}) - \boldsymbol{L}_t|_F^2 \tag{10}$$

where $\boldsymbol{L} = \{\boldsymbol{L}_1, ..., \boldsymbol{L}_T\}$ is a set of rotation matrices to be applied to each triangle. The algorithm is then trying to find a rigid transformation and the locations of the 2D vertices $\boldsymbol{w}_t$ such that areas are distorted as least as possible when the mesh is flattened. The optimization is performed in 2 stages following an approach by Sorkine and Alexa (2007). First, the rotation matrices $\boldsymbol{L}_t$ are found for fixed $\boldsymbol{w}'s$ (this involves a SVD factorization per triangle); then the $\boldsymbol{w}'s$ are found for given $\boldsymbol{L}$'s (this involves solving a linear system). The process is repeated until a convergence criterion is reached. Liu et al. (2008) show how (10) can be modified to add a term that considers also angle distortions, but from their numerical experiments, and ours, this is not necessary: the angles are preserved quite well so the resulting parameterization is close to an isometry. An inital parameterization is required to start the ARAP algorithm; following Liu et al. (2008) we used the Least Squares Conformal Map (LSCM) which provides a quick (conformal) mapping. Figure 10 shows the different steps of the ARAP algorithm applied to the 398 point CMM dataset of section 7. Note how a procrustes alignment step is required after the algorithm ends in order to align the parameterization $(u_i, v_i)$ with the original data $(x_i, y_i, z_i)$. Since only small ($2 \times 2$) SVD factorizations are required to solve (10), this algorithm scales much better with the size of the dataset than the standard Isomap method.

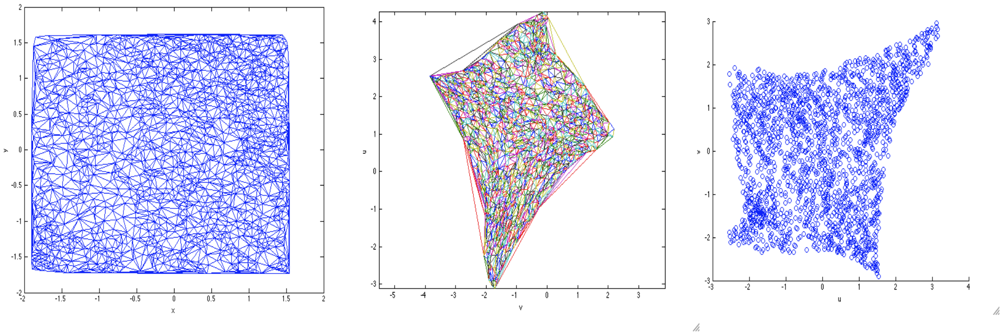

Figure 10: Illustration of the ARAP algorithm for the CMM data of section 7. Left: the triangulation of the original 3D data projected over the 2D x-y plane. Center: the 2D triangulation after rigid transformations are applied. Note how the triangles on the "tail" are most distorted. Right: a final procrustes alignment step is required to keep the orientation of the resulting $(u, v)$ mapping with respect to the original 3D data.

