Trimmed likelihood estimators for stochastic differential equations with an application to crack growth analysis from photos

Christine H. Müller and Stefan H. Meinke

Abstract We introduce trimmed likelihood estimators for processes given by a stochastic differential equation for which a transition density is known or can be approximated and present an algorithm to calculate them. To measure the fit of the observations to a given stochastic process, two performance measures based on the trimmed likelihood estimator are proposed. The approach is applied to crack growth data which are obtained from a series of photos by backtracking large cracks which were detected in the last photo. Such crack growth data are contaminated by several outliers caused by errors in the automatic image analysis. We show that trimming 20% of the data of a growth curve leads to good results when 100 obtained crack growth curves are fitted with the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross processes while the fit of the Geometric Brownian Motion is significantly worse. The method is sensitive in the sense that crack curves obtained under different stress conditions provide significantly different parameter estimates.

Christine H. Müller
Department of Statistics, TU Dortmund University, D-44227 Dortmund, Germany
✉ cmueller@statistik.tu-dortmund.de

Stefan H. Meinke
Department of Statistics, TU Dortmund University, D-44227 Dortmund, Germany
✉ stefan.meinke@uni-dortmund.de

ARCHIVES OF DATA SCIENCE (ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. -, No. -, -
ISSN 2363-9881

1 Introduction

The motivation of this paper is the analysis of micro crack growth data obtained from photos of the surface of a steel specimen exposed to cyclic load. A simple model for crack growth is given by the Paris-Erdogan equation (see e.g. Pook, 2000)

$$\frac{da}{dN} = C\,(\Delta\sigma\sqrt{a})^m, \tag{1}$$

where a is the crack length, N the number of load cycles, C and m are usually unknown constants and $\Delta\sigma = \sigma_{max} - \sigma_{min}$ is the range of the cyclic stress amplitude. Since crack growth is not a deterministic process, several stochastic versions of the Paris-Erdogan equation were developed (see e.g. Ortiz and Kiremidjian, 1988; Ray and Tangirala, 1996; Nicholson et al, 2000; Wu and Ni, 2004; Chiquet et al, 2009; Hermann et al, 2016a,b). Often only models are developed but no statistical analysis is presented. For example, the books of Sobczyk and Spencer (1992), Castillo and Fernández-Canteli (2009), Sánchez-Silva and Klutke (2016) are full of models but besides some simple statistical methods not much is provided.

One approach is to extend equation (1) with an additive stochastic term leading to a stochastic differential equation (SDE), see e.g. Lin and Yang (1983), Sobczyk and Spencer (1992), Wu and Ni (2004), Zárate et al (2012), Hermann et al (2016a,b). The advantage of a SDE is that several statistical methods were developed already, at least for some of them, see e.g. Sørensen (2004) or Iacus (2008). In particular, likelihood methods were proposed as in Pedersen (1995), Beskos et al (2006), Pastorello and Rossi (2010), Sun et al (2015), Höök and Lindström (2016). However, the choice of the SDE is still a problem. Moreover, usually the crack growth data are not so nice and numerous as those of Virkler et al (1979), who collected 68 series with 164 measurements in a laborious study. These measurements are nice since they do not include many big jumps and errors in contrast to other crack growth data as those considered in Kustosz and Müller (2014) and Hermann et al (2016a) where additionally less than ten series were observed.

With automatic detection of crack growth from photos, it is possible to obtain a much larger number of series of crack growth data, and the data collection is much less laborious. The resulting large data sets allow a better analysis of the crack growth and the corresponding fatigue behavior of the material. However, those data sets contain several errors originating from the

automatic detection process. For example, these errors are caused by scratches and contaminations of the material, blurred photos, shadows on the photos and other image processing problems, and they lead to outliers in the crack growth curves. In particular, they can generate crack growth data series whose growth curves are not strictly increasing.

To cope with such outliers, we propose to use trimmed likelihood estimators for SDEs. Trimmed likelihood estimators for independent observations were introduced by Hadi and Luceño (1997). They extend the least median of squares estimator and the least trimmed squares estimators of Rousseeuw (1984) and Rousseeuw and Leroy (1987) by replacing the likelihood functions of the normal distribution by likelihood functions of other distributions. Müller and Neykov (2003) applied them for generalized linear models and other applications can be found for example in Neykov et al (2007), Cheng and Biswas (2008), Neykov et al (2014), Müller et al (2016). Because of the trimming of a proportion of the data, trimmed likelihood estimators can deal with an amount of outliers up to the trimming proportion. We use trimmed likelihood estimators here to define two measures for the performance of a fit of a SDE to the data. One performance measure is based on the median of absolute deviations of predictions and the other is based on the coverage rate of prediction intervals.

The paper is organized as follows. In Section 2, the trimmed likelihood estimator together with its computation is introduced and the two performance measures based on the trimmed likelihood estimator are proposed. Section 3 provides the application to crack growth data obtained from photos. Therefore, at first, it is described how crack growth curves can be obtained by backtracking long cracks which were detected in the last photo in a series of photos. Then the fits of three SDEs to these curves are obtained with trimmed likelihood estimators with different trimming rates and are compared via the two performance measures. Moreover, the fits of curves from two experiments with different stress conditions lead to significantly different parameter estimates so that the approach can be used to distinguish between different stress conditions. Finally, Section 4 discusses the results and some further extensions.

2 Trimmed likelihood estimators for SDEs

2.1 Trimmed likelihood estimators and their computation

A stochastic extension of the Paris-Erdogan equation (1) is given by the stochastic differential equation

$$\mathrm{d}X_t = b(X_t,\theta) + s(X_t,\theta)\mathrm{d}B_t$$

where the time-continuous stochastic process $(X_t)_{t\geq 0}$ provides the crack size, b and s are known functions, $(B_t)_{t\geq 0}$ is the standard Brownian Motion, and θ is an unknown parameter vector. Special cases are given by

$$\mathrm{d}X_t = (\theta_1 + \theta_2 X_t)\mathrm{d}t + \theta_3 X_t^{\gamma}\mathrm{d}B_t,$$

with $\theta = (\theta_1,\theta_2,\theta_3,\gamma)' \subset \mathbb{R}^4$ which include the Ornstein-Uhlenbeck process ($\gamma = 0$), the Cox-Ingersoll-Ross process ($\gamma = 0.5$), and the Geometric Brownian Motion ($\theta_1 = 0, \gamma = 1$), see e.g. Iacus (2008). The process is observed at time points $0 \leq t_0 < t_1 < \ldots < t_N$ providing observations $x_{t_0},...,x_{t_N}$.

The idea of trimming is to use only a subset $\mathscr{I} = \{n(1),\ldots,n(I)\}$ of $\{0,1,\ldots,N\}$ with $0 \leq n(1) < n(2) < \ldots < n(I) \leq N$. Since the conditional distribution of $X_{t_{n(i+1)}}$ given $X_{t_{n(i)}}$ is often known or at least can be approximated, we set

$$p_\theta(x_{t_{n(i+1)}}|x_{t_{n(i)}})$$

for the transition density of the conditional distribution or its approximation. Then the likelihood function for the observation vector $x_{\mathscr{I}} := (x_{t_{n(1)}},...,x_{t_{n(I)}})$ is given by

$$l_\theta(x_{\mathscr{I}}) = \prod_{i=1}^{I-1} p_\theta(x_{t_{n(i+1)}}|x_{t_{n(i)}}).$$

