Crack segmentation in 3d concrete images: perspectives and challenges

Tin BARISIN¹,²,*, Christian JUNG², Anna NOWACKA¹, Katja SCHLADITZ¹, Claudia REDENBACH²

¹ Fraunhofer-Institut für Techno- und Wirtschaftsmathematik; Kaiserslautern, Germany
² Technische Universität Kaiserslautern; Kaiserslautern, Germany
*Corresponding author, e-mail address: tin.barisin@itwm.fraunhofer.de

Abstract
Concrete is one of the most frequently used building materials whose failure can have catastrophic consequences. Hence, understanding its properties is a prerequisite to wide-spread application. Usually, the (post-)cracking behaviour of concrete is investigated by mechanical tests. Computed tomography opens the opportunity to analyze the concrete’s microstructure and crack systems non-destructively. A tomography scan yields a 3d gray value image of the sample. Segmenting cracks – identifying the voxels belonging to the crack – in these images is challenging due to crack structures being very thin and hard to distinguish from the heterogeneous material composed of cement paste, aggregates, pores, and reinforcements yielding locally very strongly varying gray values. The immense size of representative image data prohibits interactive processing.

In a previous study, we developed a technique for generating realistic semi-synthetic tomograms with cracks. This helped us to identify the most promising methods for automatic crack segmentation: a convolutional neural network – 3d U-Net – and Hessian-based percolation. Here, we apply these two methods to four real tomographic images of concrete specimens with embedded rebars featuring cracks created by pull-out tests. Compared to the synthetic image data, the images of real cracks pose an additional challenge: The thickness of cracks varies continuously while the crack propagates. Hence, we have to adapt the selected methods to deal with crack widths between one and 20 voxels at the same time. Applicability, limitations, and robustness of our methods are discussed.

Keywords: 3d computed tomography, 3d segmentation, crack detection, supervised learning, percolation, region-growing, multi-scale methods.

1 Introduction
Concrete is the base material e. g. for buildings and bridges. Hence, its quality, durability, and mechanical stability have to be ensured. For investigating concrete behaviour under load, various stress tests can be applied to concrete specimens. Concrete is known to be a brittle material which tends to develop cracks under load. For example, during tensile testing, cracks occur when the applied force exceeds the tensile strength of the concrete. With the increase of force, the specimen may eventually fail. Shape, size, and location of the cracks as well as the sample’s post-cracking mechanical behaviour provides valuable information on the specific properties of the studied concrete. With the naked eye, the cracks are only visible on the sample surface and neither their full 3d structure nor the correlation of the crack with local structural features can be observed.

Micro-computed tomography (μCT) can provide the missing information. The same concrete specimen can be scanned several times during the mechanical test, and even in-situ. To gain the desired structural information, the reconstructed μCT images need to be processed and
analyzed. These images can be very large (up to $2\,000^2 \times 10,000$ voxels). Hence, manual crack segmentation is not feasible and automatic image processing is needed.

![Figure 1: From left to right: 2d slice of realistic simulated crack images (crack width 1 and 3, image size $256^2$), 3d renderings of simulated crack and 3d U-net segmentation of crack (image size $256^3$).](image)

The heterogeneous microstructure of concrete renders the segmentation of cracks a challenging task. In particular, pores forming during the concreting process and cracks are both filled with air. Consequently, they feature essentially the same gray value distribution and have to be distinguished based on shape. Furthermore, there are various types of concrete, like normal, high performance or ultra high performance concrete with different types of reinforcement. They include a wide variety of components. To be widely applicable, crack segmentation methods must be robust to changes of type and composition of the concrete.

3d crack segmentation in CT images of concrete has been studied by Paetsch and colleagues in [3, 8, 9]. In [3], three classical segmentation methods were compared based on manually segmented cracks in CT images. These methods were later applied to larger CT images of up to $2\,000^3$ voxels [9]. Limitations, robustness and reliability of automatic 3d crack segmentation methods were discussed in [8].

