POLITECNICO DI MILANO
DEPARTMENT OF MECHANICAL ENGINEERING
DOCTORAL PROGRAMME IN MECHANICAL ENGINEERING

Data Fusion for Process Optimization and Surface Reconstruction

Doctoral Dissertation of:

Luca Pagani

ID number: 785772

Supervisor:
Prof. Bianca Maria Colosimo

Tutor:
Prof. Alfredo Cigada

The Chair of the Doctoral Program:
Prof. Bianca Maria Colosimo

2014 - XXVII Cycle

Acknowledgement

I would like to express my gratitude to Professor Xiangqian (Jane) Jiang for hosting me for 6 months in her research group at Huddersfield university. I would like to thank the EPSRC centre for innovative manufacturing in advance metrology group for letting me work aside them, have been a great experience that changed my personal and professional mindset.

Contents

Introduction

Last years are characterize by the impact of global trends and manufacturing technology is faced with a number of different challenges. The topics of customized products, reduced life time, increased product complexity and global competition have become essential in production today. The changing in modern manufacturing towards a customized productions is characterized by high-variety and high-quality products, a paradigm shift in metrology is coming on (Jiang et al. 2007a; Jiang et al. 2007b). Geometric tolerances are more and more often used in technical drawings to provide a more comprehensive way for defining allowable variation for a given product geometry. Globalization has made possible for production to be linked on a worldwide basis, in which the exchange of information is becoming more and more important in securing quality. Much of this information, particularly that concerned with the state of the products and production processes, is obtained with the aid of metrology.

Due to the paradigm shift, the complexity and accuracy of the product requirements are increasing. At the same time, smart sensorization of equipments and processes is providing new opportunities, which have to be appropriately managed. A large amount of data can be available to aid production and inspection and appropriate methods to process the "big data" have to be designed.

In this scenario, multisensor data fusion methods can be employed to achieve both holistic geometrical measurement information and improved reliability or reduced uncertainty of measurement data to an increasing extent (Weckenmann et al. 2009).

Research purpose

The main purpose of this thesis is to explore new approaches for reconstructing a surface starting from different sources of information, which have to be appropriately fused. Surface is meant in a broad sense, both as the geometric pattern of a physical object to be inspected and as the surface representing a response function to be optimized.

The first part of the PhD thesis focuses on the reconstruction of the surface geometry via data fusion. In this case, it is assumed that multiple sensors are acquiring the same surface, providing different levels of data density and/or accuracy/precision. The thesis starts exploring the performance of a two-stage method, where Gaussian Processes (also known as kriging) are appropriately used as modeling tool to combine the information provided by the two sensors. Then, the thesis faces the problem of suggesting a data fusion method when large point clouds, i.e., "big data" (as the ones commonly provided by non contact measurement systems) have to be managed. In this case, the use of Gaussian Processes poses some computational challenges and this is why a different method based on multilevel B-spline is proposed. As a second contribution, the thesis presents a novel method for data fusion, where the uncertainty of the specific measurement system acquiring data is appropriately included in the data fusion model to represent the uncertainty propagation. Eventually, the thesis faces the problem of using surface modeling to quickly detect possible out-of-control states of the machined surface. Starting from a real case study of laser-textured surface, an approach to combine surface modeling with statistical quality control is proposed and evaluated.

The second part of the PhD thesis focuses on using data fusion for process optimization. In this second application, data provided by computer simulations and real experiments are fused to reconstruct the response function of a process. In this case, the aim is to find the best setting of the process parameters to maximize the process performance. It is shown how data fusion can be effectively used in this context to reduce the experimental efforts.

Part I

Surface Reconstruction

Chapter 1

Data Fusion via Gaussian Process

In this chapter a data fusion model based on Gaussian process is applied to the reconstruction of smooth free form surfaces. In order to explain why data fusion can be effective in the actual industrial scenario, this chapter will start with an overview of some modern measurement devices. Then, the use of Gaussian processes for metrology data fusion is described. Finally, the advantages provided by multisensor data fusion are shown.

1.1 Measurement Devices

A measurement device is an instrument for measuring a physical quantity. Measurement is defined as the activity of obtaining and comparing physical quantities of real-world objects.

Measurement characteristics are discussed in the International Vocabulary of Metrology (VIM 2008):

- *measurement accuracy*: *closeness of agreement between a measured quantity value and a true quantity value of a measurand.* This concept is not a quantity and is not associated to a numerical value. A measurement is said to be more accurate when it offers a smaller measurement error.

- *measurement trueness*: *closeness of agreement between the average of an infinite number of replicate measured quantity values and a reference quantity value.* Measurement trueness is not a quantity and thus cannot be expressed numerically, but measures for closeness of agreement are given in ISO5725 (e.g., mainly expressed in terms of bias).

- *measurement precision*: *closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions.* Measurement precision is usually expressed numerically by measures of imprecision, such as standard deviation, variance or coefficient of variation under the specified measurement conditions.

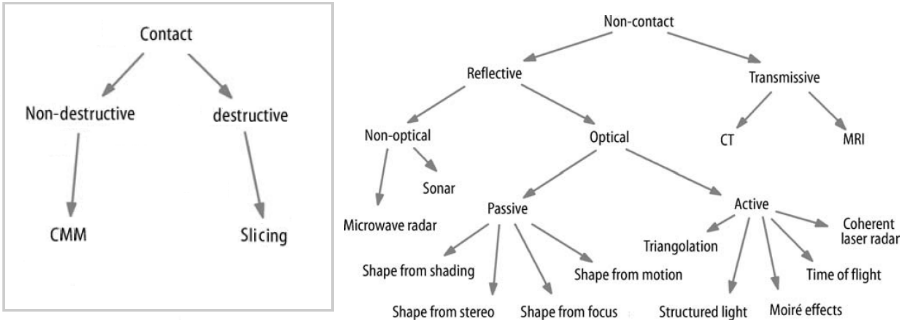
- *resolution*: *the smallest change in a quantity being measured that causes a perceptible change in the corresponding indication.*

1.1.1 Classification of Measurement Devices

In the following sections, the main advantages and disadvantages of contact and non contact sensors will be briefly presented to motivate the need of data fusion. A more detailed description of some common measuring sensors will be provided while presenting the real case studies throughout the thesis. Weckenmann et al. (2009) provides an overview on different methods for geometric data acquisition, the main classification is based on the way the measurement is taken, i.e., considering the interaction between the device and the surface or volume of interest. Instruments are divided into two sets: contact and non contact measurement systems. A hybrid category also exists, where contact and non contact devices cooperate in order to perform a better surface acquisition. A simple classification of methods is represented in Figure 1.1.

1.1.2 Contact Devices

Tactile devices are based on the principle of touching specific points of a surface using a mechanical arm and use sensing devices in the joints of the arm in order to determine the coordinate locations of the touched point. Contact methods have usually high performance (i.e., high precision, high accuracy, high repeatability, etc.), but are usually applied in cases where surface measurement requires acquiring a small set of data point, because the time to acquire each single point is usually long.



(a) Contact methods **(b)** Non contact methods

Figure 1.1: Classification of 3D acquisition systems (Rocchini et al. 2001)

1.1.3 Non Contact Devices

Non contact measurements systems allow one the acquire the point cloud on a surface without requiring mechanical interaction between the sensor and the object.

The point cloud can be computed with a variety of optical methods as triangulation, ranging, interferometry, structured lighting and image analysis.

A classification of non contact methods based on sensor technology can be found in (Tamas, Ralph, and Jordan 1997), while a classification based on data acquisition techniques is proposed in (Rocchini et al. 2001).

A further classification of optical devices is proposed by Isgro, Odone, and Verri (2005), where the systems are classified in active and passive sensors. Passive sensors use only natural or man-made illumination, while active sensors project a source of energy onto the object. They allow more accurate measurements of depth when compared to passive sensors.

Compared to contact devices, the main advantages of non contact sensors are:

- no physical contact is required, so fragile workpiece can be measured without damaging the object;
- fast acquisition of large objects;
- ability to acquire colors;
- ability to acquire complex shapes on large sizes where, for example, a contact probe can have problems to reach some features on large measurand.

On the other side, disadvantages are:

- possible limitations for transparent or reflective surfaces, holes and undercuts;
- performance influenced by the light conditions;
- lower accuracy, compared with the one of contact methods.

1.2 Multiple Sensor Data Fusion

Both contact and non contact devices have some advantages and disadvantages, so there is no technology that outperforms the other. As a consequence, the device to be used depends on the specific application. In the following, the term High-Fidelity (Hi-Fi) and Low-Fidelity (Lo-Fi) will be used to refer to data provided by contact and non contact sensors, respectively.

In modern manufacturing technology, the manufactured surfaces have often complicated features, designed to meet functional specifications. In this competitive environment the ability to rapidly design, produce and check the specifications of these pieces is becoming essential. To this aim, measurement with more than one sensor can be effective.

The main idea of a multiple sensor device is to use different sensors to acquire the same object, a classification of multiple sensor systems according to the way the information contribute to the final objective can be found in Girao, Pereira, and Postolache (2006):

- complementary: the sensors are not directly dependent but their output can be combined to yield better data, under a pre-defined criterion;
- competitive: the sensors provide independent measurements of the same quantity. They can be identical or can use different measuring methods and they compete in the sense that when discrepancies occur, a decision on which to believe must be made. Sensors are put in competition either to increase the reliability of a system or to provide it with fault tolerance;
- cooperative: the sensors' outputs are combined to produce information that is unavailable from each sensor if used alone;

- independent: all the remaining cases, basically when the sensors' outputs are not combined.

The process to combine multiple data from different sensors is defined in (Weckenmann et al. 2009) as *combining data from several sources (sensors) in order that the metrological evaluation can benefit from all the available sensor information data*. According to Weckenmann et al. (2009) the steps needed to combine data coming from multiple sensors are:

1. pre-processing: the data are transformed in the required level of abstraction. It can be at signal, feature or symbol level (Dasarathy 1997);

2. registration: this is one of the most import step in datafusion. The measured data are aligned and transformed to share a common coordinate system;

3. data fusion: at this step, a decison is taken to select which measurement data is integrated into the final data set and how data are combined together;

4. data reduction (optional): e.g., decimation of the data set in order to easily transport it;

5. meshing (optional): the point cloud can be converted into a mesh to reconstruct the surface of the object;

6. data format conversion (optional).

A machine equipped with multiple sensors is the hybrid CMM (Figure 1.2). It is able to combine the data from optical and contact sensor in order to produce the measurement of a specific object with higher precision, increased measurement areas and (possibly) reduced time with respect to traditional CMMs.

Figure 1.2: An example of hybrid measurement device

A classification of the sensors that can be mounted on a classical CMM in order to improve performance is reported in Neuman and Christoph (2007) (Figure 1.3).

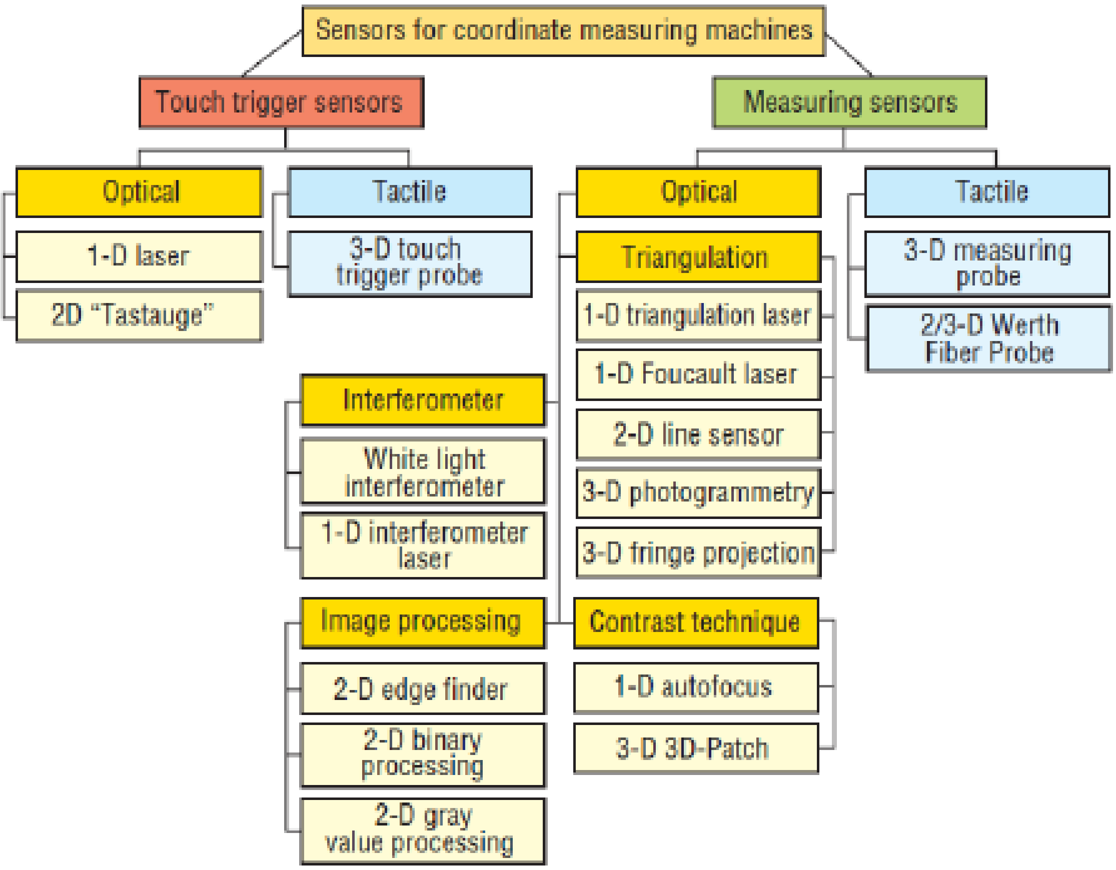


Figure 1.3: Type of sensors that can be mounted on a CMM (Neuman and Christoph 2007)

With this machine, the registration phase can be skipped because relative position of the two sensors is known and they can simply refer to a common coordinate system. This kind of measurement device allows one to partially reduce the purchasing costs (compared with costs of buying the two different solutions independently).

The literature on data fusion often suggests sequential use of the two sensors. In Carbone et al. (2001) and Shen, Huang, and C. (1999) the optical and contact sensors are sequentially used for reverse engineering applications. Firstly, the point cloud is acquired with the optical system and there is a first digitization of the object. With this information, it is possible to write the CMM part program that acquires the desired point cloud. The final CAD model is thus build by refining the first raw model (based on non contact data) with the more precise CMM data points.

An example of competitive data fusion can be found in (Ramasami, Raja, and Boudreau 2013) where a multiscale approach based on wavelet decomposition is presented. The authors show how an appropriate multiscale model can increase the measurement result in micro-scale manufacturing.

Another example where the two sensors are used in a competitive way can be found in (Xia, Ding, and Mallick 2011). The authors used a data fusion technique in order to align point clouds coming from two different sensors. In this case, the main idea is to reconstruct the information provided by the Lo-Fi sensor with a bayesian GP model. Then, the alignment with the Hi-Fi data is performed thanks to a local kernel smoothing technique and with a rigid transformation matrix (translation and rotation).

Starting form the approach proposed by Xia, Ding, and Mallick (2011), Colosimo, Pacella, and Senin (2014) presented a data fusion model to merge data from non contact (Lo-Fi) and contact (Hi-Fi) devices with a GP model. The model was applied to reconstruct a surface with application to normal and large scale metrology. The fusion process was performed using Hi-Fi data to locally correct the Lo-Fi model. This model will be presented in the next section and used as starting point of the thesis.

1.3 Process and Surface Modeling via Gaussian Process

Gaussian Process (GP) is a method originally developed in the geostatistics field, also known as kriging (Cressie 1993; Schabenberger and Gotway 2005). The approach was proposed by Danie G. Krige, a South African Mining Engineer, to evaluate mineral resources (Krige 1951). Recent interests to GP originate from applications to describe the output of a computer experiment (Santner, Williams, and Notz 2003) and to represent a smooth surface (Xia, Ding, and Wang 2008; Xia, Ding, and Mallick 2011; Colosimo et al. 2014; Colosimo, Pacella, and Senin 2014; Del Castillo, Colosimo, and Tajbakhsh 2013).

Let S represents a continuous and fixed subset of $\mathbb{R}^d$ with positive d-dimensional volume. Following Cressie (1993) and Schabenberger and Gotway (2005), a spatial process in d dimensions is denoted as $\{Z(\boldsymbol{s}) : \boldsymbol{s} \in S, \omega \in \Omega\}$ where $\{\Omega, \mathcal{F}, P\}$ is a probability space and Z is the random process.

The random process $\{Z(\boldsymbol{s}) : \boldsymbol{s} \in S\}$ is defined through the finite-dimension distribution $\boldsymbol{Z}(\boldsymbol{s}) \in \mathbb{R}^n$, where n is the number of the observed locations. If for any $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ in S the vector $\boldsymbol{Z}(\boldsymbol{s}) = (Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_n))$, has a multivariate normal distribution, $Z(\boldsymbol{s})$ is a Gaussian random function.

Gaussian random function belong to the class of probabilistic models that specify distributions over function spaces. They are completely determined by their mean function, $\mu(\boldsymbol{s}) \equiv E\{Z(\boldsymbol{s})\}$, for $\boldsymbol{s} \in S$, and by their covariance function $C(\boldsymbol{s}_1, \boldsymbol{s}_2) \equiv Cov\{Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2)\} = E[\{Z(\boldsymbol{s}_1) - \mu(\boldsymbol{s}_1)\}\{Z(\boldsymbol{s}_2) - \mu(\boldsymbol{s}_2)\}]$, for any $\boldsymbol{s}_1, \boldsymbol{s}_2 \in S$, or by their correlation function defined as:

$$R(\boldsymbol{s}_1, \boldsymbol{s}_2) = \frac{C(Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2))}{\sqrt{Var(Z(\boldsymbol{s}_1))Var(Z(\boldsymbol{s}_2))}}, \quad -1 \leq R(\boldsymbol{s}_1, \boldsymbol{s}_2) \leq 1.$$

The definition of the correlation reflects the fact that values at two sample locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are stochastically dependent or independent. If the correlation is positive, we expect high (low) values to be surrounded by high (low) values. If it is negative, high (low) values should be surrounded by low (high) values. When the spatial dependency changes as a function of the particular direction the process is called anisotropic.

Many statistical methods of spatial data analysis make some assumptions about the behavior of the covariance function $C(\cdot)$; the most used is known as stationarity assumption. A stochastic process $Z(\cdot)$ is strongly stationary if the spatial distribution is invariant under translation of the coordinates, i.e., for any $\boldsymbol{h} \in \mathbb{R}^d$, any $\boldsymbol{s}_1, \dots, \boldsymbol{s}_k \in X$ with $\boldsymbol{s}_1 + \boldsymbol{h}, \dots, \boldsymbol{s}_k + \boldsymbol{h} \in X$, then $(Z(\boldsymbol{s}_1), \dots, Z(\boldsymbol{s}_k))$ and $(Z(\boldsymbol{s}_1 + \boldsymbol{h}), \dots, Z(\boldsymbol{s}_k + \boldsymbol{h}))$ have the same distribution.

As the name suggests, strong stationarity is a stringent condition; most statistical methods for spatial data analysis are satisfied with stationary conditions based on the moments of the spatial distribution rather than the distribution itself.

Typically only a second order stationarity is required. It means that the random processes $(Z(\boldsymbol{s}_1), \dots, Z(\boldsymbol{s}_k))$ and $(Z(\boldsymbol{s}_1 + \boldsymbol{h}), \dots, Z(\boldsymbol{s}_k + \boldsymbol{h}))$ always have the same mean and the covariance is a function of the spatial separation between points only, i.e. $Cov\{Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2)\} = Cov\{Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_1 + \boldsymbol{h})\} = C(\boldsymbol{h})$ where $\boldsymbol{h} = \boldsymbol{s}_1 - \boldsymbol{s}_2$.

The assumption of second order stationarity implies that $Var[Z(\boldsymbol{s})] = \sigma^2$ is not a function of spatial location, i.e., the variance of the spatial process is the same everywhere. The covariance function does no longer depend on $\boldsymbol{s}$ and the correlation function becomes:

$$R(\boldsymbol{h}) = \frac{C(\boldsymbol{h})}{\sigma^2}$$

As previously shown, the second moment structure of a weakly stationary random field is a function of the spatial separation $\boldsymbol{h}$, but the covariance function can depend on the direction. When a GP is invariant under rotations, that is the covariance function depends only on the absolute distance between points, the function is termed isotropic. This property can be expressed as:

$$Cov(\boldsymbol{h}) = C(\|\boldsymbol{h}\|)$$

where $\| \bullet \|$ is the Euclidean norm of the lag vector, $\|(\boldsymbol{s} + \boldsymbol{h}) - \boldsymbol{s}\| = \|\boldsymbol{h}\| = (\sum_i h_i^2)^{1/2}$.

The covariance function $C(\boldsymbol{h})$ of a second order stationary random field has several other properties. In particular:

- $C(\mathbf{0}) \geq 0$;
- $C(\boldsymbol{h}) = C(-\boldsymbol{h})$, i.e., C is an even function;
- $C(\mathbf{0}) = |C(\boldsymbol{h})|$

- $C(\boldsymbol{h}) = Cov[Z(\boldsymbol{s}), Z(\boldsymbol{s}+\boldsymbol{h})] = Cov[Z(\boldsymbol{0}), Z(\boldsymbol{h})]$;

Moreover, to be a valid covariance function for a second order stationary spatial random field, C must satisfy the positive-definiteness condition:

$$\sum\sum a_i a_j C(\boldsymbol{s}_i - \boldsymbol{s}_j) \geq 0 \tag{1.1}$$

for any set of locations ($\boldsymbol{s}_i$ and $\boldsymbol{s}_j$) and real numbers (a_i and a_j).

A typical class of desirable covariance functions is one that links the correlation between errors to the distance between the corresponding points and it has to decrease with increasing spatial separation, a common fact to many statistical models for spatial data. A common choice is the exponential model. In this type of model the covariance between measurements at two locations is an exponential function of the interlocation distance, typically given by

$$d(\boldsymbol{s}_1, \boldsymbol{s}_2) = \sum_{j=1}^{d} \vartheta_j (s_{1j} - s_{2j})^{p_j}, \quad \vartheta_j > 0,\ p_j \in (0, 2]\ \forall\ j = 0, 1, \ldots d$$

Given this distance definition, a whole class of correlation functions is introduced under the name of power exponential correlation functions:

$$R(\boldsymbol{s}_1, \boldsymbol{s}_2) = \exp(-d(\boldsymbol{s}_1, \boldsymbol{s}_2)) = \exp\left\{-\sum_{j=1}^{d} \vartheta_j (s_{1j} - s_{2j})^{p_j}\right\} \tag{1.2}$$

where ϑ_j is the scale correlation parameter that controls how fast the correlation decays with the distance along the *j-th* coordinate direction, and p_j is the power parameter, which controls the smoothness of the data fitting. When the scale parameter ϑ_j increases, the correlation decreases and the random field shows a behavior closer to a white noise, conversely as ϑ_j decreases, the correlation increases until the value one and the random field becomes closer to the process mean.

The power exponential form includes two commonly used correlation functions as special cases: the exponential (when $p = 1$) and the squared exponential (when $p = 2$), also called the Gaussian correlation function. The last type of correlation function has nice property in terms of smoothness (or differentiability in the mean square sense) and thus it is a very popular choice in areas like machine learning (where is also known as radial basis kernels) and computer experiments modeling. Besides, past experience indicates that this correlation function is able to model various spatial features (Zimmerman and Harville 1991; Gaudard et al. 1999) and, as suggested by Xia, Ding, and Mallick (2011), it appears reasonable for a large number of manufactured geometric features.

For a detailed discussion on other kinds of correlation functions the interested reader can refer to (Stein 1999; Santner, Williams, and Notz 2003; Rasmussen and Williams 2006).

Without loss of generality, let us assume that the random field can be described by the model

$$Z(\boldsymbol{s}) = \mu(\boldsymbol{s}) + \eta(\boldsymbol{s}) + \varepsilon \tag{1.3}$$

where $\mu(\boldsymbol{s})$ is the mean function or rather the large-scale trend of the random field, $\eta(\boldsymbol{s})$ is a Gaussian stationary process with zero mean and covariance function $C(\boldsymbol{h}) = \sigma^2_\eta R(\boldsymbol{h})$ and the last term ε is a random component which denotes a white noise measurement error with variance σ^2_ε. We further suppose that the random variables $\eta(\boldsymbol{s})$ and ε are independent.

When computer experiments with deterministic simulations are analyzed, the term ε is not included, since no random error has to be modeled and the fitted model has to be an exact interpolator.

Typically, it is assumed that the mean function is a linear or nonlinear function of $\boldsymbol{s}$

$$\mu(\boldsymbol{s}) = \sum_h \boldsymbol{f}^T_h(\boldsymbol{s})\boldsymbol{\beta}_h. \tag{1.4}$$

In traditional linear models, the practical problem of finding the correct functional form for the regressor terms can introduce some difficulties. On the contrary, one of the main advantages of GP is its flexibility in modeling any complex pattern. This is why most of the times, the regression part shown in Equation 1.4 is limited to a constant or a linear term (i.e. $f(\boldsymbol{s}) = 1$ or $\boldsymbol{f}(\boldsymbol{s}) = [1, u, v]^T$), without affecting the whole model prediction ability.

1.3.1 Prediction

When using GP models, usually the main target is predicting the underling function at a location $\boldsymbol{s}_0 \in S$, when obervations $\boldsymbol{z}$ are available. Prediction can be required at a new or an already existing location. Usually when prediction is done at locations where data were already observed, the main objective is to filter noise, i.e. measurement error.

In order to find a predictor, a loss function $L(Z(\boldsymbol{s}_0), p(\boldsymbol{Z}; \boldsymbol{s}_0))$ has to be selected. This function measures the “cost” incurred by using $p(\boldsymbol{Z}; \boldsymbol{s}_0)$ as a predictor of $Z(\boldsymbol{s}_0)$. The squared-error loss function $L(Z(\boldsymbol{s}_0), p(\boldsymbol{Z}; \boldsymbol{s}_0)) = (Z(\boldsymbol{s}_0) - p(\boldsymbol{Z}; \boldsymbol{s}_0))^2$ is commonly used; under squared-error loss, the cost function induced is the mean squared error (MSE) and the prediction criterion is based on the minimization of the Mean Squared Prediction Error (MSPE).

It is possible to show that the predictor of $Z(\boldsymbol{s}_0)$ that minimizes the MSPE for the model (1.3) is linear in the observed data and its value is given by the conditional expectation:

$$\mathbb{E}[Z(\boldsymbol{s}_0)|\boldsymbol{Z} = \boldsymbol{z}] = \boldsymbol{f}^T(\boldsymbol{s}_0)\boldsymbol{\beta} + \sigma_\eta^2 \boldsymbol{r}_0^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{z} - \boldsymbol{F}\boldsymbol{\beta}) \tag{1.5}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ is the vector of the observed values, $\boldsymbol{F} = [\boldsymbol{f}(\boldsymbol{s}_1), \boldsymbol{f}(\boldsymbol{s}_2), \ldots, \boldsymbol{f}(\boldsymbol{s}_n)]^T$ is the so called model matrix, $\boldsymbol{r}_0 = Corr(Z(\boldsymbol{s}_0), \boldsymbol{Z}(\boldsymbol{s}))$ and $\boldsymbol{\Sigma} = \sigma_\eta^2 \boldsymbol{R} + \sigma_\varepsilon^2 \boldsymbol{I}$ where $\boldsymbol{R}$ is matrix with entries $r_{ij} = R(\boldsymbol{s}_i, \boldsymbol{s}_j)$

Another attractive property of the GP model is that the prediction accuracy can be simply expressed as a function of the conditional variance, i.e.:

$$\mathbb{V}ar[Z(\boldsymbol{s}_0)|\boldsymbol{Z} = \boldsymbol{z}] = \sigma_\eta^2 - \sigma_\eta^4 \boldsymbol{r}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{r}_0. \tag{1.6}$$

1.3.2 Parameters Estimation

The prediction model previously described assumes that the vector of parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Psi} = (\boldsymbol{\vartheta}, \sigma_\eta^2, \sigma_\varepsilon^2)$ are known. In most practical cases these quantities are unknown. The common approach is to assume a structure of the covariance function that depends on a vector of parameters $\boldsymbol{\Psi}$ and estimate all the unknown parameters using the maximum likelihood (ML) or the restricted maximum likelihood (REML) (Harville 1977).

In order to make a distributional assumption for $\boldsymbol{Z}(\boldsymbol{s})$, Mardia and Marshall (1984) described how ML estimator for spatial models can be obtained for the Gaussian case, when residuals are correlated and when the covariance among the residuals is determined by a parametric model containing unknown parameters. ML estimation is the simultaneous solution to the problem of minimizing the negative of twice the Gaussian log likelihood

$$-2\,l = n\log 2\pi + \log|\boldsymbol{\Sigma}(\boldsymbol{\Psi})| + (\boldsymbol{z}-\boldsymbol{F\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Psi})(\boldsymbol{z}-\boldsymbol{F\beta}). \tag{1.7}$$

In equation (1.7), it is possible to obtain a closed form expression for the parameters $\boldsymbol{\beta}$, enabling one to remove these parameters from the optimization (a process termed profiling) thus reducing the dimension of the solution space. In oreder to profile $\boldsymbol{\beta}$, it is necessary to take the derivatives with respect to $\boldsymbol{\beta}$ equaling the vector to $\mathbf{0}$. The result is the generalized least squares (GLS) estimator

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Psi})\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Psi})\boldsymbol{z}.$$

Harville (1977) suggested to use the REML estimation because the ML estimate of the process variance is biased. The RMLE of the parameters can be found by minimizing the function

$$-2\,l_{\widehat{\beta}} = n\log 2\pi + \log|\boldsymbol{\Sigma}(\boldsymbol{\Psi})| + (\boldsymbol{z}-\boldsymbol{F}\widehat{\boldsymbol{\beta}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Psi})(\boldsymbol{z}-\boldsymbol{F}\widehat{\boldsymbol{\beta}}) + \log\left|\boldsymbol{F}^T\boldsymbol{\Sigma}(\boldsymbol{\Psi})\boldsymbol{F}\right| \tag{1.8}$$

Since the function $l_{\widehat{\beta}}$ has many local minima (Spall 2003), a global optimization routine is preferable, unless a good starting point for the local optimization algorithm is provided. Therefore, one or more initial solutions are usually found considering a global optimization algorithm (e.g. simulated annealing, direct Monte-Carlo sampling, etc.) and then yhese solutions are refined using a local optimization algorithm (e.g. Newton-Raphson, Quasi-Newton). Even though this method provides no formal guarantee for global optimization, it is a reliable strategy and offers a reasonable computational effort in the optimization of multiparameter functions.

After the the parameter $\boldsymbol{\beta}$ has been estimated, the kriging variance (MSPE) became (Schabenberger and Gotway 2005)

$$\begin{aligned}\mathbb{V}ar[Z(\boldsymbol{s}_0)|\boldsymbol{Z}=\boldsymbol{z}] =&\sigma_\eta^2 - \sigma_\eta^4\boldsymbol{r}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{r}_0 + \left(\boldsymbol{f}_0 - \boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{r}_0\right)^T \\ &\left(\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\Psi})\boldsymbol{F}\right)^{-1}\left(\boldsymbol{f}_0 - \boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{r}_0\right).\end{aligned} \tag{1.9}$$

Test case

In order to show the prediction ability of the GP model the Matlab `peaks` function is used. This function is a mixture of translated and scaled bivariate Gaussian probability density function.

The `peaks` function is given by

$$f(\boldsymbol{s}) = 3(1-u)^2\, e^{-u^2-(v+1)^2} - 10\left(\frac{u}{5} - u^3 - v^5\right) e^{-u^2-v^2} - \frac{1}{3} e^{-(u+1)^2-v^2}. \quad (1.10)$$

Assume to take measurements from the `peaks` function where noise is added to represent measurement errors

$$Z(\boldsymbol{s}) = f(\boldsymbol{s}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.4^2) \quad (1.11)$$

in the $[-3,3] \times [-3,3]$ domain.

Figure 1.4 shows the simulated points together with the surface reconstructed assuming the following GP model

$$z(\boldsymbol{s}) = \beta_0 + \eta(\boldsymbol{s}) + \varepsilon.$$

The color map on the reconstructed surface is a map of the prediction error. It is possible to observe that the GP model with a simple trend term (β_0) can estimate even a complex unknown shape with small prediction error (the range of the absolute error is almost 5% of the surface range).

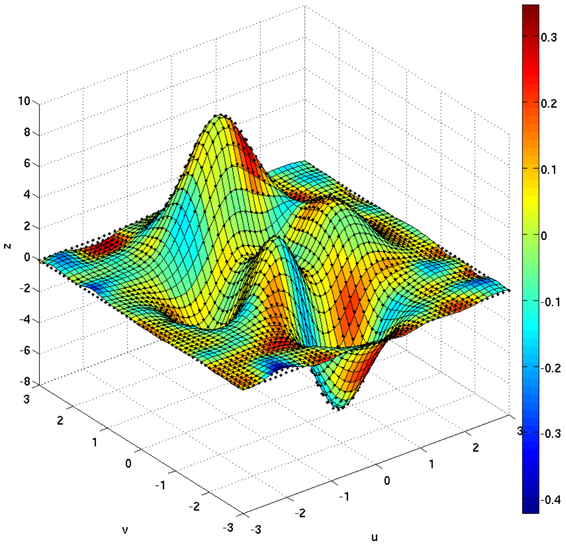


Figure 1.4: Simulated point and reconstruction with GP

1.4 Gaussian process model for data fusion

In the previous sections a GP model was used to reconstruct the single response of a process or a measurement system. In this section a model is proposed to link the data coming from different sources.

The proposed model originates from Kennedy and O'Hagan (2000), where a model to improve efficiency by combining expensive runs of a complex computer code with relatively cheap runs from one or more simpler approximations is presented. In a subsequent work, the authors used the same model in a calibration framework (Kennedy and O'Hagan 2001); i.e., to correct any inadequacy of the computer approximation which is revealed by a discrepancy between the observed data and the model predictions. Starting from these seminal papers, others data fusion models based on computer responses where proposed in (Qian et al. 2006; Qian and Wu 2008). In these works, the authors used GP to model and obtain data fusion moving from a frequentist to a bayesian approach.

(Xia, Ding, and Mallick 2011) extends the model presented by (Qian et al. 2006; Qian and Wu 2008) to deal with multisensor data alignment for surface metrology. In this case, the main idea is to reconstruct the information provided by all the different sensors to perform the fusion step. Similarly, (Colosimo, Pacella, and Senin 2014) used a similar model for normal and large-scale metrology data fusion. These approaches have the main advantage of including statistical modeling while reconstructing the information provided by different data sets, thus providing prediction intervals for the local discrepancies between different data sets as well as for the final prediction of the shape at any given location. A second advantage of these approaches consist in relaxing the assumption of acquiring all the data at the same location set.

