Automated 3D Crack Detection for Analyzing Damage Processes in Concrete with Computed Tomography

Olaf Paetsch¹, Daniel Baum¹, Karsten Ehrig², Dietmar Meinel², Steffen Prohaska¹

¹Zuse Institute Berlin (ZIB),
Takustr 7, D-14195 Berlin, Germany, e-mail: [paetsch|baum|prohaska]@zib.de

²BAM Federal Institute for Materials Research and Testing,
Unter den Eichen 87, D-12205 Berlin, Germany, e-mail: [karsten.ehrig|dietmar.meinel]@bam.de

Abstract
Analyzing damages at concrete structures due to physical, chemical, and mechanical exposures requires the application of innovative non-destructive testing method combined with 3D image processing algorithms. Since manual segmentation is too time-consuming, we need automatic segmentation methods. In this work, two methods to automatically detect cracks in computed tomograms of concrete specimen are presented and compared. These methods were tested on several datasets of size up to 2000³ voxels. Furthermore, the question is addressed of whether automatic crack detection can be used for the quantitative characterization of damage processes, such as crack area and thickness. To achieve such statistical evaluations, a method is proposed to investigate the embedding of the cracks in their surrounding material. Moreover, the representation of cracks as geometrical objects is introduced.

Keywords: computed tomography, template matching, Hessian eigenvalues, crack statistics, visualization, crack surface, ZIBAmira

1 Introduction
Applying 3D micro computed tomography (µ-CT) to specimens of cementitious building materials is a non-destructive method to analyze the inner structure of those specimens. In the resulting 3D image grey values, cracks generally appear darker than their surrounding material, because of the lower material density; hence, they can be easily detected by human beings. But a manual segmentation of cracks in 3D datasets with a typical size of up to 2000³ voxels is too time-consuming even for a single dataset. If several datasets have to be segmented, this becomes even more apparent. Hence, we need automatic segmentation methods.

Cracks can locally be considered as thin, plate-like structures with diameters of 50-200 µm. Considering the voxel size of 39 µm, cracks may be detected with a minimum width of 80 µm. Starting with these elementary properties, the crack detection methods discussed below have been developed. Due to a signal to noise ratio of 5-10 depending on the surrounding material and the presence of non-crack-like structures with similar grey values than those of cracks, threshold-based image processing methods are bound to fail. Hence, methods with an inherent shape analysis or methods that follow cracks from a given starting point are to be preferred.

One such method is template matching using a plate-like 3D reference object. Since it is well established in cell biological applications for the search of membranes, it seems to be promising to use it for crack detection, too.
Another tested method is the analysis of the eigenvalues of the Hessian at every voxel combined with an extended percolation algorithm. The eigenvalue analysis of the Hessian also has applications in medicine and biology, e.g. for finding sheet-like structures in μ-CTs of bones. In this work, results using the above mentioned algorithms are presented and will be discussed w.r.t. the applicability for the quantitative characterization of damage processes. A new approach is the geometric representation of cracks as points and medial surfaces, which helps to compute statistics on cracks. Furthermore, the embedding of cracks in their surrounding material is an important feature to assess the results of the statistical evaluation. In order to evaluate the results of a crack detection method, it is necessary to visualize the specimen as well as the cracks in 3D. Therefore, the presented methods have been implemented in the visualization and analysis system ZIBAmira. Given a large toolbox set in ZIBAmira, tools for the statistical evaluation of the identified cracks are also available, thus allowing a statistical analysis of the cracks.

2 Related Work

Although crack detection is a developing field of research, there are only a few publications discussing the detection of cracks in 3D images, whereas 2D techniques are often used. Regardless of the dimensionality, the most obvious techniques are image processing methods such as thresholding to find cracks in tomography datasets. For noisy input data, feature preserving noise reduction filters are applied followed by thresholding and artefact removal [6] [7]. For 2D images (photogrammetric pictures) taken from concrete surfaces, tracing techniques can be used afterwards starting from binary images obtained by thresholding [1]. A promising technique for detecting sheet-like structures in 3D images is template matching. It is a well-established method in the analysis of biological image data [8] [9] but to the best of our knowledge so far it has only been used in [3], which reflects an earlier state of the project described in this paper.