## Appendix B. Software implementation

We have used Matlab throughout this research due its support for graphics and optimization techniques. We made use of its built-in triangulation routines and found the MATLABMesh toolbox (Schmidt, 2009) particularly useful. The NURBS surfaces used in this work were created with the NURBS toolbox (Spink, 2010). The manifold learning algorithms listed in Table 1, including Isomap, where run using the routines written by der Mateen et al. (2009). Our ARAP routine was adapted from that by Liu et al. (2008). When $n_{fit} \leq 1600$, no covariance approximation was done and the full covariance matrix was used in the REML routine. However, when the number of points to fit the model is large, the suggested REML routine requires the inverse and determinant of a $n \times n$ matrix, and this requires special handling to overcome the computational burden, which for the inverse has $O(n^3)$ complexity. There is a large body of recent literature that address the "big n" problem in Gaussian Process model fitting. For fitting the models in the laser scanner example ($n_{fit} = 9000$) we used the method due to Sang and Huang (2012) to approximate the $9000 \times 9000$ covariance matrix. This method partitions $\boldsymbol{\delta}(\boldsymbol{w})_\bullet$ in equation 3 into $\boldsymbol{\delta}_1(\boldsymbol{w})_\bullet + \boldsymbol{\delta}_S(\boldsymbol{w})_\bullet$ where $\boldsymbol{\delta}_1(\boldsymbol{w})_\bullet$ is a low rank approximation of $\boldsymbol{\delta}(\boldsymbol{w})_\bullet$ based on $m$ "knots", used to model long-range correlations throughout the space $D$, and $\boldsymbol{\delta}_S(\boldsymbol{w})_\bullet = \boldsymbol{\delta}(\boldsymbol{w})_\bullet - \boldsymbol{\delta}_1(\boldsymbol{w})_\bullet$ is the residual of this approximation for which a taper function is applied in order to model small-range correlation. While we still used an exponential spatial covariance model with parameters $\phi_\bullet, \sigma^2_\bullet, \tau^2_\bullet$, $\bullet = \{x, y, z\}$, the resulting approximated covariance model differs from that used when no approximation to the covariance matrix was done (equation 6). Throughout the REML optimization, sparse matrix techniques were used, in particular, the `sparseinv.m` function (Davies, 2011) was used to compute the required inverses. For the laser scan data where $n_{fit} = 9000$, we used $m = 576$ equidistant knots over a grid in $D$ and a spherical tapering function with $\gamma$ =0.01 resulting in a sparsity ratio of 0.0025. For large $n$, the GGP model allows straightforward parallelization, since the models for $x(u,v)$, $y(u,v)$ and $z(u,v)$ can be fit in different CPUs in parallel once the $(u,v)$ parameterization is available, since it is used by all 3 models and the coordinate response models are assumed independent. We used a cluster of Intel Xeon 3.0 GHz processors operating in batch to fit each coordinate model separately. Fitting the 9000 point laser data set took approximately 4.5 hr of a single CPU (core) time for each of $x(u,v)$ and $y(u,v)$ and about 100 minutes for $z(u,v)$, the significant difference in time probably due to the different loads in the CPU's being used at the time (the euclidean GP model $z(x,y)$ took also about 4.5 hr. using the same sparsification techniques as for the GGP model).

The Matlab code that implements the GGP modeling of all the examples in this paper is available upon request from the first author. Matlab's Optimization, Global Optimization, and Statistics toolboxes are necessary.

## Appendix C. Differential-Geometrical analysis of the fitted surface

A benefit of the parametric surface model approach is that it allows estimation of lengths, angles and areas on the true underlying surface in the presence of measurement error (on all 3 coordinates), in contrast to finding the same surface characteristics using a triangulation of the observed data (which simply either neglects measurement noise or filters the data disregarding any statistical modeling). Here we show how to estimate the surface area.

Assume we have fitted $\hat{\boldsymbol{p}}(u,v) = (\hat{x}(u,v), \hat{y}(u,v), \hat{z}(u,v))'$, the Gaussian Processes of each of the parametric components coordinates (1). Define the vectors of partial derivatives (where in the rest of this section we omit the "hat" used to denote estimates of the reconstructed surface):

$$\boldsymbol{d}_1(u,v) \equiv \frac{\partial \boldsymbol{p}(u,v)}{\partial u}, \quad \text{and} \quad \boldsymbol{d}_2(u,v) \equiv \frac{\partial \boldsymbol{p}(u,v)}{\partial v}$$

Then the coefficients of the first fundamental form of $S$, used to determine its surface area, are equal to (see Kreyszig, 1991):

$$g_{11} = |\boldsymbol{d}_1|^2, \quad g_{12} = \boldsymbol{d}_1 \boldsymbol{\cdot} \boldsymbol{d}_2, \quad g_{22} = |\boldsymbol{d}_2|^2$$

and the area of $S$ defined parametrically on the domain $D$ of the $uv$ plane is given by

$$A(S) = \iint_D |\boldsymbol{d}_1 \times \boldsymbol{d}_2| \, du \, dv = \iint_D \sqrt{g} \, du \, dv \tag{11}$$

where $g = g_{11}g_{22} - g_{12}^2$ is the discriminant of the first fundamental form. It can be shown, after some algebra, that for the Gaussian Processes assumed in this paper (for $p_\bullet = 1$), all the partial derivatives in $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ evaluated at an specific pair $(u_0, v_0)$ have the form:

$$\frac{\partial p_\bullet}{\partial u_0} = \frac{\partial}{\partial u_0}(\boldsymbol{f}_\bullet(u_0, v_0)'\boldsymbol{\beta}_\bullet) + \sigma^2 \sum_{i=1}^{n} \frac{\partial S_{i\bullet}}{\partial u_0}, \qquad \bullet \in \{x, y, z\}$$

and

$$\frac{\partial p_\bullet}{\partial v_0} = \frac{\partial}{\partial v_0}(\boldsymbol{f}_\bullet(u_0, v_0)'\boldsymbol{\beta}_\bullet) + \sigma^2 \sum_{i=1}^{n} \frac{\partial S_{i,\bullet}}{\partial v_0}, \qquad \bullet \in \{x, y, z\}$$

