MASTER

Noise reduction in digital subtraction angiography, using morphological image processing

van Eekeren, A.

Award date:
2002

Link to publication
Philips Medical Systems, Best, the Netherlands
X-ray Diagnostics Predevelopment Department

Technische Universiteit Eindhoven, the Netherlands
Department of Electrical Engineering
Signal Processing Systems

Confidential

Noise Reduction in Digital Subtraction Angiography,
using Morphological Image Processing

Adam van Eekeren
August, 2002

Graduation report

Adam van Eekeren
Eindhoven, August 2002
Supervisors: Dr. ir. P. Rongen (PMS)
Ir. H. Stegehuis (PMS)
Prof.dr.ir. J.W.M. Bergmans (TU/e)
Dr. ir. J.A. Blom (TU/e)
Preface

This report has been written as a completion of my graduation project, which I have done within the XRD-predevelopment department of Philips Medical Systems from the 1st of November 2001 till half of July 2002. It marks as well the end of my study on 'Information technology' at the Technische Universiteit Eindhoven.

First, I would like to thank Prof.dr.ir. Bergmans and Hans Blom for offering the opportunity to get in contact with Philips Medical Systems. Also I am grateful for their guidance during the project.

Next, I would like to thank Philips Medical Systems and the XRD-predevelopment department for the opportunity to work in a professional and very interesting environment.

My special thanks goes to Peter Rongen, who was my inspirer, and Herman Stegehuis for their support and useful suggestions. I have learnt a lot from them, thanks! Also I would like to thank Marcel Quist for his assistance in understanding the COSMIC algorithm. Further, I also want to thank all other people at the department who helped me, especially the guys that made lunch more enjoyable: Arnold, Hans, Paul and Peter.

Last, but not least, I say thanks to my colleague-trainees, who I’ve ‘survived’, for the pleasant times and the many cups of tea they’ve brought me: Sanne, Zaher, Tjerk and Bart.
Summary

Digital Subtraction Angiography (DSA) is a special form of angiography where a so-called mask image is subtracted from a live image, containing vessels filled with contrast agent. In this way, the background is subtracted from the actual images, resulting in an image sequence containing only vessels. A so-called ‘vessel-over-bone’ problem arises when a vessel, which appears dark in the image, crosses another dark structure, like bone. The contrast between vessel and bone is usually very bad, so after subtraction, the part of the vessel crossing the bone is hardly visible.

In order to solve this ‘vessel-over-bone’ problem, a previous trainee at XRD predevelopment, Haupert [7], has designed a new DSA process, which makes use of a logarithmic Look-Up Table. As a consequence, a lot of noise is introduced in the dark regions of the DSA images. It was found out by Haupert that the present noise could be removed quite successfully by using morphological methods. However, it was reasoned that even better results might be obtained by improving the designed noise reduction algorithm.

In this report some new noise reduction algorithms based on the ideas of Haupert have been designed to improve the already obtained results. First the noise is analysed to get an idea of the noise statistics of DSA images. Next the basics of morphological image processing are described to give the reader some insight in the morphological operators used in the proposed morphological filters. Then, the noise reduction algorithm of Haupert is explained in detail and some self-designed algorithms are explained. Because in these filters so-called directional structuring elements are used, it seems promising to perform first a structure analysis on the input image. This information can be used to steer the already designed directional morphological filters and may give better results.

At the end of this report simulation results are given for the designed filters. From these simulation results it becomes clear that the structure-steered directional morphological filters perform better than their non-steered directional variants. So it may be concluded that the use of a structure analysis of the image improves the performance of directional morphological filters. Simulation results of some structure-steered filters show that especially the vessel walls are more sharp (more contrast) in comparison with results of their non-steered variants.

From the filters tested in this report, a structure-steered variant of the so-called MIC filter performs best. However, only a small improvement was obtained with structure analysis. Comparing this result with Haupert’s result, the vessels have a smoother look, while the contrast is more or less the same. Also the standard deviation of the noise is smaller according to a performed noise analysis. Although not much research has been done in the area of multi-scale filtering, the most promising filter (MIC) seems one with a multi-scale approach. Therefore, in this multi-scale area, more should be experimented in order to obtain even better results.
Contents

Preface ................................................................................................................. i
Summary .............................................................................................................. iii
1 Introduction ...................................................................................................... 1
2 DSA image processing ..................................................................................... 3
  2.1 Introduction ................................................................................................ 3
  2.2 Principle .................................................................................................... 3
  2.3 Explanation of current DSA process .......................................................... 5
  2.4 The ‘vessel-over-bone’ problem ................................................................. 7
  2.5 Explanation of a new DSA process .............................................................. 8
    2.5.1 Logarithmic LUT ............................................................................. 8
    2.5.2 Scatter correction .......................................................................... 11
  2.6 Additional temporal filtering ..................................................................... 13
3 Noise analysis .................................................................................................. 15
  3.1 Introduction ................................................................................................ 15
  3.2 Noise types ............................................................................................... 15
  3.3 Noise sources ........................................................................................... 16
  3.4 Noise analysis ........................................................................................... 17
    3.4.1 Phantom image with additive Gaussian noise ............................... 18
    3.4.2 Phantom image with multiplicative noise ..................................... 20
    3.4.3 Phantom image with a predefined noise curve ......................... 21
    3.4.4 X-ray noise in a static phantom image ....................................... 22
    3.4.5 X-ray noise in a Digital Subtraction Angiography image ........... 24
  3.5 Image filtering techniques ......................................................................... 26
    3.5.1 Temporal filtering ........................................................................... 26
    3.5.2 Spatial filtering ............................................................................... 26
4 Morphological image processing ................................................................. 27
  4.1 Introduction ............................................................................................... 27
  4.2 Binary erosion and dilation ....................................................................... 27
  4.3 Binary gradient ........................................................................................ 29
  4.4 Binary opening and closing ...................................................................... 29
  4.5 Greyscale opening and closing ................................................................ 30
  4.6 Greyscale gradient ................................................................................... 32
  4.7 Greyscale erosion and dilation ................................................................ 32
  4.8 Reconstruction .......................................................................................... 34
    4.8.1 Binary reconstruction ..................................................................... 34
    4.8.2 Greyscale reconstruction ............................................................... 35
  4.9 Some simple morphological filters ............................................................ 37
  4.10 Morphological decomposition ................................................................ 38
5 Noise reduction using mathematical morphology .................................... 41
  5.1 Introduction ............................................................................................... 41
  5.2 Haupert’s algorithm .................................................................................. 41
    5.2.1 Principle ............................................................................................ 41
    5.2.2 Directional closing and opening ..................................................... 41
  5.3 Filter with reconstruction .......................................................................... 45
  5.4 Filter with elements in series .................................................................... 47
  5.5 MIC algorithm ........................................................................................... 48
    5.5.1 Principle ............................................................................................ 48
    5.5.2 Original algorithm .......................................................................... 49
    5.5.3 Modified algorithm ........................................................................ 49
    5.5.4 Some considerations ....................................................................... 52
  5.6 Conclusions ............................................................................................... 53

© 2002 Philips Medical Systems Nederland B.V.
Philips Medical Systems
6 Structure-steered directional morphological filtering ................................................. 55
   6.1 Structure analysis ................................................................................................................ 55
      6.1.1 Theory of the structure tensor ................................................................................. 55
      6.1.2 Eigenvalues and vectors of the structure tensor ...................................................... 58
   6.2 The $\alpha$- and $\beta$-map .................................................................................................. 61
   6.3 Calculation of weight factors for directional morphological filters ................................. 63
   6.4 Combination of analysis with directional morphological filters ..................................... 65
      6.4.1 Global structure ............................................................................................................. 65
      6.4.2 Combination of the $\alpha$- and $\beta$-map ................................................................. 65
      6.4.3 Combination of weight factors and the new $\alpha$-map .............................................. 65
      6.4.4 Steered directional morphological filter ................................................................. 67
      6.4.5 Considerations ............................................................................................................. 68
   6.5 Conclusions ....................................................................................................................... 69

7 Simulation results ............................................................................................................ 71
   7.1 Haupert's algorithm ............................................................................................................ 73
   7.2 Reconstruction filter, based on Haupert's algorithm ...................................................... 73
   7.3 Modified MIC algorithm .................................................................................................. 74
   7.4 Structure-steered (closing, opening) filter, $K = 20$ ....................................................... 74
   7.5 Structure-steered (closing, opening) filter, $K = 5$ .......................................................... 75
   7.6 Structure-steered (dilation, erosion, erosion, dilation) filter, $K = 20$ ............................. 75
   7.7 Structure-steered (dilation, erosion, erosion, dilation) filter, $K = 5$ ............................... 76
   7.8 Structure-steered (dilation, erosion + reconstruction) filter ........................................... 77
   7.9 Structure-steered variant of Haupert's algorithm ............................................................ 78
   7.10 Structure-steered variant of modified MIC algorithm ................................................... 79

8 Conclusions and recommendations ............................................................................... 83
   8.1 Conclusions ....................................................................................................................... 83
   8.2 Recommendations ............................................................................................................ 84

List of abbreviations ............................................................................................................. 85
References .................................................................................................................................. 87
Appendix A: Matlab code noise analysis.................................................................................. 89
Appendix B: Flow diagram noise analysis ............................................................................... 91
Appendix C: Extra simulation results ...................................................................................... 93
1 Introduction

Cardiovascular diseases are the main cause of death in industrialized countries, accounting for more than 40% of all deaths. Diseases of the circulatory system can be diagnosed and treated thanks to X-ray angiography, which is an imaging technique that allows a doctor to monitor on-line injections of contrast medium in blood vessels, for the diagnostics of (partly) obstructed or dilated vessels (stenoses or aneurysm).