3 Crack Detection Methods

In this section, we shortly describe the techniques that we used in our project. For a deeper description and discussion see [3]. The first method we apply is the sheet filter proposed in [9], which detects sheet structures in volume data. It is computed from the eigenvalues of the Hessian matrix. The second method is template matching, where the image is searched for a pattern that models a crack. For details on template matching see [10]. Another method is the percolation algorithm described in [11], which is based on the physical model of liquid permeation. Although this technique has been developed for 2D and a 3D extension cannot be derived directly, we developed a 3D version [3] that can be used as a post-processing step after cracks have been detected by evaluating the Hessian matrix.

3.1 Data Acquisition and Image Preprocessing

μ-CT measurements with x-ray energy of 210 kV, 1 mm Cu prefilter, detector size 2023x1731 pixels (pixel pitch: 0.2 mm), source to sample distance of 226 mm, source to detector distance of 1153 mm, 2400 projection angles over 360° have been performed on samples with a diameter of 70 mm. With an isotropic voxel size of 39 µm, this results in datasets of 6 to 8 GB. Before the 3D image is reconstructed using standard filtered back projection, image preprocessing steps like beam hardening correction and median filtering are applied. Due to the noisy nature of images obtained from computed tomography, filtering of such images is of great importance. Since features should be retained, simple smoothing filters like the Gaussian smoothing do not perform well. As a consequence, if image filtering is needed, feature-preserving filters like bilateral filters, non-local-means filter or median filters have to be applied. According to [5],
where several filtering techniques are discussed, for the results presented in this paper, we applied an iterative median filter $med^I$ prior to crack detection. Since data filtering always comes along with loss of information [16], data filtering has to be carefully evaluated according to the used applications and calculated results.

![Unfiltered image (left), $med^I$ filtered image (right)](image)

**3.2 Hessian-Driven Percolation**

Hessian-driven percolation is a two step process, where first a sheet filter is applied to the input data. This filter, which detects surface-like structures in volume data – what cracks obviously are –, is computed from the eigenvalues of the Hessian Matrix $\Delta$, which is defined by

$$\Delta I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

where $I_{xx} = \frac{\partial^2}{\partial x^2} I$, $I_{xy} = \frac{\partial^2}{\partial x \partial y} I$, ..., $I_{zz} = \frac{\partial^2}{\partial z^2} I$.

Let $\lambda_1 \leq \lambda_2 \leq \lambda_3$ be the three eigenvalues of $\Delta$ of a point $p$ in $I$. A point $p$ denoting a sheet is characterized by one large eigenvalue and two small eigenvalues and is the base for a sheet filter $S$.

As the second step, a percolation-based algorithm is applied to $I$ starting at all points in $H$, where $H$ is the result of applying an appropriate threshold to $S$. See [3] and [9] for more details. Although the original percolation algorithm has only been developed for 2D images [12], we extended it to 3D as a post-processing step as described in [3].

Since branching cracks or cracks with regions of high curvature are often missed by the filter $S$, gaps appear in the detected cracks. To account for these gaps, the second step was introduced which closes the gaps, if the percolation process is started in their neighbourhood.

![2D slices showing cracks (white) found with the Hessian-percolation algorithm. Left: Overview, Middle: Close-up at another slice. Right: Artefact stemming from the percolation.](image)
3.3 Template Matching

The idea of template matching is to find a certain pattern, given as a small 3D image \( T \) (the template), in the image \( I \) (here the 3D CT image). The template \( T \) is moved to every voxel in \( I \) and at every voxel position the correlation coefficient \( C \) at \( (r,s,t) \) between \( T \) and the corresponding subimage of \( I \) is computed:

\[
C_L(r,s,t) = \frac{1}{N} \sum_{(j,k) \in T} \frac{(I(r+ie+je+k) - \bar{I})(T(j,k) - \bar{T})}{\sigma_I \sigma_T}.
\]

Here, \( \bar{I}, \bar{T} \) are the mean values and \( \sigma_I, \sigma_T \) are the standard deviations of \( T \) and \( I \), respectively, and \( N \) is the number of voxels in \( T \).