Recently, deep learning methods have been established for solving various image segmentation tasks. In a previous work [1], we have compared the 3d U-Net and a random forest with the classical methods from [3]. The methods were tuned in a controlled setting with fixed crack scales. As it is difficult to control the crack width during stress tests, real cracks were not suitable. Hence, a technique for simulating artificial crack structures was designed. These simulated cracks can then be combined with CT images of non-cracked concrete to get realistic semi-synthetic CT images with cracks, see Figure 1. A further advantage of this approach is that the data allows for an objective method comparison.

Based on [1], the two most promising methods were selected for further validation on CT images with real cracks: the 3d U-Net [2] representing the class of machine learning approaches and Hessian-based percolation as a classical image processing technique [3]. In [6], both methods were applied to real CT images of concrete with cracks resulting from tensile tests. The multi-scale nature of cracks, i.e. crack thickness varying from one to more than 20 voxels, was identified as main challenge for the application to real data.

In this work, we continue to explore possibilities and limitations of automatic 3d crack segmentation in CT images of concrete. To this end, we investigate a different testing setup, namely cylindrical concrete samples with an embedded rebar that are subjected to a pull-out test as described in [5]. A tensile force is applied to the rebar to pull it out of the sample. With increasing force, cracks propagate radially from the contact area between concrete and rebar to the specimen surface. Examples of cracks formed by the tensile and pull-out tests can be seen in Figure 2. Cracks forming in the two testing setups differ with respect to direction, thickness,
and topology. In the pull-out tests, also cracks resulting from delamination of the rebar are observed.

![Figure 2: 2d slices of scanned concrete specimens. From left to right: crack induced by a tensile test (two images), circular delamination region around the rebar and crack spreading from the rebar, both induced by a pull-out test.](image)

This work is structured as follows. Materials, CT imaging, and methods are described in Section 2. We analyze the segmentation results of the two selected methods on four CT images in Section 3. In Section 4, conclusions regarding robustness and limitations of these methods are drawn. They combine insights from both [6] and this study.

## 2 Materials and methods

### 2.1 Samples and experiments

Three concrete samples with an embedded rebar were prepared. On two of these specimens, pull-out tests were applied. The first specimen (Image 1) was loaded until failure, i.e. until the bar was completely pulled out of the concrete. The second specimen shows cracking but the bar stays in place. For this specimen, two scans were taken. Image 2 refers to the central part of the specimen, while Image 3 is a scan of the bottom part. Interestingly, the cracks in the two scans exhibit very different characteristics regarding shape and width. Finally, the third specimen (Image 4) has not been subjected to a pull-out test at all but contains a crack which occurred during the concreting process.

<table>
<thead>
<tr>
<th>Size [voxels]</th>
<th>Cropped size [voxels]</th>
<th>Voxel edge length [µm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>1 965 × 1 965 × 1 716</td>
<td>1 000 × 900 × 500</td>
</tr>
<tr>
<td>Image 2</td>
<td>1 979 × 1 979 × 1 547</td>
<td>1 000 × 1 000 × 700</td>
</tr>
<tr>
<td>Image 3</td>
<td>1 987 × 1 987 × 1 577</td>
<td>1 000 × 1 000 × 800</td>
</tr>
<tr>
<td>Image 4</td>
<td>959 × 959 × 1 849</td>
<td>368 × 268 × 1 500</td>
</tr>
</tbody>
</table>

### 2.2 Computed tomography imaging

The concrete specimens were scanned at the Fraunhofer ITWM, Kaiserslautern, Germany. The μCT device consists of a Feinfocus FXE 225.51 X-ray tube with maximum acceleration voltage 180.3 kV and maximum power of 12.7 W. The detector is a Perkin Elmer flat-bed detector XRD 1621 with 2 048 × 2 048 pixels. For Image 1 and 4, a tube voltage of 180 kV was used. The integration time was set to 1 second. 1 200 projections were used for acquiring tomographic reconstructions. For Image 2 and 3, a tube voltage of 160 kV, an integration time of 0.5 seconds and 400 projections were used. The CT setup is shown in Figure 3. The sizes and voxel edge
lengths of the reconstructed images are given in Table 1. The segmentation methods are applied to cropped images that contain most of the crack structure. Each image is preprocessed via min-max normalization to an 8-bit gray value range.

**Figure 3:** From left to right: side view of the specimen for Image 2 and 3, side and top view of Image 1, and CT imaging setup when scanning Image 4.

