RHEINISCH-WESTFÄLISCHE
TECHNISCHE HOCHSCHULE AACHEN
Theoretical Computer Science
Prof. Dr. Rossmanith

Automatic Detection and Segmentation of Cracks in Underground Pipeline Images

Seminar: Medical Image Processing
Summer Semester 2006

Tim Niemueller

Matriculation Number: 236104

Supervisor: Benedikt Fischer
Institut für Medizinische Informatik, RWTH Aachen

Abstract

Buried infrastructures like sewers and water mains have to be checked for their current condition. Cracks are a strong indicator for the condition of a pipe. An affordable way to detect those cracks is to take images of the pipeline and use image processing techniques to detect cracks in these images. The methods used to accomplish this task are mathematical morphology and curvature evaluation to segment images with respect to a precise geometric model to define crack-like patterns. This paper discusses a paper by Shivprakash and Iyer where this method has been proposed. It describes the method, introduces the theoretical backgrounds, discusses the evaluation of the method in the paper and evaluates the paper itself.

Contents

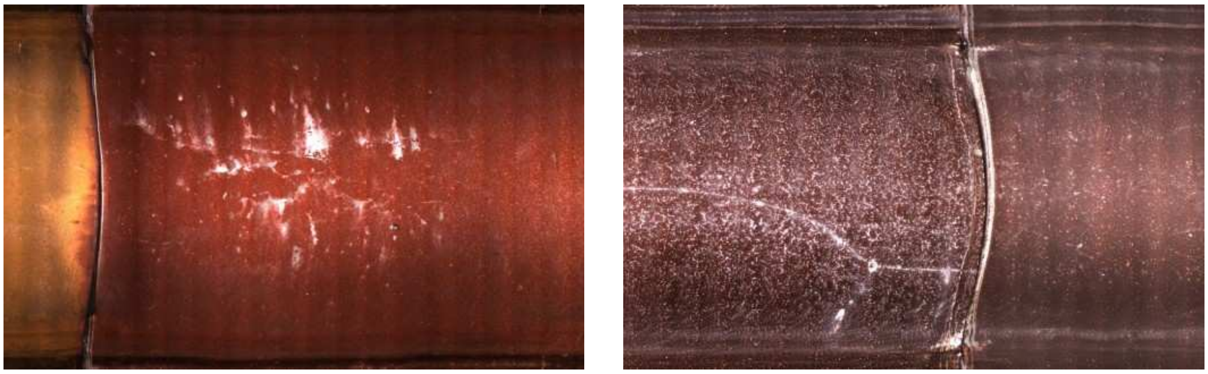
1 Introduction

The detection of cracks in underground pipelines is an important first step to keep sewer infrastructure intact. Up to now this is done by a visual inspection by a human operator. The images evaluated are usually taken with a closed circuit television (CCTV) system or with some kind of sewer scanner evaluation technology (SSET) which usually consists of a camera mounted on a robot manually controlled by the operator.

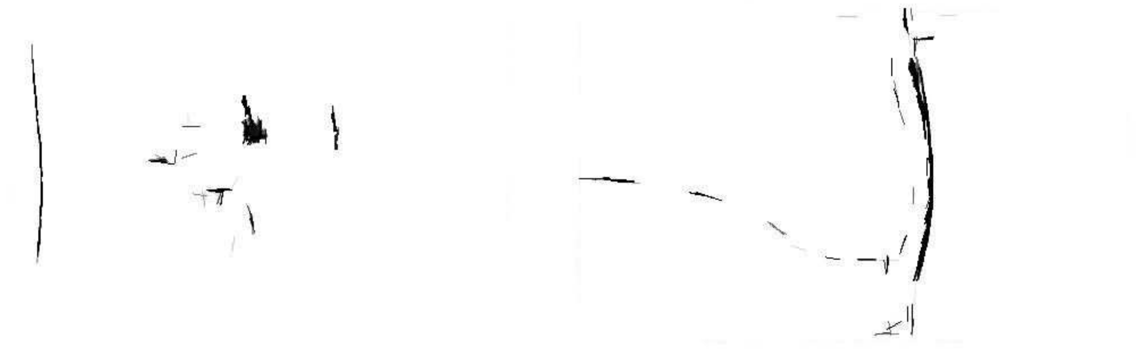
The detection of weaknesses and cracks in the pipeline is done offline after taking the images. The success of this task is influenced by the experience, the skill level and the concentration of the operator. Therefore, it is desirable to have an automated defect detection technology for reliable and reproducible results which are independent of the executing operator.

The basic task for automated condition assessment of underground pipelines is to detect cracks, holes, joints and fissures in the images taken via CCTV or SSET.

It has been observed that crack-like patterns in underground pipeline images seem to have a specific Gaussian profile. The paper that we are going to discuss ([IS05]) in more detail deals with the detection of these crack-like patterns in images. The techniques used in the paper for crack detection are mathematical morphology and linear filters.



(a) Two cracks with different crack patterns, background patterns and colors.



(b) The two binary crack maps which are the result of the discussed approach when fed with the cracks from 1(a).

Figure 1 shows an example for cracks in underground pipeline images. Note that these cracks have different crack patterns, background patterns and color. The discussed method has to deal with these problems and detect the cracks despite of these annoyances. You will also see that there are joints in the image that are characterized as crack by the algorithm. The paper states that there has to be a pre-processing step that sorts out these kind of things before we search for cracks. This is not part of the paper and not discussed here. Therefore, we do not consider this to be a problem of the algorithm.

In section 2 *Mathematical morphology and morphological operators* we are going describe the ideas behind mathematical morphology and how this can be applied to images for detecting features in an image based on their geometric model. In section 3 *Linear filters and curvature evaluation* we are going to describe briefly linear filters and how they can be used for curvature evaluation. In section 4 *Detection of crack features* we will describe how the given paper utilizes the previously described techniques for crack detection. In section 5 *Evaluation* we discuss the evaluation results mentioned in the paper and we will evaluate the paper itself. In section 6 we are going to give a short summary and a conclusion.

2 Mathematical morphology and morphological operators

2.1 Introduction

Mathematical morphology is a tool for extracting image components with respect to geometric features of these components. Instead of just manipulating an image it allows for extracting features from the image that can be used for representation and description (with enough knowledge about the image domain this can be used to get semantic information about the image). For example in the given domain cracks can be segmented from the background and can be semantically described with a set of morphological filters.

Mathematical morphology in image processing was originally developed by Matheron and Serra ([Ser82]) at the Ecole des Mines in Paris. It is a set-theoretic method of image analysis providing a quantitative description of geometrical structures. It was developed to analyse geological data and to detect the structure of the given material (for example to find inclusions in geological images).

Mathematical morphology can be used to detect the boundaries of objects, their skeletons or their convex hulls. It is also often used as a pre- and post-processing technique, for example for thinning or pruning of edges.

Morphological operations are based on simple expanding and shrinking operations with regard to a given structuring element. Originally mathematical morphology has been used for binary (black and white) images and has been extended later to be used with grayscale images as well.