Digital Subtraction Angiography (DSA), is a special form of angiography where a so-called mask image is subtracted from a live image, containing vessels filled with contrast agent. In this way, the background is subtracted from the actual images, resulting in an image sequence containing only vessels. A particular problem (called 'vessel-over-bone' problem) arises when a vessel, which appears dark in the image, crosses another dark structure, like bone. The contrast between vessel and bone is usually very bad, so after subtraction, the part of the vessel crossing the bone is hardly visible.

In order to solve this 'vessel-over-bone' problem, a previous trainee at XRD predevelopment, Haupert [7], has designed a new DSA process, which makes use of a logarithmic LUT. As a consequence, a lot of noise is introduced in the dark regions of the DSA images. It was found out by Haupert that this noise could be removed quite successfully by using morphological methods. However, it was reasoned that even better results might be obtained by improving the designed noise reduction algorithm.

The main goal of this report is to present the design of some new noise reduction algorithms based on the ideas of Haupert, which are expected to give better results. In contrast to Haupert's algorithm, before filtering, a structure analysis will be performed, which makes it possible to steer the so-called directional structuring elements in a morphological filter. Also the possibility of using the designed filters on different scales will be studied. At the end, simulations will be done and the results will be compared.

The structure of this report is as follows. In Chapter 2, Digital Subtraction Angiography image processing will be the main subject: first the current DSA process will be explained and, next, the new DSA process with the modifications to improve the ‘vessel-over-bone’ contrast.

In Chapter 3, a method to analyse noise will be discussed and some simulations are presented that show the noise statistics of some images with pre-defined noise and some X-ray images, where the noise statistics are not a-priori known.

In Chapter 4, the basics of morphological image processing will be treated and some possibilities will be given to build a multi-resolution pyramid with morphological operators. This should give the reader more insight in the noise reduction algorithms that will be explained further on in this report. The next chapter describes different noise reduction algorithms that make use of morphological operators. First the noise reduction algorithm of Haupert is explained in detail, which forms, in fact, the basis of all other algorithms discussed in this report. Also other self-designed algorithms will be explained in this chapter.

In Chapter 6, first a method to obtain structure information from an image will be explained. Next, this information will be integrated with the already known morphological filter concepts from Chapter 5. The simulations that are done for the proposed filters are given and compared in Chapter 7. Finally, in Chapter 8, conclusions will be given according to the obtained results and some recommendations will be given that may be useful for future research on this subject.
2 DSA image processing

2.1 Introduction

In this chapter a new subtraction algorithm will be explained. Also it will become clear why a noise reduction method is necessary for this new algorithm. Digital Subtraction Angiography is a commonly used technique in medical imaging. It is used to make blood vessels in a so-called angiogram better visible, in order to make a good diagnosis of the patient's condition.

The DSA technique has been developed by medical physics groups at the University of Wisconsin (USA), the University of Arizona (USA) and the Kinderklinik in Kiel (Germany) during the 1970s. Developments in this area were made possible by the introduction of cesium iodine image intensifiers and the progress in digital electronics. In 1978 the feasibility of DSA for human patients was demonstrated and the first prototype DSA system was introduced in 1980. DSA is part of the Digital X-ray image acquisition field, which involves the digitisation of the output of a video camera of an x-ray system.

2.2 Principle

Because the attenuation of X-rays is equal for blood and the surrounding soft tissue, it is necessary to use some contrast agent. Iodine is a suitable material for this purpose, because its biological compatibility is good and it is easily brought into the blood circulation of the patient. To obtain now only the vessels, image subtraction is done.
First, a so-called mask image is made without using the contrast material. Next, the contrast material is injected into the blood and another image is made, the so-called live image. If both images are subtracted from each other and no patient motion has taken place, an image is obtained which shows only the vessels. In practice, the mask image is usually one of the first two images of the run where the injection of contrast agent is done, because it usually takes a few frames before the contrast agent becomes visible in the image.

The photon attenuation that takes place due to the absorption in the body and the contrast material can be described with the following formula:

\[ I_{\text{out}} = I_{\text{in}} \exp(-\mu d) \] \hspace{1cm} (2.1)

Where
- \( I_{\text{out}} \) = the transmitted photon intensity with energy \( E \) (J)
- \( I_{\text{in}} \) = the incident photon intensity with energy \( E \) (J)
- \( \mu \) = the energy dependent absorption coefficient of the material (m\(^{-1}\))
- \( d \) = the distance travelled through the material (m)

When it is assumed that the contrast material in the vessels absorbs more radiation than the tissue around it and that no scatter radiation has been generated, the following equations are true for the intensity of a mask image and a live image, respectively.

\[ I_{\text{mask}} = I_{\text{in}} \exp(-\mu d) \] \hspace{1cm} (2.2)

\[ I_{\text{live}} = I_{\text{in}} \exp(-\mu d - \mu_v d_v) = I_{\text{mask}} \exp(-\mu_v d_v) \] \hspace{1cm} (2.3)

Where
- \( \mu_v \) = the energy dependent absorption coefficient of the contrast material (m\(^{-1}\))
- \( \mu \) = the energy dependent absorption coefficient of the other tissues in the body (m\(^{-1}\))
- \( d_v \) = the thickness of the vessel (m)
- \( d \) = the thickness of the other tissues (m)

If both images are subtracted from each other the intensity becomes:

\[ D = I_{\text{live}} - I_{\text{mask}} = I_{\text{mask}} (\exp(-\mu_v d_v) - 1) \] \hspace{1cm} (2.4)

It is clear that this intensity is dependent on the mask image as well as the live image. This means that artefacts exist due to anatomical details in the body and not only the vessels are visible. To overcome this problem, theoretically, the natural logarithm of the mask- and live image can be calculated before subtraction. This gives the following result:

\[ D_{\text{ln}} = \ln(I_{\text{live}}) - \ln(I_{\text{mask}}) = \ln(I_{\text{mask}} \exp(-\mu_v d_v)) - \ln(I_{\text{mask}}) \]

\[ D_{\text{ln}} = \ln(I_{\text{mask}}) - \mu_v d_v - \ln(I_{\text{mask}}) \]

\[ D_{\text{ln}} = -\mu_v d_v \]

(2.5)

This equation indicates that the intensity of the subtracted image is only dependent on the contrast material and that there will hardly be any information of the surrounding anatomy. Only motion of the patient or X-ray equipment can cause some artefacts in the subtracted image. Another advantage of the logarithmic subtraction is that the intensity of the subtracted image has a linear relation with the thickness of the vessel. In our case thick vessels will have a smaller intensity (darker) than thin vessels (brighter). Note that the relation in (2.5) also can be obtained by dividing \( I_{\text{live}} \) by \( I_{\text{mask}} \) and performing subsequently the natural logarithm.
2.3 Explanation of current DSA process

The DSA process as it is currently used in the cardio-vascular systems of Philips (VISUB) can be seen in Figure 2.2 and will be referred to as 'current DSA process'.

The first step in this process is the conversion from the photon intensities to a digital video signal (XTV camera and AD converter). The transfer function that describes this relation is called the Sensor Noise Curve (SNC) and has been designed to minimize the noise sources introduced by the camera. The application factor (AF), that can be adapted to a specific clinical investigation, determines the stabilisation point of the dose control loop (see Figure 2.3).

The second step is the Non-Subtraction Look-Up-Table (NS LUT) (see Figure 2.4). It has been included to compensate the SNC and its application factor (AF). Depending on the application factor
and the choice of the system grey-scale transfer, this NS LUT selects some grey levels of interest to optimise for grey level quantisation.

![Non-subtraction LUT](image1)

**Figure 2.4:** Example of a Non-Subtraction LUT.

As a third step, pixel shifting is performed to reduce global motion artefacts, usually caused by a table or patient shift, between the live and mask image.

The next LUT in the DSA chain is the Subtraction LUT (S LUT), which can be seen in Figure 2.5. This LUT is designed to approximate together with the NS LUT, a logarithmically shaped LUT, in order to make the grey levels directly proportional to the thickness of the vessels with contrast medium.

![Subtraction LUT](image2)

**Figure 2.5:** Two examples of Subtraction LUTs.

Next, the subtraction takes place and some more LUTs are used to get the final DSA image. First, the SubGain LUT is applied to correct the loss of amplitude after image subtraction by increasing the dynamics of the vessels in the dark parts. Note that this LUT has a small linear part around the background intensity to decrease the noise in that region (see Figure 2.6). This is called 'noise coring'.
Finally, a windowing LUT and a monitor LUT are used to display the DSA image in a right way on a special medical greyscale monitor. The windowing LUT performs a gain and an offset correction and clips it to an 8-bit image. The monitor LUT corrects for gamma distortion of the monitor when the image is displayed on the screen.