Since a crack may have an arbitrary orientation in 3D, the template is rotated at every voxel position in both directions (azimuth and elevation, the Euler angles) regarding approximately uniform samples. See Figure 3 for an illustration.

![Figure 3: A template at three out of 844 rotation steps](image)

3.4 Comparison of the two Methods

Although the Hessian-based percolation algorithm works much faster (up to 25x), the results are not satisfying. Tests with several realistic datasets led to the conclusion w.r.t. the resulting cracks that template matching is to be preferred. This is in contrast to [3]. One reason for this might be that we now use a different implementation of template matching, which is more flexible regarding the construction and rotation of the reference template. It is now possible to use plate-like or disk-like reference templates of different diameters or thicknesses. The computations here have been carried out with a plate-like template of 1280x640x80 µm (see Fig. 3). A comparison of Figures 2 and 4 reveals that template matching is more accurate and tends to create smoother cracks and less artefacts. Also,
thin cracks are often not detected using Hessian-driven percolation (compare Figures 2 (Middle) and 4 (Right)). In contrast to [3], where the Hessian-driven percolation turned out to be superior because of a low false-positive rate, we are now convinced of the capabilities of template matching. In [3] the results relied on only one small dataset, while we now studied more datasets in more detail as well as a complete CT dataset of 2007x2007x1201 voxels.

<table>
<thead>
<tr>
<th></th>
<th>Ratio Crack Voxels to Total Voxels</th>
<th>Num. of Cracks (Connected Components)</th>
<th>Max. Crack Volume</th>
<th>Min. Crack Volume</th>
<th>Median of Crack Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hessian-driven Percolation</td>
<td>1.4 %</td>
<td>10805</td>
<td>0.55 mm³</td>
<td>0.03 mm³</td>
<td>0.07 mm³</td>
</tr>
<tr>
<td>Template Matching</td>
<td>0.09 %</td>
<td>1130</td>
<td>6.76 mm³</td>
<td>0.03 mm³</td>
<td>0.07 mm³</td>
</tr>
</tbody>
</table>

Table 1: Comparison of detected cracks in a CT image 2007x2007x1201 voxels (Voxel size: 39 µm).

### 3.5 Crack Postprocessing

Further postprocessing steps are applied to the primary results of the presented methods to remove systematic falsely detected regions and small crack islands. Although the size of the crack islands to be removed is adaptable, a value of 500 has been choosen for practical reasons to ensure that no falsely detected regions are included in the results. Image border regions are excluded from crack detection results due to false detection artefacts. Currently, also a histogram-based exclusion of background grey values has to be defined due to falsely detected regions in porosities and on border lines. This relies on the experiences that grey values in cracks significantly differ from grey values of the image background. As further step this global exclusion should be replaced by local threshold values. The result of the postprocessing is a labelfield (of bytes) where every voxel is labeled as crack (1) or other material (0). Figure 5 shows a 3D visualization (isosurface) of the above mentioned complete dataset where the crack detection has been carried out with template matching.

