Surface Topography: Metrology and Properties

PAPER

Crossing-The-Line Segmentation as a Basis for $R_{Sm}$ and $R_c$ Evaluation

J Seewig¹, P J Scott², M Eifler³, B Barwick⁴ and D Hüser⁵

¹ Technische Universität Kaiserslautern, Institute for Measurement and Sensor-Technology, Germany
² University of Huddersfield, Centre for Precision Technologies, United Kingdom
³ Physikalisch-Technische Bundesanstalt Braunschweig, Department 5 Precision Engineering, Germany
E-mail: seewig@mv.uni-kl.de
Keywords: surface texture, feature characterization, segmentation, zero crossing

Abstract
Feature characterization of rough surfaces is of growing interest in terms of a function oriented description of technical surfaces. Feature characterization requires a segmentation of significant hills and dales of the measured profile. The segmentation can be done in several ways. One method is the so called crossing-the-line segmentation which will be part of ISO 16610 part 45 and ISO 21920 part 2. The crossing-the-line segmentation described in this publication represents an extension of the algorithm proposed by Scott (Scott P, 2006, Meas. Sci. Technol. 17, 559–564) and is based on new knowledge gathered over the last ten years. As an example, the feature parameters $R_{Sm}$ and $R_c$ according to ISO 4287:1997 are evaluated.

1. Introduction

The characterization of profile surface texture with corresponding parameters is described in the standard ISO 4287:1997 [1]. The standard includes not only the well-known amplitude parameters $R_a$, $R_q$, $R_z$ but also the mean width of all profile elements $R_{Sm}$ based on the segmentation in individual profile features for a lateral characterization [1] (see [2, 3] for an historical overview). However the current $R_{Sm}$ parameter definition in ISO 4287:1997 has revealed some issues. First, the $R_{Sm}$ parameter is defined as the mean width of profile elements, though it is intended to be a spacing parameter ($R_{Sm}$ = roughness spacing mean). This does not affect the evaluation as long as there are no portions in the profile that are neither hill nor dale. However such zero elements can be part of real profiles and this issue has not yet been addressed in the current definition. Furthermore, current recommendations in ISO 4287:1997 regarding detection thresholds cause problems when handling several profiles, as will be shown in section 2.3. Finally the current description lacks further instructions or program flow charts for an unambiguous implementation. This leads to results of different measuring instruments and evaluation routines not being comparable. For example even the direction of evaluation can lead to different results in profile feature spacing, as later illustrated in section 3.1.

Feature characterization is a powerful tool in terms of a function oriented description of technical surfaces. The corresponding analysis methods have been researched extensively for areal surface texture evaluation (see [4] for an overview) and were included in the corresponding standardization in the meantime [5, 6]. For areal feature characterization, the watershed transformation is commonly applied to segment hills and dales [3]. The method was extended towards an application in the GPS system amongst others by Wolf [7] and Scott [8]. Since the corresponding instructions have been published in the standard ISO 25178-2 in 2012 [3] they have also been developed further [9]. In the field of areal metrology, feature-based analysis is used for example for the characterization of additively manufactured surfaces [10].

It can be already predicted that feature-based evaluation methods will have a growing incidence in the future standardization: in the future profile standard ISO 21920 which is currently being developed by WG 16 of ISO TC 213 under participation of some of the authors, a paradigm shift will be included. Most of the relevant profile surface texture parameters will be defined with the aid of profile hills and profile dales rather than the spatial height values. This means, that
the feature-based analysis will be the default case in future profile surface texture evaluation. As a result, it is inevitable that the feature-based characterization of rough surfaces follows rigorously defined methods.

Feature characterization techniques require a segmentation of significant geometrical features. In the profile case, such geometrical features are e.g. hills and dales (peaks and valleys in ISO 4287:1997). Concerning the x-axis of a measured profile as a reference line for profile evaluation, hills are typically above the reference line and dales are typically below the reference line. In order to separate hills and dales the zero crossings of the profile with the reference line have to be calculated. This technique is called crossing-the-line segmentation and has been intensely discussed and developed in the WG 16 of ISO TC 213 during the last decade.

In this paper, we describe an unambiguous implementation of this type of segmentation, which will consequently also be included in the profile standard ISO 21920. We suggest a determination of profile elements based on the crossing-the-line segmentation in order to ensure a comparability of feature-based evaluations. The algorithm detects hills and dales with regard to distinct mathematical criteria, applies a vertical threshold to eliminate insignificant features, merges adjacent hills and dales and finally determines e.g. the parameter PSm, RSm, WSm based on all profile elements. The objective is the specification of an unambiguous implementation and to address the challenges of current implementations following ISO 4287:1997. The main aspects of the new algorithm can be summarized as follows: robust identification of zero crossings, definition of a zero element (points lying on the reference line) which can be assigned neither to a hill nor dale, two different vertical limits for hills and dales. The algorithm leads to an unambiguous parameter evaluation and is simple to implement. Besides a detailed analysis of the described profile features, a program flow chart and finally implementations both in Matlab and Python will be given to show the ease of use of the proposed algorithm.