If $\mathscr{I} = \{0,1,\ldots,N\}$ then the classical maximum likelihood estimator is given by

$$\widehat{\theta} := \arg\max_\theta l_\theta(x_{\mathscr{I}}).$$

We can also define the maximum likelihood for any subset $\mathscr{I} \subset \{0,1,\ldots,N\}$ as

$$\widehat{\theta}(\mathscr{I}) := \arg\max_{\theta} l_{\theta}(x_{\mathscr{I}})$$

so that $\widehat{\theta}(\mathscr{I}) = \widehat{\theta}$ if $\mathscr{I} = \{0,1,\ldots,N\}$. A H-trimmed likelihood estimator is then defined as (see e.g. Hadi and Luceño, 1997; Müller and Neykov, 2003)

$$\widehat{\theta}_H := \widehat{\theta}(\mathscr{I}_H) \quad \text{with} \quad \mathscr{I}_H \in \arg\max\{l_{\widehat{\theta}(\mathscr{I})}(x_{\mathscr{I}});\ \mathscr{I} \in \mathscr{J}_H\},$$

where $\mathscr{J}_H := \{\mathscr{I} \subset \{0,1,\ldots,N\};\ \sharp\mathscr{I} = N-H+1\}$ and $\sharp A$ denotes the number of elements of a set A. In the H-trimmed likelihood estimator, the H most unlikely observations are trimmed, i.e. not used. If $H = 0$, i.e. no observation is trimmed, then we get again the maximum likelihood estimator so that $\widehat{\theta}_0 = \widehat{\theta}$. The trimming number H should be chosen such that it is larger then the expected number of outliers. Values up to $\frac{N}{2}$ are possible, but it is very unlikely that almost the half of observations are outliers. Therefore, reasonable values for H are 10% to 30% of the sample size.

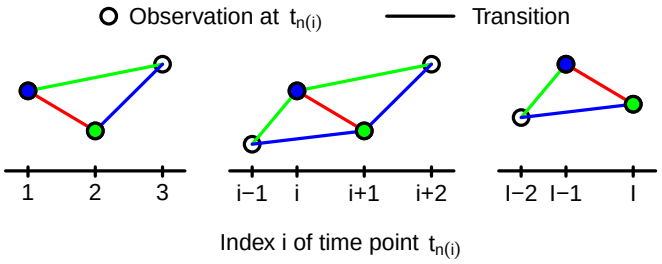

Fig. 1 Remaining transitions between observations (blue and green lines) if the observation at $t_{n(i)}$ (blue dot) or at $t_{n(i+1)}$ (green dot), respectively, is trimmed because the transition density $p_{i+1|i}$ (transition given by the red line) is small, for the case $i = 1$ (on the left), $i \in \{2,\ldots,I-2\}$ (in the middle), $i = I-1$ (on the right)

If N is small or H is very small then the H-trimmed likelihood estimator can be calculated by considering all subsets $\mathscr{I} \in \mathscr{J}_H$. If this is too time consuming then approximate algorithms based on a genetic algorithm and a concentration step as proposed in Neykov and Müller (2003) or based on a special selective iteration as proposed in Rousseeuw and Driessen (2006) can be used. However the concentration step is here much more complicated as in the case of independent observations since a single transition density $p_{i+1|i} := p_{\theta}(x_{t_{n(i+1)}}|x_{t_{n(i)}})$ is influenced by two observations, see Figure 1. Hence if $p_{\theta}(x_{t_{n(i+1)}}|x_{t_{n(i)}})$ is

small it is not clear whether $x_{t_{n(i+1)}}$ or $x_{t_{n(i)}}$ should be trimmed. Hence we propose the following procedure as concentration step.

As in the independent case, the concentration step starts with an initial subset $\mathscr{I}_0$ with $N-H+1$ elements and provides a new subset $\mathscr{I}_*$ with $N-H+1$ elements. Set for simplicity $\theta = \widehat{\theta}(\mathscr{I}_0)$ and define the transition densities

$$p_{j|i} := p_{j|i}(\mathscr{I}) := p_\theta(x_{t_{n(j)}}|x_{t_{n(i)}}) \quad \text{for} \quad j > i \quad \text{and} \quad p_i := p_{i+1|i}$$

for any set $\mathscr{I} = \{n(1),\ldots,n(I)\}$ with $0 \le n(1) < n(2) < \ldots < n(I) \le N$. The idea is now to start with the complete sample, i.e. $\mathscr{I}(0) = \{0,1,\ldots,N\}$, where no observation is trimmed. Then observations one after another are removed until a set $\mathscr{I}_*$ with $N-H+1$ elements is obtained. For that, only the transition densities $p_{j|i}$ depending on $\theta = \widehat{\theta}(\mathscr{I}_0)$ are used.

The algorithm for the concentration step is given in Algorithm 1. The last step mentioned in the last 3 lines before the output of this algorithm is necessary since $l_{\widehat{\theta}(\mathscr{I}(H))}(x_{\mathscr{I}(H)}) < l_{\widehat{\theta}(\mathscr{I}_0)}(x_{\mathscr{I}_0})$ could happen so that $\mathscr{I}(H)$ is worse than $\mathscr{I}_0$. This is in contrast to the concentration step for independent observations where the resulting trimmed likelihood is never worse then the starting trimmed likelihood.

A pseudo code of the genetic algorithm for the optimization is given by Algorithm 2.

The efficiency of the genetic algorithm depends heavily on the population size M, the maximum bound for the repetitions, and the efficiency of the concentration step. The complexity of the concentration step is mainly given by the difficulty of calculating the transition densities. If they are not given explicitly, as this is often the case but not in our approach, then they must be simulated. This could be very time consuming. Then the concentration step should not be used too often. This means that a smaller "population size" M and more "mutations" with higher k should be used. However, in general, the higher M and the maximum repetition bound is, the more likely it is to find the correct maximum. It is clear that a larger M and a larger maximum repetition bound increase the computation time drastically. Moreover, even for high M and high maximum repetition bound, it can happen that the correct maximum is not found. Hence as soon as it is possible to calculate all $\binom{N+1}{H}$ possible subsets $\mathscr{I} \in \mathscr{J}_H$ then this should be preferred.

Algorithm 1 Calculation for the concentration step

Input: $\mathscr{I}_0$, $\widehat{\theta}(\mathscr{I}_0)$, $t_0, t_1, \ldots, t_N$, $x_{t_0}, x_{t_1}, \ldots, x_{t_N}$, H.

$\mathscr{I}(0) = \{0, 1, \ldots, N\}$, $I(0) = N+1$

for $h = 0, 1, \ldots, H-1$ **do**

  $\mathscr{I} = \mathscr{I}(h)$, $I = I(h)$

  Determine the ordered transition densities inside $\mathscr{I} = \{n(1), \ldots, n(I)\}$, i.e.

$$p_{\pi(1)} \geq p_{\pi(2)} \geq \ldots \geq p_{\pi(I-1)} \text{ with } \{\pi(1), \pi(2), \ldots, \pi(I-1)\} = \{1, 2, \ldots, I-1\}.$$

  if $\pi(I-1) = 1$ **then** (see left figure in Figure 1)
    if $p_{3|1} \leq p_{3|2}$ **then** $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(1)\}$
    else $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(2)\}$
    end if
  end if

  if $\pi(I-1) = I-1$ **then** (see right figure in Figure 1)
    if $p_{I-1|I-2} \leq p_{I|I-2}$ **then** $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(I-1)\}$
    else $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(I)\}$
    end if
  end if

  if $i := \pi(I-1) \in \{2, \ldots, I-2\}$ **then** (see middle figure in Figure 1)
    if $p_{i+2|i} \cdot p_{i|i-1} \leq p_{i+2|i+1} \cdot p_{i+1|i-1}$ **then** $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(i)\}$
    else $\mathscr{I}(h+1) = \mathscr{I} \setminus \{n(i+1)\}$
    end if
  end if

  $h = h+1$ and $I(h) = I-1$.

end for

if $l_{\widehat{\theta}(\mathscr{I}(H))}(x_{\mathscr{I}(H)}) > l_{\widehat{\theta}(\mathscr{I}_0)}(x_{\mathscr{I}_0})$ **then** $\mathscr{I}_* = \mathscr{I}(H)$
else $\mathscr{I}_* = \mathscr{I}_0$
end if

Output: $\mathscr{I}_*$.