![Figure 5: 3D visualization of cracks inside the sample](image)
4 Further Investigations of Cracks

4.1 Crack Embedding

When cracks have been detected, further investigations for quantitative characterization of the damage process are needed. An important question is the crack embedding, i.e. what material is close to the crack. Cracks may cross different grains via the cement matrix resulting in different mean grey values in the crack due to the influence of the surrounding material. For further analysis, cracks should be assigned according to their surrounding materials. As an example cracks may go through grain or through the cement matrix or it may run along the boundary between aggregate and matrix. A segmentation of the specimen prior to crack detection may be a solution to assign a crack to the surrounding material. Since such a segmentation is a very difficult task on its own [4], it is much simpler to find the cracks first and afterwards perform the evaluation for every crack voxel, i.e. analyze the grey values of the voxels in a certain distance orthogonal to the crack. To do so, we want to sample the grey values along the direction perpendicular to every crack voxel. This direction is determined during template matching. Here, for every voxel the index referring to the template orientation with the highest correlation values is stored. This can be converted into the normals of the crack voxels. Then a probe line is created pointing along the normal and its inverse with a length of six times the voxel size. See Figure 6. Typical crack profiles along those lines are plotted in Figures 7 and 8 resp.

Figure 6: Probelines orthogonal to a crack

Figure 7: Five crack profiles (grey values) of cracks lying in aggregates
Figure 8: Four grey value profiles of cracks leading along boundaries of aggregates

The values at both ends of the profiles allow the discrimination between materials of both crack sides. With this information, the detected crack voxels are relabelled according to their embedding. A slice through a relabeled crack is shown in Figure 9.

4.2 Statistics on Cracks

Statistical evaluation has to be performed on detected cracks to quantize damage processes in concrete specimen. Such statistics help to assess e.g. the quality of different types of concrete under different types of stress. The statistical parameters of main interest are

- crack size
- crack orientation
- crack aperture

all of which might be computed either in the whole specimen or in a small region of interest. While the size of a crack can be considered as the number of voxels belonging to one crack (a connected component with 26-neighbourhood in the sense of voxels), crack orientation or aperture (mean crack width) cannot be directly computed from the voxel field. Therefore, converting the cracks consisting of connected voxels into a medial surface is a reasonable solution, since all desired statistics can be obtained from a surface representation of cracks. The size of a crack is then simply the sum of the area of all triangles belonging to the crack surface. Methods for determining the other two statistical parameters are currently being developed. No results are yet available.
4.3 Surfaces as Geometric Representations of Cracks

One method to generate medial surfaces uses skeletonization of the binary voxel representation of cracks with a distance map approach [13]. Applying this algorithm on crack datasets of realistic sizes showed an immense memory usage and a very long running time, which makes it unsuitable for the purpose of this project.

Therefore, we propose a different approach: First, the position of every crack voxel is shifted along its probeline (see Figure 6) into the local minimum (Fig. 10). The resulting point set is then taken as an input to a surface reconstruction algorithm. We are currently investigating several algorithms such as [14]. This algorithm belongs to the family of advancing front algorithms. Starting at an initial triangle it rolls a sphere over the point cloud connecting points nearby. One problem with this algorithm is that it may generate two layers of triangles. Hence, we are investigating other algorithms like afront [15], which also is an advancing front algorithm, augmented by global information. Since cracks locally have a clear main orientation, it might also be possible to project subsets of crack points into the plane that best approximates each subset. The triangulation of the crack points can then be done in 2D using a Delaunay triangulation algorithm.

The point set may also be used for visualizing the cracks as shown in Figure 11.

5 Conclusions and Future Work

We showed in this work suitable ways to automatically detect cracks in CT images taken from concrete specimen. In contrast to [3], we come to the conclusion that template matching is superior for this task over Hessian-driven percolation. Since quantitative analysis of detected cracks is an important
outcome to compare and evaluate the behaviour of concrete exposed to e.g. temperature variations, chemical loading and mechanical stress, we further process the label field representing the cracks to find the crack embedding as well as a surface representation of cracks.

Since the crack size resp. the crack area can be directly computed from a given crack surface, we are currently investigating several ways to compute such surfaces. The computation of the mean crack thickness as well as the orientation can then be done using this surface. However, the influence of surface smoothing on the results of statistical evaluations requires further investigations. The surface representation might also be useful for tracing cracks over time, for example, in frost-thaw-cycles.

References