2. Crossing-the-line segmentation of profile features

In this paper a profile feature is defined as a specific portion of a profile with a specific geometrical property. The task of the crossing-the-line segmentation is to separate different profile features with the aim to enable a feature based characterization in a next step. As an example, the number of features or the mean width of features are simple values for a feature based characterization. The first type of feature is called a profile hill. A profile hill is usually an outwardly directed (from material to surrounding medium) portion of the profile connecting two adjacent intersection points of the profile and the reference line (x-axis). The second type of feature is called a profile dale. Contrary to a profile hill, a profile dale is usually an inwardly directed (from surrounding medium to material) portion of the assessed profile connecting two adjacent intersection points of the assessed profile and the reference line. The third type of feature is called a zero element. A zero element is usually a continuous profile portion on the reference line. The reference line is usually defined by the corresponding ISO standard for a given type of profile, rather than being of arbitrary choice. For example the R-profile that is used for the calculation of R-parameters (like RSm) is derived by applying a λ, low pass filter, subtracting the nominal shape and applying a λ, high pass filter to eliminate waviness in the profile. The resulting R-profile according to ISO 4287-1997 therefore has a reference line equal to zero. Because of this, with no loss of generality the reference line is set to be equal to the zero line in this manuscript. However the following algorithm can be applied to any given profile with a corresponding reference line unequal zero by simply subtracting this reference line from the profile. In order to carry out the segmentation, the crossing-the-line algorithm is separated into three calculation steps:

- detection of profile features by zero crossing,
- application of a vertical limit criterion in order to delete insignificant profile features,
- merging of profile features of the same type.

2.1. Definitions

To simplify the implementation of the segmentation procedure two function calls are used. The first function is the modified signum function which calculates the sign of a real number \( z \in \mathbb{R} \) depending on the positive real numbers \( u, l \in \mathbb{R}_0^+ \):

\[
\text{sgm}(z, l, u) = \begin{cases} 
1 & \text{if } z \geq u \\
-1 & \text{if } z \leq -l \\
0 & \text{otherwise}
\end{cases}
\]

The second function is the root function which calculates the intersection of the profile with the reference line by linear interpolation:

\[
\text{root}(x_a, z_{a0}, x_b, z_b) = \begin{cases} 
\frac{(x_a + x_b)}{2} & \text{if } z_b = z_a \\
\min\left(\max\left(\frac{(x_a \cdot z_b - x_b \cdot z_a)}{(z_b - z_a)}, x_a\right), x_b\right) & \text{otherwise}
\end{cases}
\]
with \( x_0, z_0 \in \mathbb{R} \) and \( x_0, z_0 \in \mathbb{R} \) as the coordinates of the profile whose linear connection intersects the reference line. If the intersection point lies outside the interval \([x_0, x_k]\), then the associated interval limit is used instead of the intersection point. Throughout the paper a discrete representation \((x_k, z_k)\) of the continuous profile \(z(x)\) is used. It is assumed that the values \(x_k\) are sorted in ascending order. The following variables are defined:

- \( n \) number of profile samples,
- \( x_k \) position on the reference line (in ascending order) with \( k = \{1, 2, \ldots, n\}\),
- \( z_k \) profile value for a given \( x_k \) with \( k = \{1, 2, \ldots, n\}\),
- \( k_{\text{b}} \) index of the profile value indicating a profile feature height,
- \( n_{\text{PF}} \) total number of profile features,
- \( PF_i \) profile feature with four members \( i = \{-1, 0, 1\} \) indicates the feature type dale, zero element or hill,
- \( PF_{H} \) unsigned normal distance from the reference line to the extremum of the profile feature,
- \( PF_{X_1} \) left intersection point of the profile feature with the reference line,
- \( PF_{X_2} \) right intersection point of the profile feature with the reference line,
- \( H_p \) profile hill height discrimination \( \in \mathbb{R}_{D}^+ \),
- \( H_{H} \) profile dale depth discrimination \( \in \mathbb{R}_{D}^+ \),
- \( O_{H} \) threshold to suppress numerical noise \( \in \mathbb{R}_{D}^+ \) (outwardly directed),
- \( O_{D} \) threshold to suppress numerical noise \( \in \mathbb{R}_{D}^+ \) (inwardly directed).

2.2. Detection of profile features by zero crossing

The detection of profile features is based on zero crossings, i.e. the intersection of the profile with the reference line. For two adjacent profile samples, say \( z_{j-1} \) and \( z_j \), three cases have to be distinguished:

- The connecting line of two adjacent profile samples intersect the reference line in such a way, that \( z_{j-1} \) lies below or on the reference line and \( z_j \) lies above the reference line or \( z_{j-1} \) lies below the reference line and \( z_j \) lies above or on the reference line (figure 1(a)).
- The connecting line of two adjacent profile samples intersect the reference line in such a way, that \( z_{j-1} \) lies above or on the reference line and \( z_j \) lies below the reference line or \( z_{j-1} \) lies above the reference line and \( z_j \) lies below or on the reference line (figure 1(b)).
- Two adjacent profile samples lie on the reference line (figure 1(c)).