### 2.3 Crack segmentation methods

Here, we give a short introduction to the segmentation methods that were previously identified [1] to be the most suitable. Hessian-based percolation belongs to the classical image processing methods. In general, these rely on extracting gray value or shape descriptors from the local image structure. The 3d U-Net is an example of methods from the machine learning/deep learning field. These are based on learning a classifier from pairs of input and ground truth images using optimization methods. As a subset of machine learning, deep learning refers to neural networks with several stacked layers which have a large number of parameters. This enables the learning of sophisticated features but requires a large amount of training data.

#### 2.3.1 Hessian-based percolation

Hessian-based percolation [3] is a two-step algorithm. In the first step, candidate crack voxels are selected. This is realized by using filters based on the Hessian matrix of the image. The filters are designed to produce a high response in dark, planar regions in the image. To obtain the candidates, we apply hysteresis thresholding ($t_1 \geq t_2$) to the filter values: Applying a first threshold $t_1$ results in a binary image. Then, the set of foreground voxels is grown iteratively: In each step, foreground neighbours with filter response $\geq t_2$ are added to the foreground. This stops when no more voxels can be added.

In the second step, a region-growing process is started in each preselected voxel. The resulting region is then considered to be part of a crack if its shape matches a fixed shape criterion. Denote the set of candidate voxels by $H$ and the input image by $I$. The iterative region-growing algorithm consists of the following steps:
1. Select some \( p \in H \). Initialize the percolated region as \( P = \{ p \} \) and set \( t = I(p) + \epsilon \) for a predefined, real-valued \( \epsilon \).

2. Add all neighbors \( q \) of \( P \) to \( P \) that satisfy \( I(q) \leq t \).

3. Set \( t = \max(\max_{q \in P} I(q), t) + \epsilon \).

4. Repeat 2 and 3 until the region \( P \) comes in contact with the boundary of a cubic window of predefined size \( (2W + 1)^3 \), \( W \in \mathbb{N} \), centered at \( p \).

5. Compute the shape criterion \( F_{3D} = |P \cap H|/|P| \in [0, 1] \). If \( F_{3D} \geq f \) for some predefined \( f > 0 \), every voxel in \( P \) is considered to be part of the crack structure.

6. Repeat 1-5 for every voxel \( p \in H \). Voxels that have been detected less often than a predefined threshold \( \tilde{t} \in \mathbb{N} \) are not considered to be part of the crack structure.

A good set of candidate voxels is crucial for the performance of the algorithm. For this first step, Hessian-based filters such as the sheet filter [10] or the Frangi filter [4] have been applied successfully [1]. The Hessian matrix contains the second order local gray value variations around a voxel. The eigenvalues of the Hessian matrix can be composed to suitable characteristics to distinguish blob-like, plate-like, and tube-like structures. Therefore, these filters are well-suited to detect dark structures on a brighter background and to classify their local shapes.

The filters also contain a scaling parameter. In practice, filtering is performed at multiple scales, and results are combined to detect multi-scale structures.

The sheet and Frangi filter are both based on ratios of eigenvalues of the Hessian. They only differ slightly, for example in the range of output filter values. Comparing Hessian-based percolation with candidates selected by the sheet filter or the Frangi filter in [1], we obtained higher recall values when using the Frangi index for plate-like structures.

In [6], a modification of the Frangi filter was proposed. In its original form, the Frangi filter sets the significance information of local structure information in relation to that of the maximal global structure to suppress noise. However, when evaluating authentic concrete images, we observed that the Frangi filter is prone to misclassify thin structures as background. Thus, we used a quantile-based Frangi filter which cuts off outliers when the significance of the global structure information is computed. Here, we select the candidate voxels based on this quantile-based Frangi filter.

The parameters are chosen with the same reasoning as described in [6]. In particular, the scaling parameters of the Frangi filter should be chosen to be half the widths of the structures to be detected. We select their range from 0.5 to 7.5 and increase them in steps of 0.5. The two thresholds are selected as \( \tau_1 = 0.8 \) and \( \tau_2 \in [0.2, 0.3] \). The parameters for Hessian-based percolation were \( \epsilon = 0, F_{3D} = 0.2, W \in \{2, 3\} \) and \( \tilde{t} = 0 \).

