Analysis of crack growth behavior in a double cantilever beam adhesive fracture test by different digital image processing techniques

V. Richter-Trummer¹, E. A. Marques¹, F. J. P. Chaves¹, J. M. R. S. Tavares¹, L. F. M. da Silva¹, and P. M. S. T. de Castro¹

¹FEUP - Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, Porto, Portugal

date of the initial version: 16.6.2010
date of last modification: 20.9.2010, 14:07
revision: 2
state: final
contact author: valentin@fe.up.pt

Abstract

Double cantilever beam (DCB) tests are common experiments in the field of adhesive fracture mechanics. During such an experiment, both the load and the crack length have to be monitored. Digital image processing techniques offer interesting possibilities in various fields of science. Automated analyses may significantly reduce the necessary manpower for certain cumbersome tasks. The analysis of large series of images may be done in less time, since automated image processing techniques are able to work efficiently and with constant quality 24h per day. In this work, a series of images obtained by a high-speed camera is analyzed in order to determine the crack growth behavior during a DCB test. The present work represents a contribution to the effort of automating the crack growth measurement, by proposing and comparing different automated techniques.

Keywords

Automated crack growth analysis, crack tip detection, digital image processing, double cantilever beam (DCB) fracture test, image filter, image subtraction, image thresholding, optical flow
1 Introduction

Crack growth tests are mostly based on the periodical extraction of crack length information. The object of this study is a Double Cantilever Beam (DCB) test [1]. Fracture mechanics are increasingly used to design adhesive joints and for that the adhesive toughness is required [2, 3]. The critical strain energy release rate or toughness $G_c$ in mode I obtained in the DCB test is the main fracture mechanics parameter used for its simplicity. The specimens used in such tests are comprised of two thick metallic cantilever beams bonded together by an adhesive like the one shown in Figure 1.

![Figure 1: DCB specimen being tested with traditional optical microscopes.](image)

The test is performed in order to obtain the experimental compliance-crack length curve of a polymeric adhesive. Accurate and reliable crack length measurement is needed for the generation of the previously mentioned compliance-crack length curves. Although compliance based approaches exist where no crack length measurement is necessary for DCB-type specimens [4], in most cases the crack length has to be measured manually. Conventional methods to determine the crack length include real-time optical measurement by an operator where moving optical microscopes may be used to aid in this effort, solving the cantilever beam equations and manual measurement of images taken during the test [5, 4, 6].

Human operators often have difficulties in the detection of such cracks. Depending upon the quality of the instruments used for measurement and also on the operators’ skill, different results may be obtained. Keeping in mind that the whole crack detection process is a extremely time consuming and cumbersome work, more problems arise when a high speed crack growth test is performed. The procedure described in this work follows a different approach and tries to automate the measurement procedure in images. While a human operator could detect the crack in each of the digital images acquired by a camera, this is not feasible if several hundreds of images are taken every second and therefore it is deemed necessary to automate this task. As described by Dare [7] the full automation of crack detection is almost not possible due to the difficulties inherent to the crack growing in an otherwise noisy image. The roughness of the specimen surface, luminance condition, and the camera itself may influence the image quality. The detection of cracks in a concrete structure is described, but human intervention is necessary for each of the images, which is particularly not feasible in the case of high speed. Cracks in concrete structures are also of different type than in polymeric adhesive structures, being the visibility of cracks in the more brittle
concrete structure better.

Mayrhofer et al. have used a tracing algorithm which is able to trace cracks due to their different gray level. This is an interesting approach in order to distinguish between real crack patterns and normal roughness of the brick-texture which was analyzed in that work [8]. Seedpoints are automatically defined in thresholded images and after application of the tracing algorithm the startpoint and endpoint of these structures is used to approximately measure the crack length, based on which parts may be rejected in a production process. The difficulty in this process is to accurately find the crack tip, since noise around the crack tip may be interpreted as being part of the crack by the tracing algorithm. Anyway, authors have shown an application of image processing techniques for automated quality control.

A similar problem was analyzed by Elbehiery et al. [9]. Using morphological operations to discriminate the defect pixels more accurately, the crack on an otherwise rough surface was detected and could be measured from images. In morphological operations, the value of each pixel in the output image is based on the corresponding input pixel and its neighborhood, whose shape has to be chosen appropriately in order to be sensitive to specific shapes in the input image such as cracks [10]. This effectively solves the proposed problem of quality control, but the crack tip is not necessarily detected well enough for the proposed work using such an algorithm.