The three cases can be summarized mathematically as follows: a zero crossing is detected if

\[
(z_{j-1} \leq 0 \land z_j > 0) \lor (z_{j-1} > 0 \land z_j < 0) \lor (z_{j-1} > 0 \land z_j > 0) \land (z_{j-1} > 0) \land (z_j = 0).
\]

The three cases of zero crossing are changed to:

\[
(z_{j-1} \leq O_{H}) \land (z_j > 0) \lor (z_{j-1} < O_{H}) \land (z_j \geq O_{D}) \lor (z_{j-1} \geq O_{H}) \land (z_j < O_{H}) \land (z_{j-1} > 0) \land (z_j = 0).
\]
neighbour lies sufficiently far apart. For this reason the region of doubt is bounded by two positive real values $O_H$ and $O_D$. The upper bound is given by $O_H$ and the lower bound is given by $-O_D$. It must be pointed out that both $O_H$ and $O_D$ have to be chosen carefully. A good choice for $O_H$ and $O_D$ respectively is to choose them as multiples of the standard deviation of the profile samples. Alternatively the values for $O_H$ and $O_D$ can be set to fractions of the detection thresholds $H_u$ and $H_l$ which are used for detecting the feature types hill and dale. The necessity for detection thresholds will be further discussed in section 2.3.

The resulting procedure is given by the following rule: a zero crossing is detected if

$$ (z_{j-1} \leq O_H \land z_j \geq -O_D) \lor (z_{j-1} \geq -O_D \land z_j \leq O_H). $$

It is straightforward to decide which type of feature (hill, dale or zero element) is separated by each zero crossing. Index $j-1$ of a zero crossing indicates the end of a feature type and index $j$ indicates the start of a new feature type. Per definition, the first profile feature begins at index $j = 1$ and the last profile feature ends at $j = n$, with $n$ the number of profile samples. The profile samples within each profile feature region are evaluated in the following manner:

- A hill is present if at least one of the profile samples within a profile feature region is greater than or equal to the zero crossing limit $O_H$ or to the upper vertical limit $H_u$ if $H_u < O_H$.
- A dale is present if at least one of the profile samples within a profile feature region is lower than or equal to the zero crossing limit $O_D$ or to the lower vertical limit $H_l$ if $H_l < O_D$.
- Otherwise a zero element is detected.

$O_H$ and $O_D$ are usually a magnitude smaller than $H_u$ and $H_l$ as can be seen in section 3. The cases $H_u < O_H$ and $H_l < O_D$ were only included to handle rare exceptions.

The previously described algorithm can be summarized in pseudo code:

```plaintext
{initialization}
i := 1, j := 2, n_{PF} := 0
{loop over all profile samples}
while j \leq n
  {detect zero crossing}
  if $\left( z_{j-1} \leq O_H \land z_j \geq -O_D \right) \lor \left( z_{j-1} \geq -O_D \land z_j \leq O_H \right)$ then
    {increment the number of profile features}
    $n_{PF} := n_{PF} + 1$
    {calculate the profile feature height within the region i and j – 1}
    $k_h := \arg \max_{k = (j-1) \cdots 1} (\text{abs}(z_k))$
    {get the type of the feature: hill, dale or zero element}
    $PF_{n_{PF}.t} := \text{sgm} \left( z_k, \min(O_D, H_l), \min(O_H, H_u) \right)$
    {set profile feature height}
    $PF_{n_{PF}.h} := \text{abs}(z_{k_f})$
    {calculate zero crossing by linear interpolation}
    $PF_{n_{PF}.x_i} := \text{root} \left( x_{\max((i-1),1)}, z_{\max((i-1),1)}, x_i, z_i \right)$
    $PF_{n_{PF}.x_f} := \text{root} \left( x_{j-1, j-1}, z_{j-1}, x_j, z_j \right)$
  {update index i}
  i := j
endif
{next profile sample}
j := j + 1
endwhile
{add last profile feature}
$n_{PF} := n_{PF} + 1$
$k_h := \arg \max_{k = (j) \cdots n} (\text{abs}(z_k))$
$PF_{n_{PF}.t} := \text{sgm} \left( z_k, \min(O_D, H_l), \min(O_H, H_u) \right)$
$PF_{n_{PF}.h} := \text{abs}(z_{k_f})$
$PF_{n_{PF}.x_i} := \text{root} \left( x_{\max((i-1),1)}, z_{\max((i-1),1)}, x_i, z_i \right)$
$PF_{n_{PF}.x_f} := x_n$
```
2.3. Vertical limit criterion
ISO 4287:1997 recommends the use of a vertical and a horizontal detection threshold to eliminate insignificant profile features and avoid oversegmentation. However, applying implementations that use the current definition of $RSm$ on stochastic profiles shows a correlation between the choice of the horizontal threshold and the resulting $RSm$ parameter as shown in figure 3. With an increasing value of the horizontal discrimination, also the value of $RSm$ increases as the number and area of discarded profile elements increases.

Furthermore hills that are sufficiently above the reference line might not exist in every profile. Robust filtering of plateau-like surface profiles can lead to plateaus being exactly on the reference line, that cannot be detected with a vertical threshold unequal zero (see figure 4).