where

$$\frac{\partial S_{i,\bullet}}{\partial u_0} = \phi_\bullet(u_i - u_0)\gamma_{i,\bullet} \frac{\exp(-\phi_\bullet|(u_0, v_0) - (u_i, v_i)|)}{|(u_0, v_0) - (u_i, v_i)|}, \qquad i = 1, 2, ..., n,$$

and

$$\frac{\partial S_{i,\bullet}}{\partial v_0} = \phi_\bullet(v_i - v_0)\gamma_{i,\bullet} \frac{\exp(-\phi_\bullet|(u_0, v_0) - (u_i, v_i)|)}{|(u_0, v_0) - (u_i, v_i)|} \qquad i = 1, 2, ..., n$$

where $\gamma_{i,\bullet}$ is the ith entry of the vector $\boldsymbol{\gamma}_\bullet \equiv \Sigma_y^{-1}(\mathcal{M}_\bullet - \boldsymbol{F}_\bullet\boldsymbol{\beta}_\bullet)$. Integral (11) can be evaluated numerically for the values $(u_0, v_0)$ in the domain $D$ of the parameterization mapping. To illustrate, consider the half cylinder of Figure 5. The exact surface area in this case is $\pi/2 = 1.5708$. We simulated 400 and 900 points with low noise ($\sigma_\bullet^2 = \tau_\bullet^2 = 0.00001$) on this surface and use the Isomap parameterization. The estimated areas, using both a triangulation of the points and using equation (11) are shown in Table 6. As $n$ increases, the quality of the area estimates improves, with the surface integral being better for this case, although this is at a computational cost. Depending on the domain of the surface, doing a coordinate transformation may be better if the integral (11) is used, as usually done in numerically integration. Integrating over a irregular boundary will provide much more imprecise results (for both methods), as typically happens in numerical multiple integration. In general, computing areas via a triangulation is less robust to noise, since the GGP models are "filtering" the measurement error, although it is considerably faster.

## 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.

| No. of points | Triangulation | Surface integral (eq. 11) |
|---|---|---|
| 400 | 1.7398 | 1.6040 |
| 900 | 1.6879 | 1.5934 |

Table 6: Surface area estimates for half cylinder based on different number of points. True area of noise-free surface is $\pi/2 = 1.5708$.

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`, (retrieved 4/1/2012).

Boisvert, J.B., Manchuck, J.G., and Deutsch, C.V., (2009), "Kriging in the presence of locally varying anisotropy using non-euclidean distances", *Mathematical 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.

Chang, K-Y., and Ghosh, J., (2001). " A Unified Model for Probabilistic Principal Surfaces", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 23(1), pp. 22-41.

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

Colosimo, B.M., Pacella, M., and Vlaco, M., (2012), "From profile to surface monitoring: SPC for cylindrical surfaces via Gaussian Processes, Technical Paper, Politecnico di 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`. Retrieved January 2013.

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

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` (Retrieved April 2, 2012).

Donoho, D.L., and Grimes, C., (2005), "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data". *Proceedings of the National Academy 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.

Hastie, T., (1984). “Principal Curves and Surfaces”, Technical Report, STanford Linear Accelerator Center, Stanford University, Stanford, CA.

Jun, M., and Stein, M. L., (2008), “Nonstationary Covariance Models for Global Data”, *The Annals of Applied Statistics* 2(4), pp. 12711289.

Kim, H-M., Mallick, B.K., and Holmes, C.C., (2005), “Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes”, *Journal of the American Statistical Association*, 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”, *Computer Aided Design*, 27(9), pp. 663-675.

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

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

Paciorek, C.J., and Mark J. Schervish, M.J., (2006), “Spatial modelling using a new class of nonstationary covariance functions”, *Environmetrics*, 17, pp. 483506.

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

Pedone, P., Vicario, G. and Romano, D., (2009), “Kriging-based sequential inspection plans for coordinate measuring machines", *Appl. Stochastic Models Bus. Ind.*, 25, pp. 133149.

Roweis, S.T., andSaul, 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", *Journal of the American Statistical Association*, 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. and 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/`. (Retrieved 4-1-2012).

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

Sorkine, O., and Alexa, M, (2007), "As-rigid-as-possible surface modeling", *Proceedings of the 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`. Retrieved February 2012.

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.

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., Wei, G.S., and Hu, X.T., (2009), “Ultra-precision machining of sinusoidal surfaces using the cylindrical coordinate”, *J. of Micromechanics and Microengineering*, 19, pp. 1-7.