Algorithm 2 Genetic algorithm

Input: $t_0, t_1, \ldots, t_N, x_{t_0}, x_{t_1}, \ldots, x_{t_N}, H$.

Start with M sets $\mathscr{I}_1, \ldots, \mathscr{I}_M \in \mathscr{J}_H$.

repeat

  Concentration: Calculate $\mathscr{I}_{1*}, \ldots, \mathscr{I}_{M*}$ with the concentration procedure from $\mathscr{I}_1, \ldots, \mathscr{I}_M$.
  Replace $\mathscr{I}_1, \ldots, \mathscr{I}_M$ by $\mathscr{I}_{1*}, \ldots, \mathscr{I}_{M*}$, i.e. $\mathscr{I}_m \leftarrow \mathscr{I}_{m*}$, $m = 1, \ldots, M$.

  Mutation: Exchange in each set $\mathscr{I}_1, \ldots, \mathscr{I}_M \in \mathscr{J}_H$ randomly k elements.
  $\Rightarrow$ Get further sets $\mathscr{I}_{M+1}, \ldots, \mathscr{I}_{2M} \in \mathscr{J}_H$.

  Recombination: Choose randomly $N-H+1$ elements of unions $\mathscr{I}_m \cup \mathscr{I}_{M+m}$, $m = 1, \ldots, M$.
  $\Rightarrow$ Get further sets $\mathscr{I}_{2M+1}, \ldots, \mathscr{I}_{3M} \in \mathscr{J}_H$.

  Selection: Determine from $\mathscr{I}_1, \ldots, \mathscr{I}_{3M}$ the M sets with largest $l_{\widehat{\theta}(\mathscr{I})}(x_{\mathscr{I}})$.
  Rename them as $\mathscr{I}_1, \ldots, \mathscr{I}_M$.

until

$l_{\widehat{\theta}(\mathscr{I})}(x_{\mathscr{I}})$ is not improved anymore or a given number of repetitions is reached.

Output: $\widehat{\theta}(\mathscr{I})$ with largest $l_{\widehat{\theta}(\mathscr{I})}(x_{\mathscr{I}})$.

2.2 Performance measures based on trimmed likelihood estimators

To define performance measures for the goodness-of-fit of the models estimated with the H-trimmed likelihood estimator $\widehat{\theta}_H$, let be $\mathscr{I}_H = \{n(1), \ldots, n(N-H+1)\} \in \arg\max\{l_{\widehat{\theta}(\mathscr{I})}(x_{\mathscr{I}});\ \mathscr{I} \in \mathscr{J}_H\}$. If the transition density $p_\theta(x_{t_{n(i+1)}}|x_{t_{n(i)}})$ is known or is given as approximation then the conditional expectation $E_\theta(X_{t_{n(i+1)}}|X_{t_{n(i)}} = x_{t_{n(i)}})$ and the conditional quantiles

$$l_{i+1}(\theta) := F^{-1}_{n(i+1),\theta}\left(\frac{\alpha}{2}\middle| X_{t_{n(i)}} = x_{t_{n(i)}}\right),$$
$$u_{i+1}(\theta) := F^{-1}_{n(i+1),\theta}\left(1-\frac{\alpha}{2}\middle| X_{t_{n(i)}} = x_{t_{n(i)}}\right)$$

with $F_{n(i+1),\theta}(x) := P_\theta\left(X_{t_{n(i+1)}} \le x | X_{t_{n(i)}} = x_{t_{n(i)}}\right)$ can be determined.

The first performance measure is the median absolute deviation (MedAD) defined as

$$\text{MedAD} := \text{median}\left(\frac{|x_{t_{n(2)}} - E_{\widehat{\theta}_H}(X_{t_{n(2)}}|X_{t_{n(1)}} = x_{t_{n(1)}})|}{\Delta_1}, \ldots, \frac{|x_{t_{n(N-H+1)}} - E_{\widehat{\theta}_H}(X_{t_{n(N-H+1)}}|X_{t_{n(N-H)}} = x_{t_{n(N-H)}})|}{\Delta_{N-H}}\right).$$

The absolute deviations are divided by $\Delta_i := t_{n(i+1)} - t_{n(i)}$ to take into account the different time differences $t_{n(i+1)} - t_{n(i)}$ between the used observations.

The second performance measure is given by the $(1-\alpha)$-prediction intervals for $X_{t_{n(i+1)}}$ based on the former observations $x_{t_{n(i)}}$ for $i = 1, \ldots, N-H$. If θ is known then a $(1-\alpha)$-prediction interval for $X_{t_{n(i+1)}}$ is $[l_{i+1}(\theta), u_{i+1}(\theta)]$, i.e. $P_\theta(X_{t_{n(i+1)}} \in [l_{i+1}(\theta), u_{i+1}(\theta)]|X_{t_{n(i)}} = x_{t_{n(i)}}) \geq 1-\alpha$ is satisfied. If θ is unknown then an estimate for θ can be used. Here we use the H-trimmed likelihood estimator $\widehat{\theta}_H$ as plug-in estimate. As performance measure, the mean length or the coverage rate of the prediction intervals could be used. But it is better to use a combination of both. Hence the second performance measure is defined by

$$\text{IS}_\alpha := \frac{1}{N-H}\sum_{i=1}^{N-H} \frac{S_\alpha(l_{i+1}(\widehat{\theta}_H), u_{i+1}(\widehat{\theta}_H); x_{t_{n(i+1)}})}{\Delta_i}$$

where $S_\alpha(l,u;x) := (u-l) + \frac{2}{\alpha}(l-x)1_{\{x<l\}} + \frac{2}{\alpha}(x-u)1_{\{x>u\}}$ is the interval score of Gneiting and Raftery (2007) for prediction or confidence intervals. Thereby, $1_{\{\ldots\}}$ denotes the indicator function. For prediction, as used here, the interval score $S_\alpha(l,u;x)$ combines the length $u-l$ of the prediction interval $[l,u]$ with a penalty depending on α for the case that the predicted value x is not lying in $[l,u]$. Since larger time differences $t_{n(i+1)} - t_{n(i)}$ lead to larger prediction intervals and smaller coverage rates and thus larger interval scores, we again divide the interval scores by Δ_i.

Remark 1. For example, the conditional distribution of $X_{t_{n(i+1)}}$ given $X_{t_{n(i)}}$ is a normal distribution with mean $\left(x_{t_{n(i)}} + \frac{\theta_1}{\theta_2}\right)e^{\theta_2 t_{n(i+1)}} - \frac{\theta_1}{\theta_2}$ and variance $\frac{\theta_3^2(e^{2\theta_2 t_{n(i+1)}}-1)}{2\theta_2}$ for the Ornstein-Uhlenbeck process, a log-normal distribution with mean $x_{t_{n(i)}}e^{\theta_2 t_{n(i+1)}}$ and variance $x_{t_{n(i)}}^2 e^{2\theta_2 t_{n(i+1)}}\left(e^{\theta_3^2 t_{n(i+1)}} - 1\right)$ for the Geometric Brownian Motion, and non-central χ^2 distribution with mean $\left(x_{t_{n(i)}} + \frac{\theta_1}{\theta_2}\right)e^{\theta_2 t_{n(i+1)}} - \frac{\theta_1}{\theta_2}$ and variance $x_{t_{n(i)}}\frac{\theta_3^2(e^{2\theta_2 t_{n(i+1)}} - e^{\theta_2 t_{n(i+1)}})}{\theta_2} + \frac{\theta_1\theta_3^2(1-e^{\theta_2 t_{n(i+1)}})^2}{2\theta_2^2}$ for the Cox-Ingersoll-Ross process (see e.g. Iacus, 2008).