Therefore the use of horizontal discrimination is abandoned in favour of two separate vertical limits for dale detection $H_l$ and hill detection $H_u$. In doing this, the hill height discrimination limit $H_u$ and the upper noise threshold $O_H$ can both be set to zero to allow the detection of plateaus as pseudo-hills. The same can be done for $H_l$ and $O_D$ to detect flat dales on the reference line in certain profiles. Consider the $k$-th profile feature:

- Delete a profile dale $(PF_k, t = -1)$ if its height $PF_k.h$ is lower than the vertical limit $H_l$ or lower than the lower zero crossing limit $O_D$.
- Delete a profile hill $(PF_k, t = 1)$ if its height $PF_k.h$ is lower than the vertical limit $H_u$ or lower than the upper zero crossing limit $O_H$.
- Delete the zero element $(PF_k, t = 0)$.

An implementation of the vertical limit criterion is given by the subsequent pseudo code

```
{ initialization } 
k := n_{PF}
{ loop over all profile features } 
while $k \geq 1$
  { check dale } 
  if $(PF_k, t = -1) \land (PF_k.h < \max(O_D, H_l))$ then
    { delete dale and decrement the number of profile features }
    delete $PF_k$
    $n_{PF} := n_{PF} - 1$
  { check zero element }
  elseif $(PF_k, t = 0)$ then
    { delete zero element and decrement the number of profile features }
    delete $PF_k$
    $n_{PF} := n_{PF} - 1$
  { check hill }
  elseif $(PF_k, t = 1) \land (PF_k.h < \max(O_H, H_u))$ then
    { delete hill and decrement the number of profile features }
    delete $PF_k$
    $n_{PF} := n_{PF} - 1$
  endif
  { next profile feature } 
  $k := k - 1$
endwhile
```
2.4. Merging of profile features

After the insignificant profile features have been deleted, possibly adjacent features of the same type are present in the dataset. All adjacent profile features of the same type are merged to one feature. This is done since both the current ISO 4287:1997 and the future ISO 21920 require the combination of adjacent profile features of a different type to profile elements as later described in section 3. The new feature height is the maximal feature height of all merged elements and the left intersection point \( x_l \) of the first merged profile feature and the right intersection point \( x_r \) of the last merged profile feature bound the region of the new feature. Here is the entire process step as pseudo code:

\[
\begin{align*}
\text{initialization} \\
k := n_{PF} \\
\text{loop over all profile features} \\
\text{while } k \geq 2 \\
\quad \text{compare adjacent profile features and merge them if they are of the same type} \\
\quad \text{if } (PF_{k-1}, t = PF_{k}, t) \text{ then} \\
\quad \quad \text{assign the maximum height to the left element} \\
\quad \quad PF_{k-1}, h := \max(PF_{k-1}, h, PF_{k}, h) \\
\quad \quad \text{assign } x_r \text{ of the right profile feature to } x_r \text{ of the left profile feature} \\
\quad \quad PF_{k-1}, x_r := PF_{k}, x_r \\
\quad \quad \text{delete right profile feature and decrement the number of elements} \\
\quad \quad \text{delete } PF_{k} \\
\quad \quad n_{PF} := n_{PF} - 1 \\
\quad \text{endif} \\
\quad \text{next profile feature} \\
\quad k := k - 1 \\
\end{align*}
\]

3. Crossing-the-line segmentation and the profile parameters \( R_{Sm} \) and \( R_c \)

2D-roughness parameters as \( R_a \) and \( R_t \) have been used for multiple generations [2]. Based on the early parameters, from the 1940s onwards, several 2D-measures were developed leading to a great many of parameters [2]. Nowadays, feature parameters for the characterization of profile topographies can be found in the standard ISO 4287:1997 [1]. These parameters enable a more function-oriented characterization of technical surfaces and are represented by the mean height of the profile elements \( R_c \) and the mean spacing of the profile elements \( R_{Sm} \) [1]. Both parameters are calculated based on the roughness profile. In ISO 4287:1997 a profile element is defined as a profile hill followed by a profile dale or a profile dale followed by a profile hill. The profile elements are separated by the zero crossings of the profile [1]. Unfortunately, the implementation of the parameters is not clearly defined in ISO standardisation and there is no reference algorithm so far. Due to this, a comparability of the results of different measurement devices or different evaluation software respectively, is not ensured. While the parameter definition has been subject of previous research [11] the algorithmic implementation of it has not yet been described to full extent. Furthermore, in ISO 4287:1997 \( R_{Sm} \) is actually defined as mean width of profile elements, as already described in section 1, but the following algorithm will calculate profile element spacing, as will be part of the revised definition in ISO 21920.

The crossing-the-line algorithm now allows an unambiguous evaluation of the two parameters \( R_c \) and

![Figure 4. Surface profile with plateau-like surface features on the reference line.](image-url)
For both parameters, we recommend the following discrimination limits for the segmentation process:

– The upper vertical limit $H_u$ shall be 10% of the roughness parameter $R_p$ (distance between the highest hill and the reference line).

– The lower vertical limit $H_l$ shall be 10% of the roughness parameter $R_v$ (distance between the deepest dale and the reference line).

– The upper limit $O_H$ for zero crossings shall be 0.01% of the roughness parameter $R_p$.

– The lower limit $O_D$ for zero crossings shall be 0.01% of the roughness parameter $R_v$.

