

Automatisierte Risserkennung in Verbundwerkstoffen

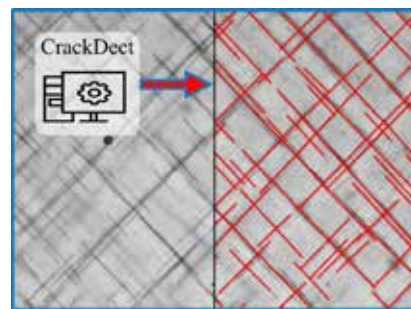
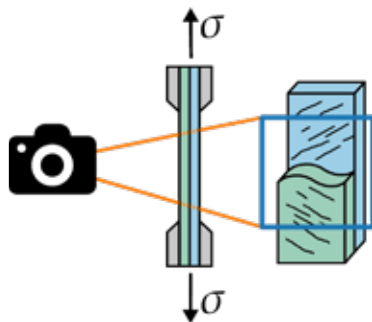
Automated crack detection in composite materials

Bei zyklisch beanspruchten Bauteilen aus Verbundwerkstoffen bilden sich oft früh Matrixrisse in Schichten, die quer zur Faserrichtung beansprucht werden. Diese Risse breiten sich entlang der Fasern aus und sind Auslöser für weitere Schädigung im Verbundwerkstoff. Um die Effekte dieser Risse genauer zu studieren, wurde CrackDect entwickelt, ein frei verfügbares Paket für die Risserkennung in Verbundwerkstoffen^[1].

CrackDect kann beliebig viele gerade Risse in eine vorgegebene Richtung erkennen und wurde speziell für den Einsatz von glasfaserverstärkten Verbundwerkstoffen entwickelt, da die transparenten Glasfasern die Erkennung von Rissen auch in eingebetteten Schichten ermöglichen. Neben mehreren Algorithmen für die Rissdetektion sind Methoden für die Vorbearbeitung der Rissbilder und das effiziente Bearbeiten von ganzen Bildserien inkludiert. Dies ist essenziell für die Auswertung von Ermüdungsexperimenten, da hier große Datenmengen anfallen. Ohne automatisierte Methoden wäre eine Auswertung, wie sie CrackDect ermöglicht, nicht in akzeptabler Zeit durchführbar. CrackDect wurde mittlerweile erfolgreich für die Auswertung von Ermüdungsexperimenten in einem FFG COMET Projekt genutzt, dessen Ergebnisse mittlerweile in mehrere Publikationen einfließen.

Die aus der Rissdetektion gewonnenen Ergebnisse fließen nun in die Weiterentwicklung eines Ermüdungsmodells für endlosfaserverstärkte Polymere ein. Gerade hier ist es wichtig, die Auswirkung von Schädigung, wie in diesem Beispiel Matrixrisse, auf die Materialeigenschaften messen zu können, um physikalisch basierte Modelle zu entwickeln. Weiters werden die Auswirkungen von Matrixrisse auf Phänomene wie hysteretische Erwärmung und dissipierte Energie sowie die Verbindung von

Matrixrisse und diffuser Delamination genauer untersucht. Dies ist nun durch CrackDect in einfacher und zugänglicher Weise für Matrixrisse möglich.



In cyclically loaded composite components, matrix cracks often form early in layers that are stressed transverse to the fiber direction. These cracks propagate along the fibers and are starters for further damage in the composite. To study the effects of these cracks in more detail, CrackDect, a freely available package for crack detection in composites, was developed^[1].

CrackDect can detect any number of straight cracks in a given direction and was developed specifically for glass fiber-reinforced composites, since the transparent glass fibers allow cracks to be detected even in embedded layers. In addition to several algorithms for crack detection, methods for the pre-processing of crack images and the efficient processing of entire image series are included. This is essential for the evaluation of fatigue experiments, as large amounts of data are generated. Without automated methods, an evaluation like CrackDect would not be feasible in an acceptable time. CrackDect has been successfully used for the evaluation of fatigue experiments in an FFG COMET project, the results of which have meanwhile been included in several publications.

The results obtained from crack detection are now being incorporated into

the further development of a fatigue model for continuous fiber-reinforced polymers. It is important to be able to measure the effect of damage, such as matrix cracks in this example, on the material properties in order to develop physically-based models. Furthermore,

the effects of matrix cracking on phenomena such as hysteretic heating and dissipated energy, as well as the connection between matrix cracking and diffuse delamination, are being studied in more detail. This is now possible in

a simple and accessible way for matrix cracks through CrackDect. ■

^[1] M. Drvoderic, M. Rettl, M. Pletz, and C. Schuecker, CrackDect: Detecting crack densities in images of fiber-reinforced polymers, SoftwareX, Bd. 16, S. 100832, Dez. 2021.



Auf einen Blick

rie

Ansprechpartner



as Drvoderic

@unileoben.ac.at