2.4 The 'vessel-over-bone' problem

With the current image processing, a particular problem (called 'vessel-over-bone' problem) arises when a vessel, which appears dark in the image, crosses another dark structure, like bone. The contrast between vessel and bone is usually poor, so that after subtraction the part of the vessel crossing the bone is hardly visible (see Figure 2.7).

Part of the problem is caused by the use of unsuitable intensity look-up tables (NS LUT and S LUT) before subtraction, as well as the presence of scatter radiation (secondary radiation). The current image
processing is optimised to minimize the noise, especially in the dark areas, which is not compatible with increasing the ‘vessel-over-bone’ contrast. Hence, the limitation of X-ray noise results in a lower contrast between vessel and bone in the dark areas (bone), than compared with the bright areas (soft tissues).

To overcome this ‘vessel-over-bone’ problem a new DSA process has been designed by Haupert [7], a former trainee at PMS. In the next section this new process will be discussed.

2.5 Explanation of a new DSA process

A block diagram of the new DSA process can be seen in Figure 2.8.

![Diagram of new DSA process]

Figure 2.8: New DSA process designed by Haupert [7].

As one can see, the difference between the new process and the previous one are the ‘Inv LUT’, ‘Scatter correction’, ‘Log LUT’ and the ‘Noise Reduction’. Also the ‘SubGain LUT’ has been adapted to the new DSA process.

The new process starts after the application of the Sensor Noise Curve (SNC) and the Non-Subtraction LUT (NS LUT) in order to keep the current pre-processing chain for non-subtracted images, which are stored on the disk. The DSA process can actually be seen as an extension of the conventional digital radiography where the radiologist can choose to perform a subtraction on already stored images.

The Inverse LUT (Inv LUT) performs an inverse transformation of the SNC and the NS LUT to get back the original images, coming out of the XTV camera. These are required before starting the new DSA process.

2.5.1 Logarithmic LUT

The main topic of the new process is the logarithmic curve. It takes the place of the Sensor Noise Curve (SNC), the non-subtraction (NS) LUT and the subtraction (S) LUT.

The natural logarithmic curve is quite different from the combined SNC, NS and S curves, as can be seen in Figure 2.9.
2. DSA image processing

The major difference between the logarithmic curve and the original LUTs is in the low intensities where the 'vessel-over-bone' problem occurs. To have a better understanding of the logarithmic effect, its gain has to be compared with the corresponding gain of the current DSA LUTs. This gain is achieved by the curve’s first derivative with respect to the input intensity (see Figure 2.10).

Figure 2.9: Comparison between the logarithmic curve and the original LUTs.

Figure 2.10: Gain of the Logarithmic LUT and the combined SNC+NS+S LUTs.
To improve the 'vessel-over-bone' problem, the contrast in the dark regions of the bone must become larger, whereas contrasts in bright regions should not change too much. The current LUTs cannot respect this condition because they combine noise reduction with the logarithmic effect. In the new process however, the noise reduction is separated from the contrast enhancement problem. Therefore a logarithmic LUT has been chosen to visualize the vessels in the best way. The logarithmic LUT is continually increasing with a very steep slope at the beginning, where the gain has to be very high to really distinguish vessels to bones. A major disadvantage of the logarithmic LUT however is that the noise present in the dark regions is boosted.

As shown in Figure 2.11, the contrast in the dark areas is already improved after applying the logarithmic LUT, without any other process. Thus, a logarithmic–like LUT seems to be a good solution of the 'vessel-over-bone' problem.

![Figure 2.11: Effect of the logarithmic LUT in the new DSA process.](image)

A second advantage of the logarithmic LUT is that the gain is smaller for vessels in the bright part (see Figure 2.10). Therefore a more uniform colour in the entire vessel may be attained. The logarithmic curve has this counterbalancing effect thanks to the high gain where the 'vessel-over-bone' problem occurs and the low gain in the bright areas where the vessels are too dark.

Briefly, the logarithmic curve has several advantages:
- Respects the DSA theory.
- Increases the gain of the structures (bones and vessels) in the dark areas where the 'vessel-over-bone' problem occurs. Thus, a better contrast between bones and vessels is obtained.
- Counterbalances the contrast between an area without and an area with bone, respectively, in order to get the same intensity for vessels with the same size.

Instead of a logarithmic curve it is also possible to make use of a Lognormal distribution as proposed by Kharboutly [8]. This distribution has the advantage that it is more flexible and it doesn't have a gain that tends to infinity at zero intensity. Because the noise in the dark areas also is increased by this distribution, an extra noise reduction step in the DSA process still seems necessary. Therefore, in this report, the Lognormal distribution has not been used.

As said before, the main disadvantage of a logarithmically shaped curve is that it increases the noise in the dark areas. Therefore a proper noise reduction method must be found to manage this noise. This will be explained in chapter 5.
2.5.2 Scatter correction

Scatter correction is the second part of the new process before the subtraction. It is very important to get the pixel value as close as possible to primary radiation in order to apply the correct gain to the right pixel, especially for the very dark pixels.

Before explaining the scatter correction, the origin and the effects of the scatter radiation have to be described.

DSA images are degraded by scattering of X-rays from within the patient. The result can be simply described by the following equation:

\[
\text{Total radiation} = \text{Primary radiation} + \text{Scatter radiation.}
\]

If we heavily simplify the problem, the scatter radiation can be considered as an invariant offset in the whole image:

\[
T_k = P_k + s
\]  

(2.6)

Where

- \( T_k \) = the intensity of the total exposure of pixel \( k \)
- \( P_k \) = the primary intensity of pixel \( k \)
- \( s \) = the intensity of the scatter radiation, modelled as a constant

In DSA image processing, the logarithm of the total exposure images (with scatter radiation) is calculated before doing the subtraction. Then the subtraction of the mask image (\( M \)) from the live image (\( I \)) is performed. The equation is given by:

\[
\ln(I_k + s) - \ln(M_k + s) = \ln \left( \frac{I_k + s}{M_k + s} \right)
\]  

(2.7)

In the dark part (where \( I \) and \( M << s \)) the signal is lost because \( \ln \left( \frac{I_k + s}{M_k + s} \right) = 0 \).

This explains why 'vessel-over-bone' contrast is very low in the dark areas, due to scatter radiation \( s \).

The problem of scatter correction is complicated because scatter radiation is not constant in the whole image and depends on the patient thickness, on the X-ray dose and on the tissue. A solution is to estimate the scatter radiation pixel by pixel from the total exposure. The simplest way to do this is by defining the scatter radiation equal to a certain percentage of the total exposure.

In the design shown in Figure 2.8 a simple method is used. This is done by subtracting a constant value from the scattered image. It is performed on each image of the image-sequence before the logarithmic transform. In this case, the scatter contribution is assumed to be uniform and not dependent on the pixel intensity.

The effect of the scatter contribution is in fact more important in the dark parts than in the bright parts. That is why the constant value is equal to a percentage of the minimum value of each image. The best percentage seems to be 50% (see Haupert [7]). The constant value is the same for each image to get the same background after subtraction. The subtraction of a constant value shifts the pixel intensity into the dark part where the 'vessel-over-bone' problem occurs.
Finally, the Gain LUT has to be adapted to the new process. It has been optimised to get an intensity as close as possible to the current process in order to compare both processes. To design this Gain LUT, some principles must be respected:

- Low gain for very dark values (big vessels) because a low contrast is sufficient to make a diagnosis.
- High gain for the values up to the background value (small vessels) because high contrast is necessary to see small details of the small vessels.
- A linear curve (gain = 1) for values around the background value. Unfortunately the very small vessels are also present, so a compromise has to be found.
- The gain for intensities higher than the background value is symmetric to the gain for values below the background value, like in the current Gain LUT. Then, the gain is zero for very bright values, which may correspond to motion artefacts.

The input image of the Gain LUT is a 1024*1024 image, containing 11 bits per pixel. Its background value is around 1024. The Gain curve's output intensity follows a quadratic function between 0 and 1020 in order to respect the statements above. Between 1020 and 1030, the curve is linear with gain one. For the values up to 1030, the Gain curve's output intensity is symmetric to the first part of the curve (see Figure 2.12).

![Gain LUT (11 bits -> 11 bits)](image)

**Figure 2.12:** New Gain LUT adapted to the new DSA images.

After applying the gain LUT, an edge enhancement is performed in the same way as in the current DSA process.

Finally, a windowing LUT is applied. It has to convert a 10-bit image into an 8-bit image. Moreover, it has to shift the background value to 80% (=204) of the maximum grey level (=255) for proper displaying. During the conversion, the bright values undergo a compression, whereas the dark values have an expansion. Both compression and expansion are linear.
2.6 Additional temporal filtering

As an extension to the DSA process, a temporal filtering operation can be performed before subtraction. This will have a noise reducing effect on the DSA images, which can be explained as follows:

The detection of X-ray quanta obeys the law of Poisson statistics. A fundamental observation in Poisson statistics is that the standard deviation is equal to the square root of the average number of detected X-ray quanta, which is a good estimation of the signal-to-noise ratio. So, if an average is taken of some live images and an average is taken of some mask images, more X-ray quanta are taken into account and therefore the signal-to-noise ratio will be higher. An example of a simple temporal filter is given in (2.8) and (2.9). Here, the mask image is obtained by taking the average of two images at the beginning of the sequence, when no contrast agent is visible yet and the live image (with contrast agent) is obtained by weighting the previous and following image with a quarter. The beginning of the sequence is indicated with \( t = 1 \) and \( I_t \) is an image of the sequence at time \( t \).

\[
\begin{align*}
\text{New mask} & = \frac{1}{2} I_t + \frac{1}{2} I_{t+1} & \text{for } t = 1 \\
\text{New image} & = \frac{1}{4} I_{t-1} + \frac{1}{2} I_t + \frac{1}{4} I_{t+1} & \text{for } 2 < t < t_{\text{max}}
\end{align*}
\]  

(2.8)  
(2.9)

This temporal filter gives good results (see Figures 2.13 and 2.14). The noise has been reduced after the subtraction.