In ISO 4287:1997 the vertical discrimination level is given with regard to a percentage of $R_z$ (vertical distance between highest hill and deepest dale). As shown above, we propose to use different levels for positive and negative profile portions and to abandon horizontal discrimination. Furthermore, the parameters $R_c$ and $R_{Sm}$ depend on the direction of evaluation. This becomes clear when the number of hills and dales are unequal. The following definition for $R_c$ and $R_{Sm}$ respectively avoid this issue.

The way to calculate $R_{Sm}$ and $R_c$:

– Apply the crossing-the-line segmentation with appropriate threshold limits.

– After segmentation, the total number $m$ of profile elements is given by $m = 2 \cdot \lfloor \frac{npf}{2} \rfloor - 1$ (the operator $\lfloor q \rfloor$ calculates the greatest integer less than or equal to $q$). Note: The first and last profile features are never considered to be part of any profile element and the total number $m$ includes the elements of both evaluation directions.

– Allocate the profile element spacing $X_{sk}$ and profile element height $Z_{tk}$ as follows

\[
\begin{align*}
\text{• take into account the evaluation from the left to the right} \\
X_{sk} &= PF_{2(k+1)} \cdot xl - PF_{2k} \cdot xl, \ k = 1, \ldots, m/2 \\
Z_{tk} &= PF_{2k} \cdot h + PF_{2k+1} \cdot h, \ k = 1, \ldots, m/2
\end{align*}
\]

\[
\begin{align*}
\text{• take into account the evaluation from the right to the left} \\
X_{sk+m/2} &= PF_{(npf+1)-2k-1} \cdot xr - PF_{(npf+1)-2(k+1)} \cdot xr, \ k = 1, \ldots, m/2 \\
Z_{tk+m/2} &= PF_{(npf+1)-2k} \cdot h + PF_{(npf+1)-(2k+1)} \cdot h, \ k = 1, \ldots, m/2
\end{align*}
\]

– Calculate $R_c$ and $R_{Sm}$

\[
R_c = \frac{1}{m} \sum_{k=1}^{m} Z_{tk}, \quad R_{Sm} = \frac{1}{m} \sum_{k=1}^{m} X_{sk}
\]

In the following the parameter calculation as pseudo code

\[
\begin{align*}
\{ \text{initialization} \} \\
m &= 2 \cdot \lfloor \frac{npf}{2} \rfloor - 1 \\
k &= 1 \\
\{ \text{calculation of spacings } X_{sk} \text{ and heights } Z_{tk} \} \\
\text{while } k \leq m/2 \\
\quad X_{sk} &= PF_{2(k+1)} \cdot xl - PF_{2k} \cdot xl \\
\quad Z_{tk} &= PF_{2k} \cdot h + PF_{2k+1} \cdot h \\
\{ \text{evaluation from left to right} \} \\
\quad X_{sk+m/2} &= PF_{(npf+1)-2k-1} \cdot xr - PF_{(npf+1)-2(k+1)} \cdot xr, \ k = 1, \ldots, m/2 \\
\quad Z_{tk+m/2} &= PF_{(npf+1)-2k} \cdot h + PF_{(npf+1)-(2k+1)} \cdot h, \ k := k + 1 \\
\} \text{ endwhile}
\]

\[
\begin{align*}
\{ \text{initialization} \} \\
R_{Sm} &= 0, \quad R_c := 0, \quad k = 1 \\
\{ \text{calculation of } R_{Sm} \text{ and } R_c \} \\
\text{while } k \leq m \\
\quad R_{Sm} &= R_{Sm} + X_{sk} \\
\quad R_c &= R_c + Z_{tk} \\
\quad k &= k + 1 \\
\} \text{ endwhile}
\]

\[
R_{Sm} = \frac{R_{Sm}}{m}, \quad R_c = \frac{R_c}{m}
\]

NOTE: If no profile elements are detected (m equal zero), $R_{Sm}$ and $R_c$ are not defined.
3.1. Example

Figure 5 shows, as an example, the segmentation of hills and dales after applying the crossing-the-line algorithm defined in section 2. The vertical limits are given by \( H_u = 0.1 \cdot R_p \) and \( H_v = 0.1 \cdot R_v \) as recommended before. Eight dales (bricks) and seven hills (hatched areas) were detected. The profile features are numerated from 1 to 15. The grey filled area (between profile feature seven and eight) is identified as a zero element.

To be independent of the evaluation direction, the profile features are evaluated from the beginning to the end of the evaluation length, figure 6, and vice versa, figure 7. The evaluation starts (figure 6) with the second profile feature \( PF_2 \), a hill. Consequently, the first profile element consists of profile feature \( PF_2 \) and \( PF_3 \). Two profile features are always combined for the next profile element. The evaluation ends when no more profile elements can be formed. A total of six profile elements can be formed in this way. The spacing of the profile elements is given by the distance of the beginning of two consecutive profile elements (here the beginning of two consecutive hills). The height of each profile element is simply the addition of the heights of the two profile features involved.