In 2000, Ryu et al. described a basic fatigue crack detection algorithm [11] showing the possibility to perform such a tedious task automatically. Only simple morphological operations are needed to eliminate the noise, without eliminating the crack to be detected in this case. Noise in binary images can for example be defined as individual 1s that are surrounded by 0s, which can easily be removed. The evolution in technology, such as a CCD camera, XYZ stage and a position controller for the camera controlled by the digital image processing algorithm, allowed to obtain good automatic measurements [12], at the price of highly sophisticated hardware. The camera was moved automatically when the crack propagated to three quarters of the acquired image width in order to guarantee the highest possible image resolution in the area of interest.

A different approach was used by Hutt et al. in order to detect cracks at holes. Digital image correlation (DIC) [13] may be used to measure differences in the displacement field around a crack as long as it is possible to load and unload the object to be measured in order to compare the two images. When a crack appears near the hole, a jump in the displacement field is noticed [14]. In his work it was shown that the technique developed is capable of detecting unmistakably notches longer than 2 mm, but even 1 mm long cracks could be detected. Further work is needed in this area, since the resolution of these systems may not be high enough for a precise crack tip detection. Furthermore, the limitation of needing images from the loaded and unloaded states of a structure limits the applicability of this method in real world problems.

Plank and Kuhn have used digital image processing for detecting a crack in mixed mode loading [15].

The present paper is organized into three main parts. After a short introduction, four image processing techniques are presented, in the second part these techniques are applied to images without noise, images with controlled noise, and real images. In the last part a performance comparison is made between all presented algorithms. At the end the pertinent conclusions are drawn and future work is suggested based on the knowledge gained in the course of this work.
2 Experimental Setup

During a DCB fracture test, a vertical force is applied on one end of the specimen, while the other end is free to pivot. The resultant load forces the specimen to open gradually as the crack initiation point develops into a crack and proceeds to advance along the adhesive layer. The main dimensions of the present specimen type are a usable length of 265.6 mm and a thickness of 12.7 mm for each of the beams, see Figure 2. The 27 mm high specimen was only recorded in 50 mm of its length due to limitations of the present system. Nevertheless, the feasibility of an automated crack tip detection system may be verified. The epoxy-based adhesive Araldite 2021 was used in the present case.

![Figure 2: Drawing of the specimen to be tested; the region of interest for the image processing is marked on the model.](image)

The specimens used in this work were painted with white colour since this was found to lead to the best contrast for crack detection, see Figure 1. The paint is expected to break as the crack advances. In order to improve the image quality, good and consistent lighting is essential. A Bosch Derwent UF500, Germany, infra-red spotlight is used for this purpose. Figure 3 shows the camera setup used for recording the images.

The image data was acquired using a PhotonFocus, Switzerland, MV-D1024 CMOS camera which has a maximum resolution of 1024x1024 pixels and can process up to 150 images per second in the used configuration. It should be noted that due to the available focal length, unlike shown by Ryu [12], the distance to the specimen was higher than 800 mm. Therefore the full resolution of the camera cannot be used, being the area of interest just a small crop within the whole recorded frame, loosing effective resolution. The specimen was therefore captured with a resolution of approximately 200x110 pixels. This leads to a scale of approximately 4 pixels per mm.

The camera was connected via CameraLink interface to a PC where a Coreco Imaging X64-CL iPro, Canada, image acquisition board is installed. For image storage, the IO Industries, Canada, Video Savant 4 software was used. This software stored a series of digital images which were used in the algorithms developed in this work. The analysis of these images was done using The MathWorks MatLab [16] and its image processing toolbox.
3 Digital Image Processing Algorithms

A variety of approaches may be used in order to detect cracks on specimens by DIP. In the present work, four different algorithms were developed in order to automatically measure the crack length and a comparison of the obtained results is made followed by a discussion on possible ways to get better results. The methods used are based on thresholding of the white area around the crack tip, filtering of horizontal details, subtraction of consecutive images and the optical flow concept [17]. While the first three algorithms described were adapted and combined from Gonzalez et al [18], the last algorithm was adapted from Horn [17]. Therefore it should be mentioned that no new image analysis algorithm was invented, but existing algorithms were adapted to be able to detect the crack tip.

In order to understand the capability of each of these algorithms, noiseless synthetic images with a crack-like shape growing horizontally were created using the GIMP image manipulation program [19]. Afterwards some noisier synthetic images were also analyzed in order to create a step between the perfect images and the very noisy and low resolution real images which were analyzed in the final step.

Some things were known about the crack and the images themselves which helped in the algorithms. The crack in the used specimen type grew as a thin dark path in a white painted area. It was known that this crack grew essentially horizontally. Since cracks never get shorter over time, the crack growth direction was also known.