In this thesis, use of bayesian approaches will not be explored. In fact, preliminsry tests allowed us to conclude that bayesian methods become computationally intractable as the number of data points increases, a situation that characterized almost all the real case studies faced in this research work.

The data fusion model in the (frequentist) approach is a two-stage model. The first stage consists of modeling the low-fidelity (Lo-Fi) response observed at location $\boldsymbol{s}_i$ using the following GP model

$$z_l(\boldsymbol{s}_i) = \boldsymbol{f}_l^T(\boldsymbol{s}_i)\boldsymbol{\beta}_l + \eta(\boldsymbol{s}_i) + \varepsilon_l, \quad i = 1, \ldots, n_l. \tag{1.12}$$

where $\boldsymbol{f}_l^T(\boldsymbol{s}_i)\boldsymbol{\beta}_l$ is the regressor term, $\eta(\boldsymbol{s}_i)$ is a Gaussian stationary process, ε_l, also known as nugget, describes the random component due to the measurement error and it is assumed to be independent and normally distributed, i.e. $\varepsilon_l \sim \mathcal{N}(0, \sigma_l^2)$.

Here the subscript l is used to indicate the Lo-Fi model. If the model is used to describe deterministic computer simulations (as in the second part of this thesis), this last term will not be included.

The main objective of the hierarchical model is to combine the Lo-Fi and Hi-Fi data in order to improve predictions achievable by using the Lo-Fi or the Hi-Fi data sets alone. The core of the data fusion model is the linkage or the second stage model, which represents the connection between Lo-Fi and Hi-Fi data and can be expressed as (Qian et al. 2006)

$$z_h(\boldsymbol{s}_i) = \rho(\boldsymbol{s}_i)\, \widehat{z}_l(\boldsymbol{s}_i) + \delta_0 + \delta(\boldsymbol{s}_i) + \varepsilon_\Delta \quad i = 1, \ldots, n_h. \tag{1.13}$$

The aim of this second stage model is to correct the Lo-Fi predictions $\widehat{z}_l(\boldsymbol{s}_i)$ (i.e., predictions done using the Lo-Fi data only) using a "scale" and a "shift" effects, represented by $\rho(\boldsymbol{s}_i)$ and $\delta_0+\delta(\boldsymbol{s}_i)$, respectively. The term ε_Δ is the random term of the linkage model, which is assumed to be independent and normally distributed, i.e., $\varepsilon_\Delta \sim \mathcal{N}(0, \sigma^2_\Delta)$. This nugget effect is included to represent randomness characterizing the Hi-Fi data, where randomness exists because Hi-Fi data are random variables.

Following Qian et al. (2006), the scale effect can be modeled as

$$\rho(\boldsymbol{s}_i) = \boldsymbol{f}^T_\Delta(\boldsymbol{s}_i)\boldsymbol{\rho}, \tag{1.14}$$

where $\boldsymbol{f}_\Delta(\boldsymbol{s}_i)$ is a known basis function acting as regressor term when computed at $\boldsymbol{s}_i$, and $\boldsymbol{\rho}$ is the vector of unknown parameters to be estimated. Usually a first-order term (linear model) for $\boldsymbol{f}_\Delta(\boldsymbol{s}_i)$ is enough to model the scale effect (Qian et al. 2006). Subscript Δ is included to show that these regressors are modeling the difference between Lo-Fi and Hi-Fi data and hence regressors can be different from the ones assumed in equation (1.12) to reconstruc the Lo-Fi data only.

The shift effect is represented by $\delta_0 + \delta(\boldsymbol{s}_i)$, where δ_0 is a constant and $\delta(\boldsymbol{s}_i)$ is a GP. In order to estimate the process parameters, the linkage model can be written in matrix form

$$\boldsymbol{z}_h = \boldsymbol{P}\widehat{\boldsymbol{z}}_l + \delta_0\mathbf{1}_{n_h} + \boldsymbol{\delta} + \boldsymbol{\varepsilon}_\Delta \tag{1.15}$$

where $\boldsymbol{z}_h = (z_h(\boldsymbol{s}_1), z_h(\boldsymbol{s}_2), \ldots, z_h(\boldsymbol{s}_{n_h}))^T$ is the vector of the observed Hi-Fi data, $\boldsymbol{P}$ is a diagonal matrix with entries $\rho(\boldsymbol{s}_i)$ (see Equation (1.14)) for $i = 1, \ldots, n_h$

and $\mathbf{1}_{n_h} \in \mathbb{R}^{n_h}$ is a vector of ones. ML parameter estimation can be performed by minimizing the function

$$-2\,l = n_h \log 2\pi + \log|\mathbf{\Sigma}_\Delta| + (\boldsymbol{z}_h - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \delta_0 \mathbf{1}_{n_h})^T \,\mathbf{\Sigma}_\Delta^{-1}\,(\boldsymbol{z}_h - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \delta_0 \mathbf{1}_{n_h}) \quad (1.16)$$

where $\mathbf{\Sigma}_\Delta$ is known as the mean squares prediction error (MSPE) or kriging variance. The kriging variance can be computed as

$$\mathbf{\Sigma}_\Delta = \boldsymbol{P}\mathbf{\Sigma}_0\boldsymbol{P}^T + \sigma_\delta^2 \boldsymbol{R}_\delta + \sigma_\Delta^2 \boldsymbol{I}_{n_h} \quad (1.17)$$

where $\mathbf{\Sigma}_0$ is the kriging variance of the Lo-Fi model (given by the first stage equation 1.13), but computed at the Hi-Fi locations. $\boldsymbol{R}_\delta = \{r_\delta(\boldsymbol{s}_i, \boldsymbol{s}_j)\}$ is the correlation matrix of the GP for the shift effect $\delta(\boldsymbol{s}_i)$, $\boldsymbol{I}_{n_h} \in \mathbb{R}^{n_h \times n_h}$ is an identity matrix and $\mathbf{1}_{n_h} \in \mathbb{R}^{n_h}$ is a vector of ones.

If the partial derivative with respect to δ_0 are computed and set to 0, the ML estimator of δ_0 is given by:

$$\widehat{\delta}_0 = \frac{\mathbf{1}_{n_h}' \mathbf{\Sigma}_\Delta^{-1}(\boldsymbol{z}_h - \boldsymbol{P}\widehat{\boldsymbol{z}}_l)}{\mathbf{1}_{n_h}' \mathbf{\Sigma}_\Delta^{-1} \mathbf{1}_{n_h}}. \quad (1.18)$$

The other unknown parameters can be found by minimizing twice the logarithm of the restricted likelihood (Harville 1977) given by

$$\begin{aligned} -2l_{\hat{\delta}_0} =& n_h \log 2\pi - \log\left(\boldsymbol{F}_\Delta^T \boldsymbol{F}_\Delta\right) + \log \mid \mathbf{\Sigma}_\Delta \mid + \log \mid \boldsymbol{F}_\Delta^T \mathbf{\Sigma}_\Delta^{-1} \boldsymbol{F}_\Delta \mid \\ &+ \left(\boldsymbol{z}_h - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \widehat{\delta}_0 \mathbf{1}_{n_h}\right)^T \mathbf{\Sigma}_\Delta^{-1} \left(\boldsymbol{z}_h - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \widehat{\delta}_0 \mathbf{1}_{n_h}\right). \end{aligned} \quad (1.19)$$

According to the assumed combination of the linkage (or second stage) equation (1.13) and the first stage equation (1.12), the hierarchical approach allows one to fuse data in order to compute a prediction at any new location $\boldsymbol{s}_0$ as (see Appendix A for details)

$$\begin{aligned} \mathbb{E}[Z_h(\boldsymbol{s}_0)|\boldsymbol{Z}_h = \boldsymbol{z}_h, \boldsymbol{Z}_l = \boldsymbol{z}_l] = \widehat{\rho}(\boldsymbol{s}_0)\,\widehat{z}_l(\boldsymbol{s}_0) + \widehat{\delta}_0 \\ + \left(\widehat{\sigma}_\delta^2\,\widehat{\boldsymbol{r}}_\delta + \widehat{\boldsymbol{\sigma}}_0\right)^T \widehat{\mathbf{\Sigma}}_\Delta^{-1} \left(\boldsymbol{z}_h - \widehat{\boldsymbol{P}}\,\widehat{\boldsymbol{z}}_l - \widehat{\delta}_0\,\mathbf{1}_{n_h}\right) \end{aligned} \quad (1.20)$$

where $\widehat{\boldsymbol{r}}_\delta = \mathbb{C}orr(\boldsymbol{z}_h, z_h(\boldsymbol{s}_0))$, $\widehat{\boldsymbol{\sigma}}_0$ is a vector with entries $\sigma_{0i} = \rho(\boldsymbol{s}_i) \cdot \rho(\boldsymbol{s}_0) \cdot \mathbb{C}ov(\widehat{z}_l(\boldsymbol{s}_i), \widehat{z}_l(\boldsymbol{s}_0))\ \forall\, i = 1, \ldots, n_h$.

The prediction, or kriging, variance can be expressed as

$$\begin{aligned}\mathbb{V}ar[Z_h(\boldsymbol{s}_0)|\boldsymbol{Z}_h = \boldsymbol{z}_h, \boldsymbol{Z}_l = \boldsymbol{z}_l] = {}& \widehat{\sigma}_\delta^2 + \widehat{\sigma}_l^2(\boldsymbol{s}_0)\ \left[\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\,\widehat{\boldsymbol{\rho}}\right]^2 \\ & - \left(\widehat{\sigma}_\delta^2 \widehat{\boldsymbol{r}}_\delta + \widehat{\boldsymbol{\sigma}}_0\right)^T \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \left(\widehat{\sigma}_\delta^2\, \widehat{\boldsymbol{r}}_\delta + \widehat{\boldsymbol{\sigma}}_0\right) \\ & + \frac{\left[1 - \boldsymbol{1}_{n_h}^T \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \left(\widehat{\sigma}_\delta^2\, \widehat{\boldsymbol{r}}_\delta + \widehat{\boldsymbol{\sigma}}_0\right)\right]^2}{\boldsymbol{1}_{n_h}^T \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \boldsymbol{1}_{n_h}}\end{aligned} \tag{1.21}$$

where $\widehat{\sigma}_l^2(\boldsymbol{s}_0)$ is the prediction variance of the Lo-Fi model in the new location $\boldsymbol{s}_0$.

Test case

In this section, the advantages of combining different datasets with a hierarchical data fusion approach are explored. To this aim, performance of four different models are compared:

- Lo-Fi model: where predictions are computed using a single GP model that considers the Lo-Fi data only;
- Hi-Fi model: where predictions are computed using a single GP model that considers the Hi-Fi data only;
- Addition model: where predictions are based on a single GP model that uses all the Lo-Fi and Hi-Fi data as they come from a single source;
- Fusion model: where predictions are based on the two-stage hierachical model presented before.

The performance indicator used throughout this thesis is the Root Mean Square (prediction) Error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} e^2(\boldsymbol{s}_i)} \tag{1.22}$$

where $e(\boldsymbol{s}_i)$ is the prediction error at location $\boldsymbol{s}_i$, given by

$$e(\boldsymbol{s}_i) = z(\boldsymbol{s}_i) - \widehat{z}(\boldsymbol{s}_i)$$

where $z(\boldsymbol{s}_i)$ is the true point at location $\boldsymbol{s}_i$ and $\widehat{z}(\boldsymbol{s}_i)$ is the value predicted at the same location using one of the aforementioned approaches.

The `peaks` surface described in section 1.3.2 is used in this section to test the ability of the data fusion approach. The Hi-Fi data are simulated from the following model

$$z_h(\boldsymbol{s}) = f(\boldsymbol{s}) + \varepsilon_h, \quad \varepsilon_\Delta = \mathcal{N}(0, 0.2^2). \tag{1.23}$$

while the Lo-Fi data model are randomly generated using the following equation:

$$z_l(\boldsymbol{s}) = f(\boldsymbol{s}) + \text{bias}(\boldsymbol{s}) + \varepsilon_l, \quad \varepsilon_l = \mathcal{N}(0, 0.4^2) \tag{1.24}$$

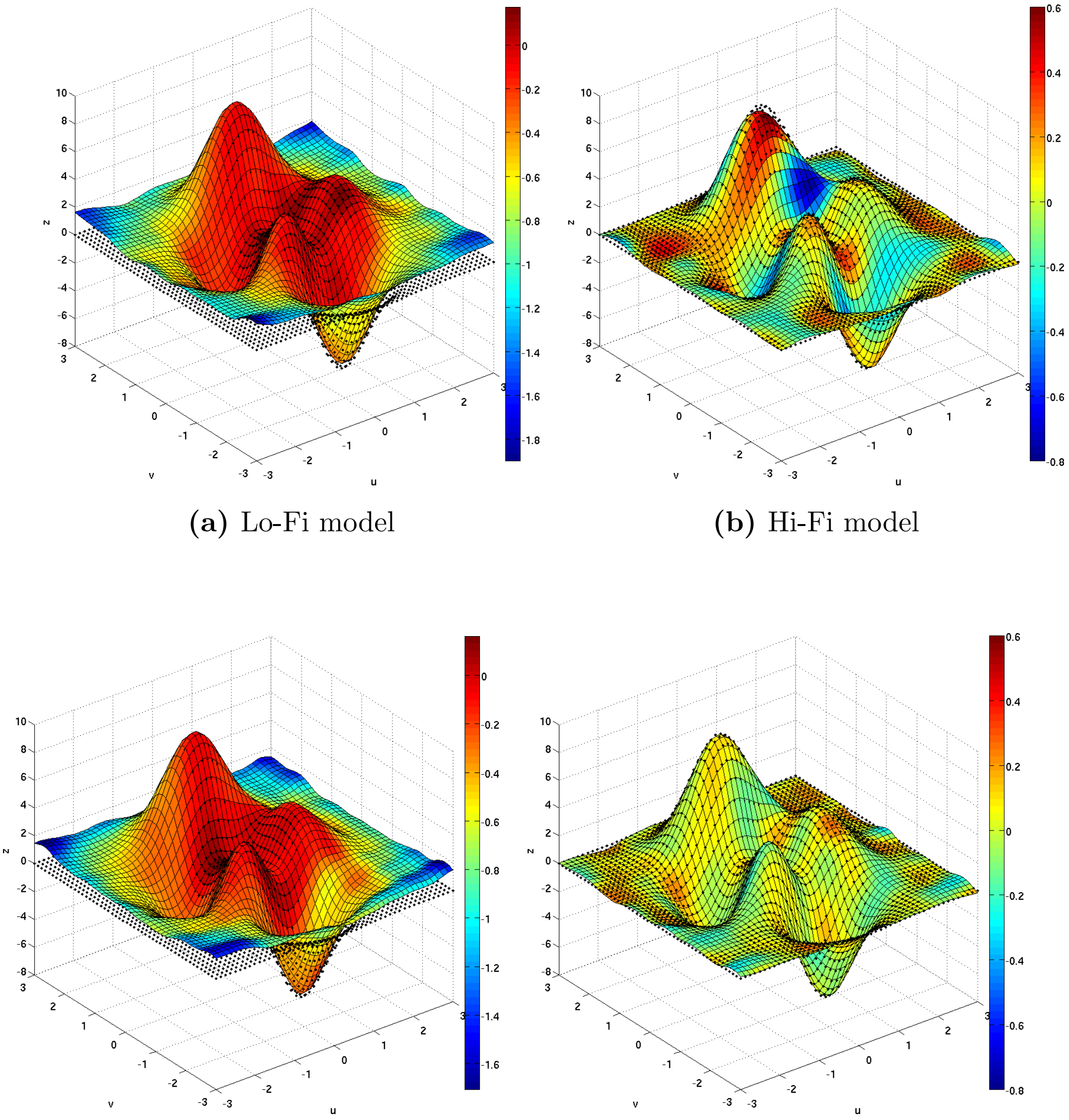
where a bias term is added. This bias term is assumed to have the following model:

$$\text{bias}(\boldsymbol{s}) = \frac{u^2}{10} + \frac{v^2}{10},$$

which describes a Lo-Fi measurement system where the bias increases as the u and v coordinates increase in module. This bias term gives evidence to the need of performing data fusion. As a matter of fact, the Hi-Fi data have to be used to correct the Lo-Fi bias where needed.

A total number of 2500 Lo-Fi points are generated from a regular grid of [50 × 50] points in u and v directions. Since the number of the available Hi-Fi points is usually lower than the Lo-Fi sample size, only 100 Hi-Fi points are generated on a regular grid of [10 × 10] points.

For each of the competitive models, an error map is computed as the difference between the predicted and the nominal value at each location ($e(\boldsymbol{s}_i)$). The error maps are reported in Figure 1.5. The shape of the error map of the Lo-Fi model shows the simulated bias affecting those data. The error map of the Hi-Fi model shows that the error is generally smaller (because those data are assumed to be characterized by zero bias). The behavior of the prediction error of the addition model is similar to the map observed for the Lo-Fi model because the number of Lo-Fi data is high compared to the Hi-Fi data sample size. In this model data are simply merged without any correction and this is why the error map is basically guided by the larger set of Lo-Fi data. The error map of the fusion model shows that a hierarchical model can correct the error of the Lo-Fi model where needed, using only a small set of Hi-Fi data.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Addition model

(d) Fusion model

Figure 1.5: Predicted surface with error map

Considering this `peaks` model as a toy example, 10 different realizations of the surface were run and the prediction ability of each model was tested using the RMSE.

The confidence interval on the mean of the RMSE computed on these 10 replications are drawn in Figure 1.6. It is clear that the fusion model outperforms all the other methods in this case study.

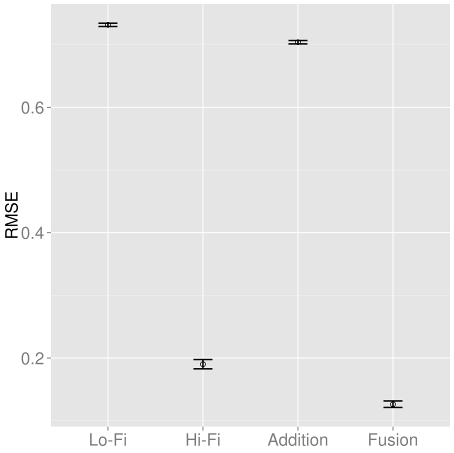


Figure 1.6: Models' RMSE

1.5 A real case study: data fusion to reconstruct a free form surface

The aim of this section is to show the advantages of the data fusion model when applied to the reconstruction of a real smooth free form surface. In this case study, the performance of different models are computed while varying the number of Hi-Fi points. Furthermore, two different sampling plans to locate Hi-Fi points are analyzed: *i)* a uniform sampling plan according to a classic regular grid; *ii)* a Latin Hypercube (LH) sampling plan, which is widely used in computer experiments (Santner, Williams, and Notz 2003) and it is also used in the metrology field in (Xia, Ding, and Mallick 2011).

The measured data of a free form surface were acquired using a Structured Light (SL) scanner (Lo-Fi) and a Coordinate Measuring Machine (CMM) Zeiss "Prismo 5 HTG VAST" equipped with a analog probe head with maximum probing error MPEP $= 2\mu m$ (Hi-Fi). A brief overview of the characteristics of these two devices is firstly presented.

1.5.1 Structured Light Scanner

SL scanner is an non contact device that projects light with a known structure on the object. An example of this device is shown in Figure 1.7. With this device, it is possible to acquire a dense point cloud with a simple and cheap hardware.

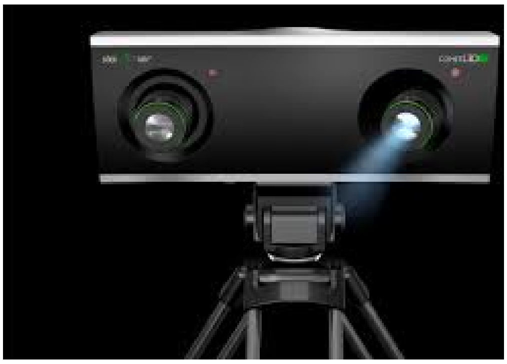
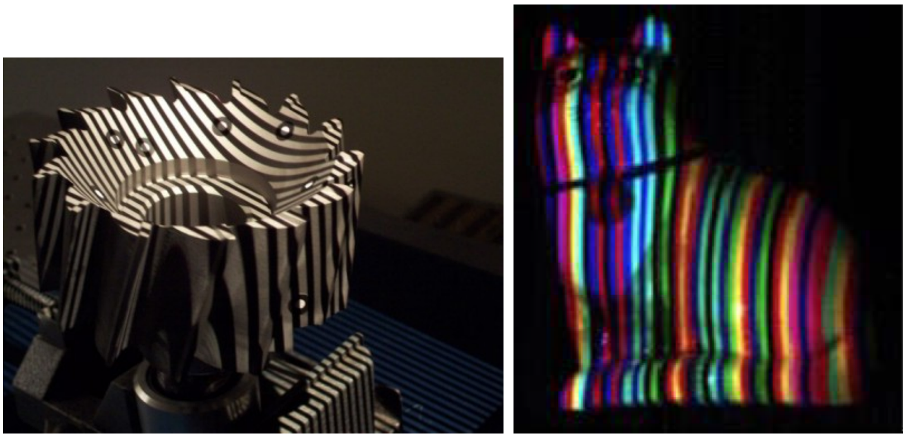


Figure 1.7: Example of a structured light device (Irwin 2012)

The light projected on the object can be made of single points, strip, grid or more complex pattern of coded light (Figure 1.8). Before acquiring the measurement, an important step is the device calibration. This is done through the acquisition of 2D figures of known shapes in order to evaluate object distortions, which are then corrected by the device software. The surface scanner distance is computed through the pattern deformation on the object computed considering the triangulation technique on each point projected on the surface.



(a) Sheet of light **(b)** Colored coded light

Figure 1.8: Example of structured light projected

The characteristics of structured light scanners are:

- low cost and easy implementation;
- readily available;

- fast and high density data acquisition;
- color information available.

Because of these advantages, there are a lot of companies that produce these devices (such as 3D3 Solutions, 3DDynamics, 3D-Shape GmbH, 4D Culture Inc., AGE Solutions S.r.l., Creaform, etc.). Although there are a lot of different devices available on the market, as all the optical methods, it is difficult to acquire reflective or transparent surfaces such as skin, marble, wax and plants.

1.5.2 Coordinate Measuring Machine

CMM is a contact device for measuring the physical geometrical characteristics of an object (Figure 1.9). The measured point is acquired by a probe mounted on the moving axis of the machine. This machine can be manually controlled by an operator or it may be computer controlled.

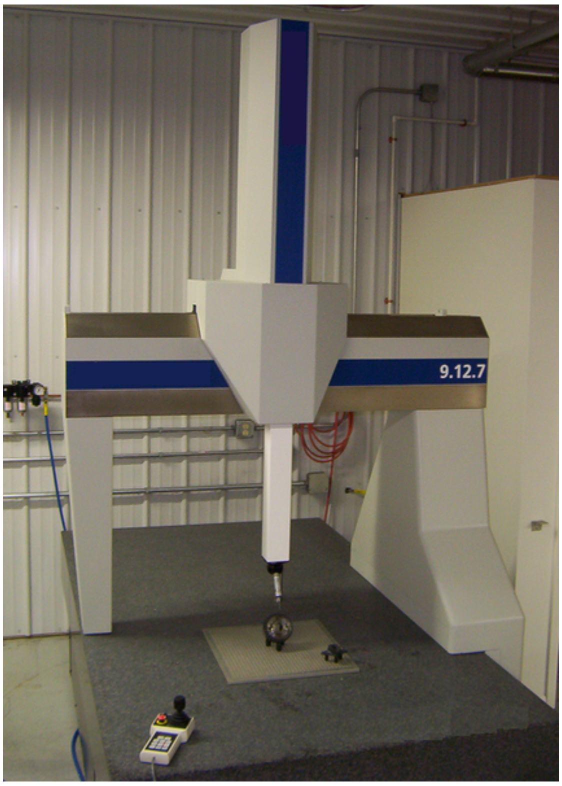


Figure 1.9: Coordinate measuring machine

The first CMM was developed in the 1950s, although this machine only had 2 axes. The first 3-axes models began appearing in the 1960s (Bosch 1995). The typical structure of a CMM is composed by three axes that are orthogonal to each other in order to form a typical three-dimensional coordinate system.

In modern machines, the gantry type superstructure has two legs and is often called a "bridge". This moves freely along the granite table with one leg following a guide rail attached to one side of the granite table. The opposite leg simply rests on the granite table following the vertical surface contour. Air bearings are the chosen method to ensure friction free travel.

Probe is responsible for point location acquisition and is usually placed on the third axis of the machine (Figure 1.10). A common probe is made by soldering a hard ball to the end of a shaft. This is ideal for measuring a whole range of flat, cylindrical or spherical surfaces. Other probes aimed at measuring specific shapes (for example the outer diameter of a screw) are also available. A typical electronic touch trigger probe has a sensitive electrical contact that emits a signal when the probe is deflected from its neutral position by the slightest amount. As the probe touches the component surface, the coordinate positions are recorded by the CMM controller, and appropriately corrected for overtravel and probe size. A CMM with a touch-trigger probe can be programmed to follow planned paths along a surface.

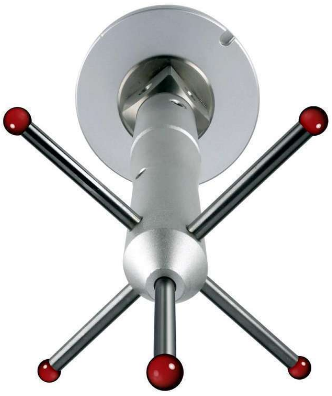


Figure 1.10: Touch probe of a coordinate measuring machine

1.5.3 Numerical Results

In their work, Petrò, Cavallaro, and Moroni (2009) acquired the point clouds of the free form surface shown in Figure 1.11 with both SL (Lo-Fi) and CMM (Hi-Fi) systems.

This case study will be used to compare perfromance of three competitive approaches:

- Lo-Fi: GP model based on the Lo-Fi (SL) points only;
- Hi-Fi: GP model based on the Hi-Fi (CMM) points only;
- Fusion: two-stage fusion model based on both the Lo-Fi and the Hi-Fi points.

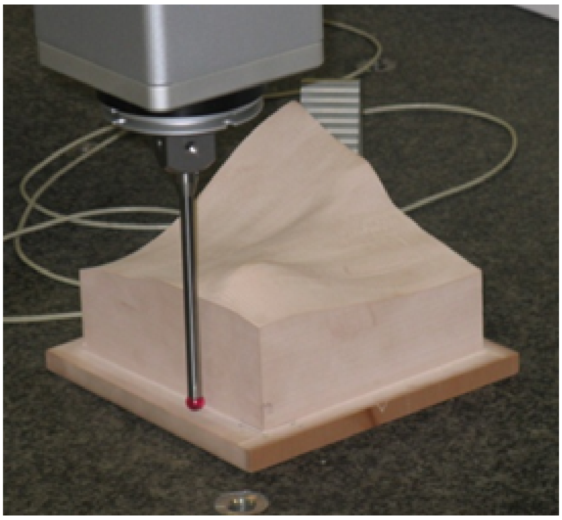
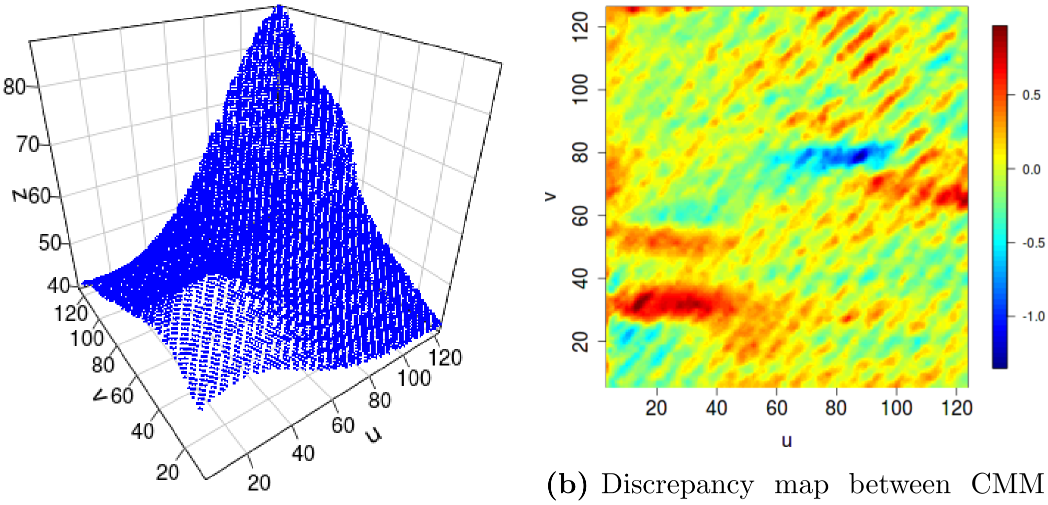


Figure 1.11: Free form surface

The number of points acquired with each device was 9635. Clearly, acquiring this sample size via CMM required a long acquisition time, which is almost unfeasible in real practice. As a matter of fact, it will assumed that only a subset of these Hi-Fi (CMM) data are effectively available to perfrom data fusion, while the remaining set will be used as test set to compute the prediction errors of all the competitive methods. In other words, $n_h = h\% \cdot 9635$ data will be used to reconstruct the surface with all the proposed methods, while the remaining set of $n_{test} = 9635 - n_h$ points will be used to represent the "real" surface. Predictions at all the locations of n_{test} data will be computed with all the competitive methods and the corresponding prediction errors $(e(\boldsymbol{s}_i))$ will be computed.

The whole set of 9635 Lo-Fi points available are shown in Figure 1.12a, while in Figure 1.12b the discrepancy map between the Hi-Fi and Lo-Fi point clouds is shown. The difference between the two point clouds cannot be described with a simple function, so the linkage model must be able to predict an unknown and complex shapes.



(a) Points acquired with the SL (Lo-Fi) scanner

(b) Discrepancy map between CMM (Hi-Fi) and SL (Lo-Fi) points (values in mm)

Figure 1.12: Lo-Fi points and discrepancy map between the Lo-Fi and the Hi-Fi points

In order to compute the performance of each model as a function of the number of Hi-Fi points, the percentage h of the Hi-Fi data used to fit the surface models was changed (to a maximum value of 4%, i.e., $n_h = 400$ Hi-Fi data points).

Furthermore, in order to study the effect of the Hi-Fi point locations on the prediction ability of the competitive methods, two sampling strategies were assumed for the Hi-Fi data: a uniform sampling and a *max-min* latin hypercube (LH) sampling (Santner, Williams, and Notz 2003). As an example, in order to select 100 points according to a uniform sampling, a regularly spaced grid of 10 points has to be drawn such that the distance along the u and v directions between the points is constant (Figure 1.13). A LH sampling for the same set of 100 Hi-Fi points is shown in Figure 1.14, where it is possible to note that the regularity is missing.

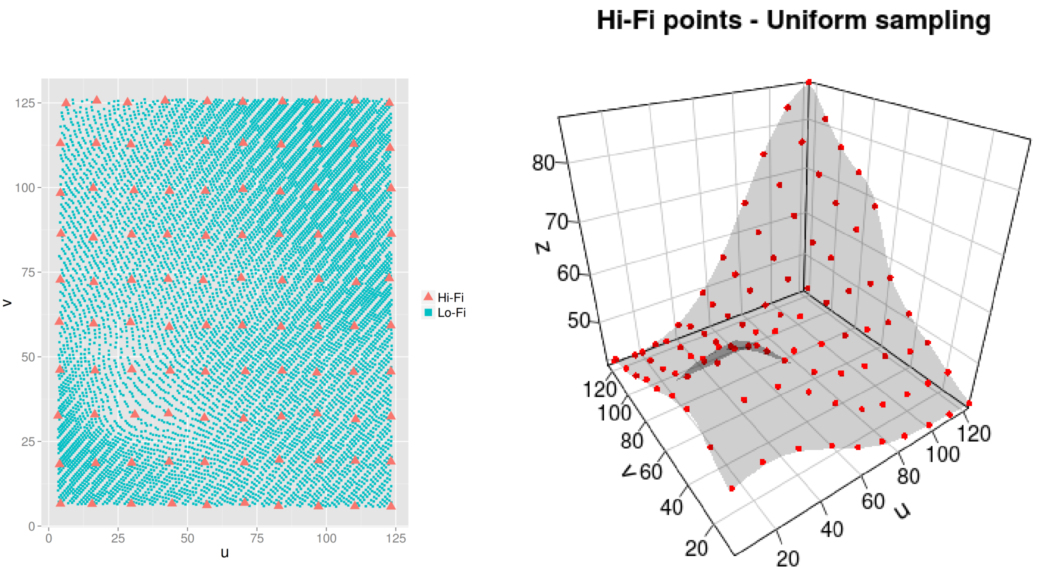


Figure 1.13: Example of 100 Hi-Fi points sampled with the uniform sampling plan (triangles figure on the left)

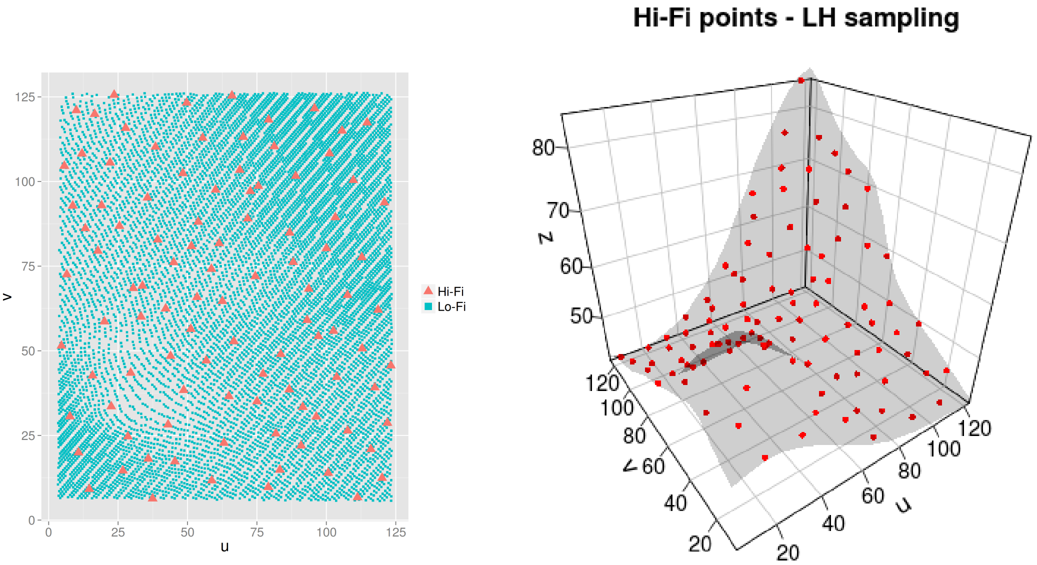
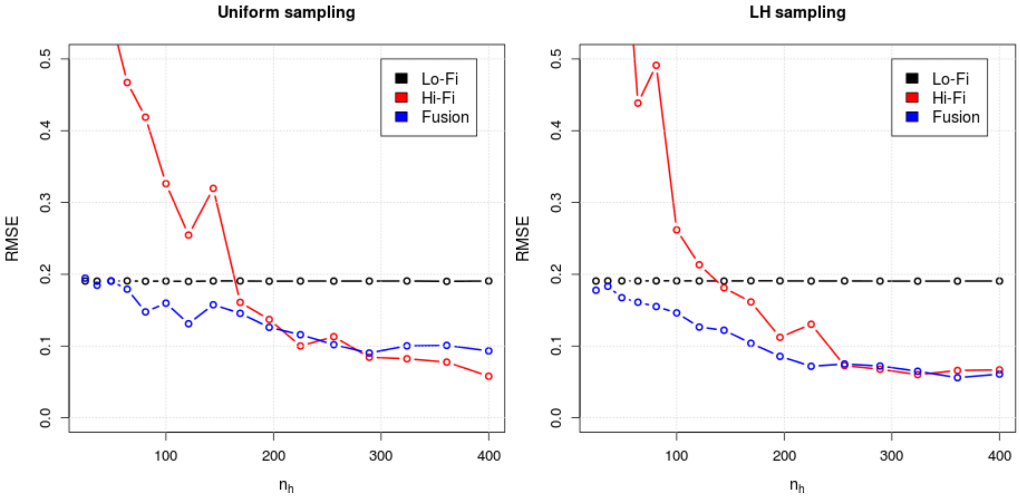


Figure 1.14: Example of 100 Hi-Fi points sampled with the LH sampling plan (triangles figure on the left)

The performance comparison of the competitor models are based on the RMSE (see Equation (1.22)). Figure 1.15 shows the RMSE of all the models as a function of the number of Hi-Fi points available, considering both the uniform (left) and the LH (right) sampling plans. Under a uniform sampling, (Figure 1.15a) the fusion model achieves better results compared with all the other approaches when the number of Hi-Fi (CMM) points available is not so high (less than 150). Then, performance of the fusion and the Hi-Fi models are comparable (i.e., adding SL data to the CMM data does not induce any advantage, since the CMM point cloud is dense enough to appropriately reconstruct the surface pattern).