Figure 7 shows the evaluation of profile features from the right to left. The procedure is the same as described before. A special case concerns profile element 10. A zero element is enclosed by a hill and a dale and is therefore part of this profile element.
3.2. Case studies

After introducing the algorithm, its application will be discussed by describing multiple examples that serve as case studies for the crossing-the-line segmentation. The first examples will describe artificial profiles to demonstrate properties of the implementation before actual engineering surfaces will be addressed. Figure 8 shows the evaluation of a sinusoidal profile with a wavelength of 1.2 mm and an amplitude of 1.0 μm. The algorithm recognizes seven profile features (four hills and three dales). The first and the last feature are border features (in this case both hills). They are deleted (indicated with the hatched areas) since they could possibly be part of larger features, which have been cut off due to the arbitrary starting point of the profile measurement. In the left to right evaluation the first dale and the second hill are combined to the first profile element (red). Furthermore the second dale and the third hill are combined to the second profile element (blue). Though it has not been deleted, the third (most right) dale cannot be assigned to a profile element, because there is no hill left over. The right to left evaluation on the other hand leaves out the most left dale respectively. The $R_{Sm}$ parameter describes a profile element spacing of 1.2 mm, which in this case is equivalent to the wavelength.
The surface in figure 9 contains sinusoidal components interrupted by zero elements that are located either exactly on the reference line or at least between the height discrimination limits. In the very first step those zero elements are deleted (hatched areas). Again the first dale and last hill (border elements) are also deleted. In the left to right evaluation the first hill is combined with the second dale (red) and the second hill with the third dale (blue) to form profile elements. Since the $RSm$ parameter evaluates profile element spacing (in opposite to width) the width of the central zero element (red and hatched) is included when calculating the spacing to the following profile element, although the zero element was deleted in the first place. In the same way the zero element that follows the second profile element (blue and hatched) is...
included when calculating the spacing to the last (border) feature.

Because zero elements are deleted before the border features are addressed the very first and the very last hill in the profile in figure 10 are deleted although not being located at the border. Therefore only one profile element can be combined from the first dale and the second hill (left to right evaluation). The zero element between this dale and this hill, though deleted in the first place, is included within this profile element and also contributes to its width. As before, when calculating elements spacing, also the width of the zero element right of the profile element is taken into account.

Figure 11 shows the merging of adjacent profile features of the same type. Therefore the first two and the last two hills are deleted as border features, after being merged into one feature each. Consequently the single profile element consists of two dales and two hills which have been merged into one dale and one hill. Over all, the ten features in this profile are merged pairwise to five features in the end.

Lowering the height discrimination limit for hills to zero allows for the evaluation of plateau like profiles, as shown in figure 12. Under the presence of noise, a plateau is recognized as a sequence of hills and zero elements which are merged into a single hill feature across the whole plateau. Depending on evaluation direction these plateaus are then combined with the adjacent grooves to profile elements. The current description of $RSm$ would not detect any profile elements as there would be no hills detected with a threshold of 10 % of $Rz$. This also allows for deleting whole plateaus at the borders as border features instead of only deleting little fractions of it.

Figure 13 shows a last example of artificial profiles with an entirely stochastic surface. It is constructed as white Gaussian noise with a 1 $\mu m$ standard deviation. For example the resulting, small $RSm$-parameter of only 2.8 $\mu m$ indicates a highly stochastic profile with quickly alternating and therefore short profile portions.

When slightly correlating neighbouring samples, one receives a profile with coloured noise. Though the resulting profile in figure 14 has roughly the same signal power (sum of squared samples) as the profile in figure 13, the $RSm$ parameter has a steep increase with about factor 4. This indicates the reduction in randomness within the profile.

Figure 15 shows a measured profile produced by a turning process. The $RSm$ parameter represents the spacing of turning grooves on the mostly deterministic surface.

Finally figure 16 shows a measured profile of a honed surface. The similarities to the profile in figure 12 can be seen based on the distinct dales and the plateau-area with a stochastic height distribution. Consequently it is evaluated with a hill height discrimination limit of zero to allow the detection of the plateaus as hills.

Figure 17 shows a measured profile of a sintered surface. The reference line for this kind of profile is determined with a morphologic filter to simulate the

Figure 11. Adjacent profile features of the same type are merged before combination into profile elements.
Figure 12. Deterministic plateau-like surface with noise. Zooming in on the plateaus reveals, that the plateaus are detected as a series of hills and zero elements und subsequently merged into one hill feature each.

Figure 13. Crossing the line segmentation on a stochastic profile with white Gaussian noise.
contact partner. In this case the profile was morphologically closed by a disc as structuring element with a 0.8 mm radius.

Table 1 gives an overview of the calculated $R_{Sm}$-parameters for the different surfaces. It also shows the $R_{Sm}$-parameters calculated with an older implementation of ISO 4287:1997 for different horizontal discrimination limits. It is obvious that the new approach delivers equal or similar results for mostly deterministic surfaces like in figures 8 to 11 or 15. Furthermore it allows the evaluation of plateau-like surfaces like the ones in figures 12 or 17, which was not possible before. The results for figures 13 to 16 show a clear correlation between $R_{Sm}$ and horizontal discrimination limit when applying the old implementation. There is no such ambiguity for the new algorithm.
3.3. Implementation
In appendix A, both a Matlab and Python code implementation of the previously given pseudo code that have been debugged and tested on various profiles are provided.