3.1 Image Thresholding

Algorithm A is based on thresholding [18] each image of the sequence in order to detect the white painted region around the crack. The algorithm available [18] was modified in order to comply with the requirements for crack detection. The crack is visible as an intensity variation in the white area. The result of this operation is a binary image as the one shown in Figure 4(a).
Vertical lines are automatically searched through the binary image from right to left in order to find the first interception of this line with the crack tip, since it is known that the crack grows from left to right approximately in horizontal direction, this is designated as vertical line scanning. It should be noted that in order to reduce errors caused by noise, only a sequence of black pixels is considered a crack and not single black pixels in the white region. Figure 4(b) shows a sample picture of the detected crack tip.

In order to start the detection process with this algorithm, the region of interest has to be manually chosen and afterwards the white point and the starting point for the region growth function have to be manually defined just once for the whole image sequence.

![Figure 4: Example of images processed by algorithm A.](image)

### 3.2 Horizontal Details

The least complex of the presented algorithms, leading to a very high processing speed, is based on the reinforcement of horizontal details, such as a crack. This algorithm is a combination of various already existing image analysis techniques [18]. Each of the images in algorithm B is therefore processed by a Sobel filter [20, 18], and afterwards isolated pixels are removed from the image using morphological cleaning [18]. The used structuring element was a one surrounded by zeros, which permits the elimination of isolated pixels as required in the present case. The Sobel filter is a discrete differentiation operator which uses the intensity values in a 3x3 region around each image point to approximately calculate the image gradient at that point. This approach is simple and leads to relatively good edge detection results while simultaneously keeping the low necessary computation load. This filter helps to identify the image edges such as the crack. It does this by returning the direction and strength of the greatest increases from light to dark. Figure 5 shows the image before applying histogram equalization [21], its histogram and the result of this operation. Figure 6(a) shows the morphologically cleaned image. The result of this technique can be seen in Figure 6(b), where the detected crack tip location is superimposed on the original image for verification.

### 3.3 Image Subtraction

Algorithm C is based on the comparison of consecutive images. The basic technique is described by Gonzalez et al [18], but some modifications were necessary in order to tune the method for crack detection. At the beginning a Gaussian filter [22] with an standard deviation of 1% is applied
to the images for smoothing. The first of these images is then subtracted from the second one in order to detect the variation between both images. Additional filtering is necessary, therefore a Sobel filter is applied to the image resulting from the subtraction. Thresholding is applied in order to reinforce the detected differences between both images. Isolated pixels are removed from the image for additional noise reduction by morphological cleaning. Therefore a 3x3 structuring element containing a 1 surrounded by 0s is used. This binary image is superimposed on the original image as shown in Figure 7. Finally the crack is detected by vertical line scanning.

Figure 7: Sample image processed by algorithm C with superimposed crack.

While this technique in theory should lead to very fast and accurate results, since the only change between two consecutive images is the crack extension, in reality problems are expectable due to vertical movement of the specimen in relation to the camera. This means that the crack may shift a pixel up or down between images, making it difficult to apply this method without some additional
filtering.

3.4 Optical Flow

Algorithm D is based on the optical flow concept developed by Horn [17]. The algorithm itself was therefore not developed, but applied to the crack growth problem. The basic idea is to determine the velocity of each pixel in a sequence of images. This technique has been applied in other applications of image processing [23], but no reference was found of the usage of such a technique for crack tip detection. Since the crack tip advances in a known direction in two consecutive images, the optical flow concept should be applicable in the present case, detecting a strong variation in velocity between the crack tip and its neighborhood.

Before calculation takes place, the original images are cropped in order to reduce the necessary processing power and a Sobel filter is applied in order to reinforce the cracks detectability. The peak velocity in the X direction corresponds to the location of the crack tip for the frame in analysis as shown in Figure 8.

![Velocity field in X direction for all pixels determined by the optical flow algorithm.](image)

**Figure 8:** Velocity field in X direction for all pixels determined by the optical flow algorithm.
In order to reduce the processing requirements, the crack growth information of the previous frames is used in order to calculate the possible next crack tip position. A linear extrapolation of the information obtained in the previous frames is used for this preview and as a consequence only a small part of the next image has afterwards to be analyzed.

4 Application of the Algorithms

The algorithms described above have different strengths and weaknesses which become clear when applying them on real image analysis problems. Three image sequences are analyzed. In a first approach, a synthetic image sequence is analyzed. Afterwards some noise is added to this sequence, and finally real low resolution images of a DCB adhesive fracture test are used for analysis.

4.1 Synthetic Images

In order to see if the algorithms themselves lead to reliable results, a series of images containing cracks is analyzed. Therefore, a series of synthetic images with a size of 1000 by 500 pixels, like the one shown in Figure 9, was made using GIMP.

![Image created in GIMP containing a simulated crack for algorithm validation.](image)

Figure 9: Image created in GIMP containing a simulated crack for algorithm validation.