### 3d U-Net

A machine learning tool which we used for semantic segmentation of real cracks in concrete is the 3d U-Net [2]. The model consists of a contracting path (encoder), which contracts the input image and the extracting path (decoder), responsible for expanding the image. The depth of the 3d U-Net is three which means it consists of three convolution blocks in the contracting path (16, 32, and 64 filters), one bottleneck (128 filters), and three convolution blocks in the extracting path (64, 32, and 16 filters).

The convolution block of the contracting path consists of two 3d convolutions of kernel size three, each followed by a batch normalization and a rectified linear unit (ReLU) activation. After those we apply a max pooling layer with stride two for downsampling. In the expansive
path, we use a 3d transposed convolution of kernel size three for upsampling at the beginning of each step. The transposed convolution is followed by a concatenation with the corresponding cropped feature map from the contracting path. At the end of each step, two convolution layers of kernel size three are used, each followed by a batch normalization and ReLU activation. After the extracting path, one more convolutional layer of kernel size three is applied. In total the 3d U-Net used in this work comprises 2 042 689 learnable parameters.

To train the 3d U-Net, 3d images of synthetic cracks of constant width one, three, and five voxels are used. 2d slices of such images are shown in Figure 1. In order to reduce the computational cost of the training, the 3d images of size 256^3 voxels are tiled into patches of size 64^3. Patches overlap by 14 voxels to reduce possible edge effects. The final dataset contains 3 456 images including augmented ones (rotated, flipped, down-scaled). The 3d U-Net is trained for 20 epochs with batch size two. We use the Adam optimizer [7] with an initial learning rate of 0.001. The learning rate is halved every five epochs. Note that we use exactly the same network trained for the study [6]. There are no adjustments or retraining for the new type of image data.

The cracks observed in the CT data are partially much thicker than five voxels, the maximal thickness covered by the training data. To take this into account, we use a multi-scale approach. That means, the prediction of the network is computed on the original image as well as on images down-scaled by factors 2 and 4. Predictions for downscaled images are then interpolated to restore the original size for all images. The final prediction is derived from the voxelwise maximum, thresholded at 0.5.

2.3.3 Postprocessing

We observe that both methods sometimes tend to misclassify small parts of the image background as cracks. In particular, this holds true for noisy regions and pore edges. Hence, we extract the largest connected components in a postprocessing step. The minimal size of the components to be extracted depends on the respective sample and is chosen manually for now.

3 Results

In Figures 4 and 5, 2d slices and 3d renderings of the segmentation results are shown. The output of the Hessian-based percolation and the multi-scale 3d U-Net are very similar, both can detect cracks of varying thickness and do not classify pores or aggregate edges as crack structures.

For Image 1, we can observe an empty space in the center where the rebar was located. Since this void is non-planar, neither method has assigned the corresponding voxels to the crack class, with the exception of some border voxels that have been detected by the 3d U-Net (first row of Figure 5). On the other hand, despite of the large number of pores, none of the models predicts them as cracks. Small artifacts are successfully removed in the postprocessing step.

Image 2 shows a larger gap in the core, between the rebar and the concrete. Both methods assign that area to the crack class, since it is a thin planar structure. The thin crack around the gap is detected, too. It seems to be smoother in the Hessian-based percolation result. The 3d U-Net detects an additional crack in the upper part of the image.
Figure 4: 2d slices of the segmentation results for Images 1-4, from top to bottom. From left to right: input image, Hessian-based percolation segmentation image, and multi-scale 3d U-Net segmentation results.

Image 3, taken from the bottom of the same specimen differs from Image 2. Here, we can observe a very thin gap between rebar and concrete. Additionally, cracks spread radially from the core to the walls of the sample. In the slices of the predictions, no strong differences are visible. Nevertheless, in the rendering (Figure 5, third row) it gets apparent that the result of the Hessian-based percolation contains more ring artifacts while the 3d U-Net turns out to be more resistant to this type of imaging artefact.