Remark 2. If the conditional distributions of $X_{t_{n(i+1)}}$ given $X_{t_{n(i)}}$ are not known, then approximations of the SDE can be used. The Euler-Maruyama approximation provides for example (see e.g. Iacus, 2008)

$$X_{t_{n(i+1)}} - X_{t_{n(i)}} \approx b(X_{t_{n(i)}}, \theta)\,\Delta_i + s(X_{t_{n(i)}}, \theta)\,\sqrt{\Delta_i}\,E_i$$

where E_i has a standard normal distribution so that $p_\theta(x_{t_{n(i+1)}} | x_{t_{n(i)}}) \approx p_{N(\mu_i,\sigma_i^2)}$, where $\mu_i := x_{t_{n(i)}} + b(x_{t_{n(i)}}, \theta)\,\Delta_i$, $\sigma_i := s(x_{t_{n(i)}}, \theta)\,\sqrt{\Delta_i}$, and $p_{N(\mu,\sigma^2)}$ is the density of the normal distribution with expectation μ and variance σ^2. In particular, we have $E_\theta(X_{t_{n(i+1)}} | X_{t_{n(i)}} = x_{t_{n(i)}}) \approx x_{t_{n(i)}} + b(x_{t_{n(i)}}, \theta)\,\Delta_i$. Hence the H-trimmed likelihood estimator can be calculated via the densities of the approximated normal distributions and the performance measures can be based on the expectations and quantiles of the approximated normal distributions.

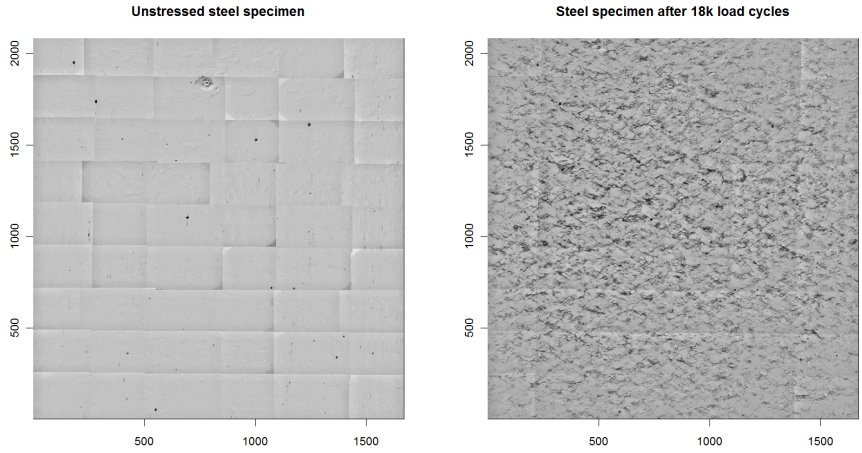

Fig. 2 Surface of an unstressed steel specimen (left) and after 18 000 load cycles (right) of a tension-compression-experiments with an external stress of 400 MPa ($= 400\,N/mm^2$)

3 Application to crack growth data from photos

3.1 Obtaining crack growth data from photos

Figure 2 shows two photos of the surface of a steel specimen (Specimen 31), one before the specimen was exposed to cyclic load and one after 18 000 load cycles. During the 18 000 load cycles, a large number of micro cracks has appeared, visible by lower (blacker) pixel values. There exist several other photos at other time points, in this case for example after 1 000, 2 000, 3 000, 4 000, 5 000, 6 000, 7 000, 8 000, 9 000, 10 000, 12 000, 14 000, 16 000 load cycles. For more details of the photos and the underlying experiment, see Müller et al (2011).

The first step is to detect the micro cracks in each of these photos by an existing crack detection algorithm as given for example by Purcell (1983), Cheu (1984), Buckley and Yang (1997), Fletcher et al (2003), Iyer and Sinha (2005), Fujita et al (2006), Yamaguchi and Hashimoto (2010), Gunkel et al (2012), Wilcox et al (2016), Amhaz et al (2016). All crack detection methods using black-and-white photos base on the fact that cracks appear as darker areas in the photos. Therefore, so called crack clusters can be defined as connected sets of pixel positions where the pixels values are lying below a given threshold value. Simple crack detection methods as the UTHSCSA Image Tool of Wilcox et al (2016) use the orientation and maximum length of smallest rectangles and ellipses containing the crack clusters as crack orientation and crack length. However, the precision concerning the crack length is not high in such approaches since the real crack lengths are often much larger. This happens in particular when the crack has a zigzag structure or a tree like structure with ramifications. Hence, more sophisticated crack detection methods are necessary to detect such structures. For example, the method of Amhaz et al (2016) is able to detect cracks with their ramifications. However, this method does not provide a measure for the length of a crack. The length of a crack is given by the crack detection algorithm of the package `crackrec` of Gunkel et al (2012), which is an R package (R Core Team, 2015) and free available at Müller (2016). This algorithm determines at first the so called *crack clusters*. Then a so called *crack path* in a crack cluster is the longest path which can be found by Dijkstra's shortest path algorithm connecting arbitrary pixel positions of the crack cluster. This method was used here in the first step for detecting the cracks.

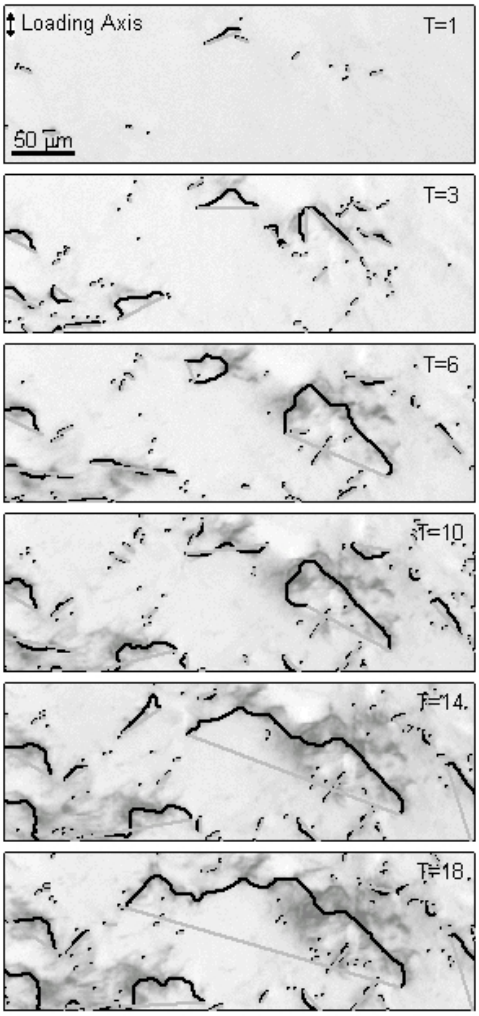

Fig. 3 Detected cracks in a cutout of the image after 1 000, 3 000, 6 000, 10 000, 14 000, 18 000 load cycles where T stands for the time in 1 000 load cycles

Figure 3 shows the detected cracks after 1 000, 3 000, 6 000, 10 000, 14 000, 18 000 load cycles in a cutout of the images. The detected paths are marked in black in Figure 3. The start and the end point of a crack path are connected by a straight line to highlight the crack paths. This figure shows clearly how the number of detected cracks increases and how existing cracks becomes longer

when the number of load cycles increases. Thereby, cracks can become longer also by the fusion of two or more cracks. Similar results will be obtained by other crack detection tools which provide crack lengths.

The second step is to backtrack large cracks which were detected at the end. Large cracks could mean large crack clusters or long crack paths if paths are detected. However, since all crack paths are surrounded by crack clusters and paths are thin, the backtracking is based on crack clusters. Hence this step can be performed also by crack detection methods which provide only crack clusters as that of Wilcox et al (2016).

