

PAPER • OPEN ACCESS

## Mathematical approach to the validation of form removal surface texture software

To cite this article: Luke Todhunter *et al* 2020 *Surf. Topogr.: Metrol. Prop.* **8** 045019

View the [article online](#) for updates and enhancements.

### You may also like

- [Study of Ferrate\(VI\) oxidation for COD removal from wastewater](#)  
Tahir Haneef, Muhammad Raza Ul Mustafa, H.M Farhan Yasin et al.
- [Long-term monitoring of water treatment technology designed for radium removal—removal efficiencies and NORM formation](#)  
Lie Hill, Siiri Suursoo, Madis Kiisk et al.
- [Experimental research of pulsed discharge plasma and TiO<sub>2</sub>/Zeolite coupling technology for formaldehyde removal](#)  
Bingyan Dong and Shuirong Lan



**IOP | ebooks™**

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

# Surface Topography: Metrology and Properties



## PAPER

# Mathematical approach to the validation of form removal surface texture software

### OPEN ACCESS

#### RECEIVED

5 October 2020

#### REVISED



29 October 2020

#### ACCEPTED FOR PUBLICATION

3 November 2020

#### PUBLISHED

10 November 2020

Luke Todhunter<sup>1</sup> , Richard Leach<sup>1</sup>  and François Blateyron<sup>2</sup> 

<sup>1</sup> Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, United Kingdom

<sup>2</sup> Digital Surf, Besançon, France

E-mail: [Luke.Todhunter@nottingham.ac.uk](mailto:Luke.Todhunter@nottingham.ac.uk)

**Keywords:** surface texture, mathematical references, form removal

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



## Abstract

A new approach to the validation of surface texture form removal methods is introduced. A linear algebra technique is presented that obtains total least squares (TLS) model fits for a continuous mathematical surface definition. This model is applicable to both profile and areal form removal, and can be used for a range of form removal models including polynomial and spherical fits. The continuous TLS method enables the creation of mathematically traceable reference pairs suitable for the assessment of form removal algorithms in surface texture analysis software. Multiple example reference pairs are presented and used to assess the performance of four tested surface texture analysis software packages. The results of each software are compared against the mathematical reference, highlighting their strengths and weaknesses.

## 1. Introduction

Surface texture characterisation is a valuable application of metrology in the field of precision engineering [1, 2]. The accurate measurement of part surface topography enables the verification of part tolerances which is crucial to high precision applications [3, 4]. Furthermore, surface texture is a primary factor in part-surface interactions and can impact attributes such as energy efficiency and component lifespan through surface properties, such as lubricant retention [5, 6].

Measured surface height information is used in surface texture analysis to calculate surface texture parameters [1]. These parameters are numerical descriptors that provide information about the statistical distribution of surface heights across the measurement range, as well as frameworks for the identification of topographical features [7, 8]. Obtaining surface texture parameters first requires preparation of the measured surface data. This involves form removal to separate the nominal shape of a measured surface from its texture, and filtration operations to isolate a finite spatial frequency band of interest from the entire measured spatial frequency range [9, 10]. A priori knowledge is required to

determine the nominal form present in the surface measurement data in order to perform effective surface texture analysis that is fit for purpose.

It is important that the surface texture analysis software used to perform each of the operations required for surface texture characterisation is properly validated to prevent systematic errors in the implementation of these operations from affecting analyses. Definitions and guidelines for form removal, filtration and parameter calculation are given in international ISO standards to facilitate meaningful international cross-collaboration in the field of surface texture characterisation by ensuring all are working to the same definitions [7, 11, 12]. National metrology institutes (NMIs) provide reference software and reference datasets as a point of comparison for the validation of surface texture analysis software [13–16]. However, these reference software packages are numerical by nature, developed using similar discrete-based algorithms as used in commercial software and are thus subject to the same approximations. Furthermore, algorithmic implementations of the operations given in the standards can vary, depending on developer, programming language, or interpretation of the standards, leading to variations in the outputs provided by different ‘reference’ software [17]. Baker *et al*

[18] performed an international comparison between sixteen laboratories using two simulated profile datasets along with real test samples and found variations between the software used by each laboratory for all of the seven surface texture parameters tested. Koenders *et al* [19] performed a similar international comparison and found different calculation methods for multiple parameters between software implementations. In 2009, an NPL report by Li *et al* [20] focussed on NMI reference software using type F1 reference data sets [13] and highlighted implementation differences between each of the software packages. Such an outcome makes the value of reference software limited, as it cannot be held as a reliable reference against which commercial software developers can validate their implementations.

These shortcomings in the current state of the art justify the need for an alternative approach to surface texture analysis software validation that is free from the inaccuracies that are introduced with discrete-based reference software. This new approach introduces a mathematical foundation in which the operations of form removal, filtration and parameter calculation are performed on continuous mathematical surface functions in line with the mathematical definitions provided in the international standards. This method ensures mathematical traceability throughout the process and can produce reliable reference pairs for the validation of surface texture analysis software [21].

This paper is part of a series that describes the development of this mathematical reference framework for different stages in surface texture analysis. Previous work has presented the mathematical software validation approach for surface texture field and functional parameter calculation and surface filtration [22–24]. The work introduced here extends the framework to surface texture form removal. A technique is presented that enables the application of a total least squares (TLS) approximation to a continuous surface function while maintaining mathematical traceability. This allows form to be removed from a surface function, revealing surface texture. Section 2 details the mathematical process for obtaining a TLS approximation on a continuous surface. Section 3 showcases the application of this technique to create reference pairs suitable for software validation. Section 4 applies the reference pairs to a range of surface texture analysis software to demonstrate their use in the validation of software performance.

## 2. Continuous total least squares approximation

Form removal is a crucial operation in the surface texture characterisation pipeline that aims to identify the nominal form, or shape, of a measured surface in order to isolate the surface texture. This could be a

slope, curve, cylinder or sphere, depending on the part or surface being measured. Specific guidelines and definitions for form removal operations have historically been vague, with areal standard ISO 25 178-2 describing the F-operation as ‘very fuzzy’ [8]. ISO 3774 gives a little more information, specifying a ‘best fit least squares form’ [11]. The new profile surface texture parameter definition standard currently under development (ISO 21 920-2 [25]), will give more information, the draft currently stating that the F-operation is ‘usually a straight line total least squares fit’, implying this is the choice most commonly taken when using surface texture analysis software. Such software also commonly offers higher order polynomials and circular/spherical form removal options. Following on from these guidelines, it was decided that a TLS fit to remove form on profile and areal continuous surface functions for linear, polynomial and circular shapes would be the most appropriate form removal reference. As the purpose of this paper is to introduce a traceable method to verify form removal software against the definitions given in the ISO specification standards, it is necessary to implement a TLS solution, rather than an ordinary least squares (OLS) solution. For surface texture applications with very small deviations from the horizontal plane, OLS solutions can achieve sufficiently similar results to TLS solutions, but with reduced computational complexity, making it a worthwhile substitution for commercial software.