2.2 Mathematical notation

Mathematical morphology is based on set-theoretic operations, namely shrinking and expanding a set (the image) based on another conditional set (structuring element). The two-dimensional (2D) image F is defined as a function:

$$F : \mathbb{Z}^2 \mapsto [I_{min}; I_{max}] \tag{1}$$

that maps $2D$ coordinates to the range $[I_{min}; I_{max}]$ which defines the possible intensity values. For a binary (black and white) image this is the range $[0; 1]$, for 8-bit grayscale images it is $[0; 255]$ which is used in the paper. Now we define two set transformations that will then make the notation for the basic morphological operations easy. The set

$$A_x = \{c \in \mathbb{Z}^2 \mid c = a + x \quad \forall a \in A\} \tag{2}$$

where $x \in \mathbb{Z}^{2\times 1}$ is the translation of A. Later we will use this notation to shift the structuring element over the image. The set

$$\hat{B} = \{d \in \mathbb{Z}^2 \mid d = -b \quad \forall b \in B\} \tag{3}$$

is the reflection of B.

In this paper only binary structuring elements are handled, so they can be defined as a function

$$E : \mathbb{Z}^2 \mapsto [0; 1] \tag{4}$$

where 0 means “don’t care” and 1 means “consider” which depends on the applied operation.

In general in images dark colors are considered to be the background of the image while white parts are the objects. For our purpose we will see that we need this the other way around. This makes the examples shown in this section confusing because they seem to have the opposite effect. This is because crack images were used for the examples and in these the white parts are the background and the dark parts are the objects.

2.3 Basic operations: dilation and erosion

The most basic morphological operations are dilation and erosion. These operations are the basic expanding and shrinking operations mentioned before. Erosion is the dual of dilation and vice versa. An interesting thing to note is that dilation can be used to enhance the white portions of an image while the erosion will help to strengthen the black portions. While we assume that in general white is the background and black the foreground the erosion effectively expands the objects while dilation thinnens them.

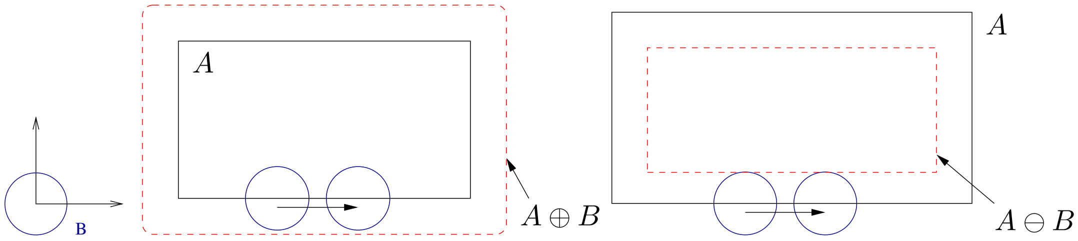


Figure 1: Schematic representation of dilation and erosion

2.3.1 Dilation

The dilation is the basic morphological expansion operation. We are now going to firstly describe that case of binary images and then describe a slight change that will be done to operate on grayscale images.

For dilation on binary images the structuring element is moved over the image. Each pixel that the mask "touches" is taken into the result image. This can be written as

$$F \oplus E = \{x \in \mathbb{Z}^2 \mid (\hat{E})_x \cap F \neq \emptyset\} \tag{5}$$

with image F and structuring element E. So the dilation is the set of points such that the structuring element and the image have common points if the structuring element's anchor point is at point x.

For grayscale images the dilation is defined differently. We define it as a function for a given image F at point P_0 and a given structuring element B:

$$\delta_B^e(F)(P_0) = \max_{P \in P_0 \cup e \cdot B(P_0)} (F(P)) \tag{6}$$

where e is a scaling factor for the structuring element (the structuring element is scaled in dimensions, not in values). This means that for grayscale images the point at coordinates P_0 will be set to the maximum of all points that are given in the structuring element.

See figure 1(b) for an example. As you see the structuring element is moved over the original set A with the structuring element B. The dashed line marks the resulting set of $A \oplus B$.

2.3.2 Erosion

The erosion is the basic morphological shrinking operation. Again we will look at the simpler case of binary images first and then give a different notation for grayscale images.

For the erosion of binary images the structuring element is moved over the images. A pixel is taken into the result image if the whole structuring element is included in the current neighborhood. This can be written as

$$F \ominus E = \{x \in \mathbb{Z}^2 \mid E_x \subseteq F\} \tag{7}$$

(a) Original image (b) Dilation (c) Erosion

Figure 2: Dilation and erosion

with image F and structuring element E. This makes the erosion the set of points where the structuring element is fully included in the original image when its origin is moved to x.

We write the extension of erosion to grayscale images as

$$\varepsilon_B^e(F)(P_0) = \min_{P \in P_0 \cup e \cdot B(P_0)} (F(P)) \tag{8}$$

with image F, structuring element B, a scaling factor e and a processing point P_0. The structuring element is moved over the image. Each pixel touched by the structuring element is considered and the minimum intensity for all these pixels is calculated. The processed pixel P_0 in the resulting image is then set to the calculated minimum of the neighbourhood with regard to the structuring element.

2.4 Combined operations: opening, closing and top-hat

Opening and closing are combinations of two basic operations. The opening is the dual of the closing and vice versa. These operations can be used to remove small objects or to close small holes. The top-hat operation can be used to remove a certain feature from the image. We only give the definition for the case of grayscale images here since this is what we need for crack detection later.

(a) Original image (b) Opening (c) Closing

Figure 3: Opening and closing

(a) Original image (b) Erosion (c) Opening (d) Inverted result

Figure 4: Hull by top-hat: Image 4(a) contains the original image, 4(b) is the erosion of 4(a), 4(c) is the opening of 4(b) and 4(d) contains the inverted result.

2.4.1 Opening and Closing

The morphological opening can be written as

$$\gamma_B^e(F) = \delta_B^e(\varepsilon_B^e(F)) \tag{9}$$

so it is a dilation of the eroded image F.

The morphological closing can be described as

$$\phi_B^e(F) = \varepsilon_B^e(\delta_B^e(F)) \tag{10}$$

so it is the erosion of the dilation of the image F. The factor e is a dimensional scaling factor (see above the dilation and erosion subsections). Note that both operations (dilation and erosion) of the opening and closing filters take the same structuring element.

The opening can be used to remove small objects from the image and the closing removes small holes. As mentioned earlier our background definition for crack images is the opposite of the usual definition in mathematical morphology so here opening and closing operations have a different effect: here the closing removes small objects while the opening removes small holes!

In figure 3 you see an example for a closing and an opening. Sub figure 3(b) shows the opening of 3(a). As you can see small holes have been closed while the smaller objects around remained unchanged. Figure 3(c) shows the closing of 3(a) where small objects have been removed and holes were retained.

2.4.2 Top-hat

This can be used to eliminate particular features from an image. The general method is to apply an opening or closing to an image followed by a subtraction with the original image with the order depending on the type of the feature. This is especially useful to subtract the background from the real object. While in many situations it is problematic to get a representation of the background it is in most cases easier to get a rough estimate of the features in the image. So to get the background you remove the feature from the image. If you now subtract this background image with the feature removed from the original image you will only get the desired feature. The order of the subtraction operation depends once more on what you consider to be the background and what the foreground.