The backtracking step is iterative. Assume that there are time points $t_0 < t_1 < \ldots < t_N$ for which detected cracks exist. For any chosen crack at time t_n with $1 \leq n \leq N$, all detected crack clusters at time t_{n-1} are calculated which intersect with the chosen crack cluster. Then the largest crack is chosen as the predecessor of the chosen crack and becomes the starting crack for the next iteration. Thereby, largest crack can mean the cluster with the largest number of pixels or the cluster with the longest detected path in the cluster. The K largest cracks at the last time point t_N are used as starting cracks.

If, for example, cracks are determined by `crackrec` of Gunkel et al (2012) then the K_n detected crack clusters $C_n(1), \ldots C_n(K_n)$ of an image at time t_n are given as a list called `crackclusters` which includes K_n matrices $M_n(k) \in \Re^{2\times n_n(k)}$ of the corresponding pixel positions for $k = 1, \ldots, K_n$. Moreover, the list element `cracks` is a $6 \times K_n$ matrix which provides, for each of the K_n clusters, the number $n_n(k)$ of pixels in the cluster, the length of the detected crack path in the cluster and the start and end points of the detected crack path. Hence, the largest cracks can easily be determined independently of whether the size is measured in number of pixels of the cluster or the length of the crack path.

Assume that $C_N(1), \ldots C_N(K)$ are the K largest clusters at the last time point t_N. Then the proposed algorithm is given by Algorithm 3.

The iteration Line 11 to Line 38 of this algorithm is demonstrated in Figure 4. On the right-hand side of this figure, the chosen crack k at a time point $t_n = 18\,000$ load cycles is given in green by its crack path. It is seen that the surrounding black area does not follow completely the path so that the corresponding crack cluster is much larger. The left-hand side of Figure 4 provides the detected crack paths at time point $t_{n-1} = 16\,000$ load cycles which is the nearest predecessor time point with an available photo. All detected crack paths are marked in blue, red, and green. All paths of the predecessor candidates, i.e. of crack clusters which intersect with the chosen crack cluster on the right-

Algorithm 3 Backtracking algorithm

Input: List of crack clusters given by $M_n(k) \in \Re^{2\times n_n(k)}$ for $k = 1,\dots,K_n$, $n = 1,\dots,N$, list of corresponding crack lengths/sizes $L_n(k) \in \Re$ for $k = 1,\dots,K_n$, $n = 1,\dots,N$, K number of largest cracks at time N which should be backtracked.

$P = NULL$ ▷ *Contains the cluster numbers of all largest predecessors of clusters* $k = 1,\dots,K$.

for $k_0 = 1,\dots,K$ **do**
  $n = N$
  $P(k_0) = NULL$ ▷ *Contains the cluster numbers of all largest predecessors of cluster* k_0.

  while $n \geq 1$ **do**

    if $n = N$ **then** $k = k_0$.
    end if
    $p = FALSE$ ▷ *Indicates whether cluster* k *at time* n *has a predecessor at time* $n-1$.
    $P_k = NULL$ ▷ *Contains the cluster numbers of all predecessors of cluster* k.

    for $\nu = 1,\dots,n_n(k)$ **do**, ▷ *i.e. for any column* $M_n(k)[,\nu] \in \Re^2$ *of* $M_n(k)$ *do,*
      for $j = 1,\dots,K_{n-1}$ **do**
        $\mu = 1$
        while $\mu \leq n_{n-1}(j)$ **do**
          if $M_n(k)[,\nu] = M_{n-1}(j)[,\mu]$ **then**,
            ▷ *i.e. if column* $M_n(k)[,\nu]$ *equals to a column of* $M_{n-1}(j)$, *i.e. if*
            ▷ cluster k of time n and cluster j of time $n-1$ contain the same pixel,
            $p = TRUE$
            $P_k = c(P_k, j)$ ▷ *Cluster number* j *is added to the vector of predecessors.*
            $\mu = n_{n-1}(j) + 1$ ▷ *Stops the loop since cluster* j *is predecessor.*
          else $\mu = \mu + 1$
          end if
        end while
      end for
    end for

    if p=FALSE **then**, ▷ *i.e. cluster* k *has no predecessor,*
      $n = 0$ ▷ *Stops the loop over the time points.*
    else
      $j_0 = \arg\max\{L_{n-1}(j);\ j \text{ in } P_k\}$
        ▷ *Determines* j_0 *as the largest crack within the predecessor candidates.*
      $k = j_0$, $P(k_0) = c(P(k_0), j_0)$, $n = n-1$
    end if

  end while

  $P = list(P, P(k_0))$
end for

Output: P, i.e. list of vectors $P_{k_0} \in \Re^{m(k_0)}$ containing the $m(k_0)$ numbers of predecessor crack clusters of crack cluster k_0 at the end, $k_0 = 1,\dots,K$.

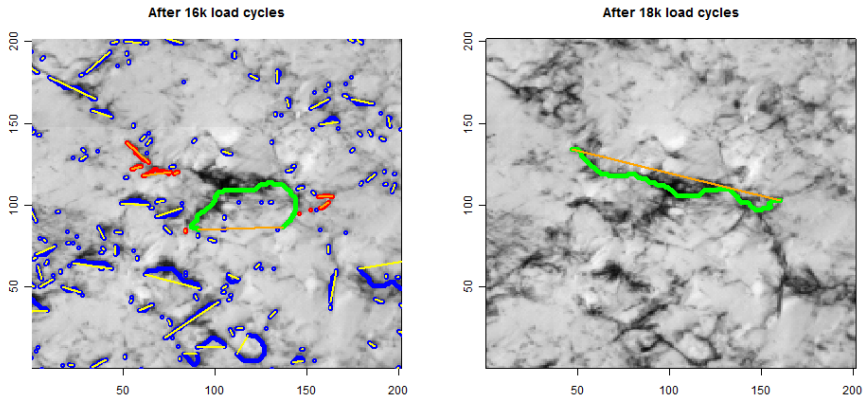

Fig. 4 Right: Chosen crack in green after 18 000 load cycles. Left: All detected cracks paths after 16 000 load cycles in blue, red, green. All predecessor cracks of the crack on the right in red and green, and the largest predecessor in green.

hand side, are marked in red and green. Clearly the green one is the longest predecessor so that it is chosen as predecessor crack and provides the new k for the next iteration. Its crack path is quite different from the green crack on the right-hand side since the crack cluster on the right-hand side is not given around one line. But it is the predecessor crack in any case, using the number of pixels in the cluster as well as the path length as crack size.

Note that a high increase of crack lengths happens in particular if two or more crack clusters become connected. The larger crack clusters are the more likely the union of several crack clusters is. Hence in the beginning (e.g. for load cycles from 6 000 to 10 000 in Figure 3), there are not much changes in the crack clusters and crack lengths, and these changes are mainly caused by growth of the single crack clusters. Much more changes appear later (e.g. for load cycles 10 000 to 18 000 in Figures 3 and 4) when crack clusters are merged.