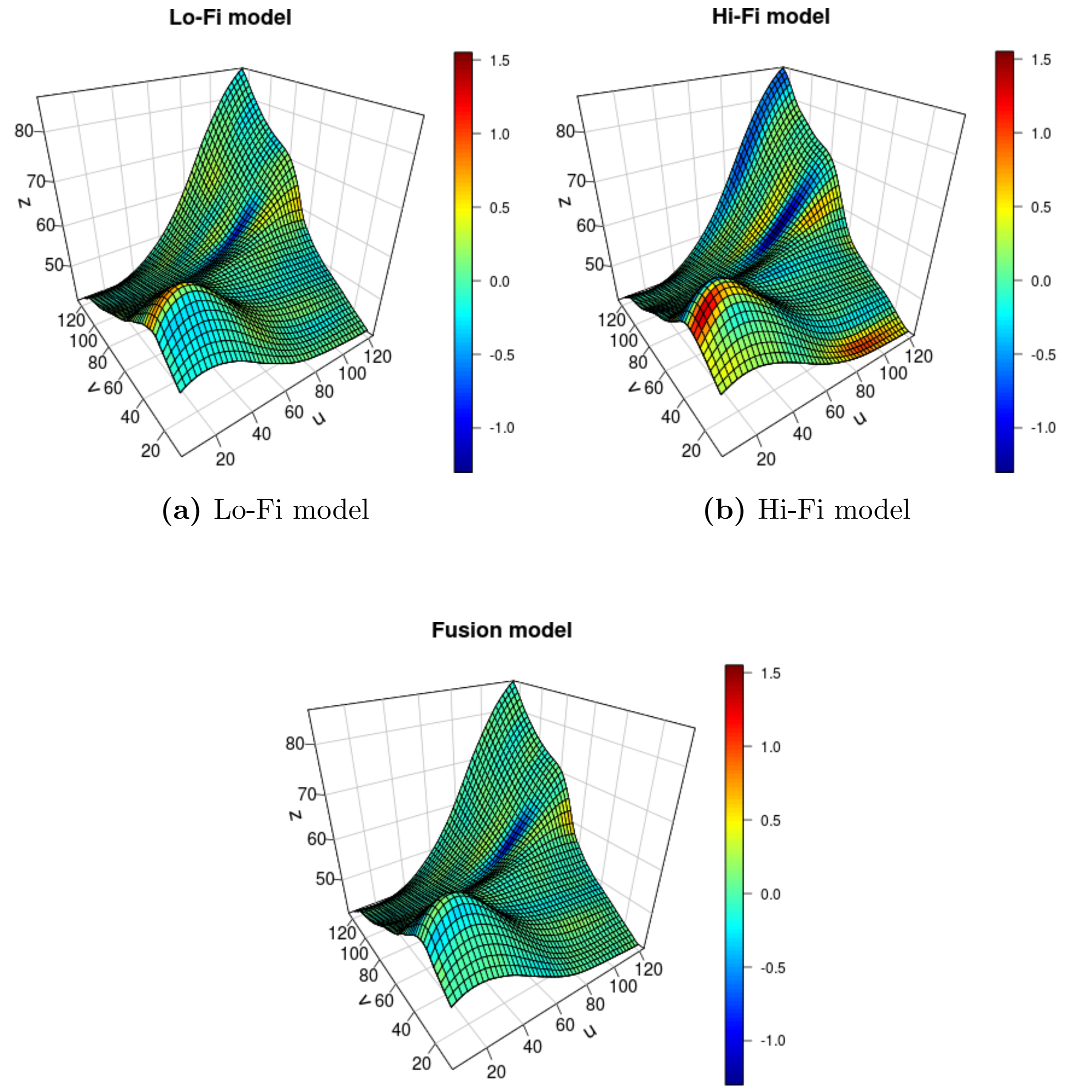
When a LH sampling is considered (Figure 1.15b), data fusion is convenient if the number of Hi-Fi data is lower than 250.



(a) Uniform sampling (b) Latin Hypercube sampling

Figure 1.15: RMSE of the GP model with different sampling strategies

An example of the surface reconstructed with all the models considered is shown in Figure 1.16. To construct these error maps, the parameters of the Hi-Fi and fusion models were estimated on 100 Hi-Fi points sampled according to the LH procedure.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Fusion model

Figure 1.16: Error map of the prediction error of the three models based on a LH sampling and 100 Hi-Fi points

Figure 1.16 shows the error maps of all the three models (considering 100 Hi-Fi data and a LH sampling strategy). It is possible to observe that the error map of the Lo-Fi model reflects the discrepancy between the Lo-Fi (SL) and the Hi-Fi (CMM) points shown in Figure 1.12b. The error map of the Hi-Fi model shows that, due to the small number of Hi-Fi points available, the model has large errors in some locations of the reconstructed surface. On the contrary, the fusion model uses the information of both the Lo-Fi model and the Hi-Fi points to better reconstruct the free form surface thus reducing the error magnitude everywhere. The GP fusion model can therefore "correct" the prediction of the Lo-Fi model using the Hi-Fi points as "attractors".

Eventually, Figure 1.17 compares the performance of the data fusion model using a uniform vs a LH sampling strategy. It is possible to conclude that a data fusion model where the Hi-Fi data are placed according to a LH sampling plan has to be preferred to the same model using a uniform sampling strategy to locate the same data set, especially when the Hi-Fi sample size is high.

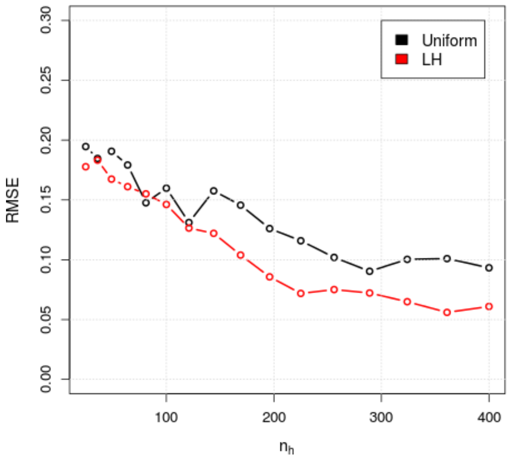


Figure 1.17: Comparison of the data fusion models with different sampling strategies for the Hi-Fi data (uniform vs LH)

Chapter 2

Surface Reconstruction in the "Big Data" Scenario

The approach described in Chapter 1 requires one to estimate all the unknown parameters of the GP models at the first and the second stages of the hierarchical data fusion method. This computation becomes slow as the number of data points increases. Clearly, the problem of estimating the GP model parameters can become easily unfeasible considering the large data set that are nowadays available via non contact sensors. Park, Huang, and Ding (2011) proposed an approach to overcome problems of the GP model when large data have to be treated. The main idea is to decompose of the initial surface in subdomains and then estimate the GP parameters on each region (assuring continuity along the subregions' borders). Although this approach can work for medium scale problems, it is not feasible in the very big data framework, because the number of regions easily increases.

A different approach for multisensor data fusion when large data est have to be managed is proposed in this chapter. In this novel method, the GP model is replaced with a Least Squares B-spline Approximation (LSBA), to be used in both the two stages of the hierarchical model for data fusion. The proposed LSBA model uses the output of a fast algorithm, the Multilevel B-spline Approximation (MBA), as a starting step to solve a least squares problem using B-spline as regressor terms. In the proposed approach, the uncertainty of the predictions is taken into account while computing the data fusion reconstruction.

In this chapter, the LSBA approach for a single stage (i.e., monosensor data) is firstly presented. Then, the two-stage model is described and its performance as fusion algorithm evaluated on a set of test cases.

2.1 Multilevel B-spline Approximation

The Multilevel B-spline Approximation (MBA) algorithm (Lee, Wolberg, and Shin 1997; Hjelle 2001) is a method, which reconstruct a surface as the sum of different levels of surface approximation

$$z(\boldsymbol{s}) = z^0(\boldsymbol{s}) + z^1(\boldsymbol{s}) + \cdots + z^q(\boldsymbol{s}) \tag{2.1}$$

where $z^i(\boldsymbol{s}) \subset S_i,\ i = 0, \ldots, q$ are surfaces where q depends on the desired approximation and $S_0, \ldots, S_q$ is a nested sequence of subspace

$$S_0 \subset S_1 \subset \cdots \subset S_q.$$

This algorithm is based on a recursive refinement of the surface depending on the level of desired accuracy (q). The approximation level of the algorithm, q, can be chosen through a k-fold cross-validation procedure (Arlot and Celisse 2010), considering for example the RMSE as performance index. In Equation (2.1), the superscript is used to highlight that the predicted surface is a sum of different surfaces computed on finer grids. The predicted surface shown in Equation (2.1), after the refinement operation is completed, is eventually given by the following expression

$$\widehat{z}(\boldsymbol{s}) = \widehat{z}(u, v) = \sum_{k=0}^{3} \sum_{l=0}^{3} B_k(r) B_l(t) \widehat{\varphi}_{(i+k)(j+l)} \tag{2.2}$$

where $\widehat{\varphi}_{(i+k)(j+l)}$ is the estimated regressor, $i = \lfloor u \rfloor - 1$, $j = \lfloor v \rfloor - 1$, $r = u - \lfloor u \rfloor$, $t = v - \lfloor v \rfloor$ and $B_\bullet$ are uniform cubic B-spline defined as:

$$B_0(t) = \frac{(1-t)^3}{6} \tag{2.3}$$

$$B_1(t) = \frac{3\,t^3 - 6\,t^2 + 4}{6} \tag{2.4}$$

$$B_2(t) = \frac{-3\,t^3 + 3\,t^2 + 3\,t + 1}{6} \tag{2.5}$$

$$B_3(t) = \frac{t^3}{6} \tag{2.6}$$

with $0 \leq t < 1$.

The main idea of the MBA algorithm is now described (for a detailed explanation, the interested reader can refer to (Hjelle 2001)).

Let Φ_k be a control lattice (see Figure 2.1) defined on a rectangular domain $\Omega = [0, m) \times [0, n)$ and $\varphi_{i,j}$ the value of the *ij-th* control point at (i, j) for $i = -1, 0, \ldots, m+1$ and $j = -1, 0, \ldots, n+1$.

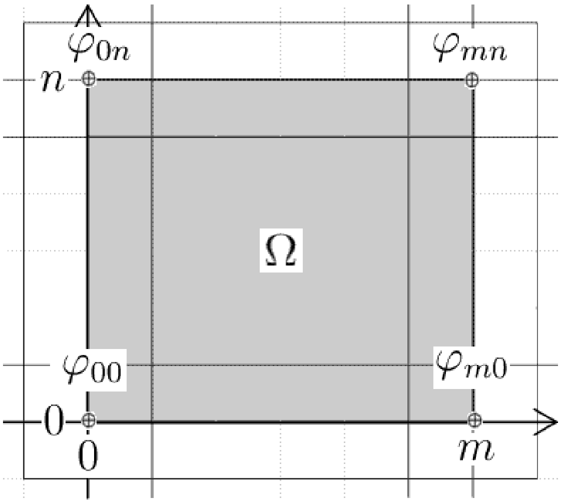


Figure 2.1: Lattice control scheme over domain

The main idea of the MBA algorithm is to estimate the B-spline coefficients on a coarse grid Φ_0. The resulting prediction function $z^0(\boldsymbol{s})$ is an initial approximation of the surface. After that, the algorithm computes the deviation between the predicted surface $z^0(\boldsymbol{s})$ and the available points and compute the B-spline coefficients on a finer grid Φ_1. After this step, the predicted surface is the sum of the surface $z^0(\boldsymbol{s})$ and the surface $z^1(\boldsymbol{s})$, which was in turn based on the discrepancies. The procedure is iterated until a desired control lattice data Φ_q is reached.

From Equation (2.2), it is possible to observe that only $4 \times 4 = 16$ coefficients (φ_{ij}) are used to predict a point $\widehat{z}(\boldsymbol{s})$ in a location $\boldsymbol{s} = (u, v)$. Since each regressor $(B_k(t)$ with $t = 0, 1, 2, 3)$ has a limited support, i.e. it is different from 0 on a closed domain, only some points contribute to its estimation (as shown in Figure 2.2). These points depend on the extension of the specific grid Φ_k, which in turn depends on the current approximation level (ranging from 0 to q). Let P_k denote the set of points influencing the estimation of the coefficients on the current grid

Φ_k.

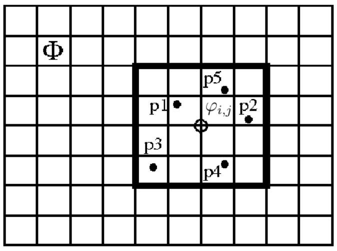


Figure 2.2: (4×4)- neighbors of a regressor coefficient and points used for its estimation

Let z_c be the value observed at one of the locations in the set P_k, this point influences all the 16 coefficients in the grid Φ_k through the equation:

$$\widehat{\varphi}_{kl}|c = \frac{B_k(r)\, B_l(t)\, z_c}{\sum_{a=0}^{3}\sum_{b=0}^{3} B_a^2(r) B_b^2(t)}, \quad (k,l) = (0,0),(0,1),\dots,(3,3) \tag{2.7}$$

If the cardinality of the set P_k is bigger than one, then an overlapping between the contributions of different points to the B-spline coefficients is present. As an example, Figure (2.3) shows that some coefficients (corresponding to positions where both a cross and a circle are drawn) are influenced by both the points $p1$ and $p2$ shown in the Figure. Therefore, the coefficient estimation should mix all the contributions due to the neighbour data, and can be computed as (Hjelle and Daehlen 2005)

$$\widehat{\varphi}_{kl} = \frac{\sum_c B_i^2(r) B_j^2(t) \widehat{\varphi}_{kl}|c}{\sum_{a=0}^{3}\sum_{b=0}^{3} B_a^2(r) B_b^2(t)}, \quad c \in P_k \tag{2.8}$$

where $\widehat{\varphi}_{kl}|c$ is the regressor estimated via equation (2.7).

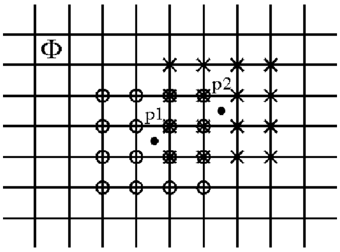


Figure 2.3: (4×4)-neighbors of coefficients in the tensor product grid of two points p_1 and p_2

2.2 Least Squares B-spline Approximation

Once the MBA algorithm is run, the surface is roughly estimated. In order to refine the reconstruction, the solution provided by MBA is taken as starting reference for a finer estimation procedure called Least Squares B-spline Approximation (LSBA) (Hjelle and Daehlen 2005).

The LSBA approach assumes that the response function can be described by the model

$$\boldsymbol{z} = \boldsymbol{F}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.9}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ is the vector of the observed data, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of coefficients, $\boldsymbol{F} \in \mathbb{R}^{n \times p}$ is the sparse matrix of B-spline regressors

$$\boldsymbol{F} = \begin{bmatrix} B_1(u_1, v_1) & \dots & B_p(u_1, v_1) \\ B_1(u_2, v_2) & \dots & B_p(u_2, v_2) \\ & \vdots & \\ B_1(u_n, v_n) & \dots & B_p(u_n, v_n) \end{bmatrix}$$

where $B_\bullet$ correspond to the B-spline regressors, $\lambda \geq 0$ is a smoothing penalty coefficient and $\boldsymbol{\varepsilon}$ is the vector the measurement errors, which are assumed to be independent and identically distributed normal random variables. The LSBA compute the unknown coefficients by minimizing the function

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^{n} (z_i - \widehat{z}_i)^2 + \lambda\,\boldsymbol{\beta}^T \boldsymbol{E}\boldsymbol{\beta} \\ &= (\boldsymbol{z} - \boldsymbol{F}\boldsymbol{\beta})^T\,(\boldsymbol{z} - \boldsymbol{F}\boldsymbol{\beta}) + \lambda\,\boldsymbol{\beta}^T \boldsymbol{E}\boldsymbol{\beta} \end{aligned} \tag{2.10}$$

where z_i and $\widehat{z}_i$ are the value observed and predicted at location $\boldsymbol{s}_i$ while the second term is a penalty term, where $\boldsymbol{E} \in \mathbb{R}^{n \times n}$ is a symmetric sparse matrix whose elements are

$$e_{ij} = \iint \nabla B_i(u, v) \circ \nabla B_j(u, v)\, dudv$$

where ∇ is the divergence operator and $\circ$ is the vector scalar product.

The minimum value of (2.10) respect to $\boldsymbol{\beta}$ can be found by solving the sparse linear system

$$(\boldsymbol{F}^T \boldsymbol{F} + \lambda \boldsymbol{E})\,\boldsymbol{\beta} = \boldsymbol{F}^T\,\boldsymbol{z}. \tag{2.11}$$

Since the matrix $(\boldsymbol{F}^T\boldsymbol{F}+\lambda\boldsymbol{E})$ is a big sparse matrix, an iterative solver has to be preferred to a direct solver. Usually, the convergence rate of an iterative algorithm depends on the initial guess of $\boldsymbol{\beta}$. Here the MBA solution comes to help. In fact, the output of the MBA algorithm is used as the initial starting point for solving the sparse linear system in Equation (2.11) with an iterative method (Hjelle and Daehlen 2005). The iterative methods used in this thesis is the conjugate gradient method implemented in the Eigen library (Guennebaud et al. 2010). The LSBA method is slow compared to the MBA algorithm, but usually produces better surface reconstruction.

In order to chose the smoothing parameter λ a k-fold cross-validation procedure with the RMSE as performance index is used. As suggested in Hjelle and Daehlen (2005), λ is chosen between the multiples of

$$\lambda_d=\frac{\|\boldsymbol{F}^T\boldsymbol{F}\|_F}{\|\boldsymbol{E}\|_F}$$

where $\|\bullet\|_F$ is the Frobenius matrix norm.

The least squares estimation of $\boldsymbol{\beta}$ should be written as (Ruppert, Wand, and Carrol 2003)

$$\widehat{\boldsymbol{\beta}}=\left(\boldsymbol{F}^T\boldsymbol{F}+\lambda\boldsymbol{E}\right)^{-1}\boldsymbol{F}^T\boldsymbol{z}, \tag{2.12}$$

while the variance covariance matrix of the vector estimate $\widehat{\boldsymbol{\beta}}_l$ is given by:

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}}=\sigma^2\left(\boldsymbol{F}^T\boldsymbol{F}+\lambda\boldsymbol{E}\right)^{-1}\boldsymbol{F}^T\boldsymbol{F}\left(\boldsymbol{F}^T\boldsymbol{F}+\lambda\boldsymbol{E}\right)^{-1}. \tag{2.13}$$

Eventually, the estimate of σ^2 required in the last equation can be given by:

$$\widehat{\sigma}^2=\frac{\|\boldsymbol{z}-\widehat{\boldsymbol{z}}\|}{n-\operatorname{tr}\left[\boldsymbol{F}\left(\boldsymbol{F}^T\boldsymbol{F}+\lambda\boldsymbol{E}\right)^{-1}\boldsymbol{F}^T\right]} \tag{2.14}$$

or (approximately) by:

$$\widehat{\sigma}^2=\frac{\|\boldsymbol{z}-\widehat{\boldsymbol{z}}\|}{n} \tag{2.15}$$

if the dataframe is big.

Once all the parameters have been estimated, the prediction at any new location point $\boldsymbol{s}_0$ can be computed as

$$\mathbb{E}[\widehat{z}(\boldsymbol{s}_0)]=\boldsymbol{f}^T(\boldsymbol{s}_0)\,\widehat{\boldsymbol{\beta}} \tag{2.16}$$

and the corresponding prediction variance is given by:

$$\mathbb{V}ar\left[\widehat{z}(\boldsymbol{s}_0)\right]=\mathbb{V}ar\left[\boldsymbol{f}^T(\boldsymbol{s}_0)\widehat{\boldsymbol{\beta}}\right]=\boldsymbol{f}^T(\boldsymbol{s}_0)\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\beta}}}\boldsymbol{f}(\boldsymbol{s}_0). \tag{2.17}$$

2.3 Least Square B-spline Approximation for Data Fusion when Big Data are Available

The previous approach for surface reconstruction can be used in both the first and the second stages of the data fusion approach. In this section, a fusion model using LSBA is shown.

As in section 1.3 the LSBA algorithm is firstly applied to the Lo-Fi data at the first stage. We can thus rewrite equation (2.9) using l as subindex to denote the Lo-Fi data:

$$\boldsymbol{z}_l = \boldsymbol{F}_l \boldsymbol{\beta}_l + \boldsymbol{\varepsilon}_l \tag{2.18}$$

Then, the second stage of the hierarchical model is used to model the discrepancy between the observed Hi-Fi points and the corresponding Lo-Fi data predicted at the same location (using to the first-stage model). The model of the second stage is thus given by:

$$\boldsymbol{z}_\Delta = \boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l = \boldsymbol{F}_\Delta \boldsymbol{\beta}_\Delta + \boldsymbol{\varepsilon}_\Delta \tag{2.19}$$

where $\widehat{\boldsymbol{z}}_l \in \mathbb{R}^{n_h}$ are the predictions computed using the first stage model in the Hi-Fi locations and $\boldsymbol{\varepsilon}_\Delta \in \mathbb{R}^{n_h}$ is a vector of independent and identically distributed normal random variables with 0 mean and variance equal to σ^2_Δ. These two vectors are assumed independent.

The covariance of the fusion model can be computed as

$$\mathbb{C}ov(\boldsymbol{z}_h) = \mathbb{C}ov(\widehat{\boldsymbol{z}}_l) + \mathbb{C}ov(\boldsymbol{\varepsilon}_\Delta). \tag{2.20}$$

The vector $\boldsymbol{\beta}_\Delta$ of estimated coefficients can be computed using the LSBA approach

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_\Delta &= \left(\boldsymbol{F}_\Delta^T \boldsymbol{F}_\Delta + \lambda_\Delta \boldsymbol{E}_\Delta\right)^{-1} \boldsymbol{F}_\Delta^T \left(\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l\right) && (2.21)\\ &= \boldsymbol{Q}_\Delta \boldsymbol{z}_h - \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h) \boldsymbol{Q}_l \boldsymbol{z}_l && (2.22) \end{aligned}$$

where $\boldsymbol{S}_h = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{h}]^T$ is the matrix of Hi-Fi locations and hence $\boldsymbol{F}_l(\boldsymbol{S}_h) = [\boldsymbol{f}_l(\boldsymbol{s}_1), \boldsymbol{f}_l(\boldsymbol{s}_2),\ \ldots, \boldsymbol{f}_l(\boldsymbol{s}_{n_h})]^T$ represents the matrix of $\boldsymbol{f}_l(\bullet)$ functions applied to the Hi-Fi locations and

$$\begin{aligned} \boldsymbol{Q}_\Delta &= \left(\boldsymbol{F}_\Delta^T \boldsymbol{F}_\Delta + \lambda_\Delta \boldsymbol{E}_\Delta\right)^{-1} \boldsymbol{F}_\Delta^T \\ \boldsymbol{Q}_l &= \left(\boldsymbol{F}_l^T \boldsymbol{F}_l + \lambda_l \boldsymbol{E}_l\right)^{-1} \boldsymbol{F}_l^T. \end{aligned}$$

The variance in Equation (2.20) can be estimated as

$$\mathbb{C}ov(\boldsymbol{z}_h) = \mathbb{C}ov(\widehat{\boldsymbol{z}}_l) + \mathbb{C}ov(\boldsymbol{\varepsilon}_\Delta) = \left(\overline{\sigma}_l^2 + \sigma_\Delta^2\right) \boldsymbol{I}_{n_h}, \tag{2.23}$$

where the variance σ_Δ^2 can be estimated with Equation (2.14) or (2.15) while the variance of the Lo-Fi data is assumed constant everywhere and thus estimated as

$$\overline{\sigma}_l^2 = \frac{1}{|S|} \int_S \widehat{\sigma}_l^2(\boldsymbol{s})\, d\boldsymbol{s} \tag{2.24}$$

where $\widehat{\sigma}_l^2(\boldsymbol{s})$ is the predicted variance in the spatial location $\boldsymbol{s} = (u, v)$, S is the surface domain and $|S|$ is its area.

According to this model, the prediction at a new location $\boldsymbol{s}_0$ is given by:

$$\begin{aligned} \widehat{z}_h(\boldsymbol{s}_0) =& \widehat{z}_l(\boldsymbol{s}_0) + \widehat{z}_\Delta(\boldsymbol{s}_0) = \boldsymbol{f}_l^T(\boldsymbol{s}_0)\widehat{\boldsymbol{\beta}}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\widehat{\boldsymbol{\beta}}_\Delta \\ =& \boldsymbol{f}_l^T(\boldsymbol{s}_0)\boldsymbol{Q}_l\boldsymbol{z}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta(\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l) \\ =& \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{z}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{z}_h \end{aligned} \tag{2.25}$$

The variance of the prediction in a new location $\boldsymbol{s}_0$ is computed by

$$\begin{aligned} \mathbb{V}ar[\widehat{z}_h(\boldsymbol{s}_0)] =& \mathbb{V}ar\left[\left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{z}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{z}_h\right] \\ =& \mathbb{V}ar\left[\left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{z}_l\right] + \mathbb{V}ar\left(\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{z}_h\right) \\ &+ 2\,\mathbb{C}ov\left[\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{z}_h, \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{z}_l\right] \\ =& \sigma_l^2\left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{Q}_l^T \\ &\cdot \left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h)\boldsymbol{Q}_\Delta^T\boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) \\ &+ \left(\overline{\sigma}_l^2 + \sigma_\Delta^2\right)\boldsymbol{f}_\Delta(\boldsymbol{s}_0)^T\boldsymbol{Q}_\Delta\boldsymbol{Q}_\Delta^T\boldsymbol{f}_\Delta(\boldsymbol{s}_0) \\ &+ 2\,\sigma_l^2\boldsymbol{f}_\Delta(\boldsymbol{s}_0)^T\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\boldsymbol{Q}_l\boldsymbol{Q}_l^T\left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h)\boldsymbol{Q}_\Delta^T\boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) \end{aligned} \tag{2.26}$$

where $\boldsymbol{F}_l(\boldsymbol{S}_h)$ is the Lo-Fi model matrix in the Hi-Fi points' locations and

$$\begin{aligned} &\mathbb{C}ov\left[\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{z}_h, \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\right)\boldsymbol{Q}_l\boldsymbol{z}_l\right] \\ &\quad= \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{Q}_\Delta\mathbb{C}ov\left[\boldsymbol{F}_l(\boldsymbol{S}_h)\boldsymbol{Q}_l\boldsymbol{z}_l + \boldsymbol{F}_\Delta\boldsymbol{\beta}_\Delta + \boldsymbol{\varepsilon}_\Delta, \boldsymbol{z}_l\right]\boldsymbol{Q}_l^T \\ &\quad\cdot \left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h)\boldsymbol{Q}_\Delta^T\boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) \\ &\quad= \sigma_l^2\boldsymbol{f}_\Delta(\boldsymbol{s}_0)^T\boldsymbol{Q}_\Delta\boldsymbol{F}_l(\boldsymbol{S}_h)\boldsymbol{Q}_l\boldsymbol{Q}_l^T\left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h)\boldsymbol{Q}_\Delta^T\boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) \end{aligned}$$

because $\boldsymbol{z}_l$ and $\boldsymbol{\varepsilon}_\Delta$ are independent.

2.3.1 Test Case

The test case of section 1.4 is used to check also the results of the model based on LSBA. The Lo-Fi data are simulated using the Matlab peaks function from

$$z_l(\boldsymbol{s}) = f(\boldsymbol{s}) + \text{bias}(\boldsymbol{s}) + \varepsilon_l, \quad \varepsilon_l = \mathcal{N}(0, 0.4^2)$$

while the Hi-Fi data are drawn from

$$z_h(\boldsymbol{s}) = f(\boldsymbol{s}) + \varepsilon_h, \quad \varepsilon_\Delta = \mathcal{N}(0, 0.2^2)$$

where $f(\boldsymbol{s})$ is the peaks function:

$$f(\boldsymbol{s}) = 3(1-u)^2\, e^{-u^2-(v+1)^2} - 10\left(\frac{u}{5} - u^3 - v^5\right) e^{-u^2-v^2} - \frac{1}{3}e^{-(u+1)^2-v^2}$$

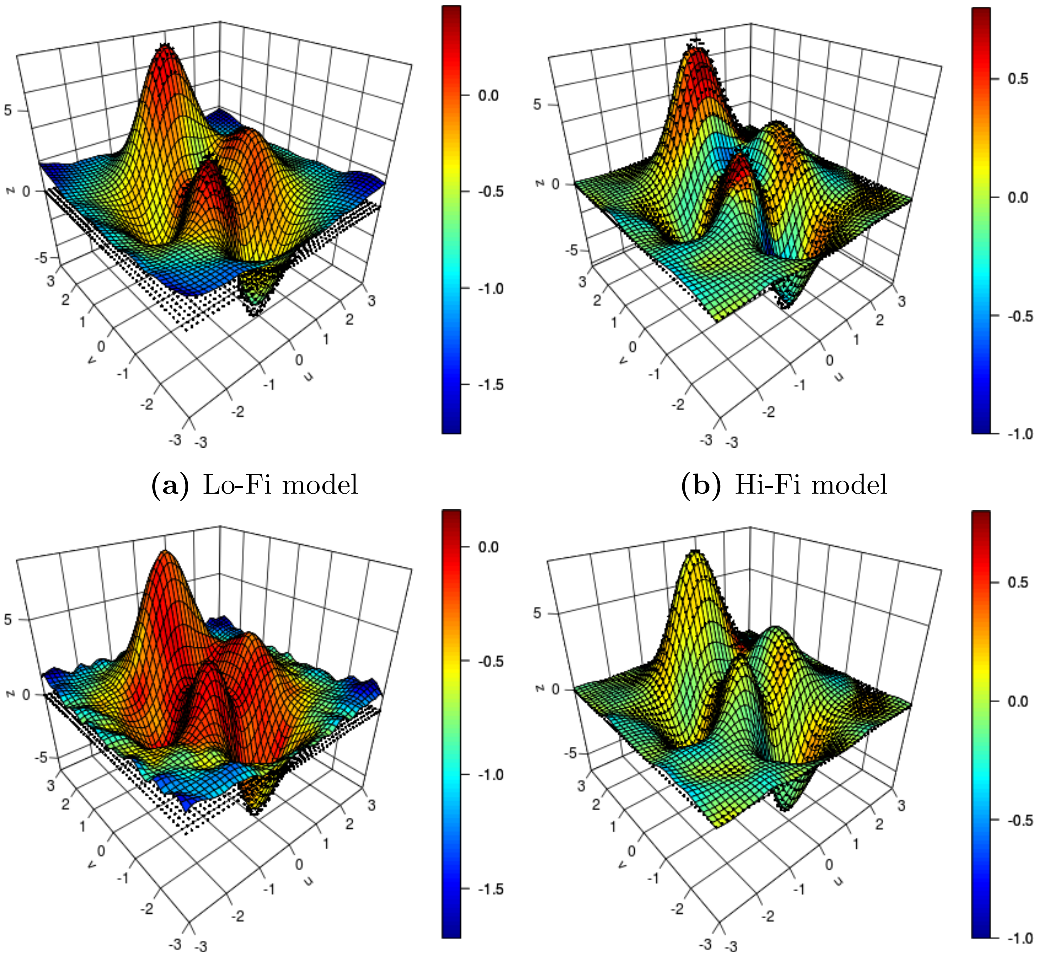
and the bias term is set to

$$\text{bias}(\boldsymbol{s}) = \frac{u^2}{10} + \frac{v^2}{10}.$$

Also in this case, competitor models considered are: *i*) (and *ii*)) the Lo-Fi (Hi-Fi) models that uses LSBA reconstruction using the Lo-Fi (Hi-Fi) data only; *iii*) the addition model, which sums up all the Lo-Fi and Hi-Fi data and use a single LSBA reconstruction as the data were provided by a single sensor; *iv*) the LSBA data fusion model presented.

Surfaces reconstructed using the different models together with the corresponding error maps are shown in Figure 2.4. As for the GP-based models, also LSBA-based models present similar results. In particular, the Lo-Fi error map reflects the systematic error (bias) of the Lo-Fi measurement system. The prediction of the addition model in the flat zone, where there is the bigger bias, has a wave form because the model try to approximate both the Lo-Fi and the Hi-Fi data. The fusion model seems to be able to correct the Lo-Fi bias.

Also in this case, the simulation was repeated 10 times and the RMSE of each model was computed. Figure 2.5 shows the 95% confidence interval on the RMSE observed using the four possible models; once again the fusion model achieves the better result also when LSBA is used instead of GP as reconstructing technique.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Addition model

(d) Fusion model

Figure 2.4: Predicted surfaces with error map

2.4 Case Study: Free Form Surface Reconstruction

In the free form case study of section 1.5, a total amount of 9635 Lo-Fi (SL) points were available. Although the GP model was able to reconstruct the Lo-Fi surface (at the first stage) using all these data, computation required about 1 hour. Just to give a quick result, the same problem faced using the LSBA approach required less than one second.

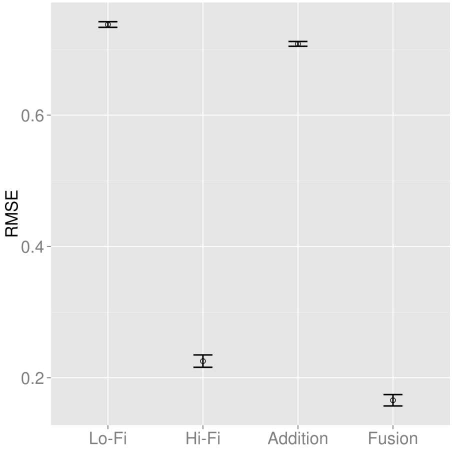
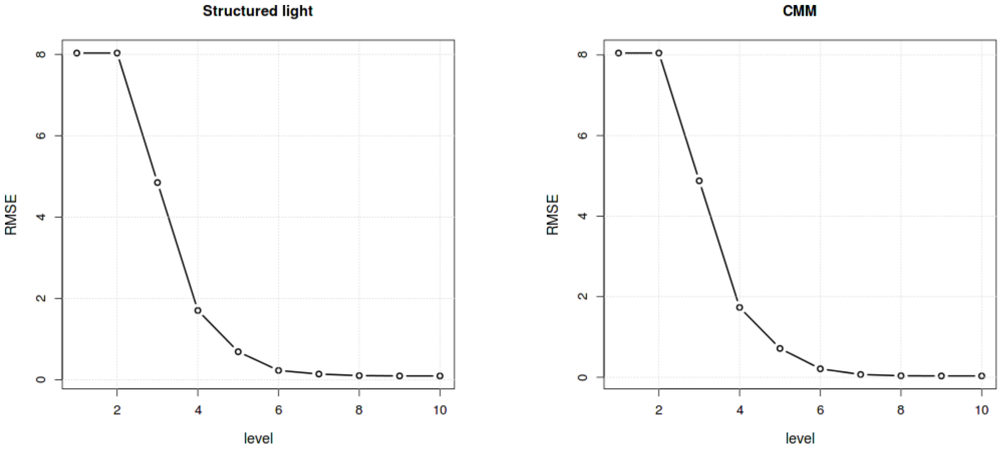


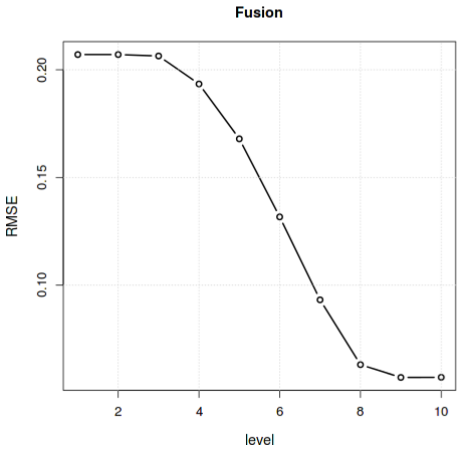
Figure 2.5: Models' RMSE

In order to show how LSBA data fusion can be applied in real case study, the first step consists in applying the MBA algorithm, thus deciding the approximation level q. As mentioned in section 2.1, a k-fold cross-validation technique can be used to this aim. We used this approach considering 70% of the available points as training set and the remaining points as test set. The k-fold cross validation procedure was repeated 5 times. The average values of the RMSE computed on these 5 runs are shown in Figure 2.6a, 2.6b and 2.6c, for the Lo-Fi, Hi-Fi and fusion model, respectively.



(a) Lo-Fi (Structured light)

(b) Hi-Fi (CMM)

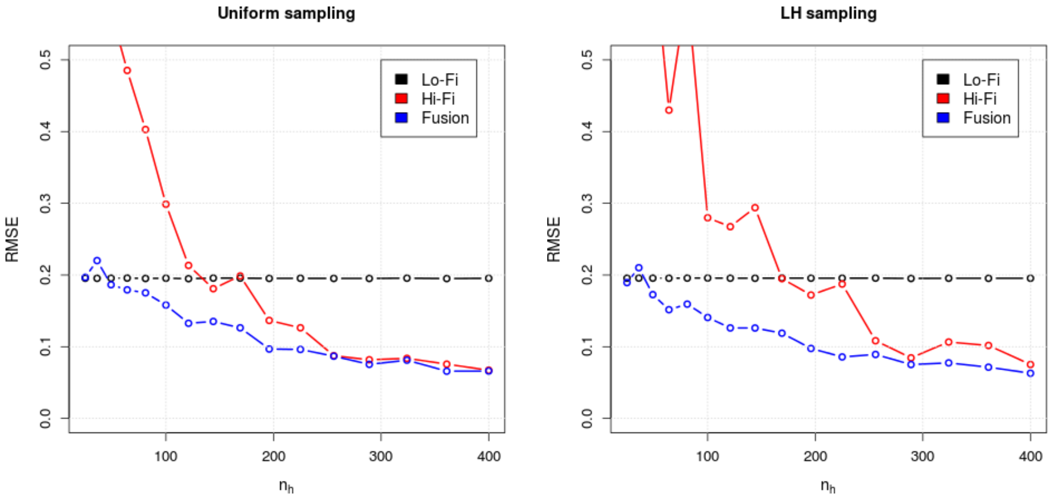


(c) Data Fusion

Figure 2.6: RMSE of the cross-validation method

According to Figure 2.6, the level 6 was selected set for the Lo-FI (SL) and Hi-Fi (CMM) models, while a level 8 was selected for the fusion model.

The RMSE of the LSBA-based models are shown in Figure 2.7. The RMSE have similar patterns observed when using the GP algorithm.

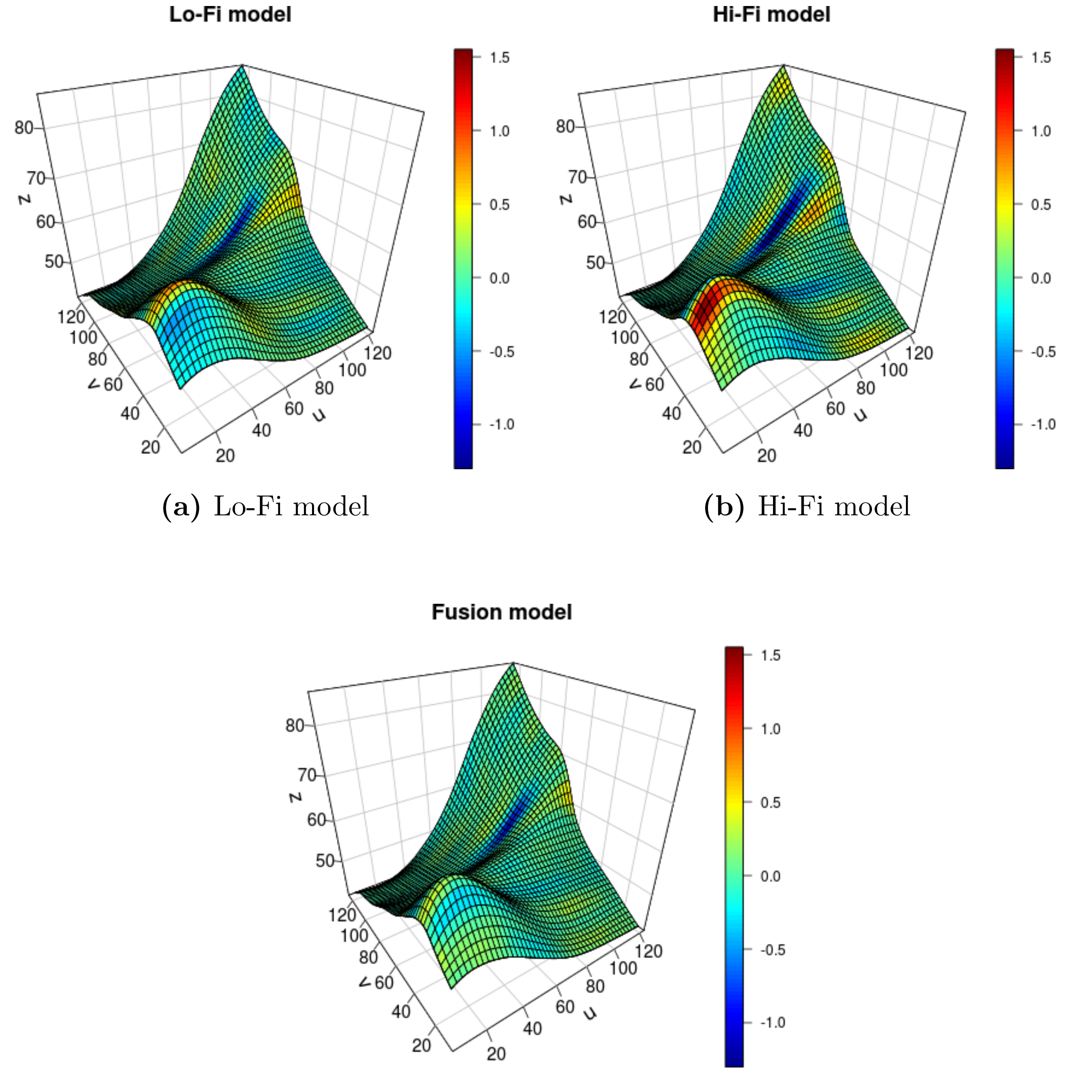


(a) Uniform sampling **(b)** Latin hypercube sampling

Figure 2.7: RMSE of the LSBA model with different sampling strategy

Examples of the LSBA-based error maps are shown in Figure 2.8. These plots are based on a number of 100 Hi-Fi points acquired according to LH sampling.

In this example the error maps of the models analyzed (Lo-Fi, Hi-Fi and fusion) have the same behavior of those of GP models (see Figure 1.16). Also in this test case the fusion model can efficiently corrects the Lo-Fi surface using the available Hi-Fi data.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Fusion model

Figure 2.8: Prediction of the free form surface with error map of the three models based on a LH sampling with 100 Hi-Fi points

In Table 2.1, computational times, in seconds, needed by the LSBA-based and GP-based fusion approaches are shown. It is possible to observe that the mean estimation time for the GP approach is approximately 23 s; it is not possible see an increasing trend when the number of Hi-Fi points increase because a lot of time is spent for the computation of the Lo-Fi model predictions in the Hi-Fi locations. The estimation time of the LSBA algorithm is about 0.16 s, almost $\frac{1}{100}$ of the time needed by the GP; also with this algorithm it is not possible to observe any trend.

n_h	GP		LSBA	
	Uniform sampling	LH sampling	Uniform sampling	LH sampling
25	22.26	21.98	0.18	0.17
36	22.78	21.65	0.17	0.16
49	22.13	22.41	0.17	0.16
64	23.06	22.91	0.15	0.16
81	21.69	22.88	0.15	0.16
100	22.93	22.84	0.15	0.16
121	21.63	22.13	0.14	0.16
144	22.80	22.66	0.14	0.15
169	24.07	22.77	0.14	0.15
196	23.80	22.75	0.24	0.14
225	23.29	25.28	0.21	0.15
289	24.81	25.13	0.19	0.14
324	23.66	26.38	0.18	0.14
361	23.68	27.28	0.18	0.14
400	25.88	28.04	0.16	0.14

Table 2.1: Time (s) of the GP and LSBA fusion models analyzed

The comparison of the RMSE values of the GP and the LSBA models are shown in Figure 2.9. It is possible to observe that, if there are few Hi-Fi points, the GP-based model can achieve better results. As the number of Hi-Fi points increases to 100, the RMSE values of the LSBA-based and GP-based models are comparable. It should also be noted that RMSE of the GP model depends on the sampling plan used. On the contrary the performance of the LSBA models does not depend on the sampling plan of the Hi-Fi points, i.e. the LSBA model is more robust with respect to the sampling plan.

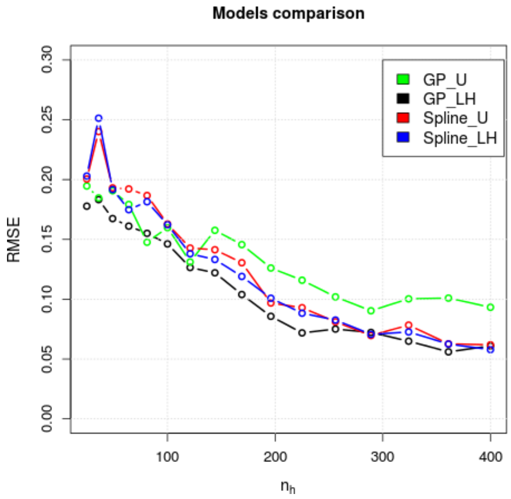


Figure 2.9: Performance comparison between LSBA and GP models

2.5 Case study: Micro Surface Reconstruction

In this section the LSBA model is applied for the reconstruction of a free form surface of the NPL Bento Box (NPL). The point clouds were acquired with a confocal microscope (Lo-Fi) and an atomic force microscope (AFM). The number of acquired points were 198 916 and 106 276, respectively for the Lo-Fi and the Hi-Fi devices. The aim of this section is to explore benefits provided by data fusion in the context of big point clouds. The performance of the fusion model will be computed as a function of the Hi-Fi data available. A brief overview of the two measurement devices is firstly presented.

Confocal Microscope

Confocal microscope is an optical imaging technique that enables the reconstruction of three-dimensional structures from the obtained images. A simple configuration of a confocal microscope is drawn in Figure 2.10.

A light source illuminates a pinhole object to the sample through the objective lens to create a point illumination. The light is reflected back to the lens and there is a second pinhole placed in front of the photo detector acting as spatial filter.

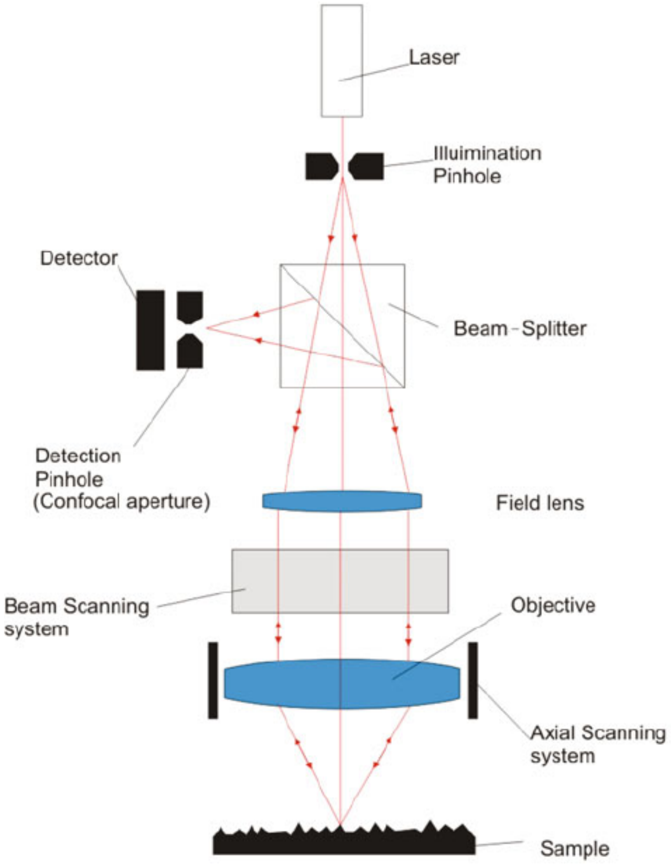
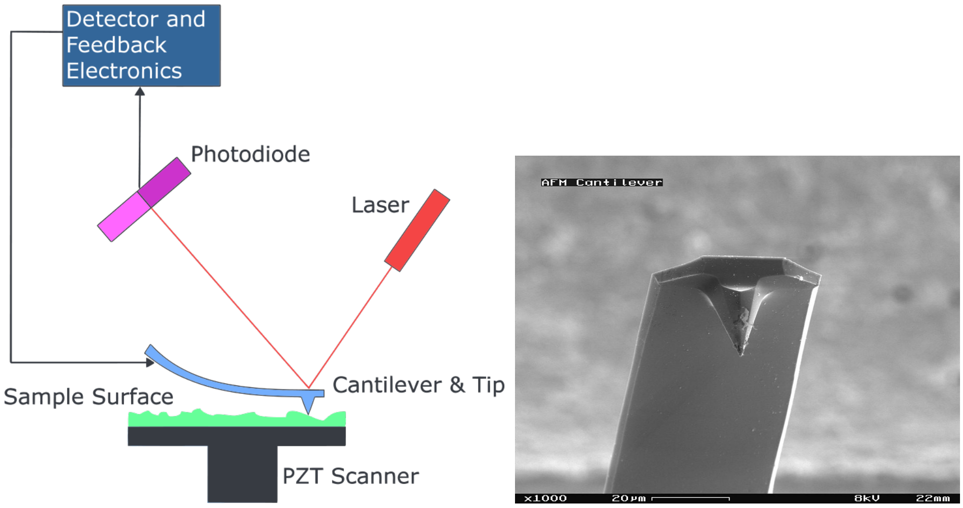


Figure 2.10: Scheme of a confocal microscope device (Leach 2013)

The microscope vertically scans the surface within a certain range. A point which is in focus position has higher intensity value on the detector. A fitting procedure is applied in order to find the point with the highest intensity, which represents the point on the measured surface.

Atomic Force Microscope

Atomic force microscope (AFM) is a very high-resolution type of contact-based scanning microscope. The resolution of AFM is in the order of fractions of a *nm*. The AFM consists of a cantilever with a probe at its end. This is used to scan the specimen surface. The cantilever is typically silicon or silicon nitride with a tip radius of curvature on the order of *nm*. When the tip is brought into proximity of a sample surface, forces between the tip and the sample lead to a deflection of the cantilever according to Hooke's law.



(a) AFM scheme

(b) Cantilever and tip

Figure 2.11: Atomic force microscope

A typical AFM probe is a consumable measuring device with a sharp tip on the free-swinging end of a cantilever that is protruding from a holder plate. The dimensions of the cantilever are in the scale of μm, while the radius of the tip is in the scale of few nm.

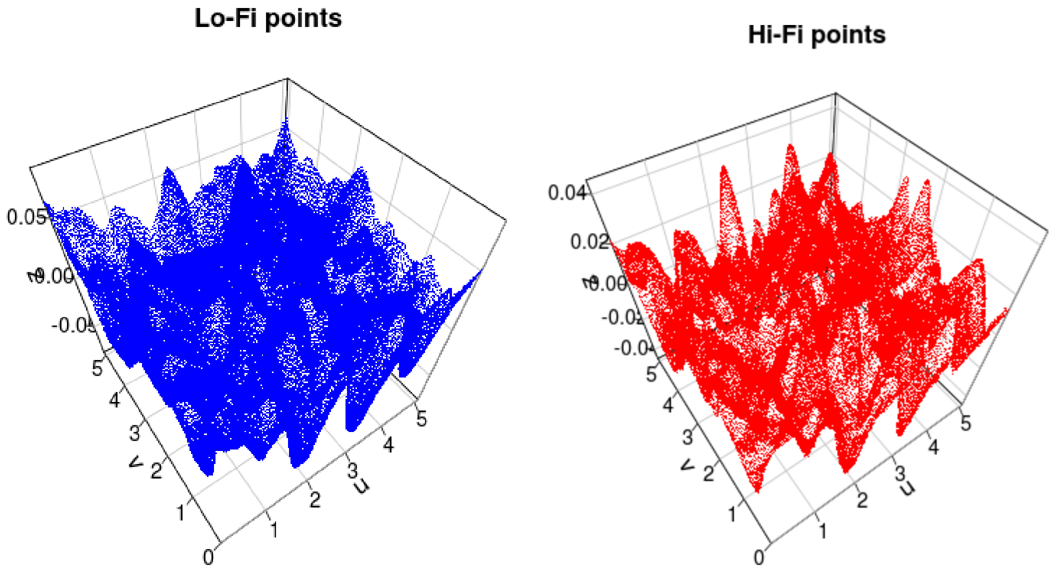
Samples viewed via AFM do not require any special treatment that would change or damage the sample.

Although it has a very good vertical and lateral resolutions, some disadvantages of the AFM are

- it can only acquire surface with a maximum height in the order of 10-20 μm and a maximum scanning area of about 150 × 150 μm;
- the scanning speed is very slow;
- there is the possibility of some artifacts, which could be induced by an unsuitable tip or a poor operating environment.

2.5.1 Numerical Results

The points acquired with the two devices are shown in Figure 2.12a and 2.12b, respectively for the confocal microscope and the AFM. All the values are expressed in μm. The discrepancy map between the point acquired point clouds is shown in Figure 2.13; it is possible to observe that the correction that the fusion model has to apply is not a simple polynomial function and that the error of the Lo-Fi device is bigger on the boundary of the u-v domain.



(a) Confocal microscope point cloud

(b) AFM point cloud

Figure 2.12: Available data

As in the previous section the following models were analyzed:

- Lo-Fi: model based only on the point cloud acquired with the confocal microscope;
- Hi-Fi: model based only on the point cloud acquired with the AFM;
- Fusion: fusion model based on both Lo-Fi and Hi-Fi point clouds.

The number of Hi-Fi points available was also considered as relevant factor. Since the Hi-Fi points are acquired on a square matrix, decimation was performed by taking a point each n in each direction. Since the sampling distance between the points in u and v directions was the same, the RMSE performance index was computed just as a function of the AFM sampling distance. Again, the performance comparison was based on the RMSE index. As in the previous test case, all the Hi-Fi points that were not used for the reconstruction (because of decimation), were used as test case (i.e., location were the predictions and real data are compared to compute errors).

Figure 2.14 shows the RMSE of all the analyzed models.

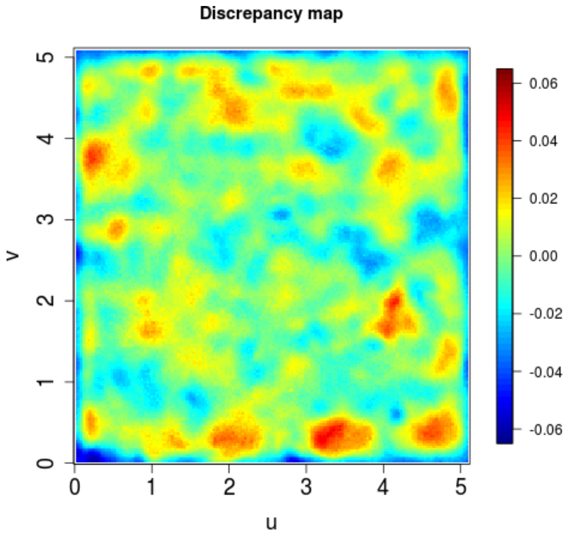


Figure 2.13: Discrepancy map between the Hi-Fi and the Lo-Fi points

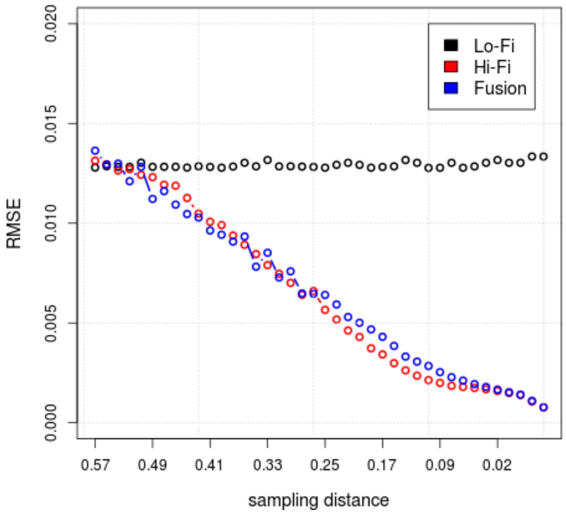
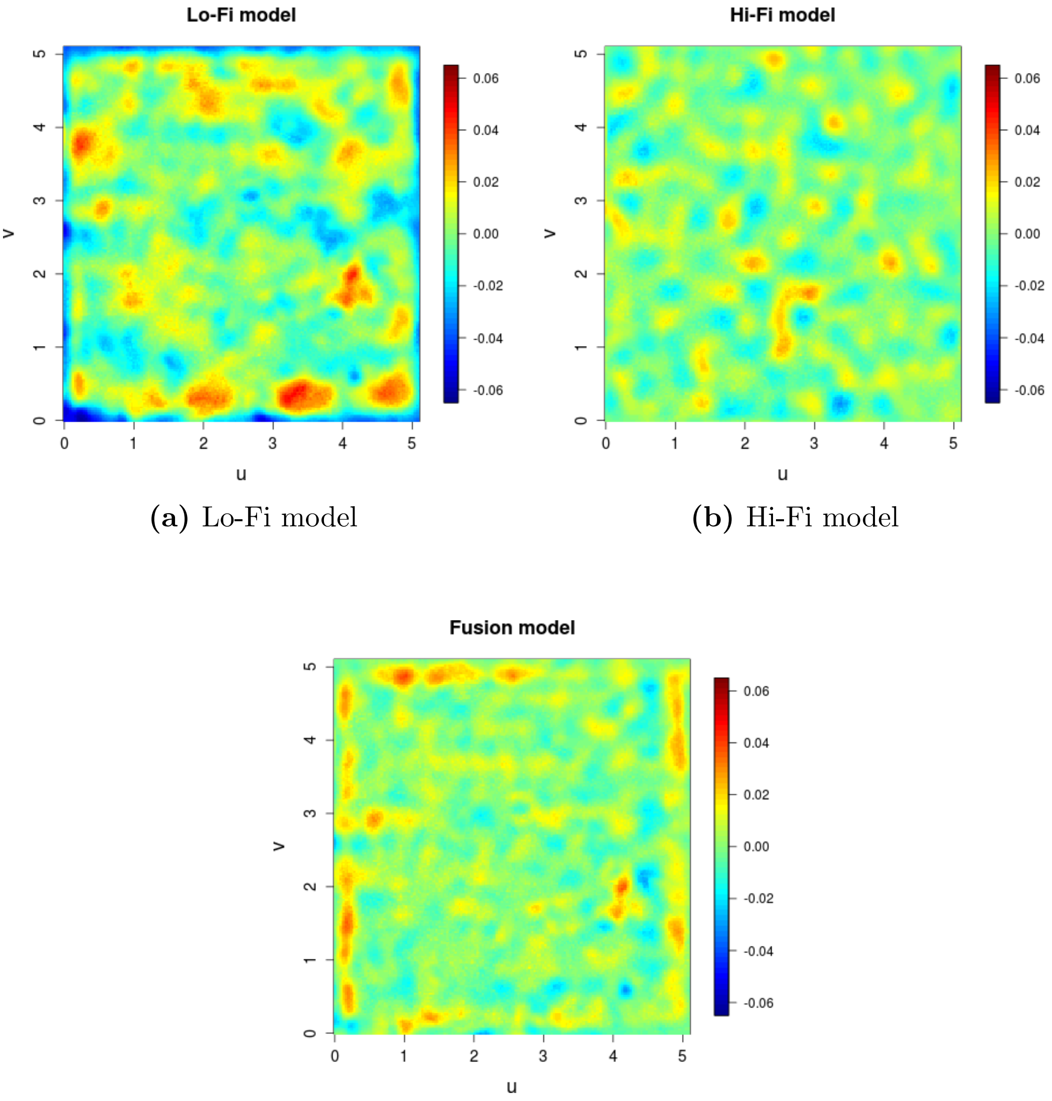


Figure 2.14: RMSE of the analyzed models

It is possible to observe that the performance of the Hi-Fi and the fusion models are comparable for all the values of the sampling distance. Both models achieve better prediction performance than the Lo-Fi model, unless the sampling distance between the Hi-Fi points is too large (i.e., Hi-Fi points are not enough to correct locally the Lo-Fi bias).

The discrepancy map between the surfaces predicted with the Lo-Fi, Hi-Fi and fusion models are shown in Figure 2.15a, 2.15b and 2.15c, respectively.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Fusion model

Figure 2.15: Error map of the surface reconstructed by competitor models

The Hi-Fi and the fusion surfaces are estimated using a sampling distance between the Hi-Fi points equal to 0.393 μm in each direction. The error map of the Lo-Fi model shows that the the larger errors are located at the border of the surface. The discrepancies of the Hi-Fi model are not located in a particular part of the surface. The Hi-Fi model does not present any specific local concentration of the errors. Eventually, the fusion model seems to be still influenced by the problem at the borders.

Since the error on the border of the Lo-Fi model was large, the range of discrepancies between the Lo-Fi and the Hi-Fi points was almost 75% of the total z range. In order to investigate the prediction ability of the fusion model is situations were less severe Lo-FI bias occurs, points on the boundary were removed (i.e., "cropped"). The discrepancy map of the cropped data sets is plotted in Figure 2.16. In this scenario, there are still some discrepancies that must to be corrected, but their magnitude is smaller if compared to the previous one.

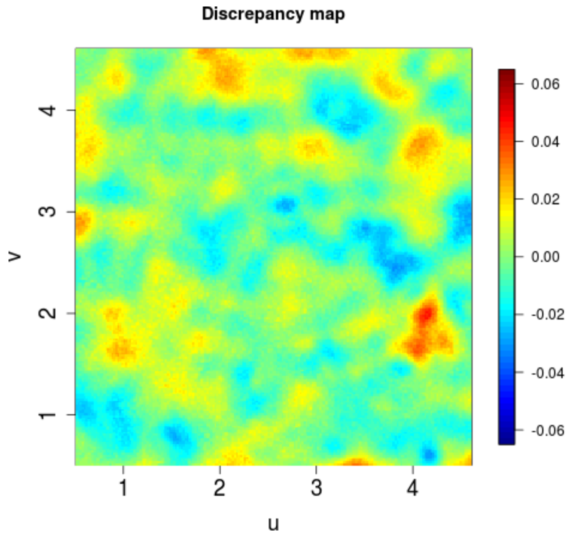


Figure 2.16: Discrepancy map between the Hi-Fi and the Lo-Fi points after deleting the border

The RMSE values of this second case are shown in Figure 2.17. It is possible to observe that the behavior of the RMSE of the LSBA models are similar of those observed for the free-form case study of section 2.4. The RMSE of the Lo-Fi model are smaller compared to the previous scenario, while those of the Hi-Fi model are comparable. The fusion model outperforms the others if the sampling distance between the Hi-Fi points is large, i.e. when there are few Hi-Fi points available.

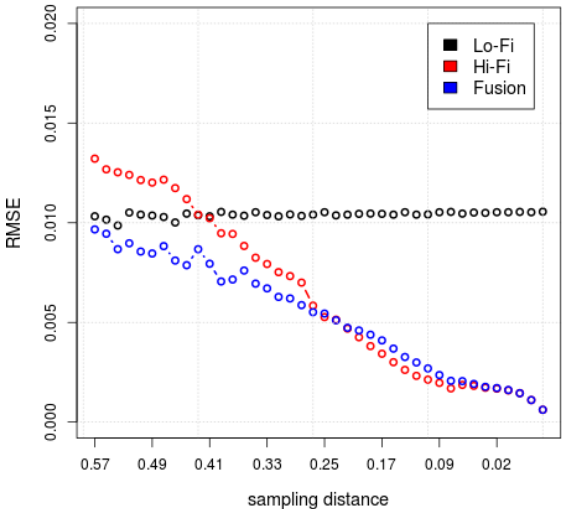
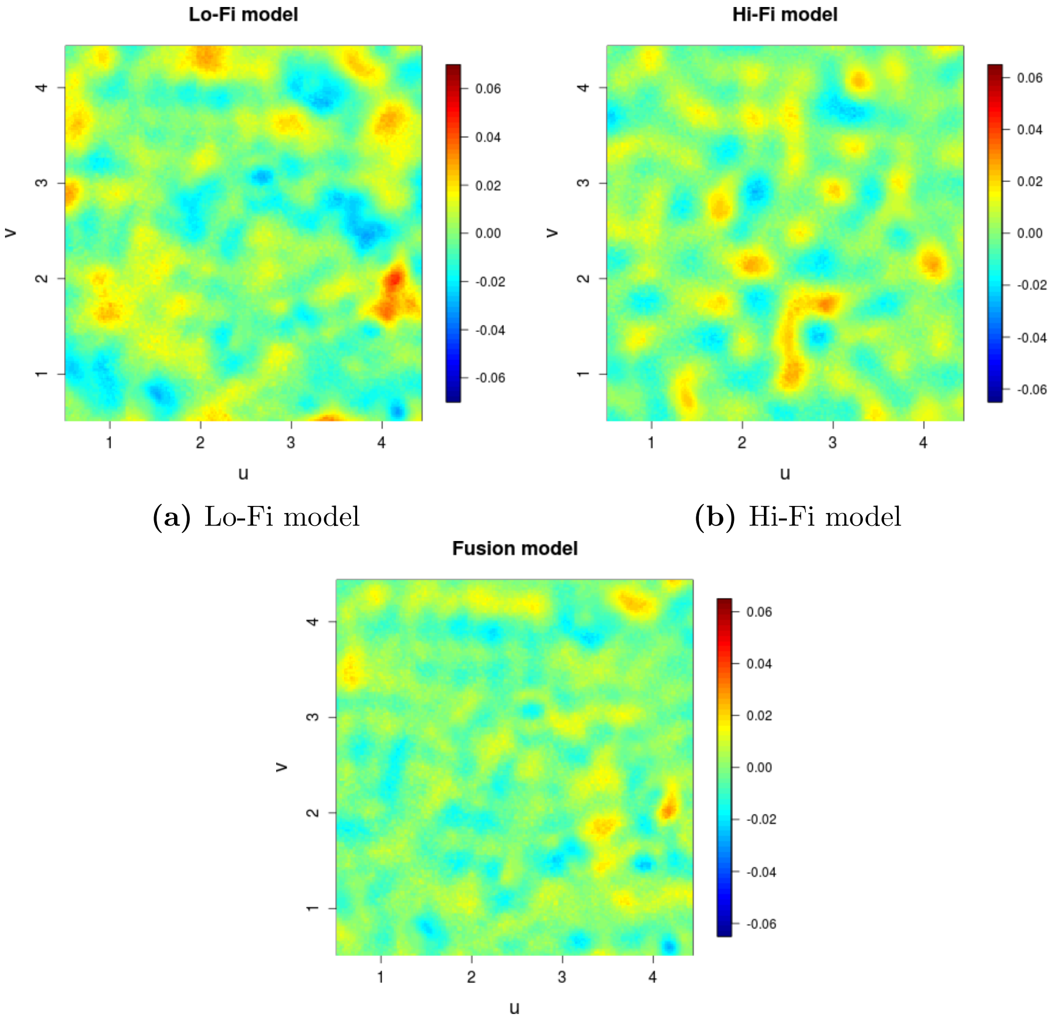


Figure 2.17: RMSE of the analyzed models

The discrepancy maps of the analyzed models (considering a sampling distance for the Hi-Fi data equal to 0.393 μm in each direction) are shown in Figure 2.18 It is possible to observe that the discrepancy maps of the Lo-Fi and Hi-Fi models (Figure 2.18a and 2.18b) are very close to the discrepancy maps of the original datasets (respectively Figure 2.15a and 2.15b). On the contrary, the discrepancy map of the fusion model shows that the bias effect on the Lo-Fi border is removed via data fusion.

In this section it has been shown that the fusion model depends on the goodness of the Lo-Fi data, i.e. if the error of the Lo-Fi device is large compared to the z range of the dataset, the fusion model can correct the Lo-Fi data, but it cannot take full advantage of the information provided by them.



(a) Lo-Fi model

(b) Hi-Fi model

(c) Fusion model

Figure 2.18: Error map between the AFM point cloud and the reconstructed surface by the analyzed models

Chapter 3

Modeling Uncertainty in Data Fusion

In the fusion LSBA model described in chapter 2, it is assumed that the variance of the predictions of the first stage (modeling the Lo-Fi data only) is constant everywhere. This assumption holds when the size of the Lo-Fi point cloud is large enough (i.e., the Lo-Fi data is a dense point cloud), when the Lo-Fi measurement system has constant error and when there are no missing points in the Lo-Fi dataset. In all the other cases, the variance of the predicted values $\widehat{z}_l$'s that enter in the data fusion model is not constant. The way in which this uncertainty propagates in the data fusion model has to be studied and appropriately modeled. In this chapter, the LSBA approach for data fusion is extended to handle the heteroscedasticity (i.e., non constant variance) of the difference between the Hi-Fi data and the Lo-Fi model. We will refer to a Weighted Least Square B-spline Approximation (WLSBA) to denote the LSBA where possible (local) changes of the variance are allowed and the fitting model is able to include this source of variability. The traditional LSBA presented in the previous chapter will be referred to simply as LSBA model, to emphasize that this model assumes constant variance everywhere (as all the approaches based on traditional Least Squares).

3.1 Weighted Least Squares B-spline Approximation for Handling Uncertainty Propagation

As in the previous chapters, it is assumed that the difference between the Hi-Fi data $z_h(u, v)$ observed at location (u, v) and the corresponding value predicted using the first-stage model $\widehat{z}_l(u, v)$ at the same location is given by

$$z_\Delta(u, v) = z_h(u, v) - \widehat{z}_l(u, v). \tag{3.1}$$

In this model, called the discrepancy model, the variance of the profile depends on the location, (u, v), because the prediction variance of $\widehat{z}_l(u, v)$, i.e., the values computed using the Lo-Fi model, depends by the location.

In order to include non constant variance in the LSBA fusion model (described in section 2.2) a weighted approach has to be considered. A small weight should be assigned to predictions with high variability (because they are less reliable). In traditional Weighted Least Squares (WLS) models (Johnson and Wichern 2007) the weight is equal to the inverse of the prediction variance at each location. We will follow the same approach to move from the LSBA to a WLSBA method for data fusion.

As described in Equation (2.19), Hi-Fi data can be modeled in the second-stage or linkage model as a function of Lo-Fi prediction $\widehat{\boldsymbol{z}}_l$ via LSBA:

$$\boldsymbol{z}_h = \widehat{\boldsymbol{z}}_l + \boldsymbol{F}_\Delta \boldsymbol{\beta}_\Delta + \boldsymbol{\varepsilon}_\Delta. \tag{3.2}$$

In the WLSBA, it is assumed that the variance of the discrepancies $\boldsymbol{z}_\Delta = \boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l$ is a multiple of the variance of $\widehat{\boldsymbol{z}}_l$, i.e., prediction variance computed via the first-stage model (by fitting the Lo-Fi data only):

$$\mathbb{C}ov(\boldsymbol{z}_\Delta) = \mathbb{C}ov(\boldsymbol{\varepsilon}_\Delta) = \mathbb{C}ov(\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l) = \sigma_\Delta^2 \boldsymbol{W}_\Delta^{-1} \tag{3.3}$$

where

$$\boldsymbol{W}_\Delta = \begin{bmatrix} \frac{1}{\widehat{\sigma}_l^2(\boldsymbol{s}_1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\widehat{\sigma}_l^2(\boldsymbol{s}_{n_h})} \end{bmatrix}.$$

Under a WLS procedure, the estimator of $\boldsymbol{\beta}_\Delta$ is given by

$$\widehat{\boldsymbol{\beta}}_\Delta = \left(\boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta \boldsymbol{F}_\Delta + \lambda_\Delta \boldsymbol{E}_\Delta\right)^{-1} \boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta \left(\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_l\right) \tag{3.4}$$

$$= \boldsymbol{Q}_\Delta \boldsymbol{z}_h - \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h) \boldsymbol{Q}_l \boldsymbol{z}_l \tag{3.5}$$

where

$$\begin{aligned}\boldsymbol{Q}_\Delta &= \left(\boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta \boldsymbol{F}_\Delta + \lambda_\Delta \boldsymbol{E}_\Delta\right)^{-1} \boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta \\ \boldsymbol{Q}_l &= \left(\boldsymbol{F}_l^T \boldsymbol{F}_l + \lambda_l \boldsymbol{E}_l\right)^{-1} \boldsymbol{F}_l^T .\end{aligned}$$

The prediction at a new location $\boldsymbol{s}_0$ can be thus computed using equation (2.25) with the previous expression of $\boldsymbol{Q}_\Delta$ and $\boldsymbol{Q}_l$

$$\begin{aligned}\widehat{z}_h(\boldsymbol{s}_0) =& \widehat{z}_l(\boldsymbol{s}_0) + \widehat{z}_\Delta(\boldsymbol{s}_0) = \\ &= \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h)\right) \boldsymbol{Q}_l \boldsymbol{z}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{z}_h\end{aligned} \tag{3.6}$$

while the prediction variance can be derived as follows:

$$\begin{aligned}\mathbb{V}ar[\widehat{z}_h(\boldsymbol{s}_0)] =& \mathbb{V}ar\left[\left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h)\right) \boldsymbol{Q}_l \boldsymbol{z}_l + \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{z}_h\right] \\ =& \mathbb{V}ar\left[\left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h)\right) \boldsymbol{Q}_l \boldsymbol{z}_l\right] + \mathbb{V}ar\left(\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{z}_h\right) \\ &+ 2\,\mathbb{C}ov\left[\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{z}_h, \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h)\right) \boldsymbol{Q}_l \boldsymbol{z}_l\right] \\ =& \sigma_l^2 \left(\boldsymbol{f}_l^T(\boldsymbol{s}_0) - \boldsymbol{f}_\Delta^T(\boldsymbol{s}_0) \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h)\right) \boldsymbol{Q}_l \boldsymbol{Q}_l^T \\ &\cdot \left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h) \boldsymbol{Q}_\Delta^T \boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) \\ &+ \boldsymbol{f}_\Delta(\boldsymbol{s}_0)^T \boldsymbol{Q}_\Delta \left(\boldsymbol{W}_\Delta^{-1} + \sigma_\Delta^2 \boldsymbol{W}_\Delta^{-1}\right) \boldsymbol{Q}_\Delta^T \boldsymbol{f}_\Delta(\boldsymbol{s}_0) \\ &+ 2\,\sigma_l^2 \boldsymbol{f}_\Delta(\boldsymbol{s}_0)^T \boldsymbol{Q}_\Delta \boldsymbol{F}_l(\boldsymbol{S}_h) \boldsymbol{Q}_l \boldsymbol{Q}_l^T \left(\boldsymbol{f}_l(\boldsymbol{s}_0) - \boldsymbol{F}_l^T(\boldsymbol{S}_h) \boldsymbol{Q}_\Delta^T \boldsymbol{f}_\Delta(\boldsymbol{s}_0)\right) .\end{aligned} \tag{3.7}$$

Eventually, the estimator of σ_Δ^2 is given by

$$\widehat{\sigma}_\Delta^2 = \frac{\|\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_h\|}{n_h - \operatorname{tr}\left[\boldsymbol{F}_\Delta \left(\boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta \boldsymbol{F}_\Delta + \lambda_\Delta \boldsymbol{E}_\Delta\right)^{-1} \boldsymbol{F}_\Delta^T \boldsymbol{W}_\Delta\right]} \tag{3.8}$$

or (in case of large Hi-Fi sample size) by the approximate expression:

$$\widehat{\sigma}_\Delta^2 = \frac{\|\boldsymbol{z}_h - \widehat{\boldsymbol{z}}_h\|}{n_h}. \tag{3.9}$$

In order to show how the WLS approach works, a simple example is now shown. Without loss of generality, we will initially play with profiles instead of surfaces to simplify the graphical display. We will use both the traditional LSBA and the WLSBA approaches to compute the RMSE, i.e., the mean reconstruction error. We will then compute the difference between the two performance indexes:

$$d_{\text{RMSE}} = \text{RMSE}_{\text{LSBA}} - \text{RMSE}_{\text{WLSBA}}. \tag{3.10}$$

This difference will be studied to understand if the WLSBA can effectively reduce the reconstruction error.

In our example, the difference between the Hi-Fi data $z_h(u)$ and the corresponding value predicted using the first-stage model $\widehat{z}_l(u)$ at the same location has a non-constant variance, given by:

$$z_\Delta(u) = z_h(u) - \widehat{z}_l(u) = u + u^2 + \varepsilon_\Delta(u), \quad \varepsilon_\Delta(u) \sim \mathcal{N}\left(0, \sigma^2(u)\right). \tag{3.11}$$

Assume that the discrepancies is available at 20 equally spaced locations, as shown in Figure 3.1, where the standard deviation of prediction at any location is also drawn. It is possible to observe that in the middle of the profile there are points with larger values of the standard deviation. This increase of the prediction uncertainty was assumed to model the possible lack of Lo-Fi points in that location.

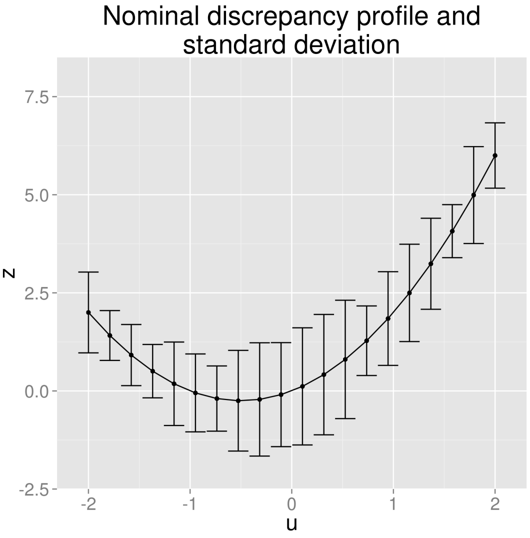
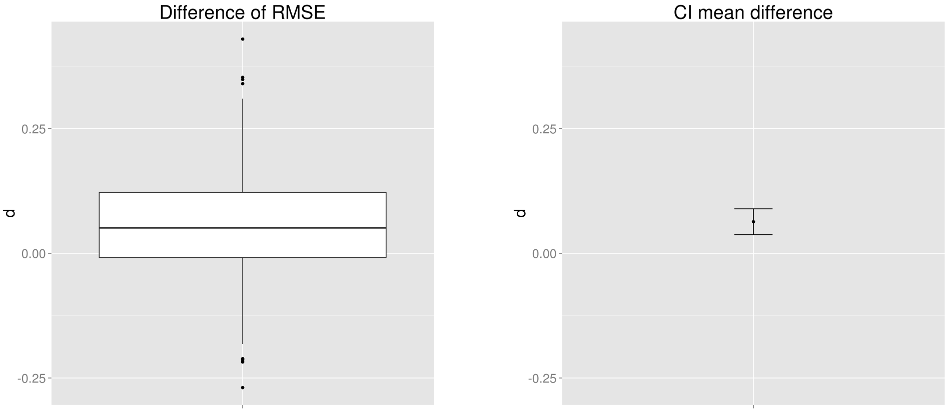


Figure 3.1: Nominal profile of the discrepancy and corresponding standard deviation assumed at each point

Starting from this model of the discrepancy, 100 profiles were simulated. Each simulation consists of 20 data observed along the discrepancy profile. For each simulation, both the LSBA and the WLSBA were used to compute RMSE and the corresponding difference was computed via equation 3.10. The boxplots of the 100 instances of d_{RMSE} is shown in Figure 3.2a, together with a confidence interval on the mean value of this difference.

Since the mean of the difference is significantly greater than zero, it is possible to conclude that the LSBA method determines a reconstruction error that is significantly greater than the WLSBA one (3.2b).



(a) Boxplot of d_{RMSE}

(b) 95% confidence interval on the mean of d_{RMSE}

Figure 3.2: Boxplot and confidence interval on d_{RMSE} (100 replicates)

3.2 An example of Surface Reconstruction via WLSBA

In this section, it is shown how the uncertainty propagates in the data fusion model. Once again, the Matlab `peaks` function is used as baseline model. In particular, equation (1.24) is used to generate the Lo-Fi data

$$z_l(\boldsymbol{s}) = f(\boldsymbol{s}) + \frac{u^2}{10} + \frac{v^2}{10} + \varepsilon_l, \quad \varepsilon_l = \mathcal{N}(0, 0.4^2)$$

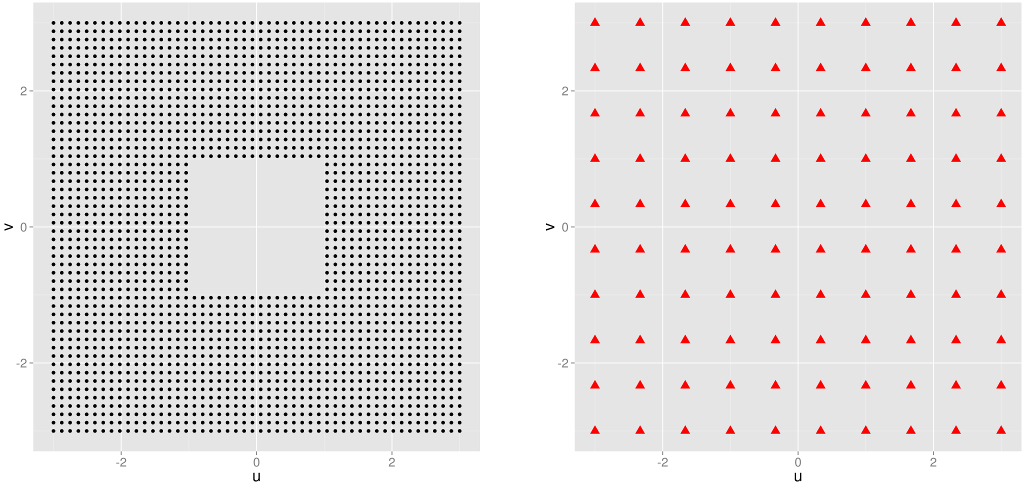
while the Hi-Fi data are generated according to equation (1.23), namely:

$$z_h(\boldsymbol{s}) = f(\boldsymbol{s}) + \varepsilon_h, \quad \varepsilon_\Delta = \mathcal{N}(0, 0.2^2)$$

where

$$f(\boldsymbol{s}) = 3(1-u)^2\, e^{-u^2-(v+1)^2} - 10\left(\frac{u}{5} - u^3 - v^5\right) e^{-u^2-v^2} - \frac{1}{3}e^{-(u+1)^2-v^2}.$$

The domain of the simulated data correspond to $[-3,3] \times [-3,3]$, but it is assumed that Lo-Fi points are acquired with a uniform grid with some missing points in the middle of the domain (Figure 3.3 - left side). This missing data will clearly reduce the reliability (i.e., increase the uncertainty) of predictions made at the first-stage using the Lo-Fi data only. A data fusion model able to capture this variable uncertainty is what we want in this case. The (u, v) locations of both the Lo-Fi and Hi-Fi simulated points are shown in Figure 3.3.



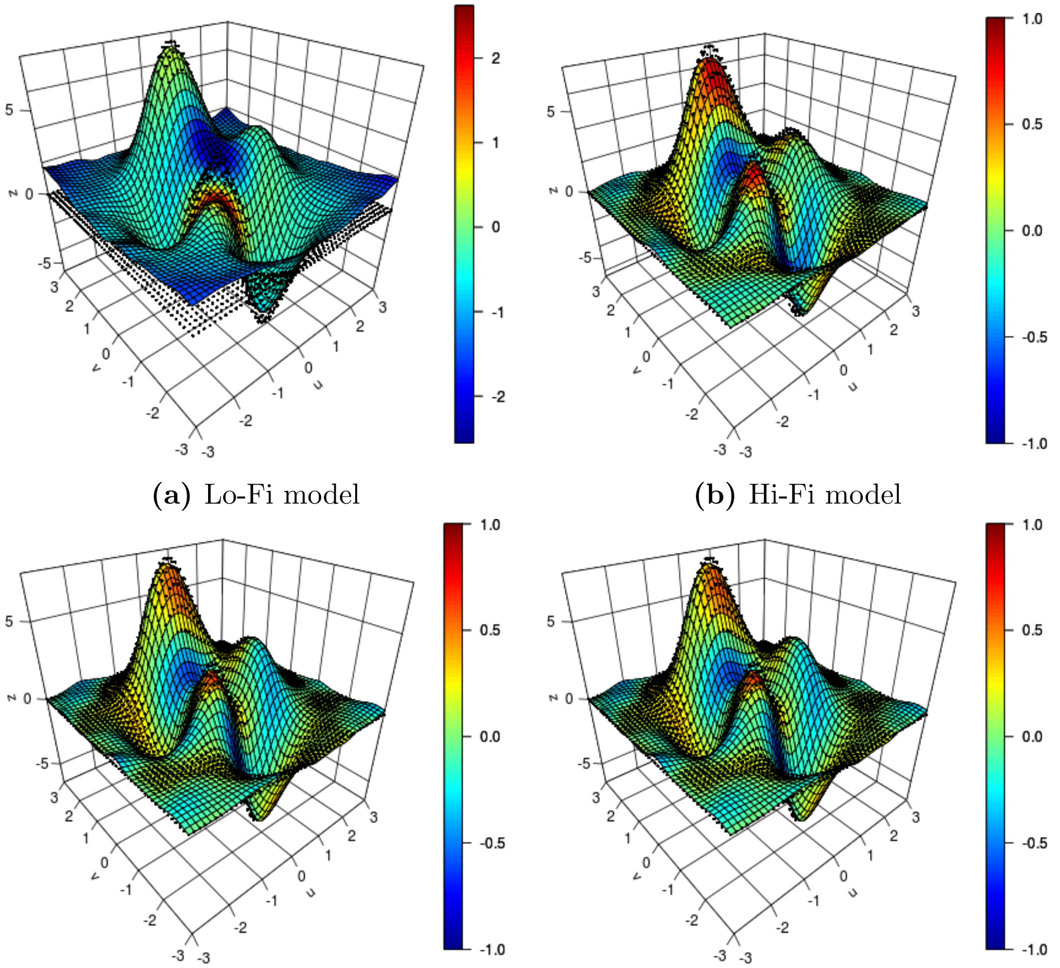
(a) Lo-Fi data **(b)** Hi-Fi data

Figure 3.3: Lo-Fi and Hi-Fi data points

Once again, the competitor models are the (LSBA-based) approaches using only the Lo-Fi data, only the Hi-Fi data or performing data fusion of the two data sets. For the sake of simplicity, these three models will be referred to as "Lo-Fi", "Hi-Fi" and "LSBA" approach. A further model performing data fusion via WLSBA will be used and referred to as "WLSBA" from now on.

In order to check how the variability of the prediction changes along the domain, the prediction variance for the Lo-Fi, Hi-Fi, LSBA and WLSBA models is computed on a regular grid of 100 × 100 points.

To visually check the prediction performance, the error maps of the models are firstly shown in Figure 3.4. The error map of the Lo-Fi model reflects the simulated bias, while the larger errors of the Hi-Fi model are mainly located on the surface peaks and valleys (because of the low density of the Hi-Fi points, peaks and valleys are "smoothed"). On the contrary, both the fusion models (LSBA and WLSBA) outperform the others, and the error map of these models look very similar. This means that the punctual predictions of the two data fusion models are very similar.



(a) Lo-Fi model

(b) Hi-Fi model

(c) LSBA model

(d) WLSBA model

Figure 3.4: Reconstructed surfaces and error maps of the models

The prediction variances of all the models are shown in Figure 3.5. The variance of the Lo-Fi model is larger at the center of the domain, because points are missing in that location. The predicted variance of the Hi-Fi model has a chessboard effect due the sampling plan (a grid). As a matter of fact, the prediction variance reduces moving closer to the observed Hi-Fi points and increases when moving away from them. The prediction variance of the LSBA model has a behavior similar to the one computed using the Hi-Fi data only. This is due to the fact that the LSBA model assumes constant variance and hence it is not able to model the variable uncertainty that the test case is representing. Eventually the WLSBA approach is able to represents the actual state of the prediction variance, which is ruled by the Hi-Fi data at the center (where no Lo-Fi data are available) and mix Lo-Fi and Hi-Fi uncertainties in the other locations. With this example it is shown how

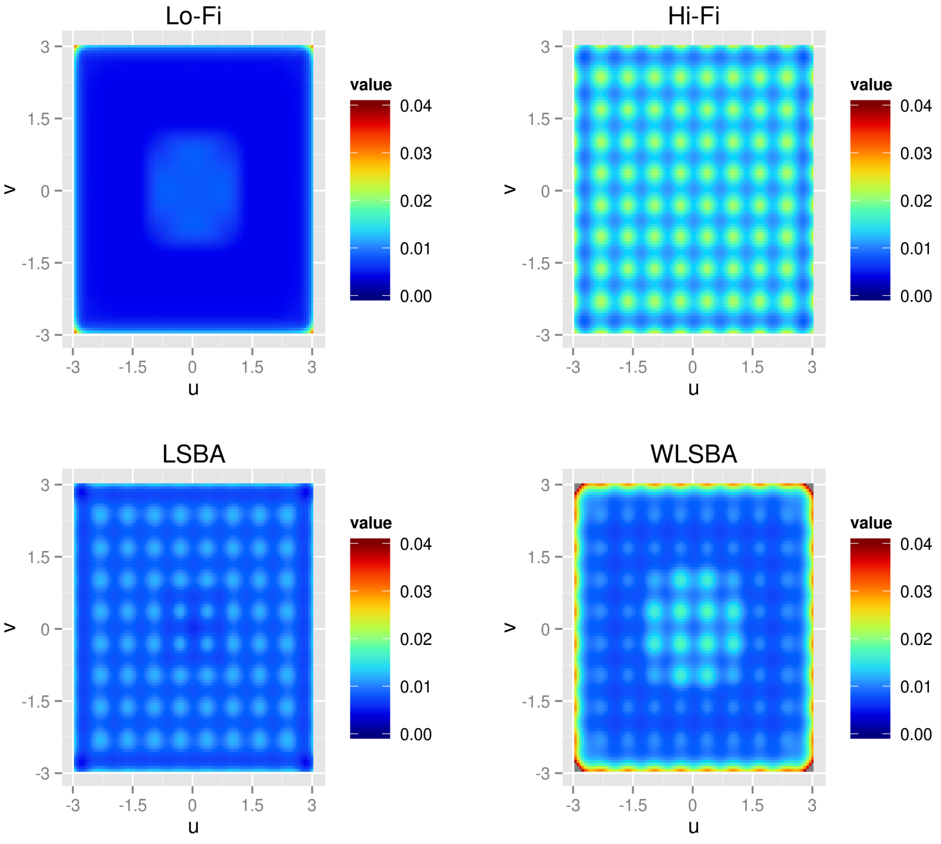


Figure 3.5: Predicted variance of the models

the WLSBA approaches is able to capture how the uncertainty propagates from the Lo-Fi (first stage) to the linkage model.

Chapter 4

Statistical Process Control of Regularly Structured Surfaces

In the previous chapters we focused on free-form surface reconstruction. In modern manufacturing processes, there is a growing interest to textured surfaces, where the regular pattern is required to enhance the functional performance of the surface.

In this chapter a Statistical Process Control (SPC, (Montgomery 2013)) procedure is presented to check the stability of a laser texturing process on the workpiece area.

4.1 Introduction

The aim of the SPC is the quick detection of an out-of-control state in order to avoid the production of nonconforming items. A SPC tool commonly used to check if a process is in control is the control chart (Montgomery 2013). The aims of the control chart are:

- reduction of the process variability;
- process control over time;
- product or process parameters estimation.

An example of a control chart is shown in Figure 4.1, where a statistic of interest is measured over time. There is a central line (CL) that is the estimated mean value, and two horizontal lines that are the upper control limit (UCL) and the lower control limit (LCL). These two limits are chosen such that the $(1-\alpha)\%$ of the items are between the limits if the process is in statistical control, where α is the type I error of the hypothesis test. This value is usually equal to 0.0023 for univariate control charts, when control limits are placed at 3 times the standard deviation from the mean (CL). If the value of the plotted statistic falls outside the limits, such as the red points in Figure 4.1, an alarm is issued. These points are called out of controls.

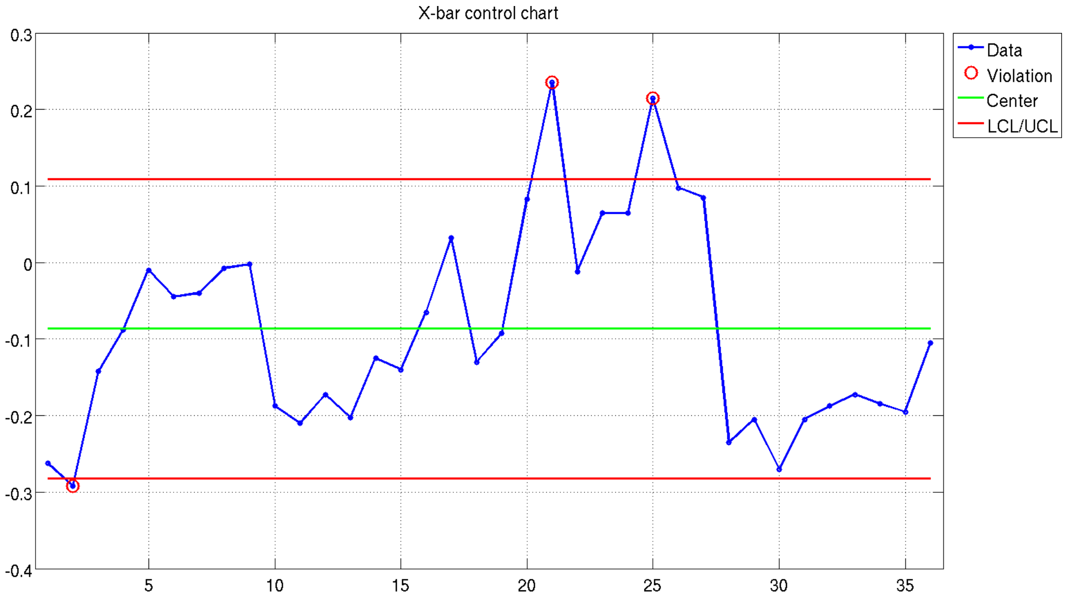


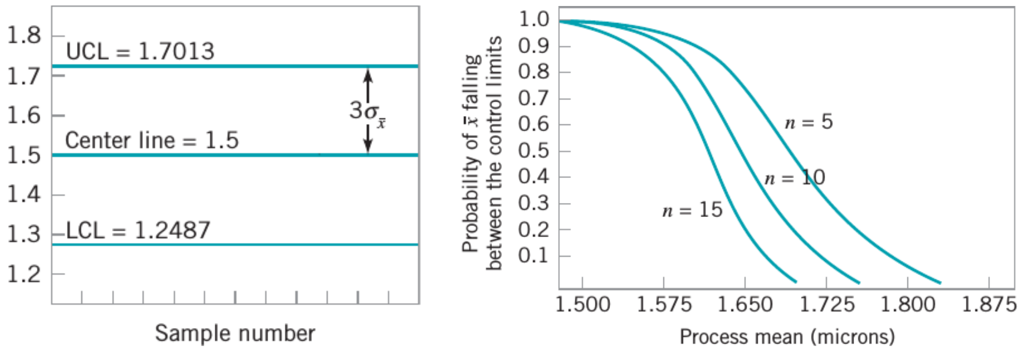
Figure 4.1: Control chart example

In order to apply control charts, two different phases called *Phase I* and *Phase II* have to be performed. In *Phase I* a statistical analysis is carried out to check if the process is stable, i.e., the presence of trend, periodical pattern etc. is checked. If the process is stable, the parameters of interest (mean, standard deviation) are estimated and control limits are computed. If one point falls outside the control limits and an assignable cause behind the out of control is found, the point is removed from the set and *Phase I* is repeated. Otherwise, *Phase I* is concluded. In *Phase II*, control charts are applied while the process is going on and any new sample statistic is plotted on the control chart to see if it is out of control. Control limits found in *Phase I* are thus simply used in *Phase II*.

In order to judge control chart performance the Average Run Length (ARL) is usually considered. ARL represents the mean value of the run lengths (RL), where the RL is the number of samples checked before detecting an alarm. ARL can be also computed also as

$$\text{ARL} = \frac{1}{1-\beta}$$

where $1-\beta$ is the power of the hypothesis test behind the control chart; i.e. the probability of detecting an out of control when it is effectively affecting the process. A common plot to evaluate the performance of a control charts is the Operating Characteristic (OC) curve; which represents the probability (β) of not detecting an out of control as a function of the size of the process shift δ. An example of the design of a $\overline{X}$ control chart for the mean and the corresponding OC curve is drawn in Figure 4.2.



(a) $\overline{X}$ control chart

(b) OOC curve

Figure 4.2: Control chart and OC curve example (Montgomery 2013)

Initially, in the 1920's, control charts were designed for small sampling frequency ad univariate characteristics. Later, multivariate control charts were introduced in order to check multiple quality characteristics simultaneously (see (Alt 1985) and (Alt and Smith 1988)). With a multivariate control chart it is also possible to detected out of control regarding the correlation between two or more quality characteristics.

A recent field of interest is profile monitoring, where the aim is to detect if a profile or pattern is stable over time. A brief overview of approaches for profile monitoring can be found in (Noorossana, Saghei, and Amiri 2012) and (Woodall et al. 2004). There are two macro-areas of profile monitoring analysis: parametric and nonparametric models. In parametric approaches a linear or non linear model describes the signature of the profile, and a control chart is based on parameters describing this signature. In nonparametric models a profile model is not assumed, and the out of control is based on the discrepancy between the observed data and the expected profile pattern.

The first profile monitoring procedures were based on parametric models to describe simple patterns (e.g. linear trend (Kang and Albin 2000)). At this point in time, both parametric and non parametric methods can be combined to represent complex patterns. As an example, Abdel-Salam, Birch, and Jensen (2010) proposed a semiparametric model based on both parametric and nonparametric methods to better describe all the feature of the profiles.

Moving form profile to surfaces monitoring, Colosimo, Mammarella, and Petrò (2010) proposed a monitoring method for cylindrical surfaces as a generalization of profile monitoring. The authors showed how a liner regression model with spatially correlated errors can model lathe-turned cylindrical items. Recently, Colosimo et al. (2014) proposed a shape monitoring approach based on GP to simplify the need of model selection. In this section the approach proposed in Colosimo et al. (2014) is followed to describe some feature parameters extracted from a regularly structured surface.

4.2 Feature Parameters Extraction

Usually, optical 3D surface measurement devices measure the height of the surface, z, in a specific location (u, v). This location refers to a matrix of pixels, so the measured locations form a regular grid in the u-v plane. ISO-25178 (ISO 2013) norm defines a procedure for features analysis. The step of the procedure are (Leach 2013):

- identify the areal feature within the areal surface topography;
- extract the feature and evaluate its relevant geometric attributes;
- take the necessary actions based on the characterization requirements.

An example of segmentation in Leach (2013) is the identification of the groove in the surface reported in Figure 4.3.

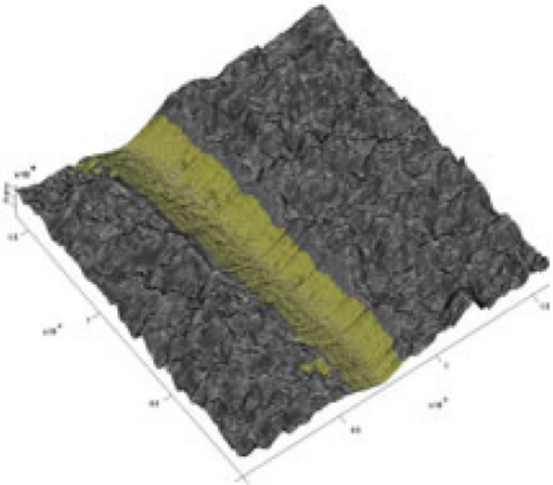


Figure 4.3: Example of surface segmentation

Firstly, the surface is subdivided into subregions. In this process hills and dales are found segmenting the surface in polygonal regions (see Figure 4.4).

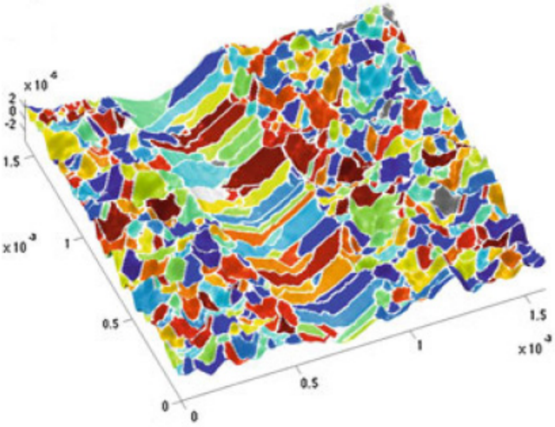


Figure 4.4: Surface segmentation

After the segmentation, sometimes, there is a problem called over-segmentation. As a matter of fact, the more the surface is irregular, the more the segmentation algorithm gives a large number of regions. An operation called *pruning* is thus needed to reduce the number of regions found. ISO-25178 norm defines criteria based on:

- wolf pruning $x\%$: pruning if the height of the feature is less than the $x\%$ of the parameter S_z (i.e. the range of the z values);

- area $x\%$: pruning if the feature area is less than the $x\%$ of a defined area;
- volume $x\%$: pruning if the feature volume is less than the $x\%$ of a defined volume.

By applying the area and wolf pruning criteria, it is possible to reduce the number of segmented regions (see Figure 4.5).

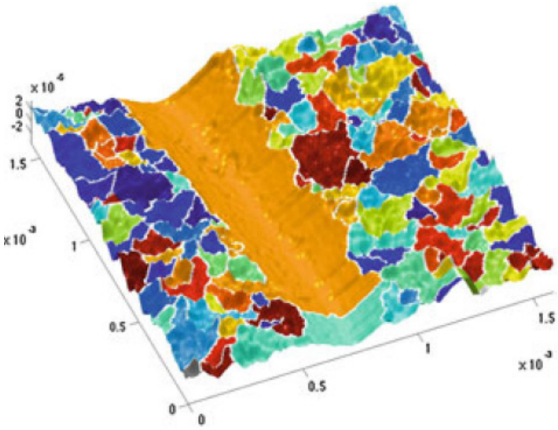


Figure 4.5: Surface segmentation with wolf pruning

After pruning, regions containing the feature of interest can be found. Once the features have been extracted, it is possible to compute some relevant descriptors such as the depth, the area and the volume of each extracted feature.