The top-hat operation for an image with black background and white features can be described as

$$\tau_B^e(F) = F - \gamma_B^e(F) \tag{11}$$

So it first calculates the opening of the image F with regard to a given structuring element B and a scaling factor e and then subtracts the result from the original image. For dark images it has to be written as

$$\tau_B^e(F) = \phi_B^e(F) - F \tag{12}$$

This fundamental difference was overlooked in the paper. They describe the top-hat filter for white features while they write in the paper that it is used for dark features.

In figure 4 we have given a non-standard example about what you can do with a slightly extended top-hat filter (we allow more operations than just an opening or closing). It is used to detect the hulls of objects in an image.

2.5 Reconstruction operations

Up to now we have seen morphological operations that take an image and a structuring element as input to apply a specific operation depending on the structuring element to the image. Now we will learn about operations that take two images as their input and always use an isotropic structuring element for a dilation or erosion operation.

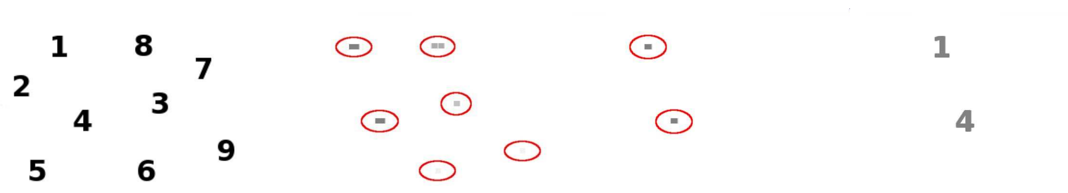
We apply the dilation or erosion with the isotropic structuring element to the first image and then use the second image to confine the result. Usually this is repeated until stability of the result has been reached and further application does not change the result anymore. This way the number of iterations does not have to be defined before running the operation.

These transformations are called geodesic reconstruction. By applying the basic morphological operations with an isotropic structuring element the original image (marker) is expanded or shrunken by one pixel in each iteration. This marker image is then confined by a so-called mask image. The number of iterations then gives a measure for the distance of the pixels.

We are now going to introduce the geodesic reconstruction by dilation and erosion. In the discussed paper they are called geodesic opening and geodesic closing. But since no opening or closing operation is performed[1] we followed other literature ([Vin93]) in their nomenclature.

Again the type of basic operation that is used depends on the color of the features you want to reconstruct. While you can use dilation for white features you will need erosion to reconstruct dark features.

[1]Opening and closing operation apply two primitive filters while the reconstruction only uses one



(a) Original image (b) Dilation $90°$, $45p$ (c) Dilation $0°$, $7p$ (d) Result

Figure 5: Reconstruction by erosion: Image 5(a) contains the original image, 5(b) is the dilated original with linear SE oriented at $90°$ with a length of 45 pixels (eroded and marked for better visibility), 5(c) is image 5(b) dilated with a linear SE oriented at $0°$ with a length of 7 pixels (eroded and marked), and finally 5(d) contains the reconstructed image.

2.5.1 Geodesic reconstruction by dilation

The geodesic reconstruction by dilation (or geodesic opening as it is called in the paper) is used to reconstruct white features in the image. The basic image is the first marker image that is used. A dilation operation is applied to the marker image and then confined with the mask image until stability has been reached and further applications of the reconstruction operation do not modify the result any more. The geodesic reconstruction by dilation is described as

$$\Gamma(F, G) = \delta^{(n)}_{B,G}(F) = \min\left(G, \delta^{(n-1)}_{B,G}\left(\delta_B(F)\right)\right) \tag{13}$$

with $\delta^{(0)}_{B,G}(F) = F$ and stability has been reached after n iterations (this means that for the given n the equation $\delta^{(n)}_{B,G}(F) = \delta^{(n+1)}_{B,G}(F)$ holds). F is the marker image (which is the original in the first iteration), G is the mask image, depending on what you want to reconstruct it is an image with the same definition range and $F \leq G$ for each iteration. B is the isotropic structuring element used throughout the operation, in general this is a 3×3 squared structuring element.

2.5.2 Geodesic reconstruction by erosion

The geodesic reconstruction by erosion (or geodesic closing as it is called in the paper) is used to reconstruct dark features in the image. It works analogously to the geodesic reconstruction by dilation but instead of the dilation an erosion is applied in each iteration.

$$\Phi(F, G) = \varepsilon^{(n)}_{B,G}(F) = \max\left(G, \varepsilon^{(n-1)}_{B,G}\left(\varepsilon_B(F)\right)\right) \tag{14}$$

with $\varepsilon^{(0)}_{g}(f) = f$ and stability has been reached after n iterations (this means that for the given n the equation $\varepsilon^{(n)}_{g}(f) = \varepsilon^{(n+1)}_{g}(f)$ holds). F is the marker image (which is the original in the first iteration), G is the mask image, depending on what you want to reconstruct it is an image with the same definition range and $F \geq G$ for each iteration. B is the isotropic structuring element used throughout the operation, in general this is a 3×3 squared structuring element.

Figure 5 shows an example of a geodesic reconstruction by erosion. The task is to isolate the 1 and the 4 in the image. As we can see these are the only numbers with a large vertical portion. So we use a long vertical linear structuring element and a dilation to extract this feature. As we see in 5(b) there are several numbers where we can find a long vertical linear portion. But as we can also see it is not as wide as in 1 and 4. So we do another dilation with a horizontal linear structuring element long enough to not fit into any numbers but only in 1 and 4. Finally we do the reconstruction by erosion and get the 1 and the 4.

With this example we have seen the basic structure that is underlying all efforts to create a morphological filter sequence to extract features. You determine what is characteristic for the feature in terms of its geometry and then this is exploited with appropriate morphological operations and structuring elements to extract these features.

3 Linear filters for curvature evaluation

Pictures of zebras and of dalmatians have black and white pixels, and in about the same number, too. The differences between the two are based on the ordering and characteristic appearance of groups of pixels in the image, rather than the individual pixel values.

We have seen before that mathematical morphology can be used to determine this information if a geometric model of the object that should be recognized is known before. Morphology segments the image by given geometrical patterns. Here we are going to introduce methods for obtaining descriptions of the appearance of a small group of pixels. We use weighted sums of pixel values and its neighbours. Depending on the weight matrix we can use it to find different image patterns.

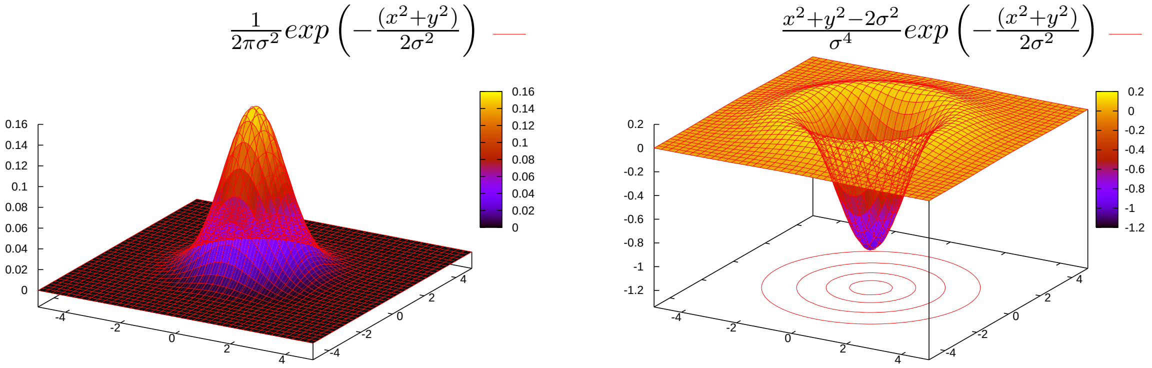
In this section we briefly describe linear filters in general and then discuss the Gaussian and Laplacian filters in more detail. As we will see in section 4 these filters are used for edge detection.