For getting crack growth curves, one can use the number of pixels of the crack cluster as well as the length of the crack path in the cluster independently how the predecessor crack was obtained. Figure 5 shows two resulting crack growth curves based on lengths of crack paths. The one on the left-hand side looks quite reasonable since it is almost strictly increasing. However, the one on the right-hand side is not a real growth curve. Deviations from a strictly

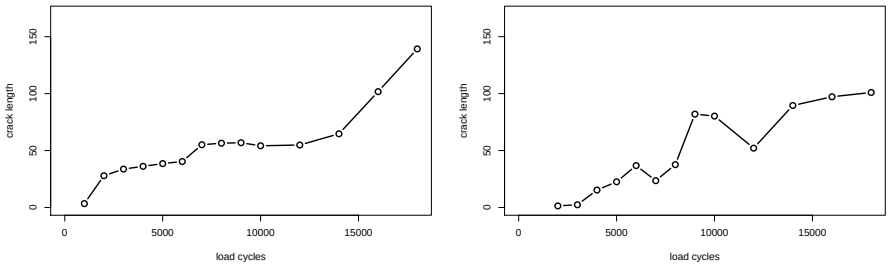

Fig. 5 Two resulting crack growth curves

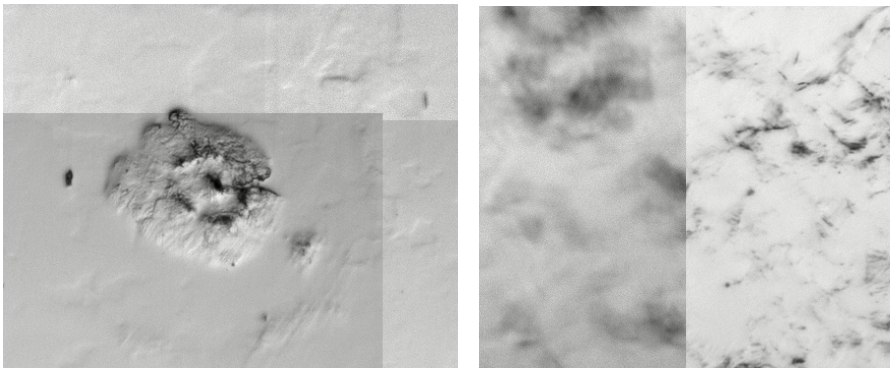

Fig. 6 A large contamination (left) and different sharpness of photos (right)

increasing growth curve are caused by several sources of errors which appear in the automatic calculation of the crack growth from the photos:

- A single image may consist of several photos as can be seen easily in the left-hand image of Figure 2, where 54 single photos were pieced together. This is sometimes necessary when the area of interest is so large that it could not be caught by one photo. The boundaries of the singles photos are clearly visible because of shadows at the boundaries and different illuminations. One error source in that particular case is that the pieces are not put together exactly. This can split a single cracks in several cracks or several disconnected cracks are detected as one crack.
- Images at different time points may differ in their location so that they have to be shifted so that the pixel positions correspond to the same part of the

image. The calculation of the shift may be erroneous. This can lead to misspecified backtracked clusters.

- Shadows and different illuminations cause problems in detecting the crack clusters. This can happen between images at different time points but also between different pieces of an image as can be seen in the left-hand side of Figure 6. This can influence the detection of cracks as well as the backtracking of clusters.
- The sharpness of the single photos can differ between time points and pieces as can be seen in the right-hand side of Figure 6. An influence on the detection of cracks and the backtracking of clusters is here also possible.
- The surface usually contains some pits, scratches and other contaminations of the material which are falsely detected as cracks by an automatic crack detection method. Pits are visible in the left-hand image of Figure 2 as black spots. A big contamination can be seen in the top of this image. This is also presented in the left-hand side of Figure 6. It provides already in the first image a large crack which shows almost no increase of growth over time. Moreover the corresponding "crack" is included in the 100 largest cracks detected at the end. Hence, some detected cracks are no real cracks. This can influence the backtracking of crack clusters. However, more important is that some long cracks detected at the end may not be cracks.

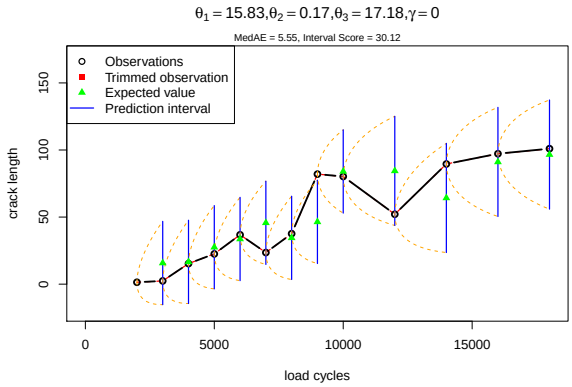

Fig. 7 Untrimmed predicted (expected) values and prediction intervals using an OU model for the curve on the right-hand of Figure 5

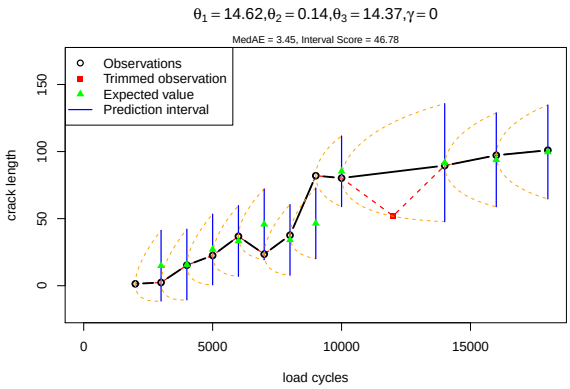

Fig. 8 10% trimmed predicted (expected) values and prediction intervals using an OU model for the curve on the right-hand of Figure 5

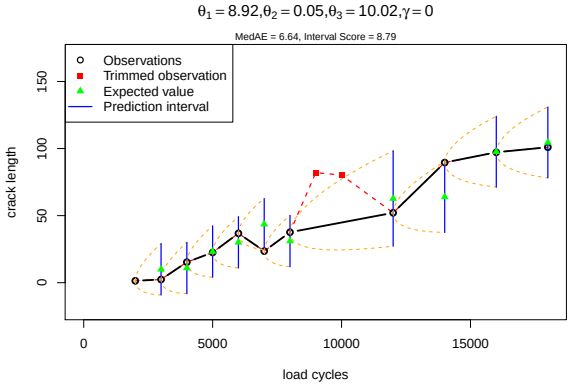

Fig. 9 20% trimmed predicted (expected) values and prediction intervals using an OU model for the curve on the right-hand of Figure 5

3.2 Performance measures for the crack growth data

Figure 7 shows the predicted (expected) values and prediction intervals using the classical untrimmed maximum likelihood estimator for an Ornstein-Uhlenbeck process (OU) for the path on the right-hand side of Figure 5. This path is not monotone because of extreme outlying observations. These outliers cause bad fits for the Ornstein-Uhlenbeck process using the classical maximum likelihood estimator so that the performance measure MedAE is equal to 5.55 and the performance measure based on the interval score $IS_{0.05}$ is 30.12. Using an 10%-trimmed estimator leads to the result shown in Figure 8 with improved MedAE of 3.45 because one observation is trimmed. However, the

performance measure based on the interval score has increased to 48.78 since the prediction intervals became smaller so that the 7'th prediction interval is more far away from the observed value as for the untrimmed estimator. If 20%, i.e. two observations, are trimmed then the performance measure based on the interval score has decreased to the very small value of 8.79 since all prediction intervals include the observations as seen in Figure 9. However, here the MedAE is worse than for using the untrimmed estimator. This extreme example shows that trimming some few observations improves the two performance measures differently. Here, a higher trimming rate is necessary to improve both performance measures simultaneously.