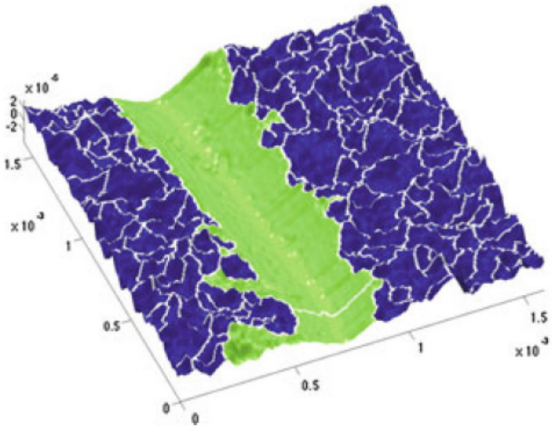


Figure 4.6: Extracted feature

4.2.1 Watershed Segmentation

A method to initially divide the surface into subregions is the watershed segmentation. Watershed segmentation is an image analysis technique that divides the pixels of an image in regions with the same properties (Leach 2013). Watershed transformation is described in Meyer (1994) where a surface with dales is flooded. Each dale forms a basin, when the water reaches the maximum basin height a dam wall is build, defining one or more regions. The phases of the flooding process are shown in Figure 4.7.

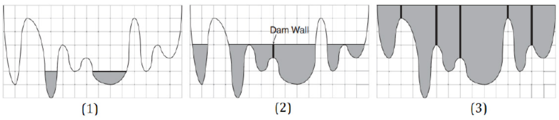


Figure 4.7: Phases of the watershed algorithm (Salman 2006)

The advantages of the watershed algorithm are:

- boundaries of the regions are closed and connected regions;
- boundaries of the regions correspond to contours which appear in the image as contours of objects;
- the union of all the regions is the whole surface or image.

4.3 Case Study: Dimpled Surface Feature Extraction and Monitoring

Surface Laser Texturing (SLT) is a technique for enhancing tribological properties of mechanical components (Etsion 2005). Surfaces of modern magnetic storage devices are commonly textured and SLT is also considered as a means for improving functional properties such as adhesion and stiction.

In this section a dimpled surface studied and developed within the project St.i.m.a. (STrutture Ibride per la Meccanica e per l'Aerospazio, founded by Regione Lombardia, Italy) is taken as reference. The aim of the research project was analyzing and controlling the effectiveness of the laser-based dimpling project.

In this research thesis, we will focus on approaches for monitoring the stability of the SLT via GP based control chart. The textured surface was made of titanium Ti64Al4V with the aim of maximizing the adhesion between different materials (Maressa et al. 2014).

In order to design the GP-based control chart, the features of interest (i.e. the dimples) were firstly identified. To this aim, the following steps were carried out:

1. data acquisition and pre-processing;
2. watershed segmentation;
3. dimple identification,
4. computation of the dimple geometry descriptors.

Since the surfaced manufactured was large, an area of $4 \cdot 10^3 \times 4 \cdot 10^3$ μm^2 was firstly processed. In this case, the acquisition time with the Alicona InfiniteFocus microscope was too long, so a smaller part of the surface was acquired at different locations. The characteristics of a generic item (portion of the surface) analyzed are:

- an area of $1.5 \times 1.5\, mm^2$;
- regular grid of $n = 10 \times 10$ dimples;
- a nominal center to center distance of $150\, \mu m$;
- a nominal depth of $4.5\, \mu m$.

The main aim of this chapter is to propose a SPC procedure to detect outlying features on the manufactured surface.

To this aim, 15 surface areas were randomly selected on the wide area of $4 \cdot 10^3 \times 4 \cdot 10^3$ μm^2 textured surface. Selected areas are shown in Figure 4.8 (left side). Each area was then constituted by a set of 10×10 dimples (Figure 4.8, right side). The aim of the SPC procedure is to detect if one or more of these examined areas can be deemed out of control by modeling the 10×10 dimples' geometry observed on it.

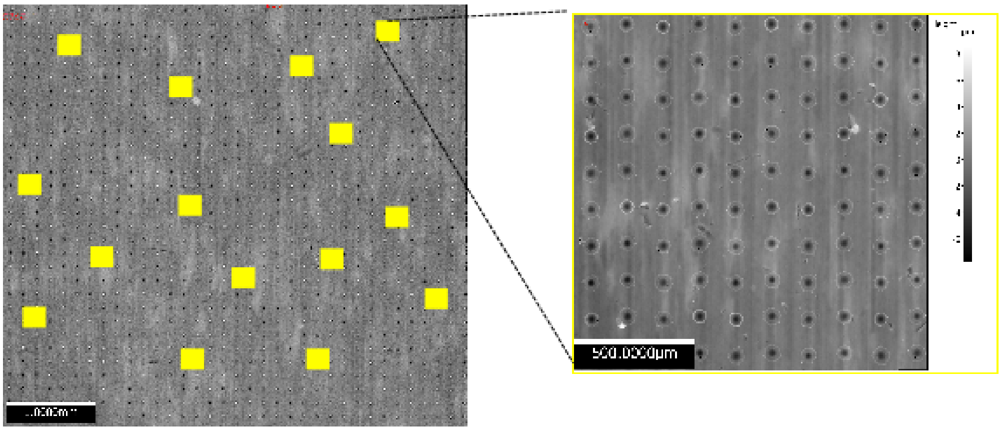
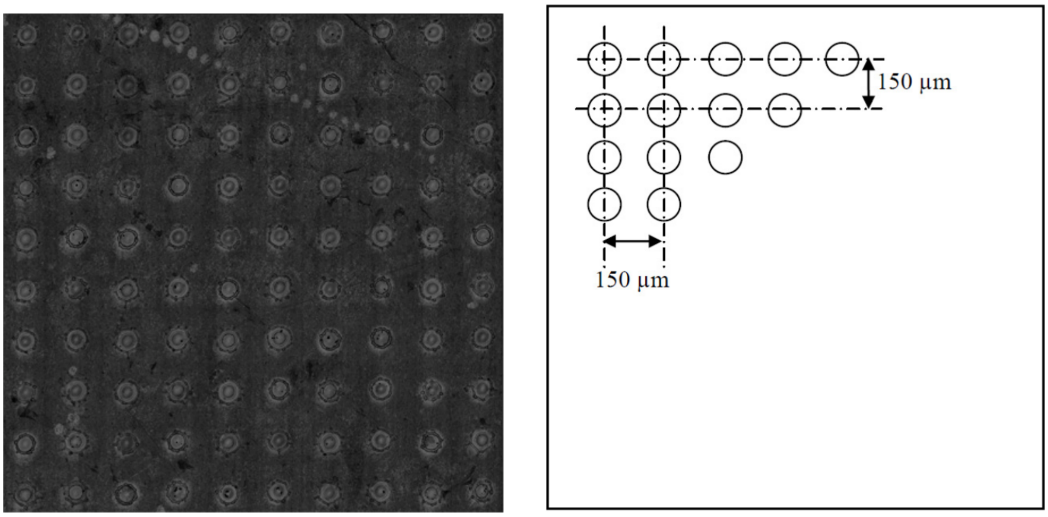


Figure 4.8: Textured surface and one of the subsample of interest

4.3.1 Data Acquisition and Pre-processing

An example of the acquired surface with the Alicona InfiniteFocus microscope is shown in Figure 4.9a, while in Figure 4.9b the nominal texture pattern is shown.



(a) Example of a dimpled surface acquired

(b) Nominal pattern of the texture

Figure 4.9: A dimple surface texture acquired and the corresponding nominal shape

Pre-processing consists of removing the form using a second degree polynomial fitting and using a robust least squares plane as reference. Then, the second pre-process step consists of reconstructing the surface using the MBA algorithm described in section 2.1 to remove high frequency irregularities. Eventually decimation of the point cloud and prediction on missing point locations (i.e. locations where the measurement system was not able to acquire the height of the surface) was performed. The reconstruction was needed to perform the segmentation step.

4.3.2 Watershed Segmentation

Surface segmentation was performed using the watershed segmentation function implemented in Matlab. Regions resulting after applying this algorithm are reported in Figure 4.9, where is clear that this algorithm produces a large number of regions. Clearly, an over segmentation is achieved and need to be solved.

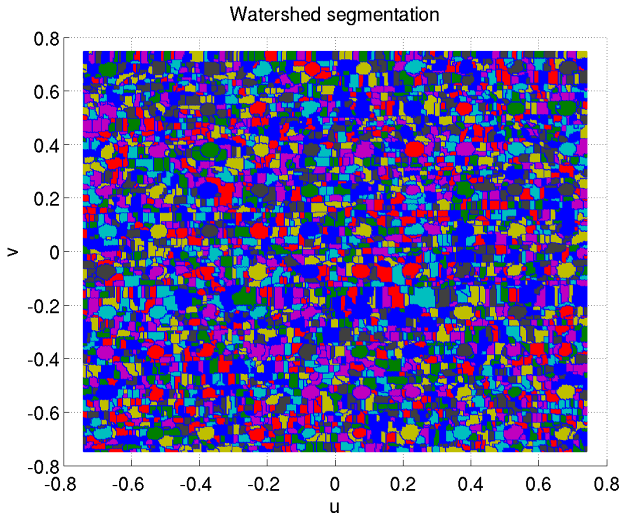


Figure 4.10: Watershed segmentation of the surface

4.3.3 Dimple Identification

In order to reduce the number of regions found, a threshold criterion was firstly applied. All the regions where the minimum z depth was lower than $-10^{-4}\,\mu m$ were neglected. A plot of the resulting dimples' regions is shown in Figure 4.11. As clear from the figure, over segmentation is not completely solved yet.

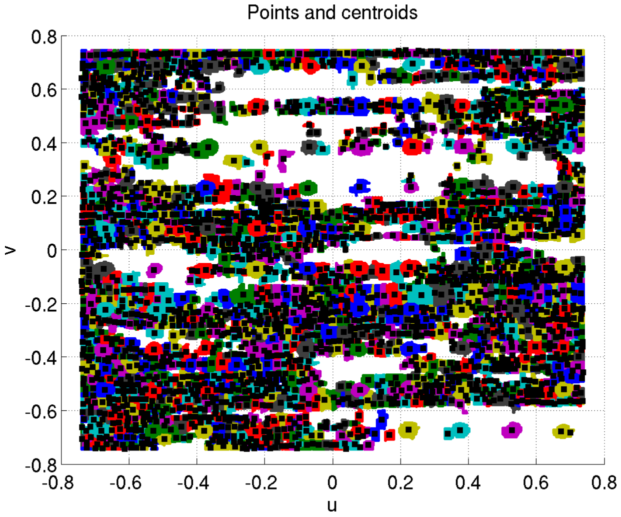


Figure 4.11: Centroids found after the first step

After this rough segmentation, the following steps were applied to eventually find the dimples of the surface:

1. compute the centroids of all the identified regions (see Figure 4.9)

$$\boldsymbol{C}_r = (u_{c,r}, v_{c,r})^T, \quad r = 1, 2, \ldots, R;$$

2. compute the nominal position of the dimples centers, considering the regular pattern used during the SLT process (see Figure 4.9b). Let $\boldsymbol{C}_j^{nom}, \quad j = 1, 2, \ldots, n$ represent these centers;

3. find the best alignment mapping the centroids computed at step 1 with the centers' positions computed at step 2 using the iterative closest points algorithm (Pandzo et al. 2001);

4. for each nominal center $\boldsymbol{C}_j^{nom}$ choose the dimple region $\boldsymbol{C}_r$ which is closer under an euclidean metric.

Dimple resulting by applying this 4-steps procedure are shown in Figure 4.12. As clear, over segmentation was eventually solved.

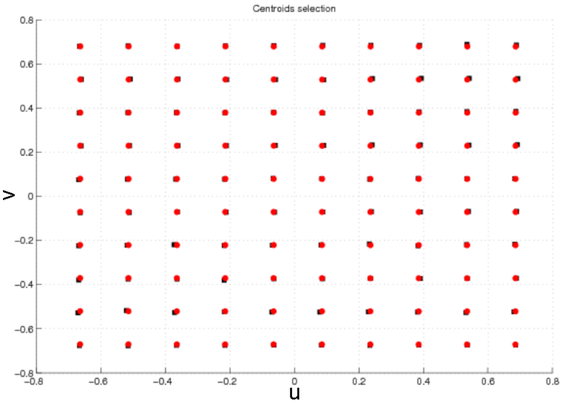


Figure 4.12: Selected dimples centroids (black) and their nominal references (red)

4.3.4 Selection of Dimple Descriptors

The reconstructed surface and the dimples found with the described procedure are shown in Figure 4.13.

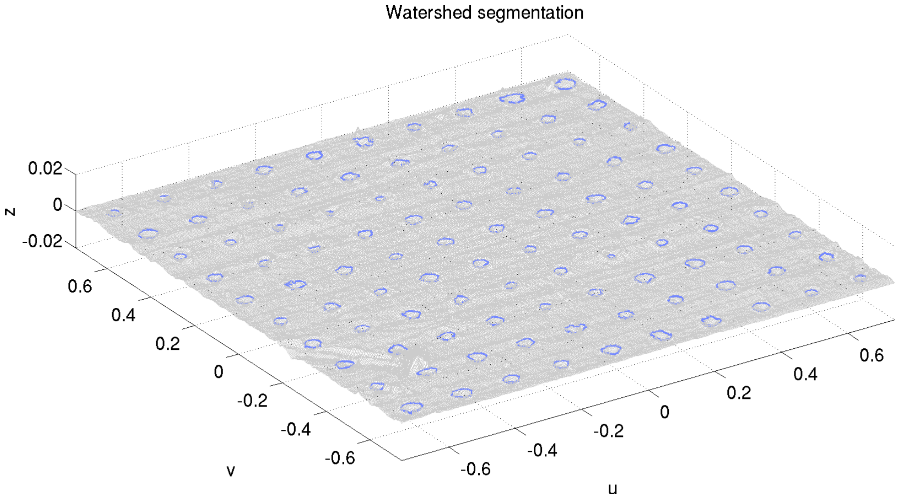


Figure 4.13: Example of reconstructed surface and segmented dimples

According to the functional requirements on dimples (i.e. to improve adhesion), the relevant descriptors selected are:

- the dimple depth (or height)

$$h_j = h\left(\boldsymbol{C}_j\right) = \min_{p \in P_j} z_p, \quad j = 1, 2, \ldots, n$$

- the dimple area

$$A_j = A\left(\boldsymbol{C}_j\right), \quad j = 1, 2, \ldots, n.$$

4.4 Gaussian Process Control Chart for Regularly Structured Surfaces

GP model described in section 1.3 can be used to design a multivariate Hotelling's T^2 control chart (Montgomery 2013) as described in (Colosimo et al. 2014). In this thesis, this approach is used to monitor a surface feature (e.g. depth or area of the features). Let $\boldsymbol{Z}_i$ collects the descriptors of interest observed on the set of locations $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n\}$ in the *i-th* sample $(i = 1, \ldots, m)$. As an example, $\boldsymbol{Z}_i$ can collect the areas of the n dimples observed on the *i-th* surface. It is assumed that

$$\boldsymbol{Z}_i \sim \mathcal{N}_n(\boldsymbol{F}_i \boldsymbol{\beta}, \Sigma_i), \quad i = 1, 2, \ldots, m \tag{4.1}$$

where $\boldsymbol{F}_i$ is the model matrix, $\boldsymbol{\beta}$ is the vector of regressors and $\boldsymbol{\Sigma}_i$ is the GP covariance matrix, i.e. $\boldsymbol{\Sigma}_i = \sigma_\eta^2 \boldsymbol{R}_{\eta,i}(\boldsymbol{\vartheta}) + \sigma_\varepsilon^2 \boldsymbol{I}_n$. Note that the subscript i is used to highlight that possible difference between locations on different items may exist because the location of the feature depends on the manufacturing process.

4.4.1 GP Control Chart Design

The goal of *Phase I* is the design of the control chart. Assuming m items (i.e. structured surface) are available in *Phase I*. Following Colosimo et al. (2014), the GP model parameters $(\widehat{\boldsymbol{\Psi}}_i = (\widehat{\boldsymbol{\vartheta}}_i, \widehat{\sigma}^2_{\eta,i}, \widehat{\sigma}^2_{\varepsilon,i}))$ are computed for each *i-th* item observed in *Phase I*. Eventually, the in-control process parameter is assumed equal to the mean of these estimated vectors:

$$\widehat{\boldsymbol{\Psi}} = \frac{1}{m} \sum_{i=1}^{m} \widehat{\boldsymbol{\Psi}}_i. \tag{4.2}$$

Let $\widehat{\boldsymbol{z}}_i$ represents predicted values of the surface descriptors on the *i-th* surface (e.g., the predicted areas of dimples of *i-th* surface). Considering all the m *Phase I* samples, the mean value of the surface descriptors in the region of interest is given by:

$$\widehat{\boldsymbol{\mu}}_{\widehat{\boldsymbol{z}}} = \frac{1}{m}\sum_{i=1}^{m} \widehat{\boldsymbol{z}}_i. \tag{4.3}$$

This vector is used as reference to detect out of control states. In fact an item where the observed descriptors will be very different from the mean value observed in *Phase I* ($\widehat{\boldsymbol{\mu}}_{\widehat{\boldsymbol{z}}}$) will be deemed out of control.

Similarly,

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i} =& \widehat{\sigma}_{\eta}^2\, \widehat{\boldsymbol{R}}_{\eta,i} + \widehat{\sigma}_{\varepsilon}^2\, \boldsymbol{I}_n - \widehat{\sigma}_{\eta}^2\, \widehat{\boldsymbol{R}}_{\eta,i}^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i} \widehat{\boldsymbol{R}}_{\eta,i} \\ &+ \left(\boldsymbol{F}_i^T - \boldsymbol{F}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i}^{-1} \boldsymbol{R}_{\eta,i}\right)^T \left(\boldsymbol{F}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i} \boldsymbol{F}_i\right)^{-1} \left(\boldsymbol{F}_i^T - \boldsymbol{F}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i}^{-1} \boldsymbol{R}_{\eta,i}\right).\end{aligned}$$

can be taken as reference to estimate the variance-covariance matrix. This matrix represents the connection between the descriptors of interest as a function of the spatial location.

In order to detect an out of control state, assume a new surface is examined and the descriptors of interest are predicted at each location considering the GP model estimated on *Phase I*. The T^2 statistic of the predicted descriptors can be computed as

$$T_i^2 = \left(\widehat{\boldsymbol{z}}_i - \widehat{\boldsymbol{\mu}}_{\widehat{\boldsymbol{z}}}\right)^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i}^{-1} \left(\widehat{\boldsymbol{z}}_i - \widehat{\boldsymbol{\mu}}_{\widehat{\boldsymbol{z}}}\right) \tag{4.4}$$

where $\widehat{\boldsymbol{\mu}}_{\widehat{\boldsymbol{z}}}$ is the mean of the predicted values (Equation 4.3).
Eventually, the upper control limit is given by

$$\mathrm{CL} = \chi^2_{n,1-\alpha} \tag{4.5}$$

where α is the first type error of a false alarm probability.

In addition to the proposal by Colosimo et al. (2014), the T^2 control chart based on the fitted value in Equation (4.4) is also augmented by a T^2 control chart based on the model residuals

$$T_i^2 = \widehat{\boldsymbol{\varepsilon}}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{\varepsilon}}_i}^{+} \widehat{\boldsymbol{\varepsilon}}_i \tag{4.6}$$

where $\widehat{\boldsymbol{\varepsilon}}$ is the vector of residuals, given by

$$\widehat{\boldsymbol{\varepsilon}}_i = \boldsymbol{z}_i - \widehat{\boldsymbol{z}}_i = \left(\boldsymbol{I}_n - \sigma_{\eta}^2\, \boldsymbol{R}_{\eta,i}\, \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i}^{-1}\right) \left(\boldsymbol{I}_n - \boldsymbol{P}_i\right) \boldsymbol{z}_i \tag{4.7}$$

with

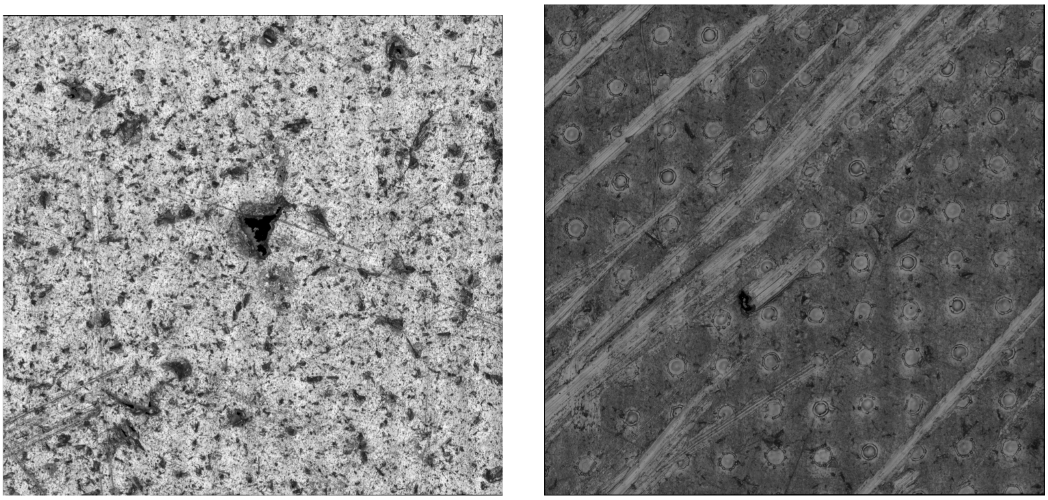
$$\boldsymbol{P}_i = \boldsymbol{F}_i \left(\boldsymbol{F}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i} \boldsymbol{F}_i\right)^{-1} \boldsymbol{F}_i^T \widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{z}}_i}^{-1}$$

and $\widehat{\boldsymbol{\Sigma}}^{+}_{\widehat{\boldsymbol{\varepsilon}}_i}$ is the Moore-Penrose pseudoinverse of the matrix $\widehat{\boldsymbol{\Sigma}}_{\widehat{\boldsymbol{\varepsilon}}_i}$. The pseudoinverse must be computed because the rank of the matrix is $n-p$, where p is the number of the columns of the matrix $\boldsymbol{F}_i$. Since the rank of the matrix $\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\varepsilon}}}$ is $n-p$ the control limit is given by

$$CL = \chi^2_{n-p,1-\alpha}.$$

4.5 Performance Study of the SPC Procedure

In this section, performance of the SPC procedure proposed in section 4.4 are evaluated on a real dimpled surface and used to design the control chart. To this aim a first set of $m = 15$ subsurfaces was measured. Then, two out-of-control surfaces were considered to test the ability of the proposed method. The first out-of-control sample (Figure 4.14a) represents a blank titanium surface, where it is assumed that the surface is not textured because of a problem occurred to the laser process. The second out-of-control surface is shown in Figure 4.14b and represents a surfaces with scratches.

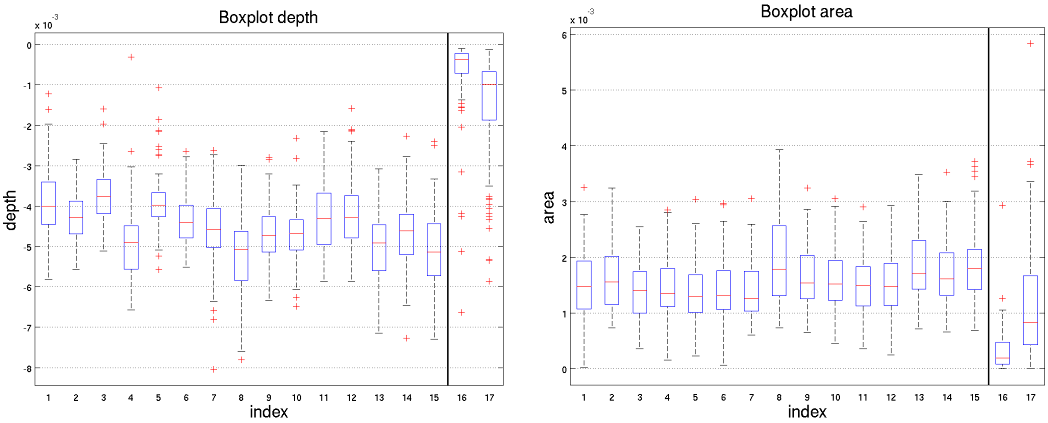


(a) Blank item

(b) Scratched surface

Figure 4.14: Out of control surfaces acquired

Figure 4.15 shown the boxplots of the 10 × 10 dimples descriptors on all the acquired items. The black vertical line divides the in-control from the out-of-control samples. It is possible to observe that out-of-control items present lower depths and lower (and more spread) areas compared with the in-control ones.



(a) Depth **(b)** Area

Figure 4.15: Boxplots of the computed features parameters of the acquired items

4.5.1 Univariate Control Chart

In order to represent the current practice, a simple alternative SPC procedure was also tested. It consists of a simple univariate control chart monitoring the mean and the variance of each descriptors on the specific area. In other words, the stability of the mean depth (area) and the related variance observed on a set of n dimples will be checked. Usually, $\overline{X} - S$ or $\overline{X} - S^2$ control charts are used in this case (Montgomery 2013). Unfortunately, assumptions underneath these control charts are not satisfied (mainly because of the correlation between the descriptors within each regions). The approach proposed by (Alwan 1999) was thus used to overcome this problem.

The mean of the dimple descriptors (depth or area) is given by

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} x_{i,j}, \quad \forall i = 1, 2, \ldots, m$$

where $x_{i,j}$ is the descriptor (either the depth or the area) of the *j-th* dimple of the *i-th* item. Thanks to the central limit theorem, it can be assumed that:

$$\mu_i \sim \mathcal{N}\left(\mu_I, \sigma_I^2\right)$$

the control limits are then computed as

$$\begin{aligned}UCL_I =& \mu_I + z_{1-\frac{\alpha}{2}} \cdot \sigma_I \\ CL_I =& \mu_I \\ LCL_I =& \mu_I - z_{1-\frac{\alpha}{2}} \cdot \sigma_I\end{aligned}$$

where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the standard normal probability distribution. In order to compute the control limits of the I chart on the variance, a transformation of the sample variance is firstly computed

$$s_i^{2*} = \frac{\left(s_i^2\right)^\lambda - 1}{\lambda}, \quad \forall i = 1, 2, \ldots, m$$

where $\lambda = 0.5$ and $s_i^2 = \frac{1}{m-1} \sum_{k=1}^{m} \left(x_{i,k} - \mu_i\right)$. Using this transformation, usually the following assumption is satisfied (Alwan 1999)

$$s_i^{2*} \sim \mathcal{N}\left(\mu_{S^{2*}}, \sigma^2_{S^{2*}}\right).$$

The control limits of the variance control chart are

$$\begin{aligned}UCL_{S^{2*}} =& \mu_{S^{2*}} + z_{1-\frac{\alpha}{2}} \cdot \sigma_{S^{2*}} \\ CL_{S^{2*}} =& \mu_{S^{2*}} \\ LCL_{S^{2*}} =& \mu_{S^{2*}} - z_{1-\frac{\alpha}{2}} \cdot \sigma_{S^{2*}}.\end{aligned}$$

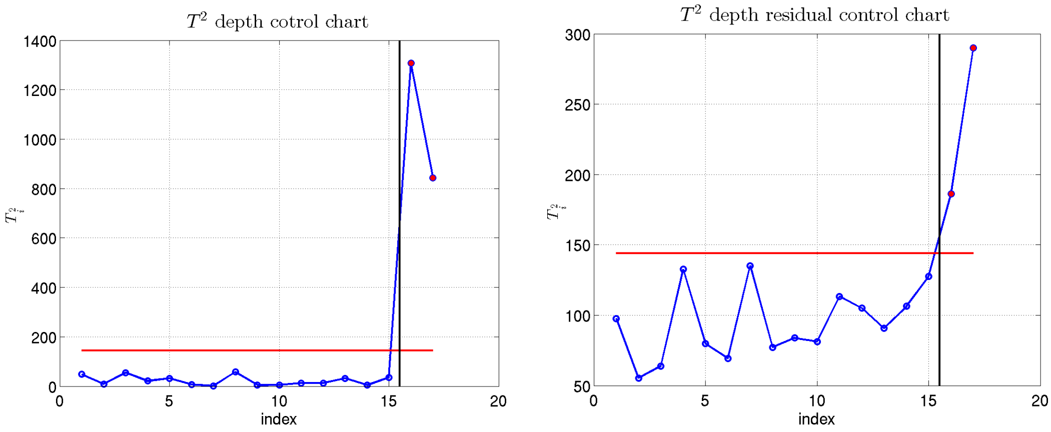
The α value of each chart is set to 0.0027, as for the T^2 charts.

Since parameters μ_I, σ_I^2, $\mu_{S^{2*}}$ and $\sigma_{S^{2*}}$ are usually not known, a first set of m samples is taken in *Phase I* and the corresponding parameters are estimated as

$$\begin{aligned}\mu_I =& \frac{1}{m} \sum_{i=1}^{m} \mu_i \\ \sigma_I^2 =& \frac{1}{m-1} \sum_{i=1}^{m} \left(\mu_i - \mu_I\right)^2 \\ \mu_{S^{2*}} =& \frac{1}{m} \sum_{i=1}^{m} s_i^{2*} \\ \sigma^2_{S^{2*}} =& \frac{1}{m-1} \sum_{i=1}^{m} \left(s_i^{2*} - \mu_{S^{2*}}\right)^2.\end{aligned}$$

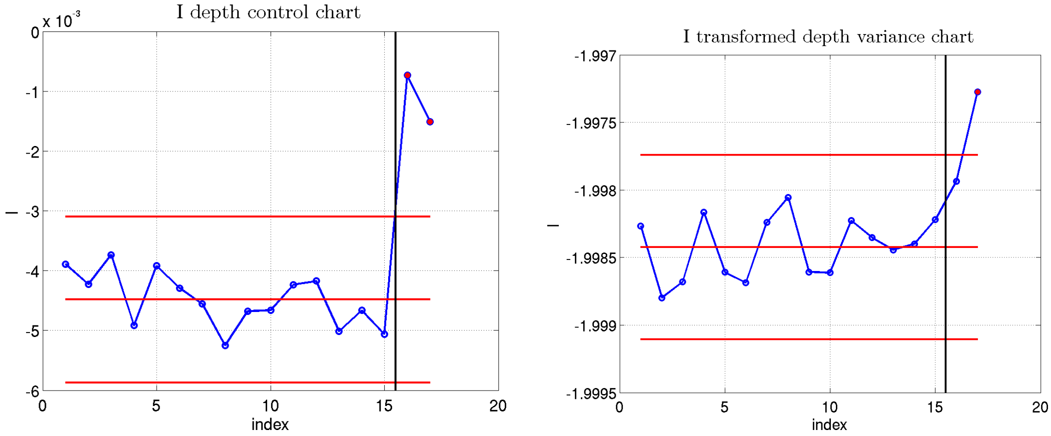
4.5.2 Performance on a Real Case Study

Control charts shown in Figure 4.16 to 4.19 presents both the in-control and the out-of-control textured surfaces. It is possible to observe that both the multivariate GP-based SPC procedure proposed and the simplest univariate control charts are performing well. As a matter of fact, they are able to detect out-of-control states using both the dimples depths and their areal descriptors.



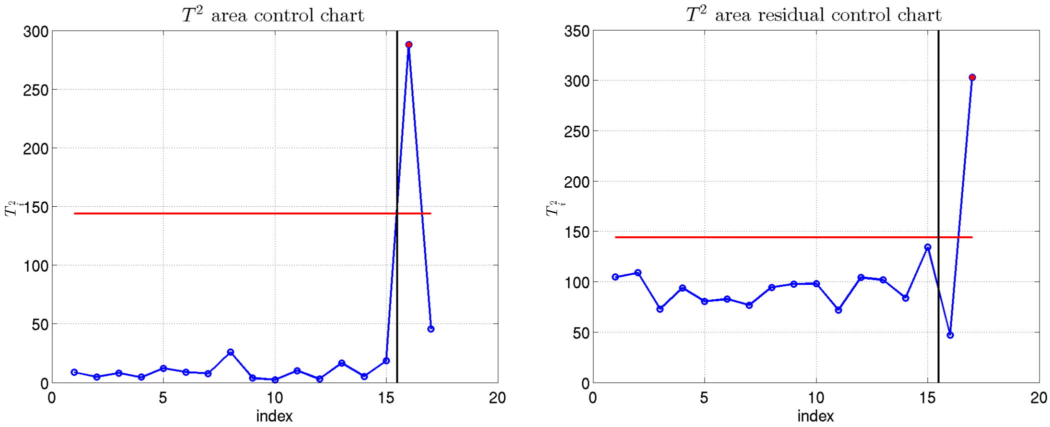
(a) Depth T^2 control chart **(b)** Depth residual T^2 control chart

Figure 4.16: GP-based control charts on the dimple depth



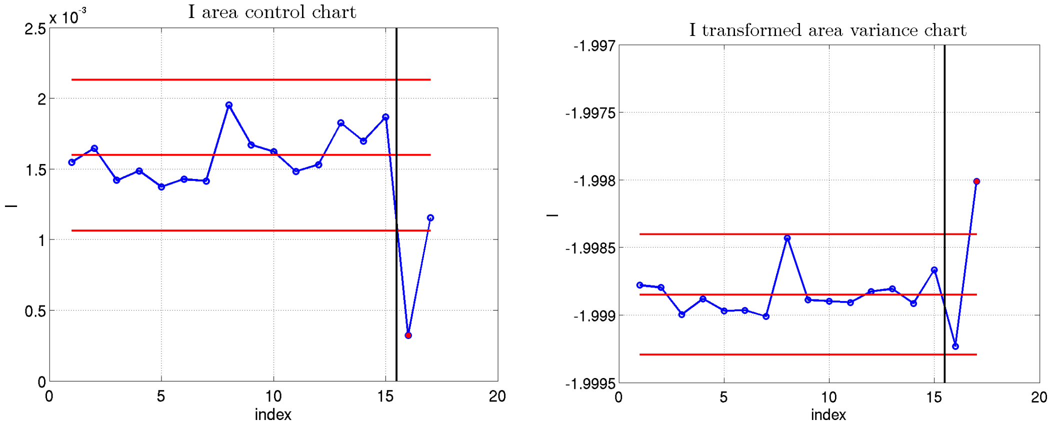
(a) Mean control chart **(b)** Variance control chart

Figure 4.17: Univariate control charts on the dimple depth



(a) Area T^2 control chart **(b)** Area residual T^2 control chart

Figure 4.18: GP-based control charts on the dimple area



(a) Mean control chart chart **(b)** Variance control chart

Figure 4.19: Univariate control charts on the dimple area

4.5.3 Performance on Simulated Case Studies

In order to compare the performance of the GP-based method with the one achieved by a simple univariate control chart, a challenging scenario was introduced. A spatially dependent out of focus of the laser was simulated. This out of control state may depend on a bad fixturing of the surface. This out of focus was simulated by adding the following linear shift to each dimple descriptor

$$\varphi = \delta \left| u_c \right|$$

where δ is a know value and u_c is the u coordinate of the dimple centroid.