### 2.1. Background

Previous literature defines the TLS fit for a finite set of points  $\{y_1, \dots, y_m\}$  at given abscissa values  $\{x_1, \dots, x_m\}$  [26, 27]. The aim of the TLS approach is to find the best solution to the linear system

$$X\mathbf{a} \approx \mathbf{y}, \quad (1)$$

where  $X \in \mathbb{R}^{m \times n}$ ,  $\mathbf{a} \in \mathbb{R}^{n \times 1}$  and  $\mathbf{y} \in \mathbb{R}^{m \times 1}$ . Here,  $\mathbf{y}$  is the  $y$  value vector,  $\{y_1, \dots, y_m\}$ , directly. For our purposes, this corresponds to the surface height values along a profile. The value of  $n$  is determined by the complexity of the model being applied to the data, given by the number of variables needed to be found in order to obtain an approximate height value,  $\hat{y}$ . For the example of a straight line fit, with equation  $ax + b$ ,  $n = 2$ . In the case  $m > n$ , the system is overdetermined and there is typically no exact solution to obtain  $\mathbf{a}$ . As  $m$  corresponds to the number of data points and  $n$  the number of variables in the fit model, in practice,  $m \gg n$ .

Suppose  $C \in \mathbb{R}^{m \times n}$ . The theory of singular value decomposition (SVD) [28] determines that  $C$  can be separated and rewritten in the form

$$C = U\Sigma V^T, \quad (2)$$

where

$$U^T U = V V^T = I, \quad (3)$$

with  $I$  being the identity matrix of length  $n$ .  $U$  and  $V$  are built from the orthonormal eigenvectors of  $CC^T$  and  $C^TC$ , respectively.  $\Sigma$  is a diagonal matrix comprised of the square roots of the eigenvalues of  $C^TC$ .

Let

$$C = [X \quad \mathbf{y}] = U\Sigma V^T, \quad (4)$$

given

$$V = \begin{bmatrix} V_{nn} & \mathbf{v}_{nq} \\ \mathbf{v}_{qn} & v_{qq} \end{bmatrix}, \quad (5)$$

where  $V_{nn}$  is an  $n \times n$  matrix,  $\mathbf{v}_{nq}$  and  $\mathbf{v}_{qn}$  are  $n \times 1$  and  $1 \times n$  vectors, respectively, and  $v_{qq}$  is a scalar. The TLS theorem [26, 27] states that a TLS solution to the system is given by

$$\hat{\mathbf{a}} = -\mathbf{v}_{nq}v_{qq}^{-1}. \quad (6)$$

## 2.2. Extension to continuous surfaces

As mentioned, existing literature defines the TLS method for finite datasets of length  $m$ . For the purpose of establishing mathematically traceable reference pairs for form removal, it is necessary to extend this to continuous surface expressions.

Let a continuous surface height function,  $f(x)$ , defined within some range  $x_l \leq x \leq x_h$ , be represented as a dataset of infinite length  $\{f(x_1), \dots, f(x_\infty)\}$  given abscissa values  $\{x_1, \dots, x_\infty\}$ . The matrix  $C$  defined in section 2.1 would, therefore, be of size  $\infty \times (n + 1)$ . In order to obtain a TLS solution, the  $V$  matrix is required. This can be found by calculating  $C^TC$ , which results in a  $(n + 1) \times (n + 1)$  square matrix,  $W$ , the elements of which are comprised of an infinite number of terms found using

$$W_{j,k} = \sum_{n=1}^{\infty} C_{n,j} C_{n,k}. \quad (7)$$

Any element in  $C$  is either from  $X$  or  $\mathbf{y}$ , so is some function of  $x$ . Given that  $x_l \leq x \leq x_h$  and each value  $x_i$  in the dataset can be defined as  $x_i = x_l + idx$ , where  $i$  is an integer and  $dx$  is the interval between them (infinitesimal in this case). Equation (7) can be rewritten as

$$W_{j,k} = \int_{x_l}^{x_h} c_j c_k dx, \quad (8)$$

where  $c_j \in \mathbf{c}$  and  $\mathbf{c}$  is equivalent to the first row of  $C$  with  $x_1$  replaced with  $x$ . This allows a finite size matrix to be defined that is comprised of definite integrals. From here, basic matrix algebra can be used to obtain the eigenvalues and corresponding eigenvectors for the matrix  $C^TC$  by evaluating the definite integrals in each element when required. To construct  $V$ , each eigenvector must also be orthogonalised. Throughout this paper, orthogonalisation was performed using the

Gram-Schmidt process via computer algebra system Mathematica 11.3. The orthogonal eigenvectors are then combined, each becoming one column, to create  $V$ . From here, the TLS solution can be found using equation (6) which defines the form calculated for the given surface function.

Consider the example of performing a linear TLS fit to a surface,  $f(x)$  to obtain a solution of the form  $a_0x + a_1$ . The linear system for this can be written as

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_\infty & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_\infty) \end{bmatrix}. \quad (9)$$

This leads to

$$C = [X \quad \mathbf{y}] = \begin{bmatrix} x_1 & 1 & f(x_1) \\ x_2 & 1 & f(x_2) \\ \vdots & \vdots & \vdots \\ x_\infty & 1 & f(x_\infty) \end{bmatrix}. \quad (10)$$

In calculating  $C^TC$ , the first element is

$$\begin{aligned} W_{1,1} &= x_1x_1 + x_2x_2 \dots \\ &= \sum_{i=1}^{\infty} x_i^2, \end{aligned} \quad (11)$$

which, following the process for equation (8), becomes

$$W_{1,1} = \int_{x_l}^{x_h} x^2 dx. \quad (12)$$

The same process can be applied for each element, leading to the matrix

$$C^TC = \begin{bmatrix} \int_{x_l}^{x_h} x^2 dx & \int_{x_l}^{x_h} x dx & \int_{x_l}^{x_h} xf(x) dx \\ \int_{x_l}^{x_h} x dx & \int_{x_l}^{x_h} 1 dx & \int_{x_l}^{x_h} f(x) dx \\ \int_{x_l}^{x_h} xf(x) dx & \int_{x_l}^{x_h} f(x) dx & \int_{x_l}^{x_h} f(x)^2 dx \end{bmatrix}. \quad (13)$$

From here, the orthogonalised eigenvectors can be obtained following evaluation of the definite integrals.