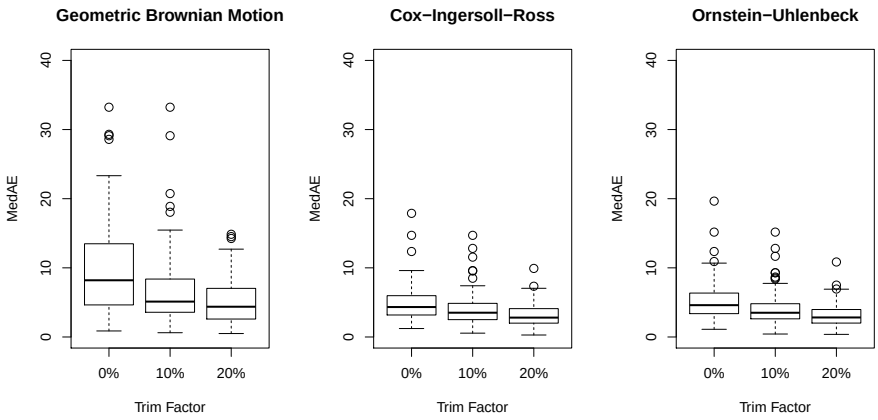

Fig. 10 Boxplots of the median absolute deviations (MedAD) of the growth curves of the 100 largest cracks in Specimen 31 fitted by three SDEs without trimming, 10% trimming and 20% trimming

However, most crack growth curves are not so contaminated as the one on the right-hand side of Figure 5. 20% trimming and even 10% trimming lead to reasonable fits of an Ornstein-Uhlenbeck process, a Cox-Ingersoll-Ross process, or a Geometric Brownian Motion for the majority of the 100 crack growth curves which are backtracked from the 100 largest detected cracks at the end. This can be seen from the boxplots in Figure 10 for the 100 obtained performance measures based on the median absolute deviation (MedAD) and in Figure 11 for the 100 obtained performance measures $\mathrm{IS}_{0.05}$ based on the interval score. Thereby, 20% trimming leads to the best result for both performance measures, and this is independent whether an Ornstein-Uhlenbeck

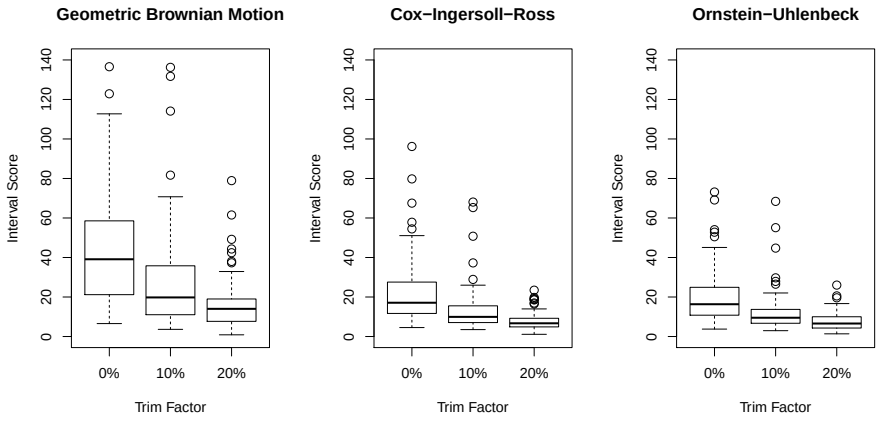

Fig. 11 Boxplots of the interval score $IS_{0.05}$ of the growth curves of the 100 largest cracks in Specimen 31 fitted by three SDEs without trimming, 10% trimming and 20% trimming

process, a Cox-Ingersoll-Ross process, or a Geometric Brownian Motion is fitted. However, the Geometric Brownian Motion provides the largest performance meausure while the performance measures for the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross process are very similar. The same result was obtained when the method was applied to a series of photos of another specimen (Specimen 10) which was exposed to lower stress so that photos are availabe at 29 time points and the crack growth curves are more flat.

The boxplots indicate that the performance measures of the growth curves of the 100 largest cracks do not follow a normal distribution which was also confirmed by Shapiro-Wilk tests. Therefore, to test whether the trimming improves the performance, a closed testing principle based on the H-test (Kruskal-Wallis test) and followed by two Wilcoxon rank-sum tests (Mann-Whitney U-tests) was applied. Table 1 clearly shows that there is a significant difference between no trimming, 10% trimming, and 20% trimming for both performance measures and all three regarded processes. Again only the results for Specimen 31 are presented in Table 1. But for Specimen 10, the P-values are either the same or even smaller.

Table 2 provides the P-values of a closed testing principle based on the H-test and followed by two Wilcoxon rank-sum tests for testing the equality of the performance measures for the three processes if 20% trimming is used. The

Table 1 P-values of H-tests and Wilcoxon-Rang-Sum tests for the performance measures obtained from Specimen 31 based on no trimmimg (P_0), 10% trimming (P_{10}), and 20% trimming (P_{20})

Null hypothesis Test procedure	$P_0 = P_{10} = P_{20}$ H-test	$P_0 \leq P_{10}$ Rank-Sum	$P_{10} \leq P_{20}$ Rank-Sum
MedAE (OU)	< 0.0001	0.0005	0.0010
MedAE (CIR)	< 0.0001	0.0012	0.0052
MedAE (GBM)	< 0.0001	< 0.0001	0.0125
Interval Score (OU)	< 0.0001	< 0.0001	< 0.0001
Interval Score (CIR)	< 0.0001	< 0.0001	< 0.0001
Interval Score (GBM)	< 0.0001	< 0.0001	< 0.0001

results are shown for Specimen 31, but are the same for Specimen 10. As indicated by the boxplots, there is no significant difference between the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross process. However, both differ significantly from the Geometric Brownian Motion. An explanation for this result is the number of unknown parameters. While the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross process possess three unknown parameters, the Geometric Brownian Motion has only two unknown parameters because the slope term satisfies $\theta_1 = 0$. This makes the Geometric Brownian Motion less flexible for fitting data and the slope term seems to be important for the fit.

Table 2 Comparison between the performance measures P_{OU}, P_{CIR}, P_{GBM} applied to the three stochastic processes with 20% trimming in Specimen 31

Nullhypothesis	Test	MedAE	Interval Score
$P_{OU} = P_{CIR} = P_{GBM}$	H-Test	< 0.0001	< 0.0001
$P_{OU} = P_{CIR}$	Rank-Sum	0.8498	0.6087
$P_{OU} = P_{GBM}$	Rank-Sum	< 0.0001	< 0.0001
$P_{CIR} = P_{GBM}$	Rank-Sum	< 0.0001	< 0.0001

Finally, it was tested whether the three parameters of the processes with best fit with 20% trimming, i.e. the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross, are different for the two specimens. The estimated parameters are given in Table 3 and the test results in Table 4. This shows that there is no significant difference in the drift term θ_2 whereas the slope term θ_1 and the diffusion term θ_3 are significantly higher for higher stress. Note that Sepcimen 10 was exposed to an external stress of 360 MPa and Specimen 31 to 400 MPa.

Table 3 Median of the estimated parameters of the Ornstein-Uhlenbeck processes and the Cox-Ingersoll-Ross processes using 20% trimming for Specimens 10 and 31

Specimen (Process)	med($\hat{\theta}_1$)	med($\hat{\theta}_2$)	med($\hat{\theta}_3$)
10 (OU)	2.2779	0.0814	1.6697
31 (OU)	6.4637	0.0799	3.4209
Difference	**-4.1858**	0.0016	**-1.7512**
10 (CIR)	2.0858	0.0709	0.4636
31 (CIR)	6.5832	0.0985	0.7547
Difference	**-4.4974**	−0.0275	**-0.2911**

Table 4 P-values for a two-sided Wilcoxon-Rank-Sum test in order to check whether the estimated parameters are significantly different between Specimens 10 and 31 for the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross process.

Nullhypothesis	$\theta_1^{(10)} = \theta_1^{(31)}$	$\theta_2^{(10)} = \theta_2^{(31)}$	$\theta_3^{(10)} = \theta_3^{(31)}$
Ornstein-Uhlenbeck	< 0.0001	0.5486	< 0.0001
Cox-Ingersoll-Ross	< 0.0001	0.9229	< 0.0001

4 Discussion

We introduced trimmed likelihood estimators for processes given by stochastic differential equations and showed how they can be computed efficiently. To study their performance on a large data set, we proposed an automatic detection method to obtain crack growth data from a series of photos by backtracking large cracks detected in the last photo. The application of the trimmed likelihood estimators to these data showed that these estimators can deal with a high amount of contamination of the data caused by the automatic detection method. In particular, the fits obtained of 100 crack growth curves measured by two proposed performance measures were significantly better for trimming 20% than 10% of the data of a growth curve. For simplicity, only the fits of the Ornstein-Uhlenbeck process, the Cox-Ingersoll-Ross process, and the Geometric Brownian Motion were studied. But similarly other processes can be fitted. Within the regarded three processes, the Ornstein-Uhlenbeck process and the Cox-Ingersoll-Ross process provided the best fits and estimated parameters which differ significantly between different stress conditions. Hence the influence of the stress conditions on these parameters may be used in fu-

ture work to predict the damage development of material by analyzing cracks detected from photos as described in this paper.