Clearly δ is positive when the dimple depth is of interest, because, the dimple depth reduces as the laser goes out of focus. An opposite behavior is expected for the dimple area

A total of 10^3 out of control items were generated for each magnitude δ, and the corresponding probability β of not detecting the out of control state was computed. In this case, only the GP-based and the univariate procedures working on the mean are used (thus neglecting the control charts on the (residual) variance).

Table 4.1 shows the performance of the β error when the dimple depth is of interest. It is possible to observe that the GP-based procedure outperforms the univariate competitor.

δ ($\times 10^3$)	β_{GP}	β_{univ}
2.6	0.23	0.70
2.8	0.15	0.65
3.0	0.10	0.51
3.2	0.08	0.36

Table 4.1: β error (shift on the dimple depth)

Similarly, Table 4.2 shows results when dimple area is considered.

δ ($\times 10^3$)	β_{GP}	β_{univ}
-0.6	0.28	0.69
-0.8	0.19	0.57
-1.0	0.08	0.37
-1.2	0.04	0.24

Table 4.2: β error (shift on the dimple area)

Part II

Process Optimization

Chapter 5

Process Optimization

The goal of this chapter is to use the prediction model presented in chapter 1 to find the optimal value of a process response (z) as function of some controllable factor ($\boldsymbol{s}$).

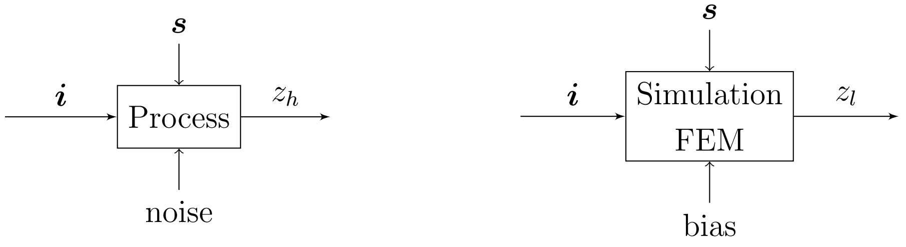
An optimization system consists on finding the best solution to this process within constraints (Biegler 2010). This task requires the following elements:

- objective function: it is a scalar quantitative performance measure that needs to be minimized or maximized (z);
- predictive model: it describes the behavior of the system (GP model). For the optimization problem this translates into a set of equations and constraints. These constraints comprise a feasible region that defines limits of performance for the system.
- variables: they appear in the predictive model and must satisfy the constraints ($\boldsymbol{s}$). This can usually be accomplished with multiple instances of variable values, leading to a feasible region that is determined by a subspace of these variables.

In order to optimize the output of a process, the response value, that depends on a particular value of the controllable factors (or locations) $\boldsymbol{s}$, can be computed through two different approaches: experiments and simulation.

In experiments a set of scientific tests is run and the response of a process is measured (see Figure 5.1a). The value of the response is a function of some non

controllable, or not controlled, input parameters ($\boldsymbol{i}$) and it is also a function of some noise affecting the measurements or due to the natural variability of the process.
If the response of the process is computed by simulation (see Figure 5.1b), the output of the process is a function of some input parameters $\boldsymbol{i}$ and of the controllable factors $\boldsymbol{s}$, but since the simulation is an approximation of the physical process, the output is biased because a simulation does not take into account all the degree of freedom of a real process.



(a) Scheme of process analysis through experiments

(b) Scheme of process analysis through simulation

Figure 5.1: Scheme of process analysis

The linkage model described in section 1.4 to fuse multisensor metrology data can be here used to merge the two types of data: the cheap (usually biased) simulations with the precise (but expensive) physical experiments (see Figure 5.2).

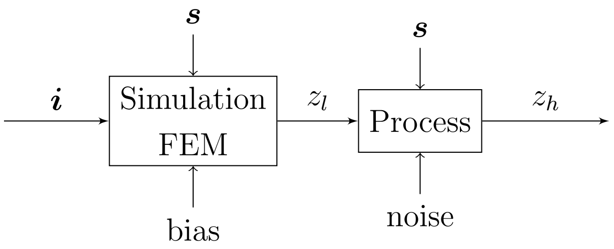


Figure 5.2: Scheme of process analysis through a fusion approach

In this chapter it is shown how an appropriate data fusion, described in section 1.4, based on combining simulated (Lo-Fi) and experimental (Hi-Fi) data can result in an efficient optimization procedure, which can significantly reduce the simulation and calibration efforts.

5.1 Process Optimization: Efficient Global Optimization Algorithm

In this work, optimization consists of finding the design point $\boldsymbol{s}$ such that the expected value of the response function is maximized (or minimized). The expected value of the response function in a new location $\boldsymbol{s}_0$ is computed via equation (1.20)

$$\begin{aligned}\mathbb{E}[Z_h(\boldsymbol{s}_0)|\boldsymbol{Z}_h = \boldsymbol{z}_h, \boldsymbol{Z}_l = \boldsymbol{z}_l] &= \widehat{\rho}(\boldsymbol{s}_0)\,\widehat{z}_l(\boldsymbol{s}_0) + \widehat{\delta}_0 \\ &+ \left(\widehat{\sigma}_\delta^2\,\widehat{\boldsymbol{r}}_\delta + \widehat{\boldsymbol{\sigma}}_0\right)^T \widehat{\boldsymbol{\Sigma}}_\Delta^{-1}\left(\boldsymbol{z}_h - \widehat{\boldsymbol{P}}\,\widehat{\boldsymbol{z}}_l - \widehat{\delta}_0\,\mathbf{1}_{n_h}\right)\end{aligned}$$

which shows prediction of the response function in any location using the data fusion model. In the following, without loss of generality, it is assumed that the response function has a unique global maximum (minimum).

A deterministic model (model without random term) is used to describe the simulation results. It is combined with a model including a nugget effect (random term) when simulated data are linked to experimental results. It is assumed that the new points added in the optimization procedure to achieve the optimum are simulated and hence not affected by noise. However, following the proposed model strategy, any new Lo-Fi simulation will be “translated” into the corresponding Hi-Fi data, thanks to the linkage model developed in the data fusion procedure (Equation 1.20). The linkage model described in section 1.4 can be interpreted as an off-line calibration step, where the Lo-Fi data are transformed to Hi-Fi data, depending on the specific locations where the points are observed.

Optimization based on a kriging model is usually performed with the Efficient Global Optimization (EGO) algorithm, originally proposed by Jones, Schonlau, and Matthias Welch (1998). The EGO algorithm looks for the global maximum (or minimum) of a generic deterministic function, using GP for computing predictions at any new location. The algorithm was extended in order to take into account also noisy functions in (Huang et al. 2006a) and in (Huang et al. 2006b).

The first step of the optimization algorithm consists of finding an initial guess of the response surface, by selecting an initial number of $10\,d$ points to roughly estimate the underlying surface, where d is the dimension of the function domain. After a first rough estimation of the function, the algorithm suggests adding a point where a criterion, called the Expected Improvement (EI), proposed by Mockus, Tiesis, and Zilinskas (1978), is maximized; this index represents the expected value of the improvement. Then, the algorithm is re-iterated until a stopping criterion is satisfied (usually when the ratio between the EI at the *j-th* step and the initial EI becomes 10^{-2}).

The details of the EGO algorithm in the version suitable to optimize the two-stage model is described below (see also (Huang et al. 2006a)).
At the *j-th* iteration of the algorithm, the current maximum is established as the maximum value predicted via the fusion model (i.e., the model combining both simulations and experimental data) in all the locations where simulations or real data are available. Let $\widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max})$ represent this current maximum at the *j-th* iteration

$$\widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max}) = \max_i \widehat{z}_h(\boldsymbol{s}_i)$$

where $\boldsymbol{s}_i$ represents the set of all the observed locations, i.e. both Lo-Fi and Hi-Fi data. A new variable $Q^{(j)}(\boldsymbol{s})$ is defined as the difference between the model $\widehat{z}_h(\boldsymbol{s})$ predicted at a generic location $\boldsymbol{s}$ and the current maximum

$$Q^{(j)}(\boldsymbol{s}) = \widehat{z}_h(\boldsymbol{s}) - \widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max}) \sim \mathcal{N}\left(\mu_{Q^{(j)}}(\boldsymbol{s}), \sigma^2_{Q^{(j)}}(\boldsymbol{s})\right). \tag{5.1}$$

In order to compute the distribution of $Q^{(j)}(\boldsymbol{s})$ at any location $\boldsymbol{s}$, a new location $\boldsymbol{s}_0$ is considered. A bi-variate vector consisting of the model prediction at this new location $\widehat{z}_h(\boldsymbol{s}_0)$ and the current maximum $\widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max})$ is built. This vector, since it is the prediction with the GP model described in chapter 1, is approximately distributed as a bi-variate normal

$$\begin{bmatrix} \widehat{z}_h(\boldsymbol{s}_0) \\ \widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max}) \end{bmatrix} \approx \mathcal{N}_2(\boldsymbol{\mu}_{ego}, \boldsymbol{\Sigma}_{ego}) \tag{5.2}$$

with

$$\begin{aligned}
\boldsymbol{\mu}_{ego} =& \widehat{\boldsymbol{P}}_{\boldsymbol{S}_0}\, \widehat{\boldsymbol{z}}_l(\boldsymbol{S}_0) + \widehat{\delta}_0\, \mathbf{1}_2 + \left(\widehat{\sigma}^2_\eta\, \widehat{\boldsymbol{R}}_{\delta_{h},\boldsymbol{s}_0} + \widehat{\boldsymbol{\Sigma}}_{l\boldsymbol{s}_h,\boldsymbol{s}_0}\right)^T \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \left(\boldsymbol{z}_h - \widehat{\boldsymbol{P}}\, \widehat{\boldsymbol{z}}_l - \widehat{\delta}_0\, \mathbf{1}_{n_h}\right) \\
\boldsymbol{\Sigma}_{ego} =& \widehat{\boldsymbol{\Sigma}}_{\Delta\boldsymbol{s}_0} - \widehat{\boldsymbol{\Sigma}}^T_{\Delta\boldsymbol{s}_h,\boldsymbol{s}_0} \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \widehat{\boldsymbol{\Sigma}}_{\Delta\boldsymbol{s}_h,\boldsymbol{s}_0} + \\
&+ \frac{\left(\mathbf{1}_2 - \widehat{\boldsymbol{\Sigma}}^T_{\Delta\boldsymbol{s}_h,\boldsymbol{s}_0} \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \mathbf{1}_{n_h}\right)\left(\mathbf{1}_2 - \widehat{\boldsymbol{\Sigma}}^T_{\Delta\boldsymbol{s}_h,\boldsymbol{s}_0} \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \mathbf{1}_{n_h}\right)^T}{\mathbf{1}^T_{n_h} \widehat{\boldsymbol{\Sigma}}_\Delta^{-1} \mathbf{1}_{n_h}}
\end{aligned}$$

where $\boldsymbol{S}_0 = [\boldsymbol{s}_0,\ \boldsymbol{s}_{max}]^T$, $\boldsymbol{S}_h = [\boldsymbol{s}_1,\ \boldsymbol{s}_2, \ldots,\ \boldsymbol{s}_{n_h}]^T$ is the matrix of Hi-Fi points, $\boldsymbol{R}_{\delta_{\boldsymbol{S}_h, \boldsymbol{S}_0}}$ is the correlation matrix between $\boldsymbol{S}_h$ and $\boldsymbol{S}_0$, $\boldsymbol{\Sigma}_{l_{\boldsymbol{S}_h, \boldsymbol{S}_0}}$ is the covariance matrix between $\rho(\boldsymbol{S}_h)\widehat{\boldsymbol{z}}_l(\boldsymbol{S}_h)$ and $\rho(\boldsymbol{S}_0)\widehat{\boldsymbol{z}}_l(\boldsymbol{S}_0)$, $\boldsymbol{\Sigma}_{\Delta_{\boldsymbol{S}_0}}$ is the covariance matrix of $\boldsymbol{S}_0$ and $\boldsymbol{\Sigma}_{\Delta_{\boldsymbol{S}_h, \boldsymbol{S}_0}}$ is the covariance matrix between $\boldsymbol{S}_h$ and $\boldsymbol{S}_0$. Clearly, the distribution of $Q^{(j)}(\boldsymbol{s})$ can be now computed starting from the aforementioned bi-variate vector, by applying standard multivariate statistical theory (Johnson and Wichern 2007)

$$Q^{(j)}(\boldsymbol{s}) \approx \mathcal{N}\Big(\mu_{Q^{(j)}}(\boldsymbol{s}) = \boldsymbol{c}^T \boldsymbol{\mu}_{ego}, \sigma^2_{Q^{(j)}}(\boldsymbol{s}) = \boldsymbol{c}^T \boldsymbol{\Sigma}_{ego} \boldsymbol{c}\Big) \tag{5.3}$$

where $\boldsymbol{c} = (1, -1)^T$. It is possible to observe that the mean and the variance of the random variable in Equation (5.3) can be rewritten as

$$\begin{aligned}
\mu_{Q^{(j)}}(\boldsymbol{s}) =& \mathbb{E}\left[\widehat{z}_h(\boldsymbol{s})\right] - \mathbb{E}\left[\widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max})\right] \\
\sigma^2_{Q^{(j)}}(\boldsymbol{s}) =& \mathbb{V}ar\left[\widehat{z}_h(\boldsymbol{s})\right] + \mathbb{V}ar\left[\widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max})\right] - 2\,\mathbb{C}ov\left[\widehat{z}_h(\boldsymbol{s}), \widehat{z}^{(j)}_{max}(\boldsymbol{s}_{max})\right]
\end{aligned}$$

The expected improvement can be thus written as a function of this new variable $Q^{(j)}(\boldsymbol{s})$, representing the difference between the objective function and the current optimum in a generic location, namely

$$\begin{aligned}
\mathrm{EI}^{(j)}(\boldsymbol{s}) =& \mathbb{E}(\max\{Q^{(j)}(\boldsymbol{s}), 0\}) = \int_0^{+\infty} \frac{q}{\sqrt{2\pi\sigma^2_{Q^{(j)}}(\boldsymbol{s})}} \exp\left\{-\frac{1}{2}\frac{\left(q - \mu_{Q^{(j)}}(\boldsymbol{s})\right)^2}{\sigma^2_{Q^{(j)}}(\boldsymbol{s})}\right\} dq \\
=& \sigma_{Q^{(j)}}(\boldsymbol{s})\ \varphi\left(\frac{\mu_{Q^{(j)}}(\boldsymbol{s})}{\sigma_{Q^{(j)}}(\boldsymbol{s})}\right) + \mu_{Q^{(j)}}(\boldsymbol{s})\ \Phi\left(\frac{\mu_{Q^{(j)}}(\boldsymbol{s})}{\sigma_{Q^{(j)}}(\boldsymbol{s})}\right).
\end{aligned}$$

From the last equation is possible to observe that the $\mathrm{EI}^{(j)}(\boldsymbol{s})$ is a weighted sum of the mean and the standard deviation of the variable $Q^{(j)}(\boldsymbol{s})$. The algorithm suggests to add a new point where the mean is high, which consists on an improvement of the current maximum, or where the standard deviation is high, i.e. where the variability of prediction is high to make better forecast on that part of the design space.

The optimization task of the function can be carried out using a Matlab global optimization algorithm proposed by Zsolt et al. (2007). It is observed that the EGO algorithm requires a new estimation of $\widehat{z}_h(\boldsymbol{s})$, $\widehat{z}^{Q^{(j)}}_{max}$ and $\mu_{Q^{(j)}}(\boldsymbol{s})$ and $\sigma_{Q^{(j)}}(\boldsymbol{s})$ at each step.

5.1.1 Example of Optimization Performance with Artificially Generated Data

In this section the EGO procedure will be shown at work, with reference to the example described in section 1.4. Without loss of generality, the number of controllable parameters will be kept equal to two in order to be able to visualize the data. The Lo-Fi data model is

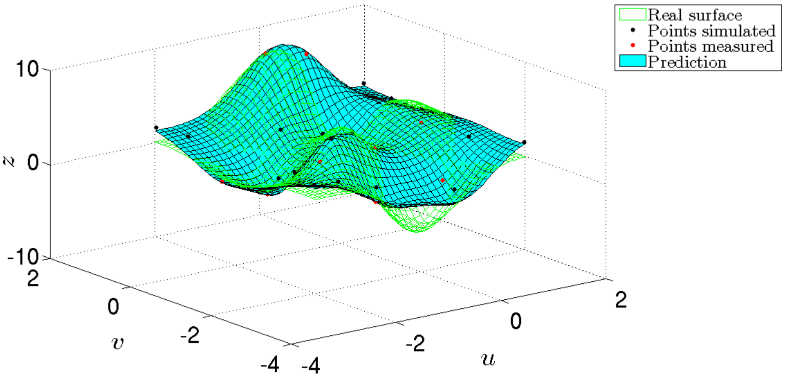
$$z_l = f(\boldsymbol{s}) + \frac{3}{10}u^2 + \frac{3}{10}v^2$$

while the Hi-Fi data are generated from

$$z_h = f(\boldsymbol{s}) + \varepsilon_h, \quad \varepsilon_h \sim \mathcal{N}(0, 0.5).$$

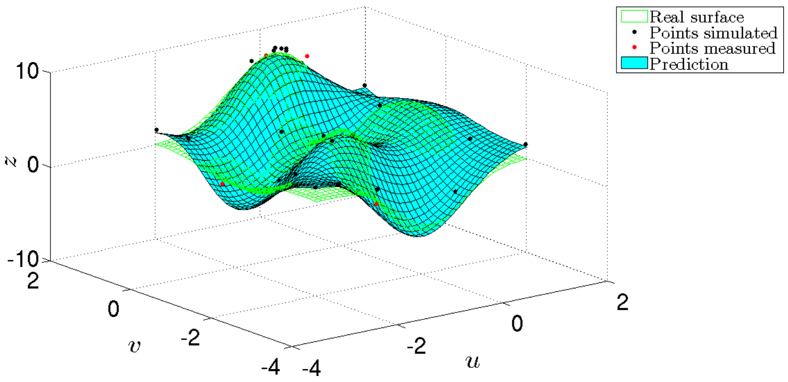
An initial design of 20 Lo-Fi points and 10 Hi-Fi points, selected according to a *max-min* latin hypercube will be considered.

Figure 5.3 shows the surface predicted before (5.3a) and after (5.3b) the EGO algorithm. As clear from the plot, the EGO algorithm concentrates simulations at the neighborhood of the maximum value (in this example after 6 steps), thus allowing for a better estimation of the true surface in that region.



(a) Before the EGO algorithm

In order to test the ability of the EGO approach to converge, the optimization based on our two-stages model was repeated 10 times, starting from different data set, generated randomly at each run. Table 5.1 summarizes results of these simulations, showing the number of iterations required before convergence together with the optimal value found and the difference between the true values of the



(b) After the EGO algorithm

Figure 5.3: Surface estimation before and after the EGO algorithm

maximum coordinates (i.e., $u = -0.009$, $v = 1.581$ and $z = 8.106$, respectively) and the values of the optimum found by the proposed method. As clear from the table and from the figures above, the error is consistently low, if compared to the range of the 3 variables.

Considering the sample of 10 replicates, statistical tests to reject the hypothesis that the coordinates of the true computed optima are, on average, equal to the true values were performed. As a final result, the null hypothesis cannot be rejected for all the three coordinates, the p-values of the tests were 0.69, 0.66 and 0.97, respectively for u, v and z.

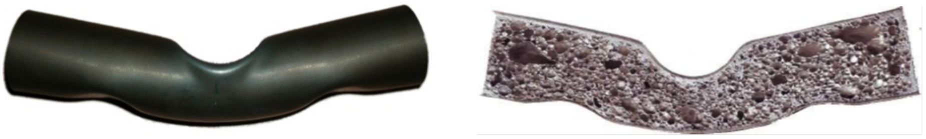
5.2 Case Study: Design of an Anti-intrusion Side bar for Vehicles

The modeling and optimization procedure described in section 5.1 is applied to a real optimization problem: the design of an anti-intrusion side bar for vehicles, made of an outer tubular steel case and a filling reinforcement made of aluminum foam (Strano, Mussi, and Monno 2010). The filling of cases made of thin metal sheets or tubes with a reinforcement made of cellular metals (or metal foams), allows production of lightweight, high performance components, particularly suited for flexural resistance in terms of amount of energy absorbed for a given maximum load (Figure 5.4).

# iterations before convergence	optimal value and difference with true optimal u		optimal value and difference with true optimal v		optimal value and difference with true optimal z	
4	-0.052	(-0.042)	1.569	(-0.012)	8.115	(0.009)
11	0.034	(0.043)	1.585	(0.004)	8.091	(-0.015)
6	-0.064	(-0.055)	1.569	(-0.012)	8.099	(-0.008)
5	-0.052	(-0.043)	1.572	(-0.009)	8.117	(0.011)
6	0.033	(0.043)	1.570	(-0.011)	8.097	(-0.010)
6	-0.001	(0.009)	1.595	(0.014)	8.113	(0.006)
4	-0.051	(-0.042)	1.572	(-0.010)	8.096	(-0.010)
5	-0.050	(-0.040)	1.568	(-0.014)	8.114	(0.008)
6	-0.051	(-0.042)	1.594	(0.013)	8.098	(-0.009)
6	0.030	(0.039)	1.565	(-0.017)	8.101	(-0.006)

Table 5.1: Number of iterations of 10 replicates required by the EGO algorithm and value and difference of the optimum coordinates found with respect to the coordinates of the true maximum value

In order to optimize the performance of the component in case of an accident, several issues must be taken into account. These refer to a closed section with a composite (bi-material) structural beam with initial length L, vertical average dimension H, horizontal average size W, initial average cross section area $S = H \cdot W$, total occupied volume $V = S \cdot L$, mass M and apparent density $\rho = \frac{V}{M}$. In lateral impact, the structure will undergo a flexural state of stress-strain. Given a load P $[kN]$ - deflection δl $[mm]$ diagram in bending of a foam filled bar, up to any value of deflection δl, the load curve profile will exhibit a maximum load value P_{max}, an average load P_{avg} and an amount of absorbed energy per volume $E_{abs} = P_{avg} \cdot \delta l$. The crash force efficiency can be written as the ratio between the mean load and the maximum load of a Force-Displacement curve (Yuen and Nurick 2008): $\eta = \frac{P_{avg}}{P_{max}}$. A body with high efficiency will have a large energy absorption, while limiting the maximum load P_{max} (and the corresponding acceleration) transmitted to the vehicle. A Specific Energy Absorption (SEA) can be also defined, as the ratio between the absorbed energy and the total mass: $SEA = \frac{E_{abs}}{M}$. In a lateral crash, for any given tubular composite structure and



(a) Stainless steel tube after 3 point bending, filled by an aluminum foam

(b) Longitudinal cross section

Figure 5.4: Steel tube

a given amount of incoming energy, it is important to achieve the competing objectives:

- increase or maximize the energy absorption E_{abs}, given a maximum deflection δl_{max}, while limiting the total mass M; this is equal to maximizing the SEA of the structure;
- increase or maximize the crash force efficiency η;
- minimize the intrusion into the vehicle δl_{max}.

This is a typical problem in engineering design, where simulations and/or experimental data are required to empirically reconstruct the relationship between the design variables and the response function thus allowing optimization. In this test case, the design vector, $\boldsymbol{s} = (u, v)^T$, is made of two design variables. The first variable, u, is a toughness indicator, related to the plastic material properties of the outer steel skin. u represents the area under the flow stress curve, it measures the fracture toughness of the skin material

$$u = \frac{n^{n+1} \cdot K}{n+1} \; MPa$$

where K and n are respectively the hardening coefficient and exponent of the flow stress power law. The second variable v is a shape factor, related to its geometry

$$v = \frac{J}{W} \; mm^3$$

where J is the moment of inertia of the tube cross section and W is the depth of the specimen, in the direction of the movement of the punch, i.e. the lateral encumbrance of the structure. In v, J is divided by W because it is important to keep the encumbrance under control, due to the limited space available inside the car door.

Let $\boldsymbol{s}$ represent a design variable influencing the aforementioned indicators, a synthetic objective function z can be built as

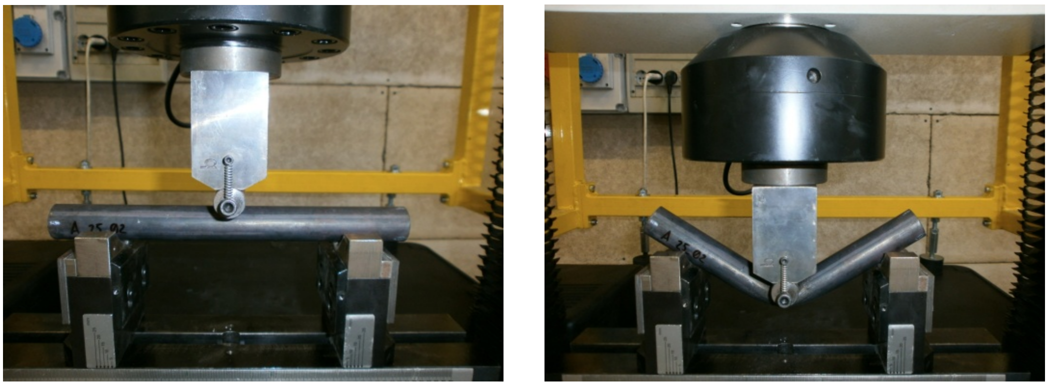
$$z(\delta l_{max}, \boldsymbol{s}) = \frac{SEA(\delta l_{max}, \boldsymbol{s}) \cdot \eta(\delta l_{max}, \boldsymbol{s})}{\delta l_{max}} = \frac{P_{avg}^2(\delta l_{max}, \boldsymbol{s})}{P_{max}(\delta l_{max}, \boldsymbol{s}) \cdot M(\boldsymbol{s})} \tag{5.4}$$

and measured in $[\frac{kN}{kg}]$. In the present case, the maximum admissible intrusion was set to $\delta l_{max} = 48\ mm$ for reasons related to design criteria of cars.

Both the design variables affect the response function $z(\delta l_{max}, \boldsymbol{s})$. Unfortunately, the way in which these two design variables affect the response is not analytically known and has to be detected via simulation and/or experimental data.

5.2.1 The Hi-Fi Data: Experimental Data

A total amount of 7 experimental conditions of anti-intrusion bar were analyzed, for a total of 21 bending tests. An example of the layout of the experiments can be found in Figure 5.5.



(a) Tube before the bending

(b) Tube after the bending

Figure 5.5: Experiment

Table 5.2 shows all the experimental conditions tested and their geometrical and dimensional specifications.

Condition	Material	Cross section shape	Length or diameter [mm]	Thickness [mm]	# of tests
1	Fe 360	Square	20	2.0	3
2	Fe 360	Square	20	2.0	3
3	Fe 360	Round	27	2.5	3
4	DOCOL800DP	Square	30	1.5	3
5	DOCOL800DP	Square	30	1.5	3
6	DOCOL800DP	Round	32	2.0	3
7	AISI 304	Round	25	1.5	3

Table 5.2: Experimental conditions

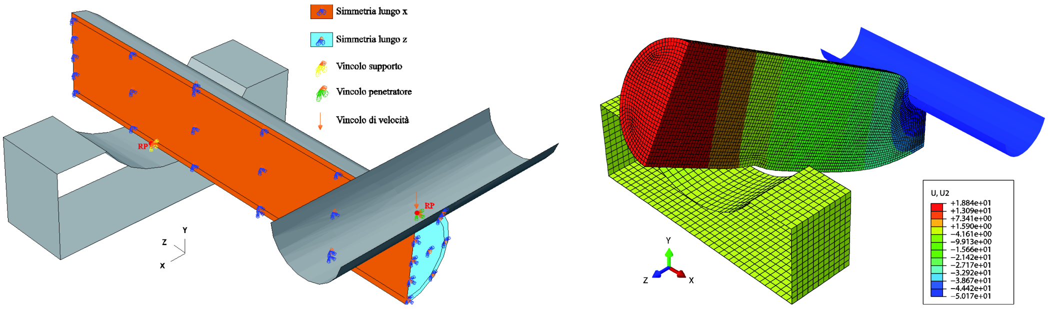
These tubes were filled with an aluminium foam, Al 6061. It was used a direct filling foam of the tubes. In this way, the precursor is inserted in the tube and the assembly and, once the two side-ends are closed, it is heated in the oven until the foaming process is not finished. The Nabertherm LT 9/11 HA was used. The density chosen for the foam was 0.56 $\frac{kg}{dm^3}$, the foaming time was between 5 and 13 minutes, according to the condition analyzed. The temperature of foaming was 750 °C for all the conditions. After the tubes filling, bending tests on three points were conducted with a 100 kN press MTS Alliance RT/100, with an indenter diameter of 20 mm and, above it, a load cell of 100 kN. The speed of the press was 500 $\frac{mm}{min}$ and the stroke was set to 48 mm.

Clearly, each experimental test is both time- and money-consuming because of the manufacturing process of foam filled tube (Strano, Mussi, and Monno 2010).

5.2.2 The Lo-Fi Data: Finite Element Method Model

Finite Element Method (FEM) simulations were run with ABAQUS software (Hibbit, Karlsson, and Sorensen 2007). Four components were modeled to reproduce a three points bending test: the tube, the foam, the indenter and the support (Figure 5.6). Only a quarter of all assembly was analyzed in order to reduce the simulation time.

Simulations of the three-point bending tests were run with an FEM model



(a) Set up of the simulation **(b)** Tube after bending

Figure 5.6: FEM simulation

with explicit time integration scheme. The foam was modeled with 3D hexahedral elements with a "Crushable Foam" isotropic material formulation. The tube was modeled with shell elements with an elastic-plastic isotropic material formulation. Contact between foam and tube was modeled with a penalty formulation and a Coulomb coefficient of friction equal to 0.6. The number of elements changed according to the simulated structure and the simulation time for each test ranges between 4 and 8 CPU hours. Each simulation was therefore quite expensive from a computational viewpoint. This is a typical situation in optimization problems where large plastic deformations are involved, due to the strong nonlinearities of the computational problem.

The common approach used in the scientific literature in this case is generally based on the following steps:

1. running a limited number of experiments,

2. building the numerical model and iterating until a satisfactory calibration of the simulation model is reached,

3. running the simulations required for modeling and optimization.

Calibration is often not described in scientific papers, and the final calibrated model is generally presented, as in Zarei and Kroger (2008). Nevertheless, step 2 is always required, although not explicitly mentioned. The aim of this section is to demonstrate how the fusion model can be used for merging steps 2 and 3, reducing the total computational time. This is why the fusion model and the EGO optimization method described in section 5.1 are applied in two different scenarios.

In the first "calibrated" scenario, step 2 of the traditional three-step procedure was applied, i.e., the simulation model was pre-calibrated. In particular, for each investigated material (Fe 360, Docol 800DP, AISI 304) with experiments available the K and n parameters of the tested materials were iteratively calibrated until the mean mismatch between experimental and numerical results felt below a 10% error threshold for the objective function $z(48, \boldsymbol{s})$. In other words, the FEM simulations were run again and again, changing K and n, until the error on $z(48, \boldsymbol{s})$ was acceptable. Data used for the three initial materials in both scenarios are given in Table 5.4. In 1 out of 7 cases, no calibration was required, i.e. the error was already at 10%. In the other 6 conditions, from 2 to 3 FEM runs were required in order to calibrate the input variables, for a total of about 15 additional computations. Calibrating the K and n values makes the u's change accordingly, i.e. the calibration has an effect on one of the two design variables. For additional materials, for which no real test are available have been evaluated by FEM only, the tensile test hardening values found using the CES$^{\text{TM}}$ software package were used to compute K and n.

In Table 5.3 are shown the difference percentage between the simulations and the experimental results.

Condition	Experiment	Simulation	Difference %
1	23 000.60	26 476.72	15.11
2	27 327.71	29 091.88	6.46
3	34 344.88	34 310.00	−0.10
4	39 178.20	45 195.27	15.36
5	50 862.09	50 503.31	−0.71
6	37 474.71	38 209.45	1.96
7	32 827.03	26 789.46	−18.39

Table 5.3: Values of $z(48, \boldsymbol{s})$ of both experiments and simulations and difference percentage after calibration

In the second "non-calibrated" scenario, no time was spent for calibrating the simulation model, saving in this case all the aforementioned 15 runs, each amounting to about 3 to 4 CPU hours with the workstation used in the study. All the materials, including the three materials with experimental values available, were simply modeled considering the nominal K and n values, calculated by the data given in CES^{TM}.

	Non calibrated			Calibrated					
	All cross sections			Round cross section			Square section		
	K $[Mpa]$	n	u $[Mpa]$	K $[Mpa]$	n	u $[Mpa]$	K $[Mpa]$	n	u $[Mpa]$
Fe 360	603	0.20	73	603	0.20	73	530	0.26	77
DOCOL800DP	944	0.18	107	700	0.17	76	944	0.18	107
AISI 304	740	0.43	154	1133	0.43	235	-	-	-

Table 5.4: K and n values for both scenarios

5.2.3 Optimization

The optimization problem can be formalized as finding the $\boldsymbol{s}$-value that maximizes $z(\delta l_{max}, \boldsymbol{s})$, within the range of the investigated values of $\boldsymbol{s}$.

Since both simulations and experiments were expensive, any mathematical technique aimed at reducing the number of design points required for optimization would be highly appreciated in the field of design and manufacturing optimization of metal foam based structures.

The method described above was applied to the crashworthiness optimization of aluminum foam filled tubular structures presented in section 5.2. For both the calibrated and non-calibrated scenarios, $n_h = 7$ experimental design locations were initially available. Every experimental combination, which is a tubular structure with its shape (summarized by v) and outer material (summarized by u) was replicated three times, for a total of $n_h \times 3 = 21$ experimental (Hi-Fi) couples of z and $\boldsymbol{s}$ values. At these design locations, FEM simulations were performed for both the scenarios, i.e., with calibrated K and n values and with non-calibrated material data. Furthermore, $n_l = 13$ additional design locations were added according to an initial random, space filling design of computer simulations (Lo-Fi data). A total of $N_{des} = n_h + n_l = 7 + 13 = 20$ locations of z and $\boldsymbol{s}$ values were therefore originally available. The EGO algorithm embedded in the optimization routine suggested $N_{ego} = 4$ additional evaluation points in the non-calibrated (Table 5.5) scenario and $N_{ego} = 5$ additional points to be simulated in the calibrated one (Table 5.6). During the optimization, every new location suggested by the EGO approach, i.e. every new combination of u and v values, was "translated" into an actual material and an actual shape. Considering that it is not realistically possible to find an engineering material with exactly the u values suggested by the EGO, the nearest available material option founded on the CES^{TM} software package to be used instead is considered. When more than one option was found by the CES^{TM} software, all the close materials were tested. Similarly, it was difficult to define a real (i.e. potentially available on the market) tube cross section with the exact shape suggested by the EGO: the closest approximations was thus considered. For this reason, the actual number of design locations tested was larger than $N_{des}+N_{ego}$ for both scenarios: 27 instead of $N_{des} + N_{ego} = 24$ locations were used in the non-calibrated scenario; 35 instead of $N_{des} + N_{ego} = 25$ locations were used in the calibrated one.