Different crack lengths between 10 and 160 pixels have been created. All four algorithms are applied to these test images and as can be seen in Figure 10, very good agreement is found between all four algorithms and the length of the drawn crack.

4.2 Synthetic Images with Noise

Noise was added to the images in the form of randomly placed dark pixels. Figure 11 shows the result of this operation around the crack tip. This is intended to simulate some of the image quality problems that appear during high-speed filming when low resolution cameras are used.

All four algorithms still detect the crack tip reasonably well, but quality differences can be seen between the different approaches, see Figure 12. As can be seen, the slight noise introduced in the image already creates difficulties in the crack tip detection. An absolute error of up to 3 pixels
Figure 10: Comparison of the detected crack length for the different algorithms developed.

Figure 11: Detail of the simulated noisy crack tip for testing the algorithms.
(equivalent to about 0.75 mm) in the four algorithms is obtained, see Figure 13.

**Figure 12:** Comparison of the detected crack length for the different algorithms with a slightly noisy crack.

**Figure 13:** Absolute error of the crack tip detection on a noisy simulated crack.

From this error comparison, it can be clearly seen that the lowest accuracy is obtained by algorithm D based on the optical flow concept which seems to be more affected by noise around the crack tip. The error is around 3 pixel, which is equivalent to 0.75 mm in the present case.

### 4.3 Real Images

The four developed algorithms have also been applied to real experimentally obtained images. For verification, two manual measurement of the crack tip on these images were performed and all algorithms were compared against this data. The first manual measurement was performed using *GraphClick* by *Arizona Design*, Switzerland, [24] and the second measurement was performed using *Photoshop* by *Adobe Systems Incorporated*, USA, [25], see Figure 14.
Figure 14: Comparison of the detected crack length for the different algorithms developed.

It should be noted that even the best working algorithm cannot detect the crack tip correctly all the time if the images to be analyzed have not enough resolution or quality. This is clearly shown by the differences in the manually measured crack lengths on the same images, simply by using different software to display the image and using different operators. Nevertheless, it has been shown that the algorithms are capable of detecting the crack tip to a similar level of accuracy as the human eye based on the available images.

In terms of required processing power, Table 1 shows the processing time of each algorithm on a PowerPC G4 867MHz, 640MB Ram running Mac OS 10.4 and MatLab 7.04 for reference.

Table 1: Required processing time for each algorithm

<table>
<thead>
<tr>
<th>algorithm</th>
<th>processing time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.87</td>
</tr>
<tr>
<td>B</td>
<td>0.33</td>
</tr>
<tr>
<td>C</td>
<td>12.80</td>
</tr>
<tr>
<td>D</td>
<td>32.64</td>
</tr>
</tbody>
</table>

It should be noted that for this work, an image sequence of only 16 images was used, whereas real-world applications may have a few thousand images in each sequence, reinforcing the need for a fast algorithm. The algorithms were not optimized for speed in the current application. Anyway, very high differences in the processing time were obtained which may influence the decision for choosing a faster algorithm despite its slightly higher error.

5 Discussion of the Results and Conclusions

While all four algorithms detect the crack tip flawlessly in the case of the “perfect” images, a small noise level already leads to complications in the crack tip detection. Anyway, the different algorithms lead to acceptable results even with the low image quality of the experimentally obtained images from the DCB experiments. It was also demonstrated that among all the algorithms considered, the optical flow method was less affected by noise around the crack tip, although this does
not translate necessarily into more accurate measurement results in real images.
It was possible to validate the usage of digital image processing techniques for adhesive crack growth analyses using standard equipment such as a relatively low resolution CCD camera and software packages such as MatLab.
One approach to get better results would be to use a different lens setup which allows to get a higher resolution and better illumination near the crack tip. This would require a considerable investment in hardware, but would most likely lead to better results.
Although the present version of the analysis software is not fully automated, it is possible do envision a solution able to receive and analyze real-time images of mechanical tests and automatically report crack growth information, since the software is already capable of analyzing a batch of images and output the crack growth curve.
This work demonstrates that such an automation is easy to implement in the crack measurement field as long as the acquired images are of sufficient quality and, with the appropriate user interface, a seamless solution to extract data may be developed which could allow for less tedious fatigue crack growth experiments.
More advanced hardware, namely higher speed cameras with higher resolution near the crack tip would most likely lead to more accurate results using the presented image processing algorithms.

6 Acknowledgments

This work was partially funded by the Portuguese Foundation for Science and Technology PhD scholarship SFRH / BD / 41061 / 2007, the Portuguese Foundation for Science and Technology project PTDC/EME-PME/67022/2006 and by the Fundação Luso-Americana project FLAD 314/06.

References