It may be concluded that the new DSA process, discussed in this chapter, is a good solution for the 'vessel-over-bone' problem. However, the use of a logarithmic LUT introduces more noise in the dark regions as well. To deal with this noise Haupert [7] has designed a morphological noise reduction method that will be discussed further on in this report. First, in the next chapter, an analysis will be made of the noise in the DSA images. Next, the theory of morphological operators will be discussed to get more insight in the operators that have been used in this noise reduction method.
3 Noise analysis

3.1 Introduction

As already is mentioned in the previous chapter, the new DSA process boosts the noise in DSA images due to the use of a logarithmic LUT. Also the noise is increased by the subtraction itself. It is hard to design a proper noise reduction algorithm to decrease this noise. But before a noise reduction algorithm can be designed in the first place, something must be said about the character of the noise present in DSA images.

In this chapter briefly some different types of noise will be described, as well as the main noise sources in an X-ray system, and a method to analyse noise. Finally, the main filter structures in image processing are indicated.

3.2 Noise types

The purpose of this chapter is to make a proper description of the noise that is present in DSA images. If a good description of the noise can be obtained, it is easier to design a noise reduction method. However, in the case of the DSA process it is difficult to say something about the noise statistics, because there are a lot of processing steps in the DSA algorithm that affect the original noise statistics.

In literature different types of noise are described. The most common types of noise are:

Additive noise
This type of noise is signal-independent. An example is Gaussian white noise, which has a constant power spectrum. For greyscale images this means that the intensity of the noise is more or less the same for all different grey intensities in the image. The intensity of an image containing Gaussian white noise can be described in the following formula:

\[ J(x,y) = I(x,y) + n \quad \text{with} \quad n \sim N(0,\sigma). \]  

(3.1)

Here \( I(x,y) \) is the original pixel intensity at pixel \((x,y)\) and \( n \) is a normal distributed noise intensity, with zero mean and standard deviation \( \sigma \). \( J(x,y) \) is the intensity of the noisy image at pixel \((x,y)\). Note that a correction is needed to keep \( J \) in the same intensity range as \( I \).

Multiplicative noise
This type of noise is signal-dependent. The amplitude of the noise will increase if the signal level increases. In image processing this means that the standard deviation of the noise becomes larger when the image intensity becomes higher (see equation (3.2)). Here, also a correction is needed to keep \( J \) in the same intensity range as \( I \).

\[ J(x,y) = I(x,y) \cdot (1+n) \quad \text{with} \quad n \sim N(0,\sigma). \]  

(3.2)
Quantisation noise
This noise is caused when an analog signal is converted into a digital signal. It can also be seen as a signal-dependent type of noise. An example is given in equation (3.3).

\[ J(x, y) = \lfloor I(x, y) \rfloor \]  \hspace{1cm} (3.3)

Here, the continuous signal \( I(x, y) \) is converted to a digital signal \( J(x, y) \), by rounding \( I \) downwards to the nearest digital level.

Impulsive noise
When an image is corrupted with individual noise pixels whose brightness differs significantly from that of the neighbourhood, it is corrupted with impulsive noise. This is a signal-independent type of noise. For greyscale images this means that some pixels are corrupted with extremely differing intensities. An example is so-called salt-and-pepper noise, which describes saturated impulsive noise, appearing as white and black noise pixels randomly distributed over the image.

The list above is, of course, not complete. There exist a lot more types of noise, which can be combinations of the types mentioned in this section. The main issue of the list above is to be able to roughly describe noise which can be present in DSA images. But before a description will be given, the different sources that cause this noise will be described in the next section.

### 3.3 Noise sources

The main cause of the noise present in DSA images is the random nature of production of X-ray quanta. This noise is called *quantum noise*. Especially during long interventions when very low dose rates are used to protect the patient and medical staff, there is a very limited number of X-ray quanta per pixel and per frame. In this case the quantum noise is very high.

Quantum noise, as it originates from the X-ray beam, is Poisson-distributed, i.e. its variance is identical to the mean number of absorbed X-ray quanta per pixel and per frame. Therefore, the relation between the noise variance and the image intensity will be linear in raw X-ray images.

The Noise Power Spectrum (NPS) of quantum noise is flat since the emissions are assumed to be spatially independent. However, because the quantum noise in an observed X-ray image has undergone filtration by the imaging system's transfer function, the NPS results in a low-pass-shaped curve.

A second cause of noise in DSA images relates to the components in an X-ray system. The XTV-camera produces, for example, *quantisation noise*, when the analog video signal is converted to a digital signal. Also electronic noise can occur when high voltages are used.

The causes above are considered as most important. Now, it must be verified if the noise statistics are more or less the same as they are expected. In the next section a method will be described to measure the noise statistics in a greyscale image.
3.4 Noise analysis

To make a proper analysis of noise in a noisy image, it is necessary to separate the noise from the features in the image. Subsequently it is possible to display e.g. the standard deviation of the noise for different grey levels. From this kind of analysis it can become clear in which regions the image is most heavily polluted with noise. To perform a noise analysis, a noise estimation algorithm [3] is used and will be described in this section (see appendix A for a Matlab version of this algorithm).

The used noise analysis algorithm can be divided in the following steps and is visualised in a flow diagram in appendix B:

1. First, the mean and standard deviation for a pixel's neighbourhood in the image is calculated. This is done for every pixel by sliding a window of fixed size over the image. The size of this window can be determined at the start of the analysis. Depending on the number of data, the data is low-pass filtered and down-sampled.

2. Secondly, the calculated standard deviations that are above a certain bound are deleted. These standard deviations most likely belong to features (edges) in the image and must not be taken into account in the noise analysis. The threshold for separation of these features must be defined by the user and is therefore a parameter of the algorithm.

3. Then, the grey level scale of the image is divided into bins and for each bin the number of pixels with a mean grey level located in that bin, is counted.

4. Since there must be sufficient observations in each bin for statistical credibility, the number of observations in each bin is thresholded. Bins with less observations than this threshold are not taken into account in the next step. The threshold is calculated by taking the median of the obtained histogram divided by a so-called threshold factor.

5. Next, the mean noise standard deviation for each bin is calculated by averaging the standard deviations of all observations in each remaining bin (alternatively, also the median STD can be calculated).

6. Finally, the following plots are made:
   - Noise STD curve (x-axis: grey level bin; y-axis: mean standard deviation of noise in each bin)
   - Histogram of observations (x-axis: grey level bin; y-axis: number of observations in each bin)
   - Relation between mean grey level and noise STD (x-axis: mean grey level of noise pixel's neighbourhood; y-axis: standard deviation of noise pixel's neighbourhood)

In order to get a better understanding of this noise analysis algorithm, some examples are given using phantom images with predefined noise included.
3.4.1 Phantom image with additive Gaussian noise

As a first example a phantom image is taken where white Gaussian noise has been added with $\mu = 0$ and $\sigma = 10$ (see equation (3.1)). This phantom image has four vertical bands with different grey levels; the leftmost band has grey level 20 and the rightmost band has grey level 80.

![Image 400x400 with 4 intensity levels: 20, 40, 60, 80. Left the original, right containing white Gaussian noise.](image)

A noise analysis with the above described algorithm of this noisy phantom image results in the next plots. Here only pixels with a neighbourhood with $\sigma \leq 20$ are taken into account. A neighbourhood window size of 4x4 pixels is used, the bin size is 2 and the threshold factor is 1.

![Relation between mean grey levels and noise STD.](image)
3. Noise analysis

The results from this analysis are quite obvious: when the noise standard deviation is plotted for different pixel intensities, 'clouds' are visible around the grey level of the corresponding band (Figure 3.2). When the STD's in each bin with enough observations are averaged, the plot in Figure 3.3 is obtained. Here, the same STD, $\sigma = 10$, that was defined for the added white noise is visible, so the analysis seems to work. In Figure 3.4 one can see the number of observations in each bin. Here, the peaks correspond with the different grey levels in the phantom image.

If, for example, the bin size in the analysis above is changed to 0.5, the following results are obtained.

As one can see the results of this analysis differ not much from the previous ones: the mean noise STD plot is almost the same. However, the threshold has decreased 4 times and one must be careful that this value is not becoming too small, otherwise the analysis loses its statistical relevance.
3.4.2 Phantom image with multiplicative noise

In a second example a noise analysis is done on the same phantom image, but now containing speckle noise with $\sigma = 0.1$ (see Figure 3.7). Speckle noise is multiplicative and therefore dependent on the image intensity. In this analysis only pixels with $\sigma \leq 0.15\mu$ are taken into account ($\mu =$ mean pixel intensity). A window size of 4x4 pixels is used, the bin size is 2 and the threshold factor is 1. See Figures 3.8 - 3.10 for the analysis results.