The crack in Image 4 occurred already during the concreting process. This explains why its appearance differs from the cracks in the other samples. There is a single, thick crack with rather sharp edges. It propagates along the z-axis – the cylinder’s axis. For this image, the predictions differ considerably: 3d U-Net preserves the shape of the crack edges better than Hessian-based percolation. This difference is easier seen in the slices than in the renderings. In general, crack shapes are better preserved in the 3d U-Net segmentation than by Hessian-based percolation. Due to a smoothing within the Frangi filter, the local crack texture can only be cap-
Figure 5: 3d renderings of segmented cracks in Images 1-4, from top to bottom for Hessian-based percolation (left) and 3d U-Net (right). The direction of the rebar and the pull-out direction is the z-axis according to the coordinate system in the bottom right corner.
tured roughly. The percolation step is mostly able to restore it. Still, the segmentation accuracy in these areas is usually inferior to that of the 3d U-Net.

The runtime for Hessian-based percolation evaluated on the four samples lies in a range of 10-35 minutes. The training step for the 3d U-Net took about 5 days while the prediction varies between 1-2 hours. Runtimes are obtained on a Red Hat Enterprise Linux Workstation 7.9 with an Intel(R) Xeon(R) CPU E5-2680 v2 2.8GHz (10 cores) and 125 gigabytes of RAM.

4 Conclusion

In this work, we investigate and analyze the performance of two 3d crack segmentation methods, namely 3d U-Net and Hessian-based percolation, in four CT images of three concrete specimens with embedded rebar. Cracking was initiated by pull-out tests. Pull-out testing was stopped in different stages: failure by complete removal of the rebar (Image 1) or at development of first visible cracks (Images 2 and 3). Additionally, we analyze an image of an unloaded sample (Image 4). Both methods yield good segmentation results when applied to these cracks, although very thin cracks remain a challenge, again for both methods. Hessian-based percolation over-smooths crack structures. On the other hand, cracks segmented by 3d U-Net seem to be more disconnected than those obtained by Hessian-based percolation: more holes in the crack surfaces can be observed.

Results from this study based on pull-out tests can be combined with those from the previous one [6] based on tensile tests to get insight into the robustness, generalization, and reliability of the two segmentation methods. Cracks from tensile tests are mostly oriented in a single plane, while cracks from pull-out tests spread radially from the rebar in the core towards the walls of the sample. Cracks from tensile tests vary more strongly in scale. On the other hand, crack shapes from pull-out tests are more complex and branching patterns differ. Nevertheless, the two selected methods are able to segment most of the cracks accurately in both cases. In particular, no major modifications of the parameters are needed to obtain satisfying results when transitioning from tensile to pull-out test concrete images. In particular, the 3d U-Net trained for a previous study was successfully applied here. We consider this an indication of the robustness of the methods with respect to varying crack structures.

We conclude that the methods which we have today are ready to be used reliably on larger samples of concrete with varying shapes and sizes. Still very thin and weakly-resolved cracks in concrete are a major challenge. These types of cracks might require special treatment. Finally, the performance of the methods on other types of concrete such as fiber-reinforced concrete needs to be analyzed in the future.

Regarding the choice of the method, standard arguments from the discussion on classical and machine learning methods still apply: Hessian-based percolation is faster, has less parameters to fine tune and is more intuitive to use for civil engineers who do not have prior expertise in image processing. This is especially useful in cases when completely new concrete types are processed. On the other hand, the 3d U-Net offers an improved localization of segmented cracks and flexibility to learn more complex features due to a large amount of parameters. However, for new types of concrete the method requires new training data and model training. This is time consuming. In addition, the training data should cover a wide range of different crack structures, shapes and sizes. This diversity in training data is particularly relevant in order to reduce the risk of model overfitting which arises from a large amount of parameters. As crack size and shape can hardly be controlled during stress tests, we need to rely on artificial training.
data. Therefore, our goal is to produce a set of realistic training data that covers the most important characteristics of real cracks.

Acknowledgements
We thank Christian Caspari from Civil Engineering department of TU Kaiserslautern for preparing samples and performing pull-out tests. We also thank Franz Schreiber from Fraunhofer ITWM for scanning the samples. This work was supported by the German Federal Ministry of Education and Research (BMBF) [grant number 05M2020 (DAnoBi)].

References