The value in this approach is that the process remains the same for a wide variety of linear algebra systems, allowing for a range of mathematically traceable form types to be obtained. For example, a 2nd order polynomial TLS fit can be represented by the system

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_\infty^2 & x_\infty & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_\infty) \end{bmatrix}, \quad (14)$$

and, following the same process as above,

$$C^T C = \begin{bmatrix} \int_{x_l}^{x_h} x^4 dx & \int_{x_l}^{x_h} x^3 dx & \int_{x_l}^{x_h} x^2 dx & \int_{x_l}^{x_h} x^2 f(x) dx \\ \int_{x_l}^{x_h} x^3 dx & \int_{x_l}^{x_h} x^2 dx & \int_{x_l}^{x_h} x dx & \int_{x_l}^{x_h} x f(x) dx \\ \int_{x_l}^{x_h} x^2 dx & \int_{x_l}^{x_h} x dx & \int_{x_l}^{x_h} 1 dx & \int_{x_l}^{x_h} f(x) dx \\ \int_{x_l}^{x_h} x^2 f(x) dx & \int_{x_l}^{x_h} x f(x) dx & \int_{x_l}^{x_h} f(x) dx & \int_{x_l}^{x_h} f(x)^2 dx \end{bmatrix}. \quad (15)$$

This technique is also easily expandable to areal form removal, dealing with surface functions of the form  $f(x, y)$ . The process is the same, however, elements in the matrices can now be a function of both  $x$  and  $y$ . A two-dimensional 2nd order polynomial fit to an areal surface can be represented by the system

$$\begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^2 & x_1 y_\infty & y_\infty^2 & x_1 & y_\infty & 1 \\ x_2^2 & x_2 y_1 & y_1^2 & x_2 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_\infty^2 & x_\infty y_\infty & y_\infty^2 & x_\infty & y_\infty & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} \approx \begin{bmatrix} f(x_1, y_1) \\ f(x_1, y_2) \\ \vdots \\ f(x_1, y_\infty) \\ f(x_2, y_1) \\ \vdots \\ f(x_\infty, y_\infty) \end{bmatrix}. \quad (16)$$

Note here that each combination of  $x$  and  $y$  values must occur, so each value of  $x$  must have a row of terms with all of the infinite values of  $y$ . That is,

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} x_i y_j. \quad (17)$$

Subsequently, when converting to an integral as in equation (8), the result is a double integral over the full area of the measurement. Therefore, a matrix for  $C^T C$  can be obtained of the form

$$C^T C = \begin{bmatrix} \int_A x^4 dA & \int_A x^3 y dA & \int_A x^2 y^2 dA & \int_A x^3 dA & \int_A x^2 y dA & \int_A x^2 dA & \int_A x^2 f_{xy} dA \\ \int_A x^3 y dA & \int_A x^2 y^2 dA & \int_A x y^3 dA & \int_A x^2 y dA & \int_A x y^2 dA & \int_A x y dA & \int_A x y f_{xy} dA \\ \int_A x^2 y^2 dA & \int_A x y^3 dA & \int_A y^4 dA & \int_A x y^2 dA & \int_A y^3 dA & \int_A y^2 dA & \int_A y^2 f_{xy} dA \\ \int_A x^3 dA & \int_A x^2 y dA & \int_A x y^2 dA & \int_A x^2 dA & \int_A x y dA & \int_A x dA & \int_A x f_{xy} dA \\ \int_A x^2 y dA & \int_A x y^2 dA & \int_A y^3 dA & \int_A x y dA & \int_A y^2 dA & \int_A y dA & \int_A y f_{xy} dA \\ \int_A x^2 dA & \int_A x y dA & \int_A y^2 dA & \int_A x dA & \int_A y dA & \int_A 1 dA & \int_A f_{xy} dA \\ \int_A x^2 f_{xy} dA & \int_A x y f_{xy} dA & \int_A y^2 f_{xy} dA & \int_A x f_{xy} dA & \int_A y f_{xy} dA & \int_A f_{xy} dA & \int_A f_{xy}^2 dA \end{bmatrix}, \quad (18)$$

where  $f_{xy} = f(x, y)$  and

$$\int_A dA = \int_{y_l}^{y_h} \int_{x_l}^{x_h} dx dy \quad (19)$$

for brevity.

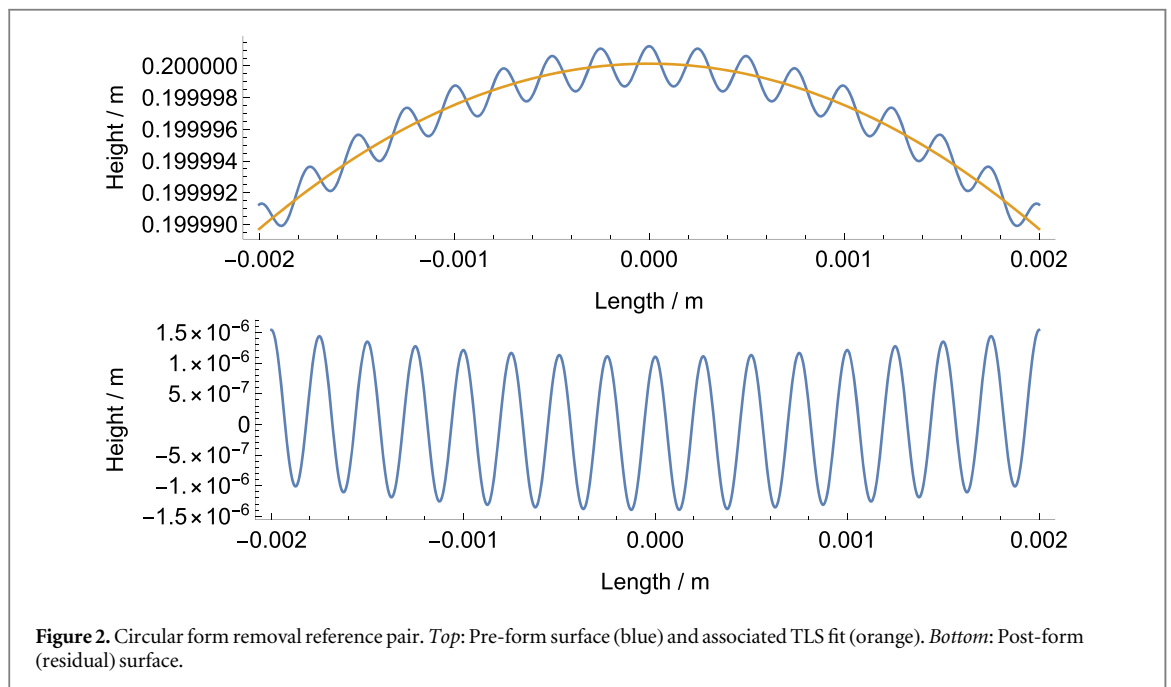
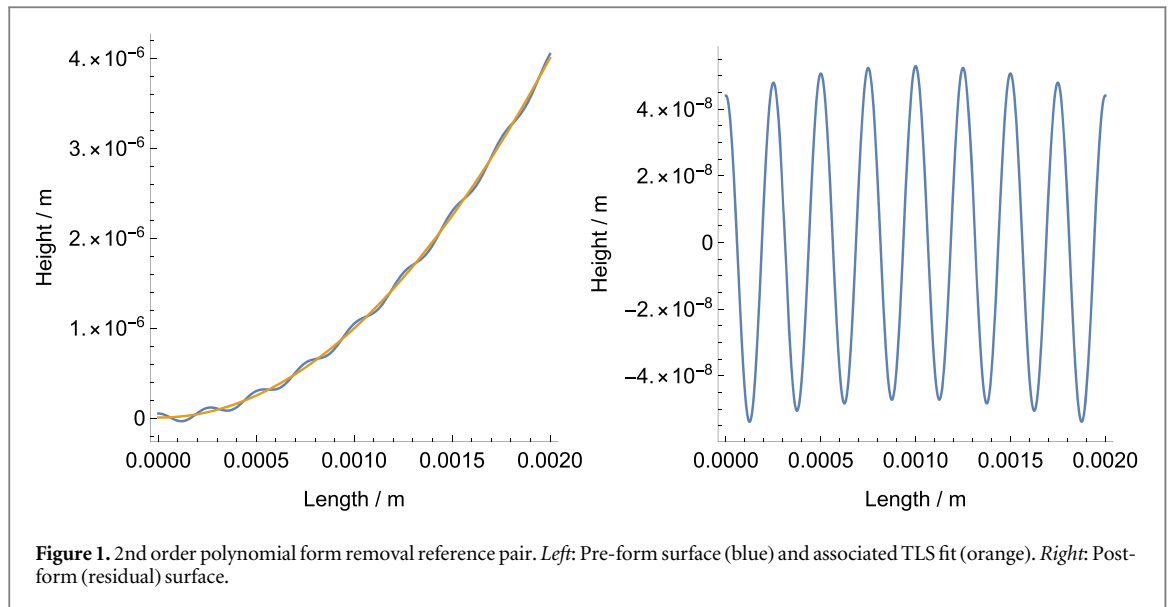
### 3. Form removal reference pairs