4. Conclusion
Crossing-the-line segmentation is a powerful tool to separate characteristic features of a profile. In this publication we have described an unambiguous core algorithm in pseudo code and have included an implementation in Matlab. A typical application for this algorithm is to calculate the feature parameters $R_Sm$ and $R_c$ defined in ISO 4287:1997. However, other parameters such as the number or spacings of particular features are also possible, just to give an example. The presented algorithm will be part of ISO 16610 part 45 and ISO 21920 part 2 which will be published in the next years. The Technical Report 23276 will also be published so that a document will soon be available at ISO level.
Table 1. Comparison of $R_{Sm}$-parameters between the proposed algorithm and a previous implementation for different horizontal discrimination limits.

| Figure, | Proposed algorithm: $R_{Sm}/\mu m$ | Old algorithm: horizontal discrimination of 0.01 mm $R_{Sm}/\mu m$ | Old algorithm: horizontal discrimination of 0.02 mm $R_{Sm}/\mu m$ | Old algorithm: horizontal discrimination of 0.05 mm $R_{Sm}/\mu m$ | Old algorithm: horizontal discrimination of 0.1 mm $R_{Sm}/\mu m$ |
|--------|----------------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
| Figure_8 | 1,200.0 | 1,200.0 | 1,200.0 | 1,200.0 | 1,200.0 |
| Figure_9 | 1,400.0 | 1,400.0 | 1,400.0 | 1,400.0 | 1,400.0 |
| Figure_10 | 1,600.0 | 1,600.0 | 1,600.0 | 1,600.0 | 1,600.0 |
| Figure_11 | 1,600.0 | 1,600.0 | 1,600.0 | 1,600.0 | 1,600.0 |
| Figure_12 | 800.0 | nan | nan | nan | nan |
| Figure_13 | 2.8 | 9.929 | 17.662 | 41.484 | 81.760 |
| Figure_14 | 11.8 | 17.657 | 24.420 | 47.958 | 86.967 |
| Figure_15 | 182.0 | 181.091 | 181.091 | 181.091 | 181.091 |
| Figure_16 | 58.9 | 88.830 | 88.830 | 97.713 | 130.283 |
| Figure_17 | 91.5 | nan | nan | nan | nan |

The required implementation by instrument manufacturers in order to enable a comparable evaluation with different instruments can be based on the provided code in the Appendix or be implemented based on the presented Pseudo-code in any other programming language.
Appendix A. Matlab and python implementation of the described pseudo-code

Subsequently both an implementation of the crossing-the-line segmentation just as well as of the $RSm$ and $Rc$ parameter determination are provided both for Python and Matlab. Note that variable calls in Python are always decremented by 1 since variable indices in Python start at 0 instead of 1.

A.1. Crossing-the-line segmentation
A.1.1. Matlab Implementation.

```matlab
function [PF, nPF] = crossing_the_line_segmentation(x, z, OD, HI, Hu)

    kH = 1; % k = 1
    k = length(x);

    % step 1: detection of profile features by zero crossing
    i = 1; j = 1; nPF = 0;
    while j < k
        if x(j) == OD && z(j) < OD || x(j-1) == OD && z(j) <= OD
            kH = kH + 1;
            kH = max(kH, kH + 1);
            PF(nPF).t = x(j);
            PF(nPF).H = OD;
            PF(nPF).z = z(j);
            nPF = nPF + 1;
        end
        j = j + 1;
    end

    % step 2: vertical limit criterion
    k = k - 1;
    while k > 1
        if PF(k).t == PF(k-1).t
            kH = kH - 1;
            k = k - 1;
        elseif PF(k).H > PF(k-1).H
            PF(k-1).t = PF(k).t;
            PF(k-1).H = PF(k).H;
            PF(k-1).z = PF(k).z;
            kH = kH - 1;
            k = k - 1;
        else
            k = k - 1;
        end
    end

    % modified signum function
    function v = sign(x, a, u)
        if x > a
            v = 1;
        elseif x < a
            v = -1;
        else
            v = 0;
        end

    % root function
    function x0 = root(xa, xa, xb, zb)
        if zb > xa
            x0 = (xa+xb)/2;
        else
            x0 = min(max((xa*zb-xb*za)/(zb-xa), xa), xb);
        end

    % argmax function
    function kh = arg_max(s)
        [~, kh] = max(s);
    End
```
A.1.2. Python implementation.
rom numpy import *

" define a class for hills, zeros and dales "
class struct_element:
    def __init__(self, t, h, xl, xr):
        self.t = t
        self.h = h
        self.xl = xl
        self.xr = xr

def __str__(self):
    return "t: %s h: %s xl: %s xr: %s" % (self.t, self.h, self.xl, self.xr)

" modified sigmoid function "
def sigmoid(z, l, u):
    if z <= u:
        return l
    elif z <= -l:
        return -l
    else:
        return 0

" root function "
def root(xa, zb, xb, zbd):
    if zb == xa:
        return (xa + xb) / 2
    else:
        return min(max(xa, zb - zb * 2) / (zb - xa), xb)