Figure 3.7: Image (400x400) with 4 intensity levels: 20, 40, 60, 80. Left the original, right containing multiplicative noise with $\sigma = 0.1$.

Figure 3.8: Relation between mean grey levels and noise STD.

Figure 3.9: Mean noise STD per bin.

Figure 3.10: Histogram of selected observations. (Th. = 97)
In Figure 3.8 the effect of multiplicative noise is clearly visible: the STD in the centre of the 'clouds' becomes bigger when the intensity grows. Also the 'clouds' become bigger for higher intensities. Note that the standard deviations above the boundary, $\sigma = 0.15\mu$, are omitted, because they belong to edges in the image.

The plot of the mean noise STD in Figure 3.9 is also what can be expected: it is increasing with the intensity and normal distributed around the grey intensities of the original image.

In Figure 3.10 the shape of the observations for each grey level is changing from high small peaks to lower broader peaks. This can also be explained by the growth of the noise STD for higher intensities.

### 3.4.3 Phantom image with a predefined noise curve

In this example noise is added to the phantom image with a predefined relation between the noise STD and image intensity (see Figure 3.11).

![Figure 3.11: Left: noisy image (400x400) with 4 intensity levels: 20, 40, 60, 80. Right: predefined relation between noise STD and intensity.](image)

In this analysis only pixels with $\sigma \leq 18$ are taken into account. A window size of 4x4 pixels is used, the bin size is 2 and the threshold factor is 1. See Figures 3.12 - 3.14 for the analysis results.

![Figure 3.12: Relation between mean grey levels and noise STD.](image)
From this example it becomes again clear that the used noise analysis method gives proper results. The mean noise STD’s in Figure 3.13 correspond with the predefined curve in Figure 3.11.

### 3.4.4 X-ray noise in a static phantom image

So far, the noise analysis algorithm has been shown for phantom images with a-priori known noise. This algorithm seems to work quite well; the noise statistics resulting from the analysis correspond with the statistics of the a-priori known noise. Now, an analysis will be done on an X-ray image (Figure 3.15) where the noise statistics are not known exactly.

After a small experimental study of the relation between the mean intensities ($\mu$) and the STD’s ($\sigma$) of the phantom image in Figure 3.15, the following bounds were used to separate the noise from the features:

- $\sigma = 0.05*\mu + 4$
- $\mu = 22$

pixels above this bound are omitted
pixels beneath this bound are omitted

A window size of 4x4 pixels is used and the calculated data is down-sampled 2 times to obtain less data. The bin size is 2 and the threshold factor is 1.2. See Figures 3.16 - 3.19 for the analysis results.
3. Noise analysis

As can be seen, the curve (stars) in Figure 3.17 is the mean of the ‘cloud’ in Figure 3.16. This curve is increasing for growing intensities. For high intensities the curve is more flattened due to white compression, which is a compression of a certain range of high intensities to a comparable smaller range.

To verify the analysis result, the curve in Figure 3.17 is compared with the result of a temporal analysis method that has been used in a report of Kunz [9] and performed on the same X-ray image. In this temporal analysis method several static X-ray images are averaged such that an almost noiseless image is obtained. This image can be subtracted from one of the static X-ray images. Now, it is easy to analyse the resulting noise image. The curve obtained with this temporal analysis method is also plotted in Figure 3.17 (line).

As one can see, the curve of the analysis method described in this section (stars) is in good agreement with the curve obtained with the temporal (and more accurate) method. Globally, the standard deviation is a bit smaller for all intensities. This may be explained by the fact that our analysis is done on only one image. A practical disadvantage of the temporal analysis method is that a sequence of images is needed with no motion artefacts between the images.
3.4.5 X-ray noise in a Digital Subtraction Angiography image

Finally a noise analysis has been done on a DSA image (see Figure 3.20).

![DSA image (512x512) with grey scale [640...1071].](image)

After a small study of all the calculated STD's in the image and the relation between the mean intensities (μ) and the STD (σ), the following bound is used to separate the noise from the features:

- σ = 20

pixels above this bound are omitted

It must be said that there exist also STD's below this bound that belong to vessel walls of small vessels, but these STD's do not have a big influence on the results, because the intensity of these vessels is near the background intensity, which has a lot more observed pixels in the analysis.

The following settings are used for this analysis: a window size of 4x4 pixels, the calculated data is 2 times down-sampled, the bin size is 5 and the threshold factor is 4. See Figures 3.21 - 3.24 for the analysis results.

![Relation mean grey level and noise std](image)

![Mean noise STD](image)

**Figure 3.20**: DSA image (512x512) with grey scale [640...1071].

**Figure 3.21**: Relation between mean grey levels and noise STD.

**Figure 3.22**: Mean noise STD per bin.
3. Noise analysis

A small study of Figure 3.22 shows that the standard deviation of the noise is more or less constant \((\sigma = 16)\) till an intensity of 960. This result is in line with what can be expected, because the noise in the dark regions is boosted with the logarithmic LUT. Therefore the noise will have a larger standard deviation for small intensities. In the regions with intensities higher than 960, the mean noise STD is decreasing. This effect is also a result of the DSA process.

However the threshold value of 20 is quite low, the calculated mean STD's in the interval \([920...960]\) are reliable. In this interval only large vessels are visible which have edges with much higher STD's.

To get a better understanding of which features belong to which intensity range, a summary has been made in table 3.1.

<table>
<thead>
<tr>
<th>Intensity interval</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>700 - 750</td>
<td>image boundaries</td>
</tr>
<tr>
<td>750 - 800</td>
<td>dark spots on vessels</td>
</tr>
<tr>
<td>800 - 850</td>
<td>dark spots on vessels + boundaries of spots</td>
</tr>
<tr>
<td>850 - 900</td>
<td>darker parts of vessels</td>
</tr>
<tr>
<td>900 - 950</td>
<td>middle and large vessels</td>
</tr>
<tr>
<td>950 - 1000</td>
<td>a lot of vessels, walls of large vessels and noise in 'hard' areas</td>
</tr>
<tr>
<td>1000 - 1050</td>
<td>mainly background, small vessels and noise</td>
</tr>
</tbody>
</table>

From table 3.1 we may conclude that the interval where the noise is decreasing is in the region where most small vessels and the background are present. This observation seems nice, because noise with a smaller variance is mostly easier to remove. In this case, however, it isn't; the intensity of the small vessels is very close to the intensity of the background and therefore it is hard to remove the noise without loosing small vessels.

Now that the noise statistics are more or less known, the question remains how to suppress the noise in DSA images. Therefore, in the following section the main filtering concepts that can be used, will be discussed.
3.5 Image filtering techniques

In image processing there are basically two ways of filtering: temporal- and spatial filtering.

3.5.1 Temporal filtering

A temporal filter uses extra information (more than one image) to suppress noise. A good example of a temporal filter is one where some images from a sequence are averaged. In the DSA process e.g. the mask and live image can be obtained by averaging (see section 2.6).

A disadvantage of temporal filters is that they are sensitive to motion. Therefore, a robust motion reduction is required to use them.

3.5.2 Spatial filtering

Spatial filtering can be divided in two parts:
- Filtering when nothing is known about the noise statistics: the local pre-processing methods.
- Filtering when a model of the noise is known in advance; also called image restoration techniques.

From both spatial filtering techniques Haupert has tried some filters (Gaussian, Median, Wiener...) on DSA images (see p.42-43 [7]). Unfortunately none of the filters seemed to be satisfying. Therefore he designed a filter based on morphological operators, which performs quite well. This filter will be discussed in Chapter 5, but before it will be discussed, in the next chapter, the basics of morphological image processing will be explained.
4 Morphological image processing

The noise reduction method that Haupert [7] has designed for the new DSA process, as will be discussed in the next chapter, makes use of so-called morphological operators. Therefore, in this chapter the theory of the basic morphological operators will be explained and some extensions will be given that are possible with these operators in image processing. First, the theory is demonstrated on binary (black and white) images and next on greyscale images. At the end of the chapter a simple morphological filter, composed with basic morphological operators, will be shown. Also a section will be dedicated to some decomposition methods, that make use of morphological operators, and can be very useful in image processing.

4.1 Introduction

Mathematical morphology is the full name of all operations that are performed with morphological operators. Its basic operators are dilation (maximum) and erosion (minimum). Mathematical morphology was first introduced in a seminal book by George Matheron, entitled 'Random Sets and Integral Geometry' [10]. This book was the foundation of mathematical morphology and introduced it as a novel technique for image processing and analysis. The big breakthrough however was by the inspiring book 'Image Analysis and Mathematical Morphology' by Jean Serra [14]. Nowadays, mathematical morphology is considered as a powerful tool for image processing and analysis and it is used in diverse applications such as industrial inspection, target detection and biomedical imaging [4].

4.2 Binary erosion and dilation

Mathematical morphology can be seen as a tool for extracting geometric information from binary and greyscale images. A structuring element is used to build an image operator whose output depends on whether or not this element fits inside a given image. Clearly, the nature of the extracted information depends on the shape and size of the element used. To illustrate this concept, we initially restrict the explanation of the morphological operators to the case of binary images.