The creation of form removal reference pairs suitable for the assessment of surface texture analysis software comprises three steps: defining a pre-form surface function,  $f(x)$ , calculating the TLS form expression,  $TLS(x)$ , using the method presented in section 2.2 and removing the TLS form from the original surface function to obtain a post-form surface,  $z(x)$ . For functions defining surface height with position, as used here, form removal can be performed by simple subtraction,

$$z(x) = f(x) - TLS(x). \quad (20)$$

For the reference pairs used throughout this paper, pre-form surface functions have been defined as a superposition of a low-order polynomial form and a high spatial-frequency sinusoid. This has the effect of adding an identifiable texture to a nominal form.

Figure 1 shows an example profile reference pair for 2nd order polynomial form removal. The pre-form surface function is an  $x^2$  function combined with a cosine with a frequency of  $4000 \text{ m}^{-1}$ , defined in the

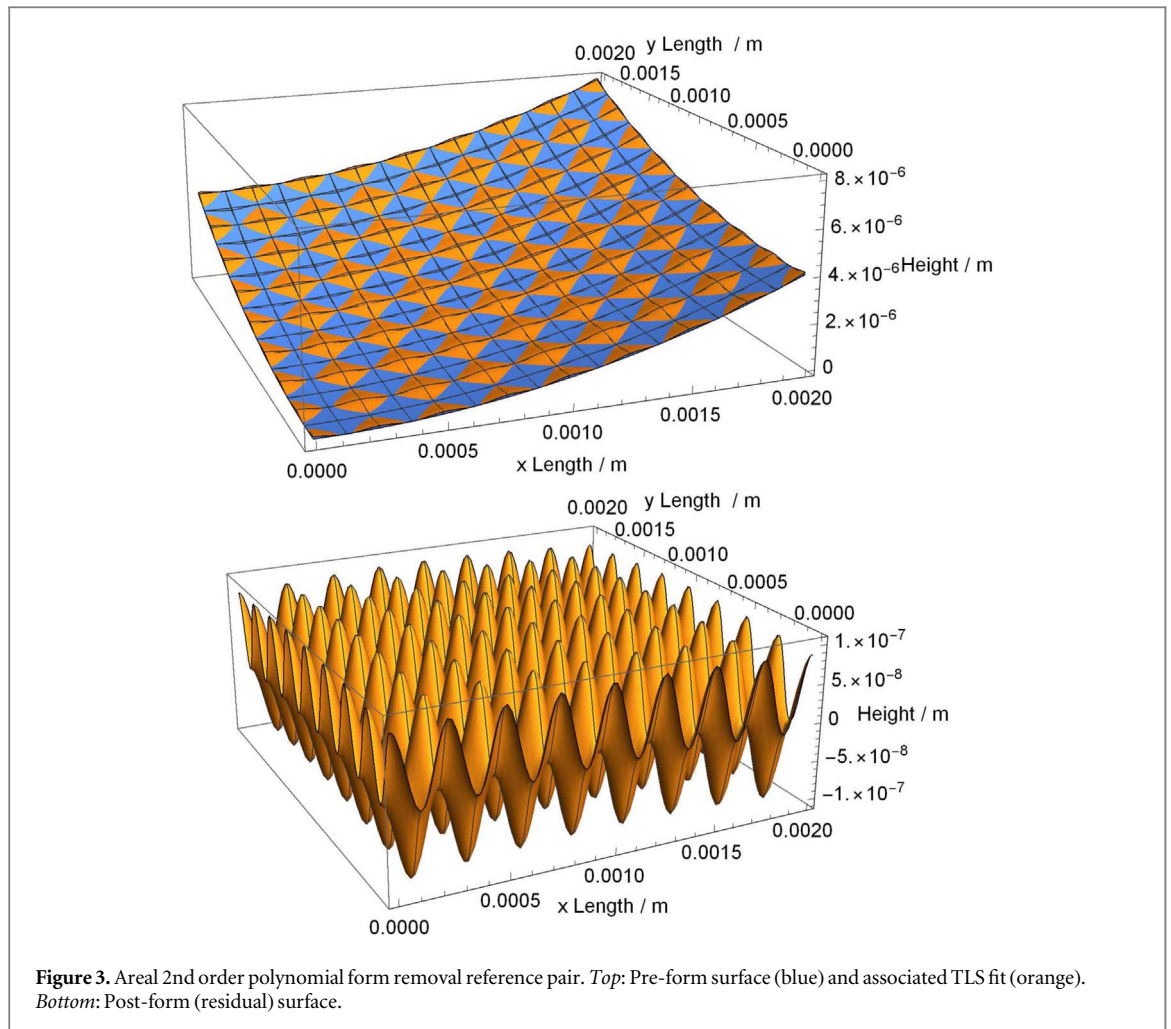


range  $0 \text{ mm} \leq x \leq 2 \text{ mm}$ . Figure 1 shows the pre-form surface in blue and the TLS fit in orange on the right. The right-hand side shows the result of subtracting the TLS form from the pre-form surface. Note that the post-form surface is not a perfect cosine and presents a slight curve, indicating presence of residual form in the final surface. This is because the amplitude of the cosine is included in the TLS fit and affects the minimisation algorithm, preventing the approximation from ignoring it entirely and only extracting the underlying  $x^2$  function. The cosine is a function of  $x$  and has been added vertically to the underlying form. Because of this, there is not an equal amount of positive and negative cosine function perpendicular to the underlying form, causing into the effect the final TLS fit. This effect is reduced as the amplitude of the

texture is reduced relative to the amplitude of the form. Figures 2 and 3 show example reference pairs for a circular TLS form removal process and an areal 2nd order polynomial TLS form removal, respectively.

#### 4. Software comparisons

To use the mathematical reference pairs to assess the performance of form removal software, it is necessary to produce discrete surface datasets that can be input into the software under test. This is achieved by sampling the surface functions  $f(x)$  and  $z(x)$  (or  $f(x, y)$  and  $z(x, y)$  in the areal case) at a series of uniformly spaced points to obtain a dataset of height values. These height values are then stored in a compatible file format, such as the .SDF format defined in ISO 25



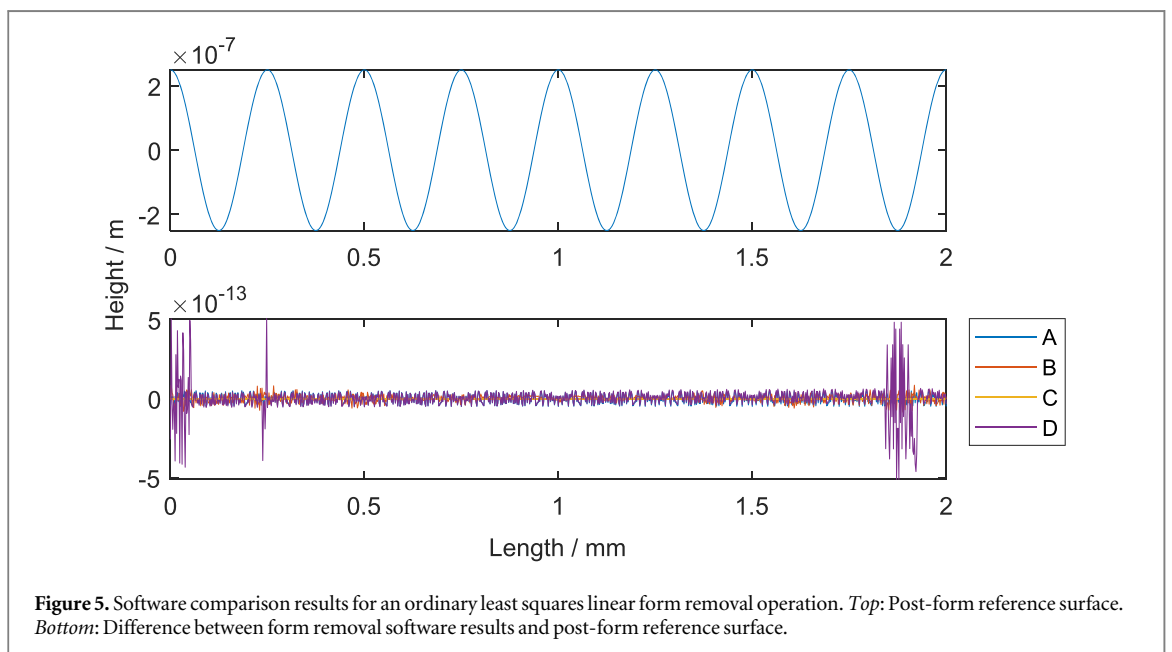
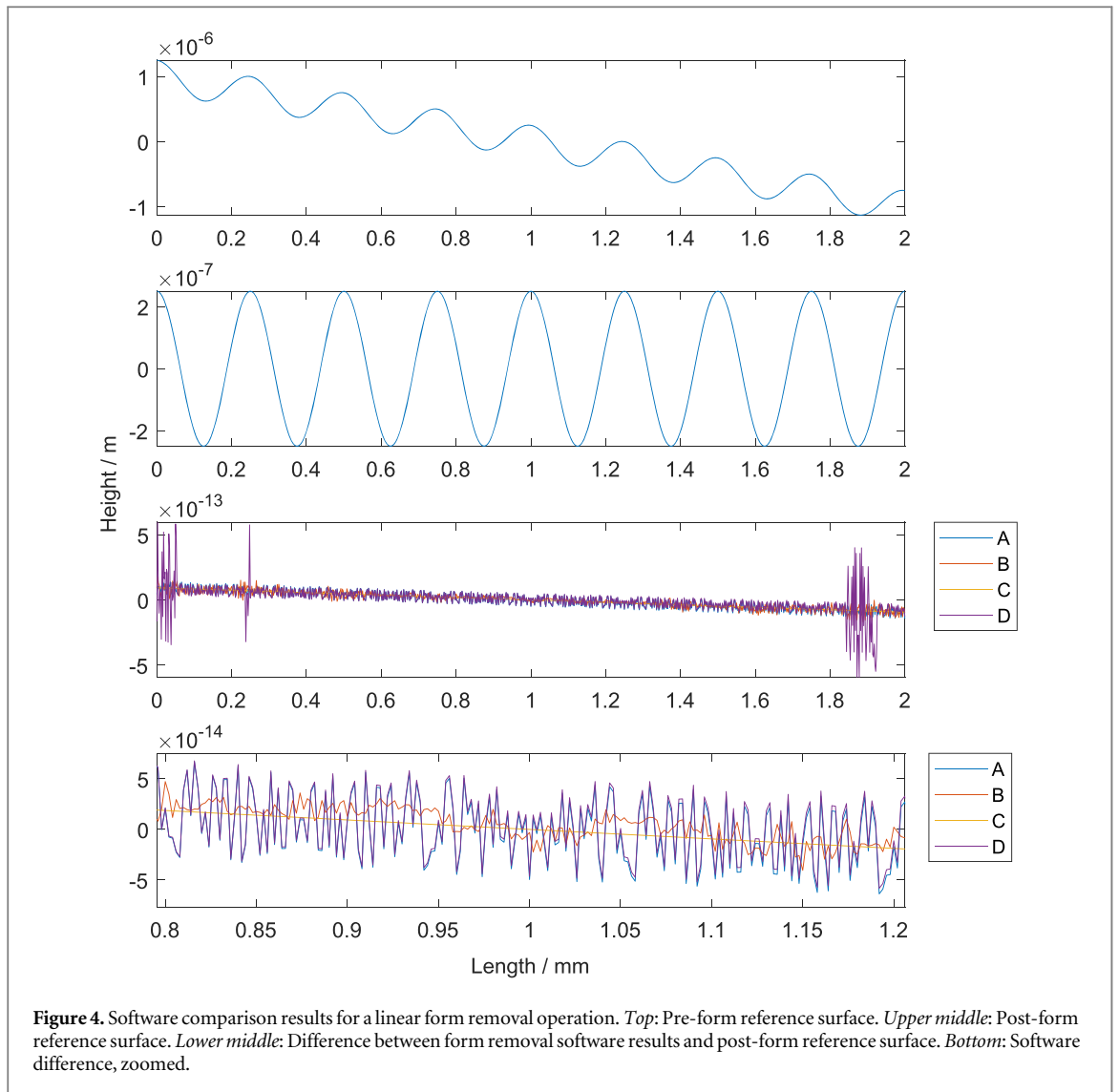
178-71 [29]. For the comparisons presented in this paper, all profile datasets were sampled at a resolution of 1001 pixels, and all areal datasets were sampled at a resolution of  $701 \times 701$ .

Mathematical form removal reference pairs were used to assess the performance of four surface texture analysis software packages, including both commercial and reference software. Each pre-form reference surface was input into the software under test, which was then used to perform the corresponding form removal operation. The resulting surface was exported and the post-form reference surface heights were then subtracted from the software results to obtain surface height difference datasets.

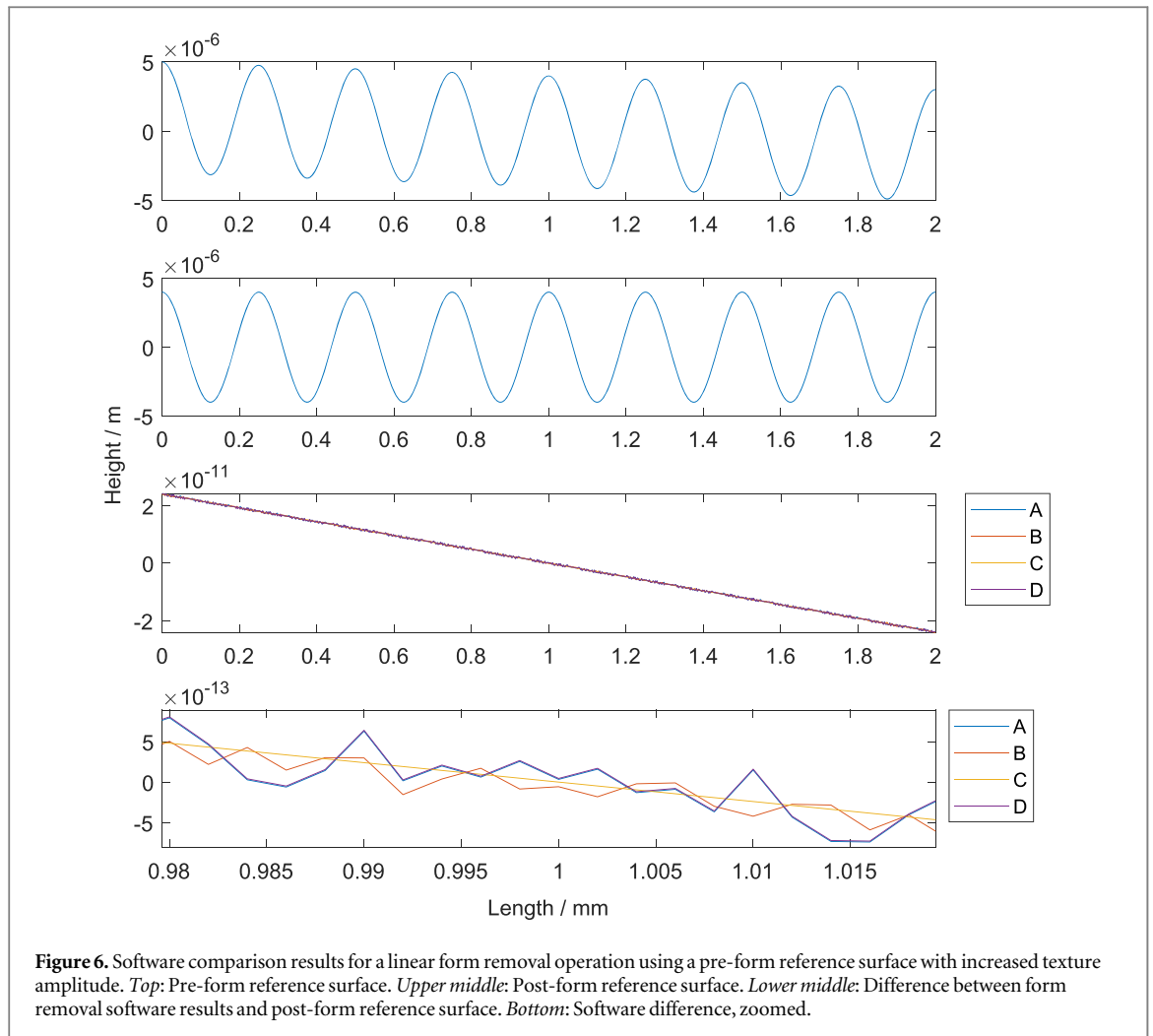
Figure 4 shows the software obtained results for a linear TLS form removal operation, subtracted from the post-form reference surface. All tested software shows good agreement with the reference, with deviation occurring on the order of the dataset precision. Software D displays small regions of larger disagreement with the reference at either ends of the profile. This could be due to an end effect in their form removal algorithm, however, the return to a small deviation from reference at the rightmost edge of the profile suggests this may not be the case and the effect is instead caused by some other aspect of their

software. Upon closer inspection when cropping in, it can be seen that software A and D both perform almost identically for the majority of the profile, suggesting a similar algorithm is used by both. While most of the software tested displays noise-like variation at small scales, software C presents a small difference from the reference across the entirety of the profile. This implies some form of averaging or interpolation to account for the small-scale variations in the results, which may arise from the precision of the data used during operation.

All tested software displayed a consistent linear slope to the difference results, indicating a variation in the gradient of the form obtained by the software compared to that of the reference surface. To investigate, an ordinary least squares (OLS) mathematical form removal was calculated for the same surface. This surface was then compared to the software results, as shown in figure 5. This shows no linear slope as seen in the comparison between the software and mathematical TLS form removal surface results, strongly indicating that all test software packages have implemented an OLS form removal algorithm instead of a TLS form removal algorithm, even NMI reference software, despite the guideline given in the ISO specification standards. This result highlights the importance of full







disclosure by the NMI reference software providers of the form removal algorithms implemented, as variations from the methods mentioned in the ISO specification standards can lead to differences. For commercial software, implementation details are industrial secrets and so are not expected to be disclosed. However, deviations from reference values, such as those presented here, can be disclosed, in the same way that surface measurement instrument manufacturers state noise level on a data sheet. This effect is even more pronounced when the amplitude of the texture on the surface relative to the form is increased and the differences between the outputs of TLS and OLS algorithms become more significant. This effect is shown in figure 6, which presents a surface with the same underlying linear form but with the sinusoidal texture amplitude increased by a factor of four. Here, the gradient present in the software difference results is several orders of magnitude larger than in figure 4.

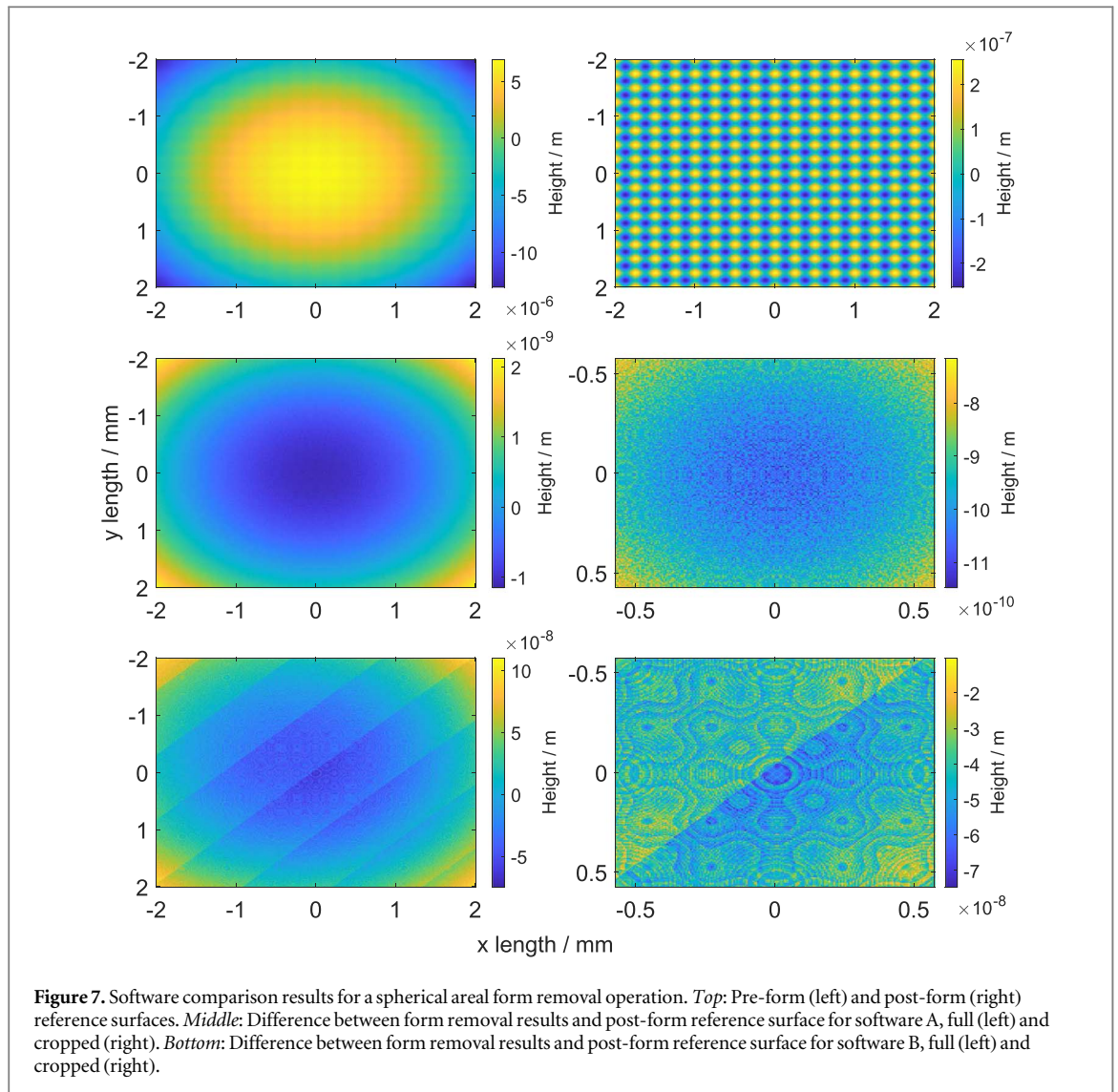
Figure 7 presents the results for an areal spherical TLS form removal operation. Only software A and B gave the option to perform spherical form removal. Both software packages show reasonable agreement with the post-form reference surface, however, clear evidence of the sphere is present on both images,

indicating an imperfect spherical fit to the data when compared to the TLS approach. In addition to a slightly higher amplitude of deviation, software B also presents a regular diagonal striation across the surface. This is some form of artefact due to the form removal algorithm applied by the software, as no presence of such striations are found in the pre-form reference surface. Cropping closer into the centre, software B also presents interesting radial patterns; again, artefacts of the form removal algorithms used.

## 5. Conclusions

The methods presented in this paper enable the calculation of TLS solutions for mathematically traceable surfaces to facilitate the creation of reference pairs for the assessment of form removal algorithms. The methods are applicable to both profile and areal surfaces and can be used for a variety of linear form equations.

The technique was applied to assess four surface texture analysis software packages. The results found generally good agreement with the references, however, they revealed the use of OLS form fitting rather



than the ISO specified TLS form fitting. This is likely due to the increased simplicity of the OLS method over TLS, allowing for both speed benefits and ease of implementation. Alternatively, this could be due to a misinterpretation of the standards, as earlier standards are less clear about the type for least-squares fitting to be used. Another important consideration is that explicit mention of TLS is only given in the most recent standard, meaning software developers may not have yet been able to update their algorithms. Nevertheless, it is important that the deviations of each software from the reference values are disclosed to users to ensure transparency and awareness of how the results of the software may differ from ISO recommendations. In surface texture measurement data, it is typical that the surface form and lateral range are orders of magnitude higher than the vertical topography height,  $z$ . The effects of texture variations are small relative to the lateral range, meaning the effect of including lateral variations in a form fitting algorithm by using a TLS method may be negligible compared to the results of a simpler OLS method. A possible avenue for future

work on this topic is to investigate the effect of TLS versus OLS numerical form removal methods on real measurement data to understand where the differences between the two approaches are significant in practice. Such an investigation could be valuable in justifying the use of the simpler OLS form removal methods in certain scenarios for commercial software.

A limitation of this new approach is its relative complexity compared to traditional discrete dataset TLS algorithms. This complexity increases as the order of the form equation increases, potentially leading to prohibitive scenarios for very high-order form fitting requirements. However, typical form removal operations use relatively simple models, ensuring this new approach is still of benefit to the industry. A further limitation is the disconnect between a continuously defined reference pair and discrete-based software. For assessment of the software to occur, appropriate sampling of the reference pair is required. While this has the benefit of enabling a reference dataset of theoretically infinite resolution to be produced, at practical resolutions there will be a disparity between the

reference and the dataset due to the information lost by finite sampling. To account for this, it is important that the sampling method and resolution are stated alongside any software assessment using a continuous original reference.

This paper is the final part in a series that introduces the use of continuous surface topography definitions to obtain mathematically traceable references for the assessment of surface texture analysis software. By utilising the methods presented in this series, the full surface texture characterisation pipeline, including form removal, surface filtration and surface texture parameter calculation, can be validated against a mathematically traceable reference pair.

## Acknowledgments

We would like to acknowledge funding from Digital Surf and the EPSRC Nottingham Impact Accelerator. Thanks also to colleagues in ISO technical committee 213 working group 16 for continuous discussions on these matters.

## Data accessibility

Due to confidentiality agreements with research collaborators, supporting data can only be made available to researchers subject to a non-disclosure agreement on an individual basis. Please visit <https://doi.org/10.5281/zenodo.4063716> to request access.

## ORCID iDs

Luke Todhunter  <https://orcid.org/0000-0003-4767-3088>

Richard Leach  <https://orcid.org/0000-0001-5777-067X>

François Blateyron  <https://orcid.org/0000-0001-7048-4045>

## References

- [1] Leach R K 2014 *Fundamental Principles of Engineering Nanometrology* 2nd edn (Binghamton: Elsevier Science)
- [2] Brinksmeier E, Karpuschewski B, Yan J, Riemer O and Schoenemann L 2020 Manufacturing of multiscale structured surfaces *Ann. CIRP* **69** 717–39
- [3] Bruzzone A A G, Costa H L, Lonardo P M and Lucca D A 2008 Advances in engineered surfaces for functional performance *Ann. CIRP* **57** 750–69
- [4] Brown C A et al 2018 Multiscale analyses and characterizations of surface topographies *Ann. CIRP* **67** 839–62
- [5] Holmberg K and Erdemir A 2017 Influence of tribology on global energy consumption, costs and emissions *Friction* **5** 263–84
- [6] Menezes P L et al 2011 Role of surface texture, roughness, and hardness on friction during unidirectional sliding *Tribol. Lett.* **41** 1–15
- [7] ISO 4287 1998 *Geometrical Product Specifications (GPS) - Surface Texture: Profile Method—Terms, Definitions and Surface Texture Parameters* (Geneva: International Organisation for Standardisation)
- [8] ISO 25 178-2 2012 *Geometrical Product Specifications (GPS) - Surface Texture: Areal Part 2: Terms, Definitions and Surface Texture Parameters* (Geneva: International Organisation for Standardisation)
- [9] Forbes A B 2013 Areal form removal *Characterisation of Areal Surface Texture* ed R K Leach (Berlin: Springer) pp 107–28
- [10] Seewig J 2013 Areal filtering methods *Characterisation of Areal Surface Texture* ed R K Leach (Berlin: Springer) pp 67–106
- [11] ISO 3274 1996 *Geometrical Product Specifications (GPS) - Surface Texture: Profile Method—Nominal Characteristics of Contact (stylus) Instruments* (Geneva: International Organisation for Standardisation)
- [12] ISO 16 610-21 2011 *Geometrical Product Specifications (GPS) - Filtration—Part 21: Linear Profile filters: Gaussian Filters* (Geneva: International Organisation for Standardisation)
- [13] ISO 5436-2 2012 *Geometrical Product Specifications (GPS) - Surface Texture: Profile Method; Measurement Standards—Part 2: Software Measurement Standards* (Geneva: International Organisation for Standardisation)
- [14] Blunt L, Jiang X, Leach R K, Harris P and Scott P 2008 The development of user-friendly software measurement standards for surface topography software assessment *Wear* **264** 389–93
- [15] Bui S H, Renegar T B, Vorburger T V, Raja J and Malburg M C 2004 Internet-based surface metrology algorithm testing system *Wear* **257** 1213–8
- [16] Jung L, Krüger-Sehm R, Spranger B and Koenders L 2004 Reference software for roughness analyses—features and results *XI International Colloquium on Surfaces (Chemnitz, Germany)* p 164
- [17] Todhunter L D, Leach R K, Lawes S D A and Blateyron F 2017 An analysis of type F2 software measurement standards for profile surface texture parameters *Meas. Sci. Technol.* **28** 065017
- [18] Baker A et al 2013 Final report on apmp.l-k8: International comparison of surface roughness *Metrologia* **50** 04003
- [19] Koenders L, Andreasen J L, De Chiffre L, Jung L and Krüger-Sehm R 2004 Euromet project 600—comparison on surface roughness standards *Metrologia* **41** 04001
- [20] Li T, Leach R K, Jung L, Jiang X and Blunt L 2009 Comparison of type f2 software measurement standards for surface texture *NPL Report ENG 16*
- [21] Linares J, Goch G, Forbes A, Spraul J, Clément A, Haertig F and Gao W 2018 Modelling and traceability for computationally-intensive precision engineering and metrology *CIRP Ann.* **67** 815–38
- [22] Todhunter L D, Leach R K, Lawes S, Harris P M and Blateyron F 2019 Mathematical approach to the validation of functional surface texture parameter software *Surface Topography: Metrology and Properties* **7** 015020
- [23] Todhunter L D, Leach R K, Lawes S, Harris P M and Blateyron F 2020 Mathematical approach to the validation of field surface texture parameter software *Surface Topography: Metrology and Properties* **8** 015010
- [24] Todhunter L, Leach R K and Blateyron F 2020 Mathematical approach to the validation of surface texture filtration software *Surface Topography: Metrology and Properties* **8** 4
- [25] ISO/DIS 21 920-2 2020 *Geometrical Product Specifications (GPS) - Surface Texture: Profile—Part 22: Terms, Definitions and Surface Texture Parameters* (Geneva: International Organisation for Standardisation)
- [26] Golub G H and Van Loan C F 1980 An analysis of the total least squares problem *SIAM J. Numer. Anal.* **17** 883–93
- [27] Markovsky I and Van Huffel S 2007 Overview of total least-squares methods *Signal Process.* **87** 2283–302
- [28] Golub G H and Reinsch C 1971 Singular value decomposition and least squares solutions *Linear Algebra* (Berlin: Springer) pp 134–51
- [29] ISO 25 178-71 2012 *Geometrical Product Specifications (GPS) - Surface texture: Areal—Part 71: Software Measurement Standards* (Geneva: International Organisation for Standardisation)