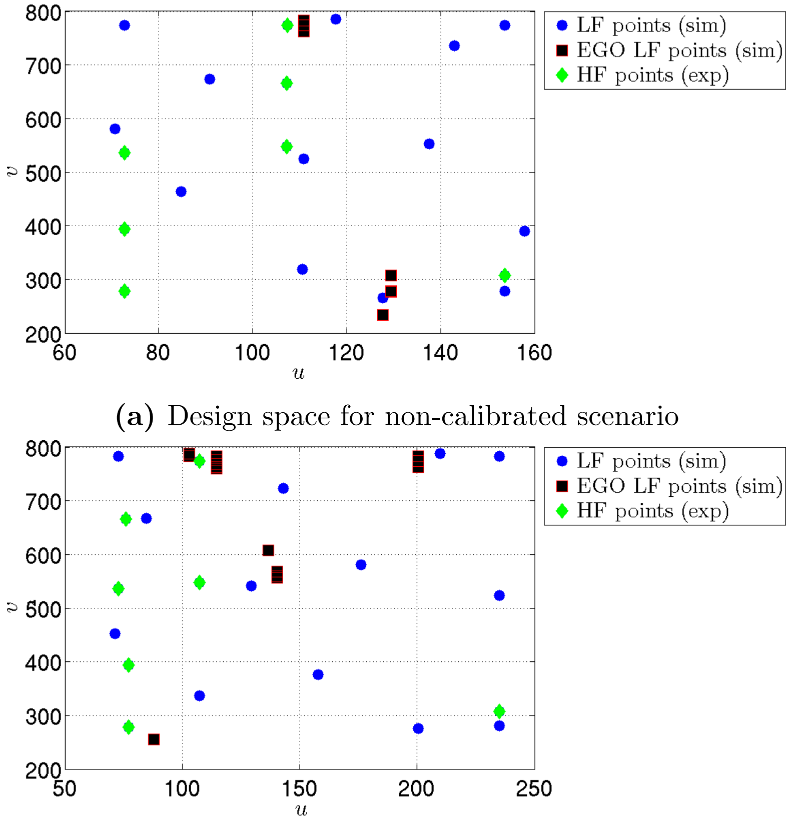
Iteration	u	v
1	128.03	233.21
2	131.79	276.52
3	110.00	774.00
4	129.48	306.69

Table 5.5: Evolution of the EGO algorithm in the non-calibrated scenario

Iteration	u	v
1	198.00	774.00
2	141.46	556.25
3	102.33	786.53
4	113.52	768.36
5	114.15	780.72

Table 5.6: Evolution of the EGO algorithm in the calibrated scenario

It is useful to further clarify the differences between the two scenarios, with reference to Figure 5.7. In the design space, the variable u is a function of the material parameters under calibration, K and n. As a consequence, in the calibrated scenario, the experimental z values are placed in the design space at calibrated locations along the u axis. In the non-calibrated scenario, the experimental z values are still "reliable", i.e. Hi-Fi, but they are located in the design space at non-calibrated u locations.



(a) Design space for non-calibrated scenario

(b) Design space for calibrated scenario

Figure 5.7: Design spaces

The optimal values of the non-calibrated and the calibrated scenario for each step of the EGO algorithm are reported in Table 5.7 and in Table 5.8, respectively.

Iteration	u	v	z
0	107.61	782.42	50 614.35
1	129.40	298.21	51 766.99
2	105.22	782.64	51 200.74
3	107.61	782.56	50 617.58
4	104.38	781.22	51 690.14

Table 5.7: Evolution of the optimum during the EGO steps in the non-calibrated scenario

Iteration	u	v	z
0	107.05	777.39	50 376.07
1	107.16	770.56	50 130.40
2	107.12	776.63	50 376.73
3	113.52	785.20	49 129.89
4	113.15	786.77	52 490.92
5	114.57	774.76	52 712.46

Table 5.8: Evolution of the optimum during the EGO steps in the calibrated scenario

The resulting interpolated surface and the optimal solution for both the scenarios are shown in Figures 5.8 and 5.9, respectively. The values of the optimum found are not in a ascending order because during the steps of the algorithm there is also a re-estimation of the response function z.

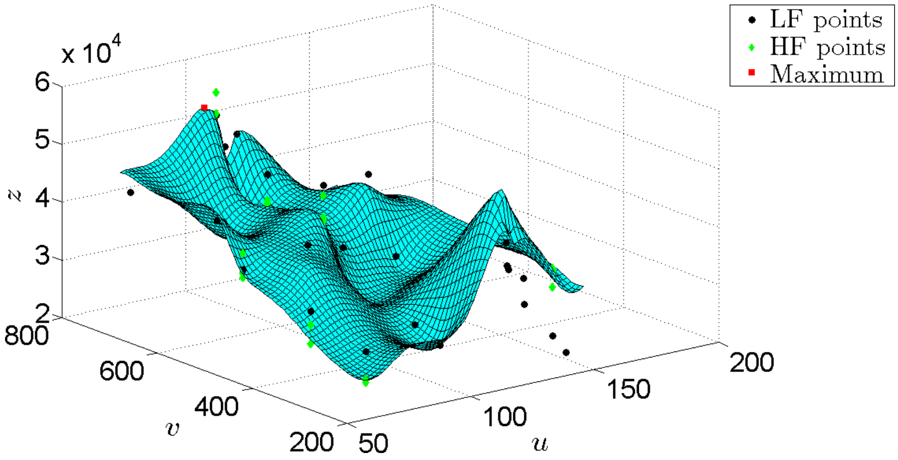


Figure 5.8: Surface of the model for the non-calibrated scenario; optimal solution values are given in Table 5.7

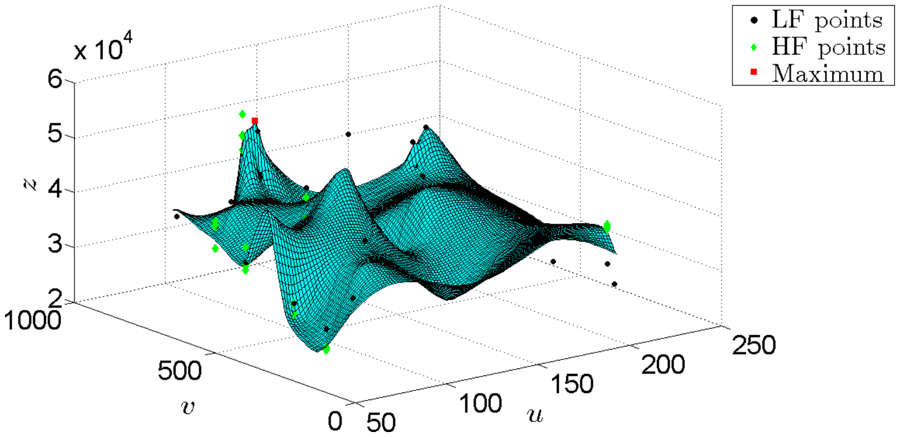


Figure 5.9: Surface of the model for the calibrated scenario; optimal solution values are given in Table 5.8

By comparing the mentioned figures, it is observed how the two optimal $\boldsymbol{s}$-values are very similar, i.e. both scenarios yield more or less the same optimal solution. Furthermore, the estimated $z(48, \boldsymbol{s})$ values are similar too, with a difference of only 0.1%. The expected error of the optimal value of $z(48, \boldsymbol{s})$ is smaller for the calibrated *vs.* the non-calibrated scenario (1764 *vs.* 5558 $\frac{N}{kg}$), which is not surprising, as the prediction yielded by the calibrated model is obviously more accurate. Although the two scenarios yield the same optimal solution, the modeled surfaces are not very similar in shape. This is mainly because the calibration affects the u values and range.

Conclusion and Future Research

The focus of the work was the reconstruction of response functions (namely a real machined surface or the performance of a process as a function of the controllable parameters) using a fusion model that links data coming from two different sources of information.

We showed how a unifying framework can be used to solve very different reconstructions problems. In particular we proposed:

1. an approach to deal with the "big data" problem (since non contact data can arrive to some millions);
2. an approach to manage uncertainty propagation in the data fusion step;
3. an approach to combine surface modeling to quality control for regularly structured surfaces;
4. an approach to combine FEM simulations and real experimental data to optimize the performance of a manufacturing process.

Possible future improvements can be foreseen.

Firstly, the reconstruction of shapes from large dataset can be performed by combining segmentation (to identify surface patches) to surface fitting via data fusion. New approaches for data fusion of multi-patches surfaces have to be developed.

Secondly, appropriate sampling schemes for the reconstruction via data fusion should be designed.

Appendices

Appendix A

Proof of GP fusion model

In this Section, the linkage model proposed by Qian et al. (2006) is extended to take into account also the uncertainty of prediction of the Lo-Fi model.

The linkage model is reported in matrix form in Equation (1.15)

$$\boldsymbol{z}_h = \boldsymbol{P}\widehat{\boldsymbol{z}}_l + \delta_0 \mathbf{1}_{n_h} + \boldsymbol{\delta} + \boldsymbol{\varepsilon}_\Delta \tag{A.1}$$

If $\boldsymbol{\delta}$, $\boldsymbol{\varepsilon}_\Delta$ and $\widehat{\boldsymbol{z}}_l$ are assumed to be independent, it is possible to show that the distribution of $\boldsymbol{z}_h$ is:

$$\boldsymbol{z}_h \sim \mathcal{N}_{n_h}(\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \delta_0 \mathbf{1}_{n_h}, \boldsymbol{\Sigma}_\Delta) \tag{A.2}$$

where:

$$\boldsymbol{\Sigma}_\Delta = \boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P}^T + \sigma_\delta^2 \boldsymbol{R}_\delta + \sigma_\Delta^2 \boldsymbol{I} \tag{A.3}$$

where $\boldsymbol{\Sigma}_0$ is the prediction covariance matrix of the first stage model in $\boldsymbol{S}_h$.

A.1 Prediction of new Point and its Propriety

The prediction value and the uncertainty interval of a new observations at location $\boldsymbol{s}_0$ is now computed. As for the classic GP model, the starting point is the joint distribution between $\boldsymbol{z}_h$ and $Z_h(\boldsymbol{s}_0) = Z_0$:

$$\begin{pmatrix} \boldsymbol{z}_h \\ Z_0 \end{pmatrix} |\boldsymbol{z}_h, \boldsymbol{z}_h \sim \mathcal{N}_{n_h+1}\left(\begin{pmatrix} \boldsymbol{P}\widehat{\boldsymbol{z}}_l + \delta_0 \mathbf{1}_{n_h} \\ \rho(\boldsymbol{s}_0)\widehat{z}_l(\boldsymbol{s}_0) + \delta_0 \end{pmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_\Delta & \sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0 \\ \sigma_\delta^2 \boldsymbol{r}_\delta^T + \boldsymbol{\sigma}_0^T & \sigma_\delta^2 + \sigma_{00}\rho^2(\boldsymbol{s}_0) + \sigma_\Delta^2 \end{bmatrix}\right)$$

where $\boldsymbol{r}_\delta = Corr(\boldsymbol{z}_h, Z_0)$, $\sigma_{00} = Var[\widehat{Z}_h(\boldsymbol{s}_0)]$ and $\boldsymbol{\sigma}_0$ is the vector whose *i-th* component is $\sigma_{0i} = \rho(\boldsymbol{s}_i)\,\rho(\boldsymbol{s}_0)\cdot Cov(\widehat{z}_{h_i}, Z_0)$.
The conditional distribution of the new prediction is:

$$\begin{aligned} Z_h(\boldsymbol{s}_0)|\boldsymbol{z}_h, \boldsymbol{z}_l \sim & \mathcal{N}\Big(\rho(\boldsymbol{s}_0) + \delta_0 + \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}(\boldsymbol{z}_l - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \delta_0 \mathbf{1}_{n_h}), \\ & \sigma_\delta^2 + \sigma_{00}\,\rho^2(\boldsymbol{s}_0) + \sigma_{\varepsilon_h}^2 - \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\left(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\right)\Big) \end{aligned}$$

Even in this case, since the real parameter values are unknown, we can substitute them with their estimates (plug-in procedure).

Substituting $\widehat{\delta}_0$ from Equation (1.18) the prediction in a new location $\boldsymbol{s}_0$ can be rewritten as

$$\begin{aligned} \widehat{Z}_h(\boldsymbol{s}_0) =& \mathbb{E}(Z_h(\boldsymbol{s}_0)|\boldsymbol{z}_h, \boldsymbol{z}_l) = \rho(\boldsymbol{s}_0)\widehat{z}_l(\boldsymbol{s}_0) + \widehat{\delta}_0 + \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\Big(\boldsymbol{z}_l - \boldsymbol{P}\widehat{\boldsymbol{z}}_l - \widehat{\delta}_0 \mathbf{1}_{n_h}\Big) \\ =& \rho(\boldsymbol{s}_0)\widehat{z}_l + \big[1 - (\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0)^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}\big]\widehat{\delta}_0 - (\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0)^T \boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \\ & + \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{z}_l \\ =& \Bigg\{\widetilde{\boldsymbol{f}}_\Delta^T(\boldsymbol{s}_0) - \Big[1 - \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}\Big]\frac{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}\widetilde{\boldsymbol{F}}_\Delta}{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}} - \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \cdot \\ & \cdot \boldsymbol{\Sigma}_\Delta^{-1}\widetilde{\boldsymbol{F}}_\Delta\Bigg\}\boldsymbol{\rho} + \Bigg\{\Big[1 - \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}\Big]\frac{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}}{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}} + \\ & + \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\Bigg\}\boldsymbol{z}_l \\ =& \widetilde{\boldsymbol{f}}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{\rho} - \boldsymbol{a}_*^T \boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}_*^T \boldsymbol{z}_l \end{aligned}$$

with

$$\boldsymbol{a}_*^T = \Big[1 - \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}\Big]\frac{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}}{\mathbf{1}_{n_h}^T \boldsymbol{\Sigma}_\Delta^{-1}\mathbf{1}_{n_h}} + \big(\sigma_\delta^2 \boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\big)^T \boldsymbol{\Sigma}_\Delta^{-1} \tag{A.4}$$

and $\widetilde{\boldsymbol{f}}_\Delta(\boldsymbol{s}_0) = \boldsymbol{f}_\Delta(\boldsymbol{s}_0)\,\widehat{z}_l(\boldsymbol{s}_0)$ and $\widetilde{\boldsymbol{F}}_\Delta = [\boldsymbol{f}_\Delta(\boldsymbol{s}_1)\,\widehat{z}_l(\boldsymbol{s}_1), \ldots, \boldsymbol{f}_\Delta(\boldsymbol{s}_{n_h})\,\widehat{z}_l(\boldsymbol{s}_{n_h})]^T$.

The estimator has the form $a_0 + \boldsymbol{a}_1^T \boldsymbol{z}_l$ and since it is not possible to find a closed-form expression for $\hat{\boldsymbol{\rho}}$, it is not the best linear unbiased prediction (BLUP) (Santner, Williams, and Notz 2003), but, as we will see in a while, $\boldsymbol{a}_*$ is the vector which minimizes the Mean squared prediction error (MSPE) of $\widehat{Z}_h(\boldsymbol{s}_0)$.

First of all, we look for the unbiased condition:

$$
\begin{aligned}
&\mathbb{E}\left[\widehat{Z}_h(\boldsymbol{s}_0)\right] = \mathbb{E}\left[\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} - \boldsymbol{a}^T_*\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}^T_*\boldsymbol{z}_l\right] = \\
&\mathbb{E}\left[\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} - \boldsymbol{a}^T_*\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}^T_*\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}^T_*\boldsymbol{1}_{n_h}\widehat{\delta}_0 + \boldsymbol{\delta} + \boldsymbol{\varepsilon}_h\right] = \\
&= \boldsymbol{f}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} + \boldsymbol{a}^T_*\boldsymbol{1}_{n_h}\delta_0 = \mathbb{E}\left[Z_h(\boldsymbol{s}_0)\right] = \boldsymbol{f}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} + \delta_0 \\
&\Longrightarrow \boldsymbol{a}^T_*\boldsymbol{1}_{n_h}\delta_0 = \delta_0 \Longrightarrow \boldsymbol{a}^T_*\boldsymbol{1}_{n_h} = 1 \qquad \text{(A.5)}
\end{aligned}
$$

Then, starting from the definition we calculate the MSPE, or the so called kriging variance

$$
\begin{aligned}
&\mathbb{E}\left[\widehat{Z}_h(\boldsymbol{s}_0) - Z_h(\boldsymbol{s}_0)\right]^2 \\
&= \mathbb{E}\Bigg\{\Bigg[\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} + \boldsymbol{a}^T_*\boldsymbol{\delta} + \underbrace{\boldsymbol{a}^T_*\boldsymbol{1}_{n_h}\delta_0 - \delta_0}_{\overset{(A.5)}{=0}} + \\
&\quad - \boldsymbol{a}^T_*\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}^T_*\boldsymbol{P}\widehat{\boldsymbol{z}}_l + \boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta - \boldsymbol{f}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho} - \delta(\boldsymbol{s}_0) - \varepsilon_\Delta\Bigg]^2\Bigg\} \\
&= \mathbb{E}\Bigg\{\Bigg[\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right) - \boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right) + \boldsymbol{a}^T_*\boldsymbol{\delta} + \boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta + \\
&\quad - \delta(\boldsymbol{s}_0) - \varepsilon_\Delta\Bigg]^2\Bigg\} \\
&= \mathbb{E}\Bigg\{\left(\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\right)^2\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)^2 + \boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)^T\boldsymbol{P}\boldsymbol{a}_* + \\
&+ \boldsymbol{a}^T_*\boldsymbol{\delta}\boldsymbol{\delta}^T\boldsymbol{a}_* + \boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta\boldsymbol{\varepsilon}^T_\Delta\boldsymbol{a}_* + \delta^2(\boldsymbol{s}_0) + \varepsilon^2_\Delta + \\
&- 2\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)^T\boldsymbol{P}\boldsymbol{a} + 2\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)\boldsymbol{a}_*\boldsymbol{\delta} + \\
&+ 2\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)\boldsymbol{a}^T\boldsymbol{\varepsilon}_\Delta - 2\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)\delta(\boldsymbol{s}_0) + \\
&- 2\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\left(\widehat{z}_h(\boldsymbol{s}_0) - z_l(\boldsymbol{s}_0)\right)\varepsilon_\Delta - 2\boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)\boldsymbol{a}^T_*\boldsymbol{\delta} + \\
&+ 2\boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)\delta(\boldsymbol{s}_0) - 2\boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)\boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta + 2\boldsymbol{a}^T_*\boldsymbol{P}\left(\widehat{\boldsymbol{z}}_l - \boldsymbol{z}_h\right)\boldsymbol{\varepsilon}_\Delta + \\
&+ 2\boldsymbol{a}^T_*\boldsymbol{\delta}\boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta - 2\boldsymbol{a}^T_*\boldsymbol{\delta}\delta(\boldsymbol{s}_0) - 2\boldsymbol{a}^T_*\boldsymbol{\delta}\boldsymbol{\varepsilon}_\Delta - 2\boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta\delta(\boldsymbol{s}_0) - 2\boldsymbol{a}^T_*\boldsymbol{\varepsilon}_\Delta\varepsilon_\Delta - 2\delta(\boldsymbol{s}_0)\varepsilon_\Delta\Bigg\} \\
&= \left(\widetilde{\boldsymbol{f}}^T_\Delta(\boldsymbol{s}_0)\boldsymbol{\rho}\right)^2\sigma_{00} + \boldsymbol{a}^T_*\boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P}\boldsymbol{a}_* + \sigma^2_\delta\boldsymbol{a}^T_*\boldsymbol{r}_\delta\boldsymbol{a}_* + \sigma^2_\Delta\boldsymbol{a}^T_*\boldsymbol{a}_* + \sigma^2_\delta + \sigma^2_\Delta - 2\boldsymbol{a}^T_*\boldsymbol{\sigma}_0 \\
&\quad - 2\sigma^2_\delta\boldsymbol{a}^T_*\boldsymbol{r}_\delta
\end{aligned}
$$

$$= \left(\widetilde{\boldsymbol{f}}_{\Delta}^{T}(\boldsymbol{s}_0)\boldsymbol{\rho}\right)^2 \sigma_{00} + \sigma_\delta^2 + \sigma_\Delta^2 + \boldsymbol{a}_*^T \left(\boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P} + \sigma_\delta^2\boldsymbol{r}_\delta + \sigma_\Delta^2\boldsymbol{I}_{n_h}\right)\boldsymbol{a}_* - 2\boldsymbol{a}_*^T(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0)$$
$$= \left(\widetilde{\boldsymbol{f}}_{\Delta}^{T}(\boldsymbol{s}_0)\boldsymbol{\rho}\right)^2 \sigma_{00} + \sigma_\delta^2 + \sigma_\Delta^2 + \boldsymbol{a}_*^T\boldsymbol{\Sigma}_\Delta\boldsymbol{a}_* - 2\boldsymbol{a}_*^T(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0).$$

The MSPE is asymptotically correct due to the fact that $\widehat{\delta}_0$ is a maximum likelihood estimator.

Finally, we want to show that $\boldsymbol{a}_*$ minimizes the MSPE of $\widehat{Z}_h(\boldsymbol{s}_0)$.

The function to be minimized is:

$$\begin{aligned}&\min_{\boldsymbol{a}} f(\boldsymbol{a}) = \min_{\boldsymbol{a}} \boldsymbol{a}^T\boldsymbol{\Sigma}_\Delta\boldsymbol{a} - 2\boldsymbol{a}^T\left(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\right)\\ &\text{subject to:}\\ &\boldsymbol{a}^T\boldsymbol{1}_{n_h} = 1\end{aligned}$$

where we have dropped the terms $\left(\boldsymbol{f}_\Delta^T(\boldsymbol{s}_0)\boldsymbol{\rho}\right)^2\sigma_{00}$, σ_δ^2 and σ_Δ^2, since they are constant with respect to $\boldsymbol{a}$.

This can be rewritten as an unconstrained minimization problem using the Lagrange multiplier:

$$\min_{\boldsymbol{a},\lambda} L(\boldsymbol{a},\lambda) = \min_{\boldsymbol{a},\lambda} \boldsymbol{a}^T\boldsymbol{\Sigma}_\Delta\boldsymbol{a} - 2\boldsymbol{a}^T\left(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0\right) + 2\lambda(\boldsymbol{a}^T\boldsymbol{1}_{n_h} - 1)$$

taking derivatives with respect to $\boldsymbol{a}$ and λ and setting these to zero leads to the system of equations

$$\begin{aligned}\frac{\partial L(\boldsymbol{a},\lambda)}{\partial\boldsymbol{a}} =& 2\boldsymbol{\Sigma}_\Delta\boldsymbol{a} - 2(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0) + 2\lambda\boldsymbol{1}_{n_h}\\ \frac{\partial L(\boldsymbol{a},\lambda)}{\partial\lambda} =& \boldsymbol{a}^T\boldsymbol{1}_{n_h} - 1\end{aligned}$$

The solutions turn out to be:

$$\bar{\boldsymbol{a}} = \left(\boldsymbol{\Sigma}_\Delta^{-1} - \frac{\boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{1}_{n_h}\boldsymbol{1}_{n_h}^T\boldsymbol{\Sigma}_\Delta^{-1}}{\boldsymbol{1}_{n_h}^T\boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{1}_{n_h}}\right)(\sigma_\delta^2\boldsymbol{r}_\delta + \boldsymbol{\sigma}_0) + \frac{\boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{1}_{n_h}}{\boldsymbol{1}_{n_h}^T\boldsymbol{\Sigma}_\Delta^{-1}\boldsymbol{1}_{n_h}}$$

which is equal to the vector $\boldsymbol{a}_*$ of Eq. (A.4) and, since the Hessian is positive definite, MSPE attains its minimum at $\boldsymbol{a}_*$.

Bibliography

Abdel-Salam, A.G., J.B Birch, and W.A. Jensen. 2010. "Nonparametric and semiparametirc mixed models methods for Phase I profile monitoring." Technical Report, Technical Report, Department of Statistics, Virginia Polytechnic Institute & State University.

Alt, F.B. 1985. *Multivariate Quality Control - Encyclopedia of Statistical Sciences.* Volume 6. John Wiley & Sons.

Alt, F.B., and N.D. Smith. 1988. "Multivariate Process Control." *Handbook of Statistics* 7:333–351.

Alwan, L.C. 1999. *Statistical Process Analysis.* McGraw-Hill/Irwin.

Arlot, S., and A. Celisse. 2010. "A survey of cross-validation procedures for model selection." *Statistics Surveys* 4:40–79.

Biegler, L.T. 2010. *Nonlinear Programming.* Society for Industrial and Applied Mathematics.

Bosch, J.A. 1995. *Coordinate Measuring Machines and Systems.* Taylor & Francis.

Carbone, V., M. Carocci, E. Savio, G. Sansoni, and L. De Chiffre. 2001. "Combination of a Vision System and a Coordinate Measuring Machine for the Reverse Engineering of Freeform Surfaces." *The International Journal of Advanced Manufacturing Technology* 17 (4): 263–271.

Colosimo, B.M., P.C. Cicorella, M. Pacella, and M. Blaco. 2014. "From Profile to Surface Monitoring: SPC for Cylindrical Sufaces via Gaussian Process." *Journal of Quality Technology* 46:95–113.

Colosimo, B.M., F. Mammarella, and S. Petrò. 2010. "Quality Control of Manufactured Surfaces." In *Frontiers of Statistical Quality Control.* Springer.

Colosimo, B.M., M. Pacella, and N. Senin. 2014. "Multisensor Data Fusion via Gaussian Process Models for Dimensional and Geometric Verification." *Precision Engineering*, vol. In press.

Cressie, N. 1993. *Statistics for spatial data.* New York: John Wiley & Sons.

Dasarathy, B.V. 1997. "Sensor Fusion Potential Exploitation - Innovative Architectures and Illustrative Applications." *IEEE.*

Del Castillo, E., B.M. Colosimo, and S. Tajbakhsh. 2013. "Geodesic Gaussian Process for the Reconstruction of a Free-Form Surface." *In Pint.*

Etsion, I. 2005. "State of the Art in Laser Surface Texturing." *Journal of Tribology* 127:248–253.

Gaudard, M., M. Karson, E. Linder, and D. Sinhai. 1999. "Bayesian spatial prediction." *Environmental and Ecological Statistics* 6:147–171.

Girao, P.S., J.D. Pereira, and O. Postolache. 2006. Chapter Multisensor Data Fusion and its Application to Decision Making of *Advanced Mathematical & Computational Tools in Metrology VII*, edited by P. Ciarlini, E. Filipe, A.B. Forbes, F. Pavese, C. Perruchet, and B.R.L. Siebert, 47–59. World Scientific Publishing Co. Pte. Ltd.

Guennebaud, Gaël, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Harville, D.A. 1977. "Maximum likelihood approaches to variance component estimation and to related problems." *Journal of the American Statistical Association* 72:320–338.

Hibbit, Karlsson, and Sorensen. 2007. *ABAQUS/Standard Analysis User's Manual.* USA: Hibbit, Karlsson, Sorensen Inc.

Hjelle, Ø. 2001. "Approximation of Scattered Data with Multilevel B-splines." Technical Report, SINTEF.

Hjelle, Ø., and M. Daehlen. 2005. "Multilevel least squares approximation of scattered data over binary triangulations." *Computing and Visualization in Science* 8:83–91.

Huang, D., T.T. Allen, W.I. Notz, and R.A. Miller. 2006a. "Sequential kriging optimization using multiple-fidelity evaluations." *Structural and Multidisciplinary Optimization* 32(5):369–382.

Huang, D., T.T. Allen, W.I. Notz, and N. Zheng. 2006b. "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models." *Journal of Global Optimization* 34:441–466.

Irwin, J.T. 2012. Structured Light Scanning Technology Gives 3D Engineering Solutions Full Range of Tools for Increased Accuracy in 3D Scanning Service.

Isgro, F., F. Odone, and A. Verri. 2005. "An open system for 3d data acquisition from multiple sensor." *Proceedings of the seventh international workshop on Computer Architecture for Machine Perception (CAMP'05).*

ISO. 2013. Geometrical product specifications (GPS) - Surface texture: areal.

Jiang, X., P.J. Scott, D.J. Whitehouse, and L. Blunt. 2007a. "Paradigm shifts in surface metrology Part I. Historical philosophy." *Proc. R. Soc. A* 463:2049–2070.

———. 2007b. "Paradigm shifts in surface metrology. Part II. The current shift." *Proc. R. Soc. A* 463:2071–2099.

Johnson, R.A., and D.W. Wichern. 2007. *Applied Multivariate Statistical Analysis.* Pearson International Edition.

Jones, D., D.R. Schonlau, and W.J. Matthias Welch. 1998. "Efficient global optimization of expensive black-box functions." *Journal of Global optimization* 13:455–492.

Kang, L., and S.L. Albin. 2000. "On-Line Monitoring when the Process Yields a Linear Profile." *Journal of Quality Technology* 32:418–426.

Kennedy, M.C., and A. O'Hagan. 2000. "Predicting the output from a complex compute code when fast approximations are available." *Biometrika* 87 (1): 1–13.

———. 2001. "Bayesian calibration of computer models." *J. R. Statist. Soc.* 63 (3): 425–464.

Krige, D. G. 1951. "A statistical approach to some basic mine valuation problems on the Witwatersrand." *J. of the Chem., Metal. and Mining Soc. of South Africa* 52:119–139.

Leach, R. 2013. *Characterisation of Areal Surface Texture.* Springer.

Lee, S., G. Wolberg, and S.Y. Shin. 1997. "Scattered Data Interpolation with Multilevel B-Spline." *IEE Transactions on Visualization and Computer Graphics* 3:229–244.

Mardia, K.V., and R.J. Marshall. 1984. "Maximum likelihood estimation of models for residual covariance in spatial regression." *Biometrika* 71:135–146.

Maressa, P., L. Anodio, A. Bernasconi, A.G. Demir, and Previtali B. 2014. "Effect of Surface Texture on the Adhesion Performance of Laser Treated Ti6Al4V Alloy." *The Journal of Adhesion* 91:518–537.

Meyer, F. 1994. "Topographic distance and watershed lines." *Signal Processing* 38:113–125.

Mockus, J., V. Tiesis, and A. Zilinskas. 1978. "The application of Bayesian methods for seeking the extremum." *Towards Global Optimization.*

Montgomery, D.C. 2013. *Introduction to statistical quality control.* 7th. Hoboken, NJ: Wiley.

Neuman, H.J., and R. Christoph. 2007. Chapter 7 of *Coordinate Measuring Machines and Systems*, 125–151. CRC Press.

Noorossana, R.R., A. Saghei, and A. Amiri. 2012. *Statistical Analysis of Profile Monitoring.* Wiley.

NPL.

Pandzo, H., S. Mahadevan, M. Bennamoun, and J.A. Williams. 2001. "A 3d acquisition and modeling system." *IEEE international conference on Acoustics, Speech and Signal Processing.*

Park, C., J.Z. Huang, and Y. Ding. 2011. "Domain Decomposition Approach for Fast Gaussian Process Regression of Large Spatial Data Sets." *Journal of Machine Learning Research* 12:1697–1728.

Petrò, S., M. Cavallaro, and G. Moroni. 2009. "Performance evaluation of non contact measuring systems considering bias." *Proceedings of the 4th International Conference on Advanced Research on Rapid Prototyping.*

Qian, P.Z.G, C.C. Seepersad, V.R. Joseph, J.K. Allen, and C.F.J. Wu. 2006. "Building Surrogate Models Based on Detailed and Approximate Simulations." *ASME Journal of Mechanical Design* 128:668–677.

Qian, P.Z.G., and J.C.F. Wu. 2008. "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments." *Technometrics* 50 (2): 192–204.

Ramasami, S.K., J. Raja, and B.D. Boudreau. 2013. "Data Fusion Strategy for Multiscale Surface Measurement." *Journal of Micro and Nano-Manufacturing* 1:1–6.

Rasmussen, C.E., and C.K.I. Williams. 2006. *Gaussian Processes for Machine Learning.* The MIT Press.

Rocchini, C., P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. 2001. "A low cost scanner based on structured light." *Computer Graphics Forum (Eurographics 2001 Conf. Proc.)*, Volume 20. 299–308.

Ruppert, D., M.P. Wand, and R.J. Carrol. 2003. *Semiparametric Regression.* Cambridge University Press.

Salman, N. 2006. "Image segmentation based on watershed and edge detection techniques." *The International Arab Journal of Information Technology.*

Santner, T.J., B. Williams, and W.I. Notz. 2003. *The Design and Analysis of Computer Experiments.* New York: Springer.

Schabenberger, O., and C.S. Gotway. 2005. *Statistical methods for spatial data analysis.* Chapman & Hall/CRC.

Shen, T., J. Huang, and Menq C. 1999. "Multiple-sensor integration for rapid and high-precision coordinate metrology." Edited by IEEE/ASME International Conference, *Advanced Intelligent Mechatronics.* 908–915.

Spall, J.C. 2003. *Introduction to Stochastic Search and Optimization.* Wiley.

Stein, M.L. 1999. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer.

Strano, M., V. Mussi, and M. Monno. 2010. "Non-conventional technologies for the manufacturing of anti-intrusion bars." *International Journal of Material Forming* 3:1111–1114.

Tamas, V., R.M. Ralph, and C. Jordan. 1997. "Reverse engineering of geometric models - an introduction." *Comput Aided Des.* 9 (4): 255–268.

VIM, ISO. 2008. "International vocabulary of basic and general terms in metrology (VIM)." *International Organization.*

Weckenmann, A., X. Jiang, K.D. Sommerc, U. Neuschaefer-Rubec, J. Seewigd, L. Shawa, and T. Estler. 2009. "Multisensor data fusion in dimensional metrology." *CIRP Annals - Manufacturing Technology* 58 (2): 701–721.

Woodall, W.H., D.J. Spitzner, D.C. Montgomery, and S. Gupta. 2004. "Using control charts to monitor process and product quality profiles." *Journal of Quality Technology* 36:309–320.

Xia, H., Y. Ding, and B.K. Mallick. 2011. "Bayesian hierarchical model for combining misaligned two-resolution metrology data." *IIE Transactions* 43 (4): 242–258.

Xia, H., Y. Ding, and J. Wang. 2008. "Gaussian Process model for form error assessment using coordinate measurement." *IEEE Transactions.*

Yuen, S.C.K., and G.N. Nurick. 2008. "The Energy-Absorbing Characteristics of Tubular Structures With Geometric and Material Modifications: An Overview." *Transactions of the ASME: Applied Mechanics Reviews* 61:1–15.

Zarei, H.R., and M. Kroger. 2008. "Bending behavior of empty and foam-filled beams: Structural optimization." *International Journal of Impact Engineering* 35:521–529.

Zimmerman, D.L., and D.A. Harville. 1991. "A Random Field Approach to the Analysis of Field-Plot Experiments and Other Spatial Experiments." *Biometrics* 47 (1): 223–239.

Zsolt, U., L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí. 2007. "Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization." *Journal on Computing* 19:328–340.