Acknowledgements The research was supported by the DFG Collaborative Research Center SFB 823 *Statistical modeling of nonlinear dynamic processes*. We also thank the reviewers and the editor for their helpful comments and suggestions which improved the presentation of the paper.

References

Amhaz R, Chambon S, Idier J, Baltazart V (2016) Automatic crack detection on 2d pavement images: An algorithm based on minimal path selection. IEEE Transactions on Intelligent Transportation Systems 17:2718–2729

Beskos A, Papaspiliopoulos O, Roberts GO, Fearnhead P (2006) Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. Journal of the Royal Statistical Society Series B 68:333–382

Buckley M, Yang J (1997) Regularised shortest-path extraction. Pattern Recognition Letters 18:621–629

Castillo E, Fernández-Canteli A (2009) A Unified Statistical Methodology for Modeling Fatigue Damage. Springer, Dordrecht

Cheng TC, Biswas A (2008) Maximum trimmed likelihood estimator for multivariate mixed continuous and categorical data. Computational Statistics & Data Analysis 52:2042–2065

Cheu YF (1984) Automatic crack detection with computer vision and pattern recognition of magnetic particle indications. Materials Evaluation 42:1506–1510, 1514

Chiquet J, Limnios N, Eid M (2009) Piecewise deterministic Markov processes applied to fatigue crack growth modelling. Journal of Statistical Planning and Inference 139:1657–1667

Fletcher DI, Franklin FJ, Kapoor A (2003) Image analysis to reveal crack development using a computer simulation of wear and rolling contact fatigue. Fatigue & Fracture of Engineering Materials & Structures 26:957–967

Fujita Y, Mitani Y, Hamamoto Y (2006) A method for crack detection on a concrete structure. Proceedings - International Conference on Pattern Recognition 3:901–904

Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association 102(477):359–378

Gunkel C, Stepper A, Müller AC, Müller CH (2012) Micro crack detection with Dijkstra's shortest path algorithm. Machine Vision and Applications 23:589–601

Hadi AS, Luceño A (1997) Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. Computational Statistics & Data Analysis 25:251–272

Hermann S, Ickstadt K, Müller CH (2016a) Bayesian prediction for a jump diffusion process with application to crack growth in fatigue experiments. Reliability Engineering and System Safety DOI 10.1016/j.ress.2016.08.012

Hermann S, Ickstadt K, Müller CH (2016b) Bayesian prediction of crack growth based on a hierarchical diffusion model. Applied Stochastic Models in Business and Industry 32:494–510

Höök LJ, Lindström E (2016) Efficient computation of the quasi likelihood function for discretely observed diffusion processes. Computational Statistics & Data Analysis 103:426–437

Iacus SM (2008) Simulation and Inference for Stochastic Differential Equations. With R Examples. Springer, New York

Iyer S, Sinha SK (2005) A robust approach for automatic detection and segmentation of cracks in underground pipeline images. Image and Vision Computing 23:921–933

Kustosz CP, Müller CH (2014) Analysis of crack growth with robust, distribution-free estimators and tests for nonstationary autoregressive processes. Statistical Papers 55:125–140

Lin YK, Yang JN (1983) On statistical moments of fatigue crack propagation. Engineering Fracture Mechanics 18:243–256

Müller CH (2016) Software-Pakete / Software Packages. Fakultät Statistik, Technische Universität Dortmund, URL `https://www.statistik.tu-dortmund.de/1253.html`

Müller CH, Neykov N (2003) Breakdown points of trimmed likelihood estimators and related estimators in generalized linear models. Journal of Statistical Planning and Inference 116:503–519

Müller CH, Gunkel C, Denecke L (2011) Statistical analysis of damage evolution with a new image tool. Fatigue & Fracture of Engineering Materials & Structures 34:510–520

Müller CH, Szugat S, Celik N, Clarke BR (2016) Influence functions of trimmed likelihood estimators for lifetime experiments. Statistics 50:505–524

Neykov N, Müller CH (2003) Breakdown point and computation of trimmed likelihood estimators in generalized linear models. In: Dutter R, Filzmoser P, Gather U, Rousseeuw PJ (eds) Developments in Robust Statistics, Physica-Verlag, Heidelberg, pp 277–286

Neykov N, Filzmoser P, Dimova R, Neytchev P (2007) Robust fitting of mixtures using the trimmed likelihood estimator. Computational Statistics & Data Analysis 52:299–308

Neykov N, Filzmoser P, Neytchev P (2014) Ultrahigh dimensional variable selection through the penalized maximum trimmed likelihood estimator. Statistical Papers 55:187–207

Nicholson DW, Ni P, Ahn Y (2000) Probabilistic theory for mixed mode fatigue crack growth in brittle plazes with random cracks. Engineering Fracture Mechanics 66:305–320

Ortiz K, Kiremidjian AS (1988) Stochastic modeling of fatigue crack growth. Engineering Fracture Mechanics 29:317–334

Pastorello S, Rossi E (2010) Efficient importance sampling maximum likelihood estimation of stochastic differential equations. Computational Statistics & Data Analysis 54:2753–2762

Pedersen AR (1995) A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations. Scandinavian Journal of Statistics 22:55–71

Pook L (2000) Linear Elastic Fracture Mechanics for Engineers: Theory and Application. WIT Press, Southhampton

Purcell D (1983) Automatic crack detection. Sensor Review 3:130–131

R Core Team (2015) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL `https://www.R-project.org/`

Ray A, Tangirala S (1996) Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. IEEE Transactions of Control Technology 4:443–451

Rousseeuw PJ (1984) Least median of squares regression. Journal of the American Statistical Association 79:871–880

Rousseeuw PJ, Driessen K (2006) Computing LTS regression for large data sets. Data Mining and Knowledge Discovery 12:29–45

Rousseeuw PJ, Leroy AM (1987) Robust Regression and Outlier Detection. John Wiley, New York

Sánchez-Silva M, Klutke GA (2016) Reliability and Life-Cycle Analysis of Deteriorating Systems. Springer, Heidelberg

Sobczyk K, Spencer BF (1992) Random Fatigue. From Data to Theory. Academic Press, Boston

Sørensen H (2004) Parametric inference for diffusion processes observed at discrete points in time: a survey. International Statistical Review 72:337–354

Sun L, Lee C, Hoeting JA (2015) A penalized simulated maximum likelihood approach in parameter estimation for stochastic differential equations. Computational Statistics & Data Analysis 84:54–67

Virkler D, Hillberry B, Goel P (1979) The statistical nature of fatigue crack propagation. Journal of Engineering Materials and Technology 101:148–153

Wilcox CD, Dove SB, McDavid WD, Greer DB (2016) UTHSCSA Image Tool. Department of Dental Diagnostic Science, University of Texas Health Science Center, San Antonio, URL `http://compdent.uthscsa.edu/dig/itdesc.html`

Wu WF, Ni CC (2004) Probabilistic models of fatigue crack propagation and their experimental verification. Probabilistic Engineering Mechanics 19:247–257

Yamaguchi T, Hashimoto S (2010) Fast crack detection method for large-size concrete surface images using percolation-based image processing. Machine Vision and Applications 21:797–809

Zárate BA, Caicedo JM, Yu J, Ziehl P (2012) Bayesian model updating and prognosis of fatigue crack growth. Engineering Structures 45:53–61