The most elementary set operators of interest to mathematical morphology are operators that are increasing. Increasing operators must satisfy the following condition:

\[ F_1 \subseteq F_2 \Rightarrow \Psi(F_1) \subseteq \Psi(F_2). \tag{4.1} \]

Here \( F_1 \) and \( F_2 \) are image objects and \( \Psi \) is an increasing operator. It says that if object \( F_1 \) is occluded by object \( F_2 \), the increasing operator may not make \( F_1 \) visible after processing.

Another important property in image processing is translation invariance. This means that an image object must be processed in the same way no matter where it is located in the image plane. Here, the following condition must be satisfied:

\[ \Psi(F + h) = \Psi(F) + h \quad \text{with translation } h \in E. \tag{4.2} \]

An operator not necessarily has both mentioned properties, but mostly we are interested in operators that are both increasing and translation invariant.

As said before the elementary operators of mathematical morphology are erosion and dilation. Now both operators are discussed when performed in the binary domain.
A set operator for binary images $F_1$ and $F_2$ that satisfies equation (4.3) is called a binary erosion:

$$\Psi_e(F_1 \cap F_2) = \Psi_e(F_1) \cap \Psi_e(F_2).$$  \hspace{1cm} (4.3)

Similarly, a set operator for binary images $F_1$ and $F_2$ that satisfies equation (4.4) is called a binary dilation:

$$\Psi_d(F_1 \cup F_2) = \Psi_d(F_1) \cup \Psi_d(F_2).$$  \hspace{1cm} (4.4)

Note that both operators are increasing (e.g. if $F_1 \subseteq F_2$, then equation (4.3) implies $\Psi_e(F_1) = \Psi_e(F_1 \cap F_2) = \Psi_e(F_1) \cap \Psi_e(F_2) \subseteq \Psi_e(F_2)$).

Let us now define a translation invariant erosion with structuring element $B$:

$$\Psi_e(F) = \bigcap_{b \in B} F - b = F \Theta B.$$ \hspace{1cm} (4.5)

In set theory the equation above is also known as the Minkowski subtraction:

$$F \Theta B = \{h \in E \mid (B + h) \subseteq F\}.$$ \hspace{1cm} (4.6)

This equation suggests that the translation invariant erosion of a set $F$ by a structuring element $B$ comprises all points $h$ of $E$ such that the structuring element $B$ located at $h$ fits entirely inside $F$. A visualisation can be seen in Figure 4.1 when $B$ is a disc.

![Figure 4.1: Binary erosion (left) and binary dilation (right).](image)

Similarly, every translation invariant dilation is of the form:

$$\Psi_d(F) = \bigcup_{b \in B} F + b = F \oplus B.$$ \hspace{1cm} (4.7)

In set theory the equation above is also known as the Minkowski addition:

$$F \oplus B = \{h \in E \mid (\bar{B} + h) \cap F \neq \emptyset\}.$$ \hspace{1cm} (4.8)
4. Morphological image processing

Here \( \tilde{B} = \{ -b | b \in B \} \) is the reflection of \( B \) in the origin. This equation suggests that the dilation of a set \( F \) by structuring element \( B \) comprises all points \( h \) such that the reflected structuring element \( \tilde{B} \) translated with \( h \) intersects with \( F \). See the visualisation in Figure 4.1.

4.3 Binary gradient

It is also possible to construct a morphological gradient with the elementary morphological operators (dilation and erosion). See the definition in the next equation:

\[
\Psi_{\text{grad}}(F) = F \ominus B \setminus F \ominus B. \tag{4.9}
\]

This morphological gradient may be seen as an operator that detects the boundary of an object \( F \). Here the operator '\( \setminus \)' represents the set difference. By using variants of equation (4.9) it is possible to detect external or internal boundaries. See the equations below respectively:

\[
\Psi_{\text{grad}}^+(F) = F \ominus B \setminus F \quad \text{and} \quad \Psi_{\text{grad}}^-(F) = F \setminus F \ominus B. \tag{4.10}
\]

4.4 Binary opening and closing

The erosion and dilation operators do not have a mostly desired property, known as idempotence. Idempotent operators perform a whole operation at one time, while consecutive repetitions of the operator will not change the obtained result anymore. The erosion and dilation operator on the other hand keep modifying the image at each successive operation.

When a translation invariant erosion \( F \ominus B \) is composed with a translation invariant dilation \( F \oplus B \), the resulting operator is idempotent, regardless of the order of composition. The composition \( (F \ominus B) \oplus B \) is called an opening, while the composition \( (F \oplus B) \ominus B \) is called a closing. Note that an opening is an anti-extensive operator (i.e. \( \Psi(F) \subseteq F \)) and a closing is an extensive operator (i.e. \( F \subseteq \Psi(F) \)).

If a geometrical interpretation is given of a structural opening with structuring element \( B \), the following equation emerges:

\[
F \circ B = \bigcup \{ B + h | h \in E \ \text{and} \ (B + h) \subseteq F \}. \tag{4.11}
\]

It says that \( F \circ B \) is the union of all translated structuring elements \( B + h \) that fit inside \( F \). The effect of an opening with a disk structuring element can be seen in Figure 4.2. It is clear that an opening with structuring element \( B \) removes all components of \( F \) that are 'smaller' than \( B \). Therefore it acts as a sort of smoothing filter depending on the shape of the structuring element used.
Also a geometrical interpretation can be given of a structural closing with structuring element $B$ (see equation (4.12)):

$$F \circ B = \{ u \in E \mid u \in \tilde{B} + h \Rightarrow (\tilde{B} + h) \cap F \neq \emptyset \}.$$  \hfill (4.12)

This equation says that $F \circ B$ is the collection of all pixels $u$ such that all translated structuring elements $\tilde{B} + h$ which contain $u$ intersect $F$. The effect of a closing (with disk structuring element) can be seen in Figure 4.2. Instead of an opening this operator removes all components of $P$ (complement of $F$) that are ‘smaller’ than $\tilde{B}$. This is a direct consequence of the duality between structural openings and closings (see equation (4.13)):

$$F \bullet B = (F' \cap \tilde{B})'.$$  \hfill (4.13)

### 4.5 Greyscale erosion and dilation

In the binary case it is opted to use operators that distribute over unions and intersections. In the greyscale case, however, it is desirable to distribute over suprema and infima. A greyscale erosion distributes over infima as can be seen in equation (4.14).

$$\psi_\epsilon(f_1 \wedge f_2) = \psi_\epsilon(f_1) \wedge \psi_\epsilon(f_2) \quad \text{with} \quad \wedge \equiv \inf.$$  \hfill (4.14)

A greyscale dilation distributes over suprema:

$$\psi_d(f_1 \vee f_2) = \psi_d(f_1) \vee \psi_d(f_2) \quad \text{with} \quad \vee \equiv \sup.$$  \hfill (4.15)

Because this report only deals with greyscale images containing a finite number of grey levels, maximum and minimum are used instead of supremum and infimum, respectively.

For a translation invariant greyscale erosion, the following relation is valid:

$$\psi_\epsilon(f)(x,y) = \min_{\xi,\eta}[\psi_\epsilon(f + \xi, y + \eta) - b(\xi, \eta)].$$  \hfill (4.16)

Here $b$ is a greyscale image known as the structuring function. When $b$ is a flat structuring function (see equation (4.17)),

---

© 2002 Philips Medical Systems Nederland B.V.

Philips Medical Systems
4. Morphological image processing

\[ b(x,y) = \begin{cases} 
0 & (x,y) \in B \\
-\infty & \text{otherwise}
\end{cases} \quad (4.17) \]

the expression of the corresponding flat greyscale erosion becomes:

\[ \psi_e(f)(x,y) = \min_{(\xi,\eta) \in B} [f(x+\xi, y+\eta)]. \quad (4.18) \]

Flat greyscale erosion replaces the value of an image \( f \) at a pixel \((x, y)\) by the minimum value of \( f \) that lies within the area of structuring element \( B \). See Figure 4.3 for the visualisation of flat greyscale erosion of an one-dimensional function \( f(x) \).

\[ \text{Figure 4.3: Flat greyscale erosion, for structuring element } B. \]

When the greyscale dilation is translation invariant, the following relation is valid:

\[ \psi_d(f)(x,y) = \max_{\xi,\eta} [f(x-\xi, y-\eta) + b(\xi,\eta)]. \quad (4.19) \]

This relation becomes with flat structuring function \( b \):

\[ \psi_d(f)(x,y) = \max_{(\xi,\eta) \in B} [f(x-\xi, y-\eta)]. \quad (4.20) \]

Flat greyscale dilation replaces the value of an image \( f \) at a pixel \((x, y)\) by the maximum value of \( f \) that lies within the area of structuring element \( B \). See Figure 4.4 for the visualisation of flat greyscale dilation of a one-dimensional function \( f(x) \).