" argmax function "
def argmax(z):
    return argmax(z) + 1

def crossing_the_line_segmentation(x, z, OD, OH, fl, ff):
    PF = []
    nPF = 0
    i = 1
    j = 2
    while j <= n:
        if z[i - 1] == OD and z[j - 1] == OH or z[i - 1] == OH and z[j - 1] == OD:
            nPF = nPF + 1
            t = sigmoid(z[kh - k - 1], min(OD, fl), min(OH, OH))
            xl = root(xl[min(i - 1), i + 1], z[i - 1], z[i + 1], k - 1)
            xr = root(xr[i - 1], z[i - 1], z[i + 1], k - 1)
            PF.append(struct_element(t, absolute(z[kh - k - 1]), xl, xr))
            i = i
            j = j + 1
            kh = arg_max(absolute(z[kh - k - 1]), i + 1)
        else:
            nPF = nPF + 1
            k = FF
            while k >= 2:
                if PF[k - 1].t == -1 and (PF[k - 1].h < max(OD, OH)):
                    del PF[k - 1]
                else:
                    PF[k - 1].t = 0;
                    del PF[k - 1]
                    nPF = nPF - 1
                if PF[k - 1].t == 1 and (PF[k - 1].h < max(OD, OH)):
                    del PF[k - 1]
                    nPF = nPF - 1
                k = k - 1

            return [PF, nPF]
A.2. RSm and Rc parameters

A.2.1. Matlab Implementation.

```matlab
function [PF, nPF, RSm, Rc] = iso21950_stmp(x, y, OD, OH, H1, H0);
    [PF, nPF] = crossing_the_line_segmentation(x, y, OD, OH, H1, H0);
    m = 2*(floor(nPF/21)+1);
    Xs and Zs
    for k=1:n/2
        Xs(k) = PF(2*k-1,:); xl = PF(2*k,:);
        Zs(k) = PF(k,:); h = PF(2*k+1,:);
        Xs(k+m/2) = PF((nPF+1)-2*k,:); xl = PF((nPF+1)-2*k,:);
        Zs(k+m/2) = PF((nPF+1)-2*k,:); h = PF((nPF+1)-2*k,:);
    end
    RSm = sum(Xs)/m;
    Rc = sum(Zs)/m;
end
```

A.2.2. Python Implementation.

```python
def parameters_rsm_rc(m, x, y, OD, OH, H1, H0):
    [PF, nPF] = crossing_the_line_segmentation(x, y, OD, OH, H1, H0)
    m = int(2 * (floor(nPF / 2) + 1))
    Xs = [None] * m
    Zs = [None] * m
    while k < m // 2:
        Xs(k) = PF[2 * k + 1 - 1] - PF[2 * k - 1]; mx = PF[2 * k - 1]; xl = PF[2 * k];
        Zs(k) = PF[2 * k - 1 + 1] - PF[2 * k]; h = PF[2 * k + 1];
        Xs(k+m/2) = PF((nPF+1)-2*k+1); mx = PF((nPF+1)-2*k+1); xl = PF((nPF+1)-2*k+1);
        Zs(k+m/2) = PF((nPF+1)-2*k+1); h = PF((nPF+1)-2*k+1);
        k = k + 1
    RSm = sum(Xs) / m
    Rc = sum(Zs) / m
    return [PF, nPF, RSm, Rc]
```

ORCID iDs

P J Scott @ https://orcid.org/0000-0002-6092-3101
M Eifler @ https://orcid.org/0000-0001-6628-7284
D Hüser @ https://orcid.org/0000-0002-2208-4490

References

[1] ISO 4287: 1997 Geometrical Product Specifications (GPS) – Surface texture: Profile method – Terms, definitions and surface texture parameters (ISO 4287:1997)
[2] Stout K 2000 Development of methods for the characterisation of roughness in three dimensions Rev. Repr. (London: Penton Press)
[3] Blint L and Jiang X 2003 Advanced techniques for assessment surface topography: development of a basis for 3D surface texture standards ‘SURFSTAND’ (London: Kogan Page Science)
[4] Blateyron F 2013 The areal feature parameters ed R Leach Characterisation of Areal Surface Texture (Berlin: Springer) pp 45–66
[5] ISO 25178-2 2012 Geometrical product specifications (GPS) – Surface texture: Areal – Part 2: Terms, definitions and surface texture parameters (ISO 25178-2:2012)
[6] Blanc J, Grime D and Blateyron F 2011 Surface characterization based upon significant topographic features J. of Physics: Conference Series 311 012014
[7] Wolf G 2017 Scale independent surface characterization: geography meets precision surface metrology Precis. Eng. 49 465–80
[8] Scott P J 1998 Foundations of topographical characterization of surface texture Int. J. of Mach. Tools Manuf. 38 559–66
[9] Blateyron F 2016 Segmentation and Feature Parameters – A correction of watershed segmentation and feature parameters defined in ISO 25178-2 5th Int. Conf. on Surface Metrology
[10] Senin N, Thompson A and Leach R K 2018 Feature-based characterization of signature topography in laser powder bed fusion of metals Meas. Sci. Technol. 29 045009
[11] Scott P 2006 The case of surface texture parameter RSm Meas. Sci. Technol. 17 559–64