In general for a linear filter you have a matrix of weights of an arbitrary size which is called kernel. Then each pixel in the destination image is set to a weighted sum of the pixel in the original image and its neighbourhood depending on the weights defined in the kernel. This process is called convolution. For a given image F and a kernel K we say that K has been convolved with F.

We have seen that in the case of mathematical morphology you take a priory knowledge about object geometries into account. In the case of linear filters we use linear transformations in order to extract certain features.

3.1 Mathematical notation

We are going to use a different mathematical notation for linear filters compared to the description for morphology based filters. We define an input image F and an output image R as a two



(a) Gaussian

(b) Laplacian of Gaussian

dimensional matrix.

$$F \in \mathbb{Z}^{m \times n} \quad \text{and} \quad R \in \mathbb{Z}^{m \times n} \tag{15}$$

For linear filter we now define another matrix W of the same size that will contain the weights for the weighted sum for each pixel. This matrix will then be "moved" over the image with the center element being at the processed pixel at position (i, j). The pixel at position (i, j) in the resulting image R will then be set to the weighted sum of the original image depending on the weight matrix W.

As an example we give the equation to compute the local average over a fixed region. Here we use a block of $2k + 1 \times 2k + 1$ pixels around the processed pixel. For an input image F this gives the output

$$R_{ij} = \frac{1}{(2k+1)^2} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} F_{uv} \tag{16}$$

In this example each pixel is weighted by the same constant.

For further notion we define this weight matrix W now in a more compact form. We assume that all weights that are not explicitly stated are 0. This way we can have a small weight matrix of values which we call *kernel*. The process of applying a filter with a given kernel is called *convolution*. With a kernel H and an image F we get the result

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv} \tag{17}$$

if H is convolved with F to yield R.

3.2 Gaussian filter

We have already described a simple averaging filter as an example before. In many situations it seems to be more appropriate to use a kernel which has large weights in the center and and that fell of sharply with increasing distance from the center. This models the kind of smoothing that occurs with a defocused lens system. A symmetric Gaussian kernel fits these criteria and can be written as

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \tag{18}$$

and therefore a Gaussian kernel for a $2k+1 \times 2k+1$ block can be defined as

$$G_{ij} = \frac{1}{2\pi\sigma^2} exp\left(-\frac{((i-k-1)^2+(j-k-1)^2)}{2\sigma^2}\right) \tag{19}$$

In the Intel Integrated Performance Primitives (IPP) the Gaussian kernel is specified from the Gaussian distribution with a standard deviation of $\sigma = 0.85$ for a 3×3 matrix:

$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$

3.3 Laplacian of Gaussian

In grayscale images edges can be modeled as fast changes in brightness – for example a switch from black to white. For a one dimensional signal one can easily see that the derivative magnitude is extremal if the second derivative is zero. This means that is is a good idea to look where the second derivative is zero to find large changes and thus edges. This can be extended to two dimensions. For this we need an analogue to the second derivative which is rotationally invariant.

In [MH80] Marr and Hildreth proposed the Laplacian operator which has these properties. For the 2D case it is defined as

$$(\nabla^2 f)(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{20}$$

It is natural to smooth the image before applying the Laplacian. So if we set f in (20) to G_σ from (18) and remove the constant normalizing factor $\frac{1}{2\pi\sigma^2}$ we get for the Laplacian:

$$\mathcal{L}_\sigma(x,y) = \frac{(x^2+y^2-2\sigma^2)}{\sigma^4} exp\left(-\frac{(x^2+y^2)}{(2\sigma^2)}\right) \tag{21}$$

To get a kernel appropriate for linear filtering we now define a matrix L_σ^w. This matrix has a size of $w \times w$ and is filled with values calculated like the following:

$$L_\sigma^w(i,j) = \mathcal{L}_\sigma\left(i - \frac{w}{2}, j - \frac{w}{2}\right) \tag{22}$$

In the IPP the Laplacian with a 5×5 kernel is given as

$$L_1^5 = \begin{bmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4- & 3 & -1 \end{bmatrix}$$

To apply the Laplacian of Gaussian to an image this kernel L_σ^w is then convolved with the image F:

$$LoG_\sigma^w(F) = F \circ L_\sigma^w \tag{23}$$

4 Detection of crack features

In the previous sections we have described the techniques of mathematical morphology and linear filters for curvature evaluation. Now we are going to describe the way [IS05] combines these techniques to detect crack-like patterns in underground pipeline images.

Some information that is needed for reproducing the results is missing in the paper, i.e. the dimensions and resolution of the taken images. Therefore, we had to re-experiment with the parameters of the algorithm to reproduce the results with our own image processing pipeline (see 4.5). But it is likely that this would have been necessary in any case since it is unlikely that all the parameters like image size and camera resolution are the same. Also we could not get the original database used in [IS05].

The pipeline consists of three steps:

1. Improve the contrast of the RGB pipe image by enhancing the dark (crack) pixels from the "background" image.
2. Perform crack enhancement by applying two morphological filters and a linear filter combination for edge detection.
3. Detect the cracks by applying a set of morphological filters with a rotating linear structuring element.

We are now going to give an explanation what cracks are in terms of image processing using mathematical morphology and curvature evaluation with linear filters. After this we will describe the three steps of the image processing pipeline in more detail.

4.1 Cracks - properties and parameters

Up to now we have an intuition what cracks are. The next step is to explain what geometric features we are going to exploit with our filters to detect cracks in images. The properties of a crack a very closely related to the properties given for vessel-like structures in [ZK01]. A general assumption in the paper is the fact that the tree-like geometry of cracks is the only feature of interest in the image and everything else can be considered to be background. This reveals a major shortcoming. In images of underground pipelines you often have additional

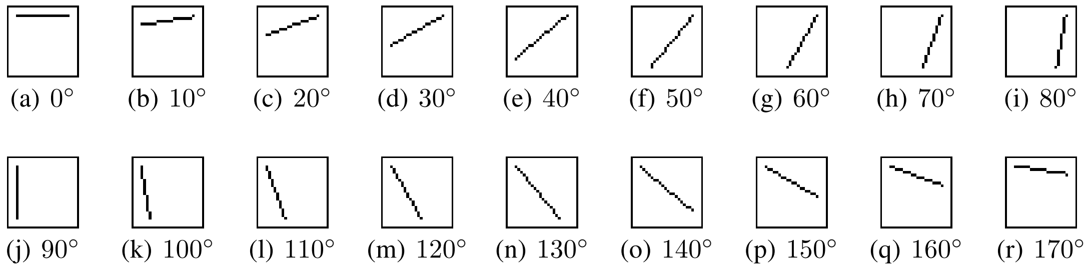


Figure 6: Linear structuring elements used with a length of 13 pixels oriented at every $10°$ from $0°$ to $180°$

features like junctions and joints. It is assumed that there has been another processing step before that has removed these features. This is not handled in [IS05]. It could be solved with a priori knowledge[2] or via other image processing step carried out before the crack detection.

The paper names three basic properties of cracks:

- intensity distribution of crack feature cross-section looks like a specific Gaussian curve
- they branch like a tree
- more or less have a constant width

Another important assumption is not explicitly named: the cracks have a high rate of linear parts. This means that the tree-like branching occurs in lines and is continued by lines. This is based on the observation made viewing a few images of the cracks.

As we know the structuring elements depend on the a priori knowledge gained about the object that we want to detect. So with the knowledge about crack features linear structuring elements have been chosen with a length of 12 pixels and a width of 1 pixel and oriented at every $10°$ from $0°$ to $180°$ which makes it 18 structuring elements. To combine the results of a morphological operation for every structuring element to one final result an appropriate operation (minimum, maximum, sum) has to be carried out which depends on the applied filter. In section 5 we will show how these parameters have been determined. See figure 6 for the linear structuring elements. We will refer from now on with B_i to these linear structuring elements.

The kernel for the Laplacian of Gaussian has been chosen appropriately for the expected width of the cracks.

4.2 Preprocessing

The preprocessing step is only described very briefly in the paper and it only consists of a few steps.

[2] If a map of the sewer network is available it may be used to know whether there are such features in the image

(a) Original image (b) Median (c) Compared

Figure 7: Preprocessing of the image: The original image 7(a) is first treated with a median filter 15×15 which results in 7(b). This is then compared to the original image and the minimum is calculated in 7(c)

First a median filter is applied to each of the R, G and B planes of the RGB image. The window size used for the media filter is 15×15 which makes it a strong smoothing filter after which the images are quite blurry. Small features are removed by this filter from the image and this can help to reduce the noise in the image. This median image is now the background image.

The next step is a comparison of the original (foreground) and the background image. This basically is a minimum filter which takes the minimum of the background and the foreground image.

In general this procedure slightly extends the scratches with a blurry surrounding. This enhances the contrast only very little. The median will in general narrow the distance between the highest and the lowest intensity. The following minimum comparison will restore the darkest features from the original image while preserving the higher lower intensity bound and thus lowering the contrast. After the preprocessing small holes have been closed. Objects tend to have a softer border. It is interesting to note that during the experiments good results could be produced by using a morphological smoothing (closing followed by an opening for cracks).

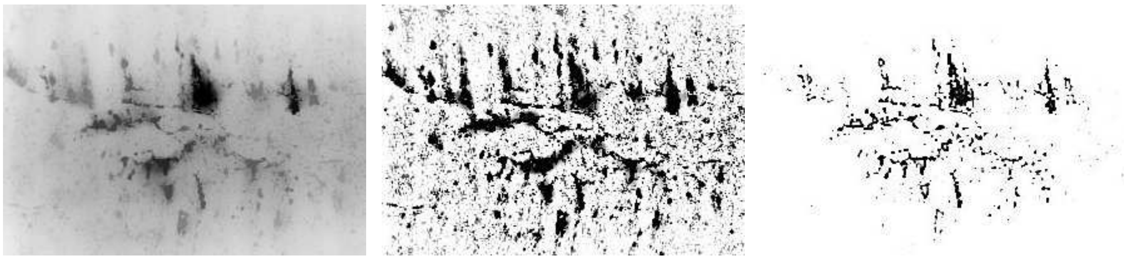
4.3 Crack enhancement

Before finally detecting the cracks some steps of enhancement are applied to the preprocessed image material. But de-facto the last of these steps also is the basic detection step as we will see below.

The first enhancement operation on the image is a morphological closing. From the results of the closing for each linear structuring element the minimum is taken for the result image. After this a geodesic reconstruction by erosion is carried out. This can be represented as

$$F_{Cl} = \Phi\left(\min_{i=1,\ldots,18}\{\phi_{B_i}(F_0)\}, F_0\right) \tag{24}$$

This will remove the first non-linear elements from the image or will weaken them. Small holes will be closed by this operation.



(a) Closing (b) Sum of top-hats (c) Laplacian of Gaussian

Figure 8: Enhancement of the image: 8(a) the compared image 7(c) is closed with the linear SEs, 8(b) is the result of the proposed sum of top-hats, 8(c) is the Laplacian of Gaussian of the closed image 8(a).

After this the paper describes a sum of top-hats. After experimental evaluation and studying more literature it seems clear that the operation stated in the paper does not yield the promised result. The paper requires the following top-hat operation:

$$F_{sum-th} = \sum_{i=0}^{17} \tau_{B_i}\left(F_{Cl}\right) = \sum_{i=0}^{17}\left(F_{Cl} - \gamma_{B_i}(F)\right) \tag{25}$$

This results in a dark image since you basically subtract the image from itself. A more useful operation for the top-hat would be

$$F'_{sum-th} = \left(\sum_{i=0}^{17}\left(\phi_{B_i}(F) - F_{Cl}\right)\right)^{-1} \tag{26}$$

This will calculate a sum of top-hats, where we use a top-hat with closing. We subtract the original image from the closing of the image. After this an inversion has to be done to have objects and background in the correct intensity range again. But since this introduced a lot of noise and false positives during the experiments this step was skipped. Figure 8(b) shows the result of the proposed corrected operation but the further pipeline used image 8(a).

The last enhancement step can already be considered to be part of the detection process. For this we apply the Laplacian of Gaussian to the closed image F_{Cl} (in the paper to the sum of top-hats image F_{sum-th}).

$$F_{lap} = LoG_2^{12}(F_{Cl}) \tag{27}$$

This will highlight all edges in the image irrespective of their direction. Since the size of the kernel has been specifically taken so that it is wide enough to include the full width of cracks we do not have holes in-between the edges. The parameters of the Laplacian of Gaussian depend on the expected width (which is by assumption almost constant) of the cracks in the image.

There is an example of the enhancement image processing steps in figure 8. First the closing is applied to the preprocessed image (8(a)). 8(b) shows the sum of top-hats proposed as correction to the paper above. But since this introduced noise 8(c) shows the Laplacian of Gaussian of the closed image.

(a) Closing (rec.) (b) Opening (rec.) (c) Large closing

Figure 9: Detection of cracks: Alternating filters are applied, 9(a) is the Laplacian image after a closing and reconstruction, 9(b) is opening (reconstructed) of the reconstructed closed image, 9 is the final result after applying a large closing operation.

4.4 Crack detection

The final detection and segmentation of the cracks is done with a set of alternating filter operations based on mathematical morphology.

First we do a geodesic reconstruction by erosion on the minimum of closings for the linear structuring elements:

$$F_1 = \Phi\left(\min_{i=1,\dots,18}\{\phi_{B_i}(F_{lap})\}, F_{lap}\right) \tag{28}$$

This will remove small objects from the Laplacian image F_{lap}. If we compare figure 8(c) and 9(a) we see that this reduces the amount of noise for the future treatment of the image. Then we do geodesic reconstruction by dilation on the maximum of openings for the linear structuring elements:

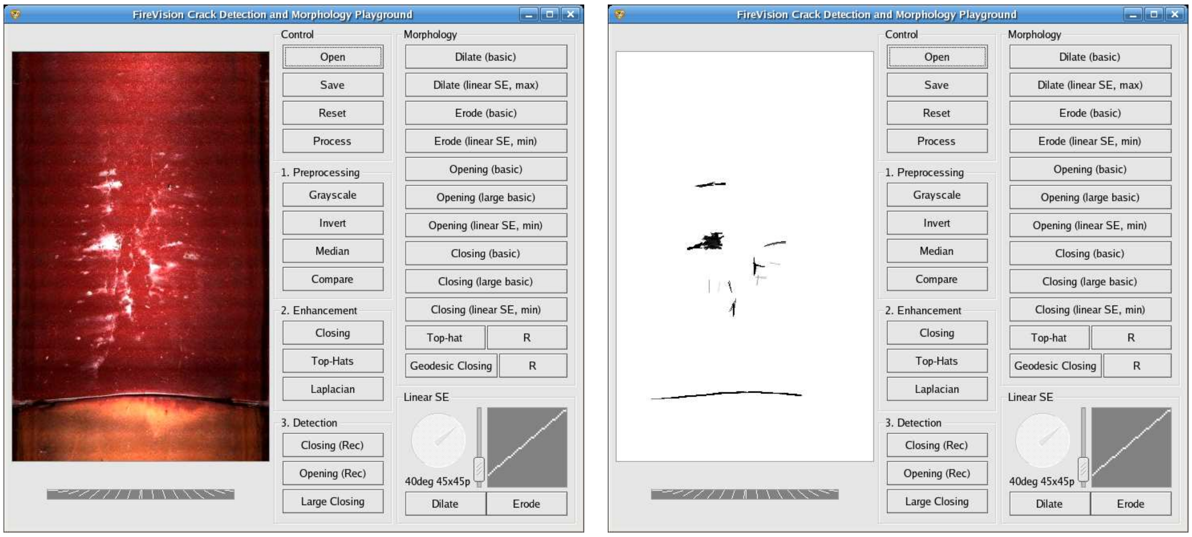
$$F_2 = \Gamma\left(\max_{i=1,\dots,18}\{\gamma_{B_i}(F_1)\}, F_1\right) \tag{29}$$

This will close small holes in the found cracks. After this we calculate the minimum of closings with linear structuring elements twice as long as the ones we used up to now (so if we used structuring elements B_i with a length of 12 pixels we now use a scaling factor of $e = 2$ for the filters effectively resulting in structuring elements with a size of 24 pixels):

$$F_{final} = \left(\min_{i=1,\dots,18}\{\phi^2_{B_i}(F_2)\}\right) \tag{30}$$

Everything that is not white in the resulting image is then considered to be a crack and we get a binary crack map.

In figure 9 you see the final steps of the processing. 9(c) is the final image, the so-called crack map.



(a) Original image

(b) Result crack map

Figure 10: FireVision Crack Detector and Morphology Playground

4.5 Experiments

For our own experiments we wrote a small application based on the FireVision framework from the AllemaniACs RoboCup team ([All]) of the RWTH Aachen. We implemented the morphology based and linear filters on top of the Intel Integrated Performance Primitives (IPP) library ([IPP]). The FireVision framework provided us with all the basic image processing functionality we needed while the IPP provided fast implementations for the dilate and erode operations that we could use to build up more complex filters like opening, closing, top-hat and reconstruction operations.

The experiments revealed quite a few problems in the paper that are not obvious if you do not try to reproduce the results. These problems are discussed in more detail in the evaluation section.

The discussed process has been slightly extended and modified for our own experiments by the following steps:

- The whole pipeline works on the YUV color space. Since the shown algorithm works on grayscale images this does not influence the algorithms performance. But it does change the first step since the preprocessing cannot work on individual R, G, and B planes.
- The example images have been inverted since the images provided by the Institute of Medical Informatics (IMI) show cracks which are the lightest features in the image and not the darkest.

Figure 10 shows the application before and after the algorithm has been applied to the shown underground pipeline image.

5 Evaluation

After showing the methods used to detect cracks and introducing the backgrounds and techniques used we are now going to evaluate the method and the paper.

First we will discuss the evaluation results mentioned in the paper, give information about expected detection, false positive and false negative rates and discuss the comparison of the proposed approach to other methods.

In the second part we will discuss the paper itself. We will show similarities to other papers and try to find measures for the overall quality of the paper.

5.1 Method evaluation results

The evaluation of the method has been split into two parts. First the optimal parameters for a given database of images have been identified. The second step was to compare the proposed approach to different methods, namely Otsu's thresholding and Canny's edge detector.

For both of these evaluation steps the cracks have been classified into three classes. They were chosen with respect to the most critical differences in images that can make or break an algorithm. The classes are different crack patterns, different background patterns (due to changing illumination and maintenance conditions while capturing the image or vegetation and algae in the pipes) and different colors of the images (which depends on the material that the underground pipeline is made of).

The images have been manually classified to have a ground truth image. These ground truth images have been used for the evaluation of the results.

5.1.1 Evaluation of parameter combinations

First the paper discusses the evaluation of different parameter combinations. As we have seen in section 4.1 the geometric features demand linear structuring elements. So the parameters that are of special interest here are the length and the degree of rotations of the linear structuring elements that have been chosen for the morphological filters. The degree of rotations determines the number of linear structuring elements that are used and combined in the morphological operations.

The interesting criteria for the evaluation are:

- probability of detection: how likely is it that a crack in the image is correctly detected
- probability of false positives: how likely is it that a crack in the image has been identified where there is none

- probability of false negatives: how likely is it that a crack is not detected in the image

The expected operation mode is that the machine does a pre-classification of underground pipeline images with regard to cracks. If a crack has been detected a human operator will inspect the image again and decide on the appropriate maintenance operations. So the basic idea is to reduce the workload by reducing the amount of images that the operator has to handle. Therefore it is bad to have false negatives. Since the idea is not to present images to the operator where no crack has been detected a crack would go unnoticed. On the other hand a false positive should be classified as "not a crack" by the human operator on manual inspection and the only problem is the increased workload of the operator (while this is not desirable as this raised the costs in the first place). Based on this the desired maximum probabilities have been stated in the paper as 7% for the false positive probability and 2% for the false negative probability.

In figure 11 charts from the paper have been reproduced. On the X axis are the different parameter combinations (D for the degree of rotations and S for the length of the linear structuring elements). For each combination the three classes are shown and the appropriate probabilities. The two columns that meet all desired criteria have been marked. In the paper the combination with a structuring element length of $S = 12$ pixels and a a degree of rotations $D = 10$ has been chosen.

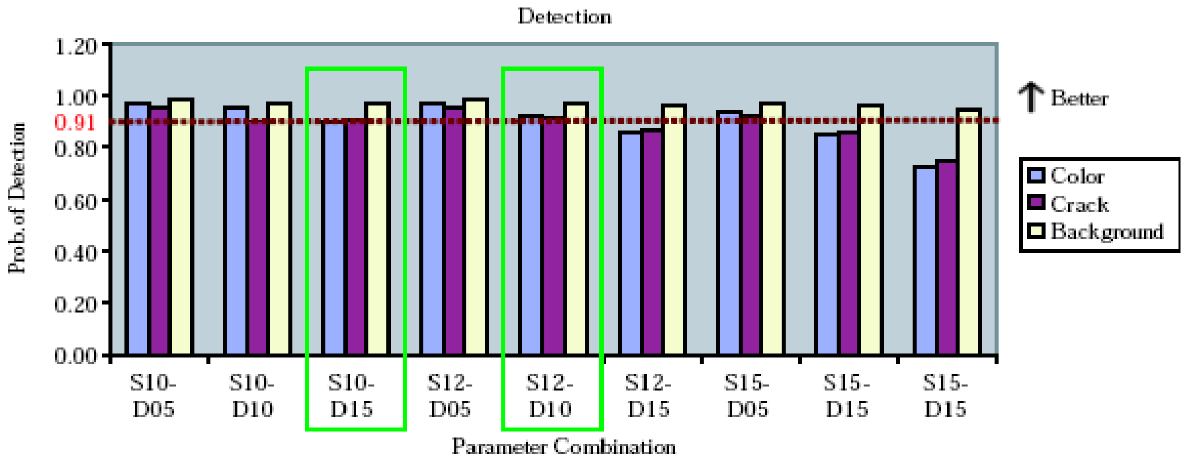
The evaluation was done on a database of 225 images from various cities. Since no comparable database was available and the authors did not answer requests for their database the results could not be verified.

The parameters directly depend on the image data that is fed to the algorithm. It depends on the image size and the resolution and the average length of cracks in the supplied images. So it is likely that in a different setup (different camera, image resolution etc.) this parameter evaluation has to be done again. Since these parameters were not mentioned in the paper basic information for a complete reproduction of the algorithm was missing.

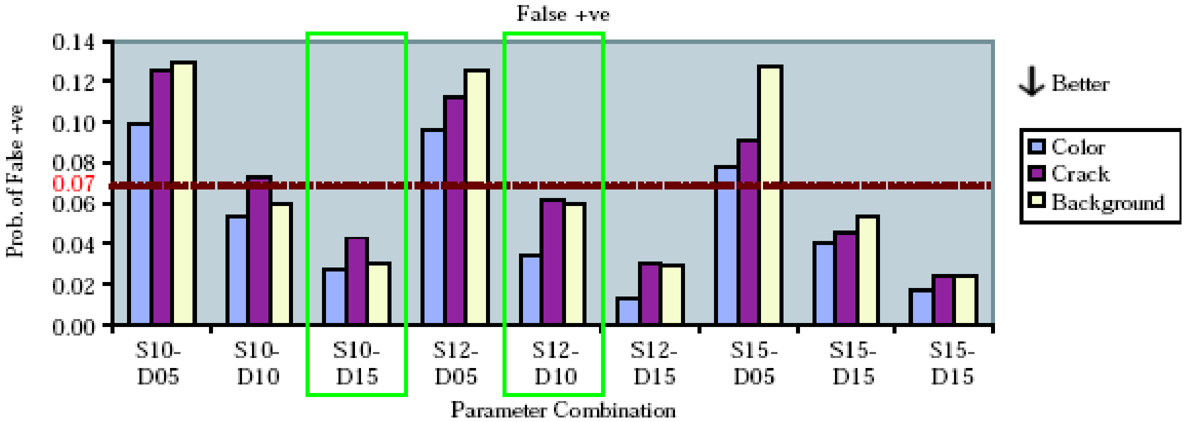
5.1.2 Comparison to different approaches

The authors compared the morphology-based approach to other approaches that have been proposed in the municipal pipeline infrastructure community before. Namely these are Otsu's thresholding and Canny's edge detector. Based on the mentioned ground truth image four criteria have been identified which are used to measure the performance of the different methods to have quantitative data for comparison.

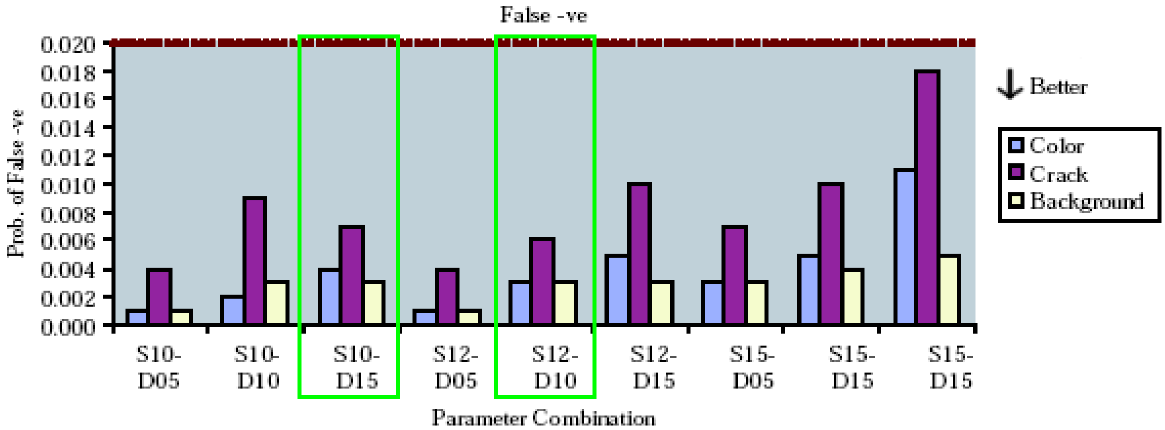
For the evaluation several different types of images are used. First the ground truth image (reference) as described above. Then we have the resulting crack map image from the algorithm (extraction). Both of these images are then eroded by a 5×5 square structuring element to create a buffer around the crack. The part of the method's crack map image that lies inside the buffer of the reference is called matched extraction, what lies outside is the unmatched extraction. The



(a) Probability of detection



(b) Probability of false positives



(c) Probability of false negatives

Figure 11: Charts for probabilities of detection of a crack, false positives and false negatives with regard to different crack classes and parameters

(a) Evaluation results for Otsu's thresholding

Class	Cracks	Background	Color
Completeness	0.98	0.61	0.62
Correctness	0.37	0.45	0.08
Quality	0.37	0.35	0.08
Redundancy	0.22	0.23	0.24

(b) Evaluation results for Canny's edge detector

Class	Cracks	Background	Color
Completeness	0.92	0.61	0.62
Correctness	0.20	0.44	0.07
Quality	0.20	0.34	0.07
Redundancy	0.15	0.17	0.14

(c) Evaluation results for proposed method

Class	Cracks	Background	Color
Completeness	0.95	0.88	0.90
Correctness	0.98	0.94	0.91
Quality	0.93	0.83	0.83
Redundancy	0.00	-0.01	0.00

Table 1: Quality measures for different methods and different classes of cracks.

part of the reference data that lies inside the extracted buffer is called matched reference, what lies outside is called unmatched reference.

The criteria that have been used are:

- $completeness = \frac{\text{length of matched reference}}{\text{length of reference}} \approx \frac{\text{number matched crack pixels of ref.}}{\text{number crack pixels of reference}}$

  The completeness is the percentage of the ground truth crack that is extracted as crack, i.e. the percentage of true crack pixels that could be extracted by the applied method and that lie in the extracted buffer.
  Completeness $\in$ $[0;\ 1]$ with 1 being the optimal value.

- $correctness = \frac{\text{length of matched extraction}}{\text{length of extraction}} \approx \frac{\text{number matched extracted crack pixels}}{\text{total number extracted pixels}}$

  The correctness describes the percentage of the correctly extracted crack data, i.e. the percentage of the extraction that matches the ground truth image.
  Correctness $\in$ $[0;\ 1]$ with 1 being the optimal value.

- $$redundancy = \frac{\text{length of matched extraction - length of matched reference}}{\text{length of matched extraction}}$$
  $$\approx \frac{\text{number of matched extraction - number of matched reference}}{\text{number of matched extraction}}$$

  The redundancy represents the percentage to which the matched extraction is redundant, i.e. it overlaps itself. Remember that a buffer method was used for evaluation and thus there can be different matched extraction and reference. This would not be possible without the buffer method.
  Redundancy $\in$ $]-\infty; 1]$ with 0 being the optimal value.

- $quality = \frac{\text{length of matched extraction}}{\text{length of extraction + length of unmatched reference}}$

  $\approx \frac{\text{number matched extraction pixels}}{\text{number extraction pixels + number unmatched reference pixels}}$

  The quality is a more general measure that combines completeness and correctness into a single measure to get a better feeling for the performance of an algorithm.
  Quality $\in\ ]0;\ 1]$ with 1 begin the optimal value.

Table 1 shows the evaluation results. The three sub tables represent the three compared approaches. Each table shows the results from the evaluation of the given four criteria for the three different classes of cracks. The green values mark the best value for the three methods. As it turns out only the completeness value is higher in Otsu's thresholding, all other values are closer to the optimal for the morphology approach. This completeness value does not count very much considered that the correctness of of Otsu's thresholding is just 20%. If we have a look at the combined quality measure it is easy to spot that the morphology approach is much better than Otsu's thresholding and Canny's edge detection.

These results could not be verified due to the lack of a comparable database. Also it is not clear what parameters were used for Otsu's thresholding and Canny's edge detector. So even with the database the results could not be reproduced reliably.

5.2 Paper evaluation

For the evaluation of the paper ([IS05]) one other paper is of special interest. In [ZK01] Zana and Klein present a method to detect vessel-like structures with the example of blood vessels in retinal images. The method presented is basically the very same method presented by Iyer and Sinha in [IS05]. Since Zana and Klein's paper is older than the presented paper by Iyer and Sinha it seems that it was used as a template. The proposed methods are the same with the only difference being that while vessels are the brightest structure in the element cracks are the darkest feature of the image. Since morphology has dual operations depending on the foreground-background relation it is easy to convert the method to work on this feature. The Laplacian of Gaussian kernel can easily be modified by changing the sign of all values in the kernel to operate the desired way.

The formula for the top-hat is wrong (see section 2.4.2). While thinking about this we came up with the idea that actually the Zana and Klein method was implemented and ran on inverted crack images and afterwards the notation was adapted to dark features. This is just a wild guess, but it is supported by the observation that in the section 5 "Summary of the proposed algorithm" of [IS05], the threshold applied to the final image is "intensity > 1". This would consider the white parts to be the cracks.

There are several points in the paper where the origin becomes obvious. One is the description of vessel and crack structures, which is almost literally the same. At another point Iyer and

Sinha use Δ in the description of the formula of a geodesic reconstruction by dilation (geodesic opening). But this is never defined. However, it is defined in the paper by Zana and Klein.

Some very important pieces of information are missing, for example the used image size and resolution, typical crack length and width and the parameters of the other algorithms used in the evaluation. This makes it hard to reproduce and check the results. An email requesting that information remained unanswered.

What is much more detailed in [IS05] is the evaluation section. While Zana and Klein give almost no evaluation results Iyer and Sinha give reasonable results and explain their methods. They compared the morphology approach to Otsu's thresholding and Canny's edge detector. What is yet missing is a comparison to a human operator. Of course this varies but there should be some data to get an estimate what human operators perform like on this task and what level of performance has to be reached for the automatic detection and segmentation to be a real alternative to the manual crack classification procedure.

6 Conclusion

In this seminar paper we have discussed the [IS05] with a method for automatic crack detection using mathematical morphology and linear filters. We introduced mathematical morphology and linear filters as tools for image processing in the domain of crack detection.

The presented approach has three steps. First the contrast of the image is enhanced and noise is reduced. In the second step the cracks in the image are enhanced. The Laplacian of Gaussian linear filter is used as a simple edge detection to fully separate the foreground from the background. In the last step a set of alternating morphological filters is applied to the image. Afterwards everything in the image that is below a given threshold is considered to be a crack and thus we have a binary crack map that tells us for every pixel if it belongs to a crack (dark pixel) or not (bright pixel).

The results mentioned in the paper are promising. They excel the other mentioned methods such as Otsu's thresholding and Canny's edge detection. It has to be stated that with reasonable efforts basic results from the algorithm could be reproduced. Because of the lack of a comparable image database and the needed parameters the results could not be reproduced in full.

The discussed paper ([IS05] by Iyer and Sinha) is very close to the paper [ZK01] by Zana and Klein. The described methods are the same used for both, cracks and retinal blood vessels. After all it seems that [IS05] was written pretty quickly which could explain the smaller and bigger errors that are in the paper.

In the discussion of the paper the question came up, if mathematical morphology is a worthwhile field of research for automatic crack detection. Since good results could be achieved after just a few hours of coding it seems promising that with more fine tuning mathematical morphology can produce adequate results. But in any case it is needed to compare the approach to modern (and often computationally expensive) methods to be able to make a justified decision.

References

[All] AllemaniACs RoboCup Team (RWTH Aachen). `http://robocup.rwth-aachen.de` (visited June 1st 2006).

[DL04] Edward R. Dougherty and Roberto A. Lotufo. *Hands-on Morphological Image Processing*. Tutorial texts in optical engineering. Society of Photo-Optical Instrumentation Engineers Press, 2004.

[FP03] David A. Forsyth and Jean Ponce. *Computer Vision – A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Pearson Education International, International edition, 2003.

[IPP] Intel Integrated Performance Primitives. `http://www.intel.com/cd/software/products/asmo-na/eng/perflib/ipp/index.htm` (visited July 31st 2006).

[IS05] Shivprakash Iyer and Sunil K. Sinha. A robust approach for automatic detection and segmentation of cracks in underground pipeline images. *Image and Vision Computing*, 23:921–933, 2005.

[Jäh97] Bernd Jähne. *Digitale Bildverarbeitung*. Springer Verlag, 4th edition, 1997.

[MH80] D. Marr and E. Hildreth. Theory of Edge Detection. In *Royal Society of London Proceedings Series B*, volume 207, pages 187–217, February 1980.

[Ser82] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[Soi98] Pierre Soille. *Morphologische Bildverarbeitung*. Springer, 1998.

[Vin93] Luc Vincent. Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, April 1993.

[ZK01] Frederic Zana and Jean-Claude Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing*, 10(7):1010–1019, July 2001.