In the figure below the effect of a flat greyscale erosion and a flat greyscale dilation with a structuring block element is shown on a phantom image.

\[ \text{Figure 4.5: Left: phantom image, middle: eroded phantom image, right: dilated phantom image.} \]
4.6 Greyscale gradient

The morphological gradient for greyscale images can be defined in the following way:

$$\Psi_{\text{grad}}(f) = \frac{1}{2} [f \oplus B - f \ominus B].$$  \hspace{1cm} (4.21)

It is in fact half of the difference of a dilation and an erosion. So edges of objects in the image are detected. For the greyscale case also an external and internal gradient exist (see equation (4.22)):

$$\Psi_{\text{grad}}^+(f) = \frac{1}{2} [f \oplus B - f] \quad \text{and} \quad \Psi_{\text{grad}}^-(f) = \frac{1}{2} [f - f \ominus B].$$  \hspace{1cm} (4.22)

For the phantom image of Figure 4.5 the morphological gradient, as well as the external and internal gradient, are calculated and displayed in Figure 4.6.

![Figure 4.6: Left: morphological gradient of Figure 4.5, middle: external gradient, right: internal gradient.](image)

4.7 Greyscale opening and closing

Like in the binary case, the greyscale opening- and closing operator are idempotent. A greyscale opening is also anti-extensive, while a greyscale closing is extensive (see Figure 4.7).

A greyscale opening of an image $f$ with a structuring function $b$ is given by $f \ominus b$. This operator is called a structural opening and can be seen as a smoothing filter that approximates an image from below, since $f \ominus b \leq f$. The amount and type of smoothing is of course determined by the selected structuring function $b$. When $b$ is a flat structuring function, the structural opening is called a flat structural opening and is denoted by $f \ominus B$.

A geometrical interpretation of a flat greyscale opening is shown in Figure 4.7 and its expression is given in equation (4.23):

$$\Psi_{\text{open}}(f)(x,y) = \max \{t \in R \mid (x,y) \in F(t) \ominus B\}.$$  \hspace{1cm} (4.23)

It shows that the maximum cross-section $t$ must be found where pixel $(x, y)$ still belongs to the binary opening $F(t) \ominus B$.

When an opening is subtracted from the original image $f$, only the peaks of the image remain. Therefore this operator is called tophat $(f - f \ominus B)$. 

© 2002 Philips Medical Systems Nederland B.V

Philips Medical Systems
4. Morphological image processing

Similarly, a greyscale closing of an image \( f \) with a structuring function \( b \) is given by \( f \bullet b \). This operator is referred to as a structural closing and operates as a smoothing filter that approximates a shape from above, since \( f \bullet b \geq f \). When \( b \) is a flat structuring function, the structural closing is called a flat structural closing and is denoted by \( f \bullet B \).

In equation (4.24) the geometrical interpretation of a flat greyscale closing is given:

\[
\Psi_{\text{clo}}(f)(x,y) = \max \{ t \in \mathbb{R} | (x,y) \in F(t) \bullet B \}. \tag{4.24}
\]

It shows that the maximum cross-section \( t \) must be found where pixel \( (x, y) \) still belongs to the binary closing \( F(t) \bullet B \). An illustration of a flat greyscale closing for the one-dimensional case is shown in Figure 4.8. When the original image \( f \) is subtracted from its closing, only the 'valleys' of the image remain. Therefore this operator is called bothat \((f \bullet B - f)\).

In Figure 4.9 the effect of a flat greyscale opening and a flat greyscale closing with a block structuring element can be seen on a phantom image. The effect of the closing is not very well visible here, because there aren't small dark features present in this phantom image, except for some noise.
4.8 Reconstruction

Reconstruction is a powerful tool for object extraction from binary and greyscale images. It is an iterative tool that extracts regions of interest from an image marked by a set of markers. First the reconstruction of binary images is discussed, then this will be extended to the greyscale domain.

4.8.1 Binary reconstruction

Let $F$ be a binary image containing several objects with some objects marked with markers $F^m$ (see Figure 4.10). To extract the objects $F^m$ containing the markers, a *conditional dilation* must be performed which is defined in equation (4.25):

$$\delta^1_B( F^m | F ) = ( F^m \oplus B ) \cap F .$$

(4.25)

So the expansion of the marker $F^m$ is restricted inside $F$. If this conditional dilation is performed $k$ times, the definition becomes:

$$\delta^k_B( F^m | F ) = \delta^1_B( \delta^1_B( \ldots \delta^1_B( F^m | F ) ) \ldots ) .$$

(4.26)

*Binary morphological image reconstruction* can be achieved by taking the union of all conditional dilations (see equation (4.27)):

$$\hat{F} = R_B( F^m | F ) = \bigcup_{k \geq 1} \delta^k_B( F^m | F ) .$$

(4.27)

This reconstruction operator finally stops when the union of all the conditional dilations doesn't change anymore. A graphical interpretation of a binary reconstruction is shown in Figure 4.10. Here, the closest circles around the marker $F^m$ indicate the *first* conditional dilation.

![Figure 4.10: Binary reconstruction, for structuring element B.](image)

An often-used way to obtain the markers in an image is to perform an opening. After that a reconstruction of the image can be made. This operator is called an *opening-by-reconstruction*:

$$\Psi_{\text{openrec}}(F) = R_B( F \circ A | F ) .$$

(4.28)

Here $A$ is a structuring element which removes all objects that not accommodate with $A$. 
4.8.2 Greyscale reconstruction

Morphological image reconstruction can be extended to the greyscale case via threshold decomposition. Let \( f \) be a greyscale image with several peaks. If we want to remove a predefined selection of those peaks and keep the remaining peaks intact, greyscale reconstruction is a very useful operator. The following equation describes the greyscale reconstruction on basis of threshold decomposition:

\[
\hat{f}(x, y) = r_B(f^m | f)(x, y) = \max \{ t \in R \mid (x, y) \in R_B(F^m(t) | F(t)) \}. \tag{4.29}
\]

This definition says that the maximum cross-section \( t \) must be found such that a pixel \((x, y)\) is within the binary reconstruction \( R_B \). Where \( F^m(t) \) is the marker at cross-section \( t \) and \( F(t) \) is the cross-section of the image \( f \). See Figure 4.11 for the graphical interpretation of a reconstruction of a one-dimensional function \( f(x) \).

![Diagram of greyscale reconstruction](image)

**Figure 4.11:** Greyscale reconstruction, for structuring element \( B \). \( \hat{f}(x) \) is the reconstruction of \( f(x) \) using marker \( f^m(x) \).

Here, \( F(t) \) is the cross-section at \( t \) and \( F^m(t) \) is the marker at that cross-section. \( f^m(x) \) represents the marker image and \( \hat{f}(x) \) represents the reconstructed image.

For the greyscale case an opening-by-reconstruction can be defined as well:

\[
\Psi_{\text{open-re}}(f) = r_B(f \circ A \mid f). \tag{4.30}
\]

In Figure 4.12 the effect of a greyscale opening-by-reconstruction can be seen. In this case, the reconstructed image is almost the same as the original phantom image. To see the effect of the reconstruction, this result must be compared with the opening in Figure 4.9

![Diagram of opening-by-reconstruction](image)

**Figure 4.12:** Left: phantom image, right: opening-by-reconstruction.
As one can see there are a lot of extensions possible with the elementary morphological operators. The most important operators have been discussed in this chapter and will be used in the morphological noise reduction algorithms, which will be discussed further on in this report. Note that the morphological operators used in these algorithms will always perform on greyscale images, since we operate mainly on X-ray images.
4.9 Some simple morphological filters

A well known filter in mathematical morphology is a closing followed by an opening or the other way around. This filter has a smoothing effect on an image, as can be seen in the following example. A black square on a white background is smoothed with a closing followed by an opening with a disk structuring element with a diameter of 5 pixels. See Figures 4.13 and 4.14 for the results.

![Original image](image1.png) ![Filtered image with closing, opening (disk element)](image2.png)

Except for smoothing, the closing-opening filter can also be used as a noise reduction filter. To illustrate this, a simple example is shown in the next figures. A black square with a grey background is corrupted with 'salt & pepper' noise (Figure 4.15). When a closing followed by an opening is performed with a 3x3 block structuring element, the image in Figure 4.16 results.

![Original image with 'salt & pepper' noise.](image3.png) ![Filtered image with closing, opening (block element).](image4.png)

As can be seen, a ‘closing, opening’ filter has a strong noise reducing effect for this amount of ‘salt & pepper’ noise. Another very important feature of a noise reduction algorithm is the preservation of objects in an image. Note that in the last example the shape of the black square is preserved after filtering. This is due to the use of a block structuring element on a block-shaped feature. So, if the right structuring element is used, objects will be preserved while noise is removed. Therefore, morphological filters are very interesting for processing of noise corrupted images. In the next chapter more enhanced morphological filters will be discussed.
4.10 Morphological decomposition

Finally, another tool that is possible with morphological operators will be discussed: morphological decomposition.

In image processing it can be useful to perform the same operation or filter on different scales. A widely accepted approach to split a signal in different levels, and recover it, is multi-resolution signal decomposition. This decomposition method has analysis/synthesis operators that map information between different levels. The analysis operators are designed to reduce information, whereas the synthesis operators are designed to undo as much as possible this loss of information.

In an article of Goutsias and Heijmans [6] a general multi-resolution signal decomposition scheme, the pyramid transform, is presented. This scheme must satisfy an important condition: the pyramid condition, also called perfect reconstruction property. This condition says that the consecutive operation of an analysis- and synthesis operator must be idempotent and therefore no information will be lost.

From section 4.4 it is known that an opening- and closing operator are idempotent. Therefore, it is possible to build a so-called morphological pyramid with these operators.

In literature several morphological pyramids can be found [5], [6]. Some of them will be shown in this section. A first example of a morphological pyramid can be seen in Figure 4.17.

Bothat/tophat pyramids

![Diagram](image)

Figure 4.17: Morphological bothat pyramid.

This is a bothat pyramid; each subband contains a bothat of the input image $I_k$ at the corresponding level $k$. In this example the dilation followed by a down-sampling is the analysis operator and the up-sampling followed by the erosion is the synthesis operator. This bothat pyramid can be used for filtering purposes, where the filtering is performed on the subband images $I_{sub}$.
A tophat pyramid has the same set-up as a bothat pyramid, but now the dilation and erosion operators are switched. Here, the analysis operator is composed of an erosion followed by a down-sampling and the synthesis operator is composed of an up-sampling followed by a dilation (see Figure 4.18).

Instead of performing consecutively erosion and dilation or dilation and erosion, respectively, it is also possible to combine these operations in the analysis operator. An example of such a morphological pyramid is given in Figure 4.19. Here, an alternative tophat pyramid is shown.

A more advanced morphological decomposition method can be found in an article of Metzler [12]. On each level of this pyramid two different decompositions are performed. Another interesting concept here is that the synthesis operators are the reconstructive dual of the corresponding analysis operators. An example of such a pyramid, without reconstructive synthesis, is shown in Figure 4.20. In this pyramid each level contains a tophat subband as well as a bothat subband.
4. Morphological image processing

A linear pyramid

Of course, a pyramid can also be built with linear components. A well-known linear pyramid is the Laplacian pyramid (see Figure 4.21). Here, the analysis operator consists of a low-pass filter followed by down-sampling and the synthesis operator consists of an up-sampling followed by a low-pass filter. The subband images will contain more or less separate frequency subbands, whereas the image with the lowest resolution is a low frequency image. This last image represents approximately the background of the original.

Figure 4.20: Advanced morphological (tophat, bothat) pyramid.

Figure 4.21: Laplacian pyramid.

In this chapter a very broad view has been given on the possibilities of morphological tools in image processing. It was shown that with a simple morphological filter noise can be removed, without affecting image features. Also morphological operators appeared very useful in performing a decomposition to split an image in subbands of different scales.

In the next two chapters some of the tools, described in this chapter, will be used in more advanced morphological filters.
5 Noise reduction using mathematical morphology

5.1 Introduction

From Chapter 3 it is clear that the task of removing noise, present in DSA images, is a difficult one. Also the fact that the noise is mixed up with the structure makes it hard to remove. Of course, it is not desired to remove any information present in the image; only noise must be removed.

By using the described morphological operators in the previous chapter in a smart way, it is possible to remove quite a lot of noise while keeping most of the structure in the image. Such a noise-reducing algorithm has been designed by Haupert [7] and will be described in this chapter. Also some other noise reducing algorithms will be described, which use morphological operators as well and perform even better.

5.2 Haupert's algorithm

5.2.1 Principle

The principle of the noise reduction algorithm of Haupert is the use of the bothat operator (closing(image) − image) in order to remove dark noise pixels in the image. Note that the bothat operator detects the valleys in an image, which correspond with dark noise pixels. In this case, also small dark vessels will be detected as valleys and therefore be removed. To prevent this situation a directional closing is used to separate noisy pixels from small vessels as much as possible.

5.2.2 Directional closing and opening

This directional closing is a combination of a set of dilations followed by a set of erosions. In each set, different dilations and erosions are performed by using different structuring elements. In the first set of dilations, 4 dilations are done in parallel with 4 different line structuring elements (see Figure 5.1). These four dilations are averaged and form the input for the subsequent set of erosions. Let us define this set of dilations as 'averaged directional dilation'.

![Directional closing diagram](image)

**Figure 5.1:** Directional closing, consisting of an averaged directional dilation with line elements and an averaged directional erosion with block elements.

© 2002 Philips Medical Systems Nederland B.V

Philips Medical Systems
The structuring line elements have a length of 3 pixels (see Figure 5.2). This length is chosen because the noise pixels are 1 or 2 pixels square and the size of the small vessels is more or less the same. Line elements have been chosen because vessels can be modelled as line segments, while the noise pixels are more block shaped.

$$\begin{bmatrix} 000 \\ 111 \\ 000 \end{bmatrix} \quad \begin{bmatrix} 010 \\ 010 \\ 010 \end{bmatrix} \quad \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix} \quad \begin{bmatrix} 001 \\ 010 \\ 100 \end{bmatrix}$$

Figure 5.2: Line structuring elements $A$, as used in the directional closing of Figure 5.1.

During the dilation the maximum is taken among the 3 pixels of each line element. If the central pixel is a noisy pixel (with low intensity), it is expected to be isolated. In this case the dilation of each line element will result in a higher grey level than the noisy pixel and therefore the mean of the 4 dilations will also result in a new pixel value with a higher grey level. However, if the central pixel belongs to a vessel, also other pixels will be found with the same intensity in at least one direction. The final pixel value of this dilation will therefore be based on the original intensity in at least one direction. In this way small vessels are more or less preserved when line structuring elements are used.

In the next step a set of erosions is performed, also in four different directions (see Figure 5.3). This set of structuring elements is composed by block elements in order to prevent the extensive effect of the directional closing with the line elements. Because the four structuring elements are symmetric, no translation or shape artefacts are introduced.

$$\begin{bmatrix} 110 \\ 110 \\ 000 \end{bmatrix} \quad \begin{bmatrix} 011 \\ 011 \\ 000 \end{bmatrix} \quad \begin{bmatrix} 000 \\ 110 \\ 110 \end{bmatrix} \quad \begin{bmatrix} 000 \\ 011 \\ 011 \end{bmatrix}$$

Figure 5.3: Block structuring elements $B$, as used in the directional closing of Figure 5.1.

During this erosion the minimum is taken of four pixels for each structural element and the four results are averaged. This set of erosions is called an 'averaged directional erosion'. If there are still some noise pixels left from the dilation step, these will not influence the erosion much. If for example an isolated noise pixel (low intensity) is in the left bottom corner of the structuring area it will only influence the final value of the central pixel for a quarter.

Now that the directional closing has been described, a bothat operation can be done by subtracting the original image from its directional closing. This results in an image with a lot of noise visible, but also some vessels. So, not only dark noise pixels are removed by the directional closing, also some vessel structure is lost. To remove the remaining vessels in the bothat image, a directional opening is performed on this image with the same concept as the directional closing. First an averaged directional erosion is performed with structuring line elements and after that an averaged directional dilation is performed with structuring block elements (see Figure 5.4).
After performing this directional opening a tophat transform is done; the resulting image of this directional opening is subtracted from its input image (= bothat image). Now, an image containing only noise in the inverse greyscale is left. To get an image with only the vessels, the bothat image is subtracted from the tophat image.

Finally, an image with less noise is obtained by adding the tophat image (inverse noise) to the original DSA image. Also the vessel image may be added to enhance the vessels. By means of parameters $c_1$ and $c_2$ the algorithm can be tuned. See the flow diagram of this noise reduction algorithm in Figure 5.5.

A simplified visualisation of this algorithm can be seen in Figure 5.6 - 5.10 on the next page.

Although the algorithm designed by Haupert seems to work quite well, it could possibly be improved to remove even more noise in DSA images. Therefore, other methods must be tried and compared with this algorithm.

In addition to the noise reduction algorithm described in this section, some variants of this algorithm have been tried. These will be explained in the following sections.
Figure 5.6: Original DSA image.

Figure 5.7: Bosphat image.

Figure 5.8: Vessel image.

Figure 5.9: Tophat image.

Figure 5.10: Filtered DSA image.

Figure 5.6-5.10: Visualisation of the different steps in Haupert's algorithm.
5.3 Filter with reconstruction

The first self-designed variant on the noise reduction algorithm of Haupert is one with reconstruction. It is expected that with reconstruction the shape and size of the original features, such as vessels, are better preserved. See the flow diagram of this algorithm in Figure 5.11.

![Flow diagram of the noise reduction algorithm using reconstruction.](image)

The first part of this algorithm is similar to the directional closing of the algorithm of Haupert and is meant to remove dark noise pixels. In the second part the resulting image after the directional closing is eroded with line elements to remove bright noisy pixels. This result is now used as the marker image for reconstruction of the resulting image after the directional closing.

The performance of this algorithm is more or less the same as the algorithm of Haupert, as can be seen in the filter results shown in Figures 5.14 and 5.15. Note that these results are not directly the output of the noise reduction algorithm, but a subsequent SubGain LUT, edge enhancement and Windowing LUT (see Chapter 2) have been performed to obtain the final filtered DSA image. Figure 5.12 shows the original DSA image, which was the input image for both filters. Its corresponding noise analysis can be seen in Figure 5.13.

![Original DSA image.](image)

![Mean noise STD per bin, original image.](image)

© 2002 Philips Medical Systems Nederland B.V.

Philips Medical Systems
Although the differences between the results above are not very large, the result in Figure 5.15 looks smoother. Also, this has the effect that the vessel walls look more homogeneous.

Another comparison can be made through the noise analysis of both images (see the figures below).

If these figures are compared with the noise analysis of the original image in Figure 5.13, it is clear that both algorithms have reduced noise. The algorithm using reconstruction performs even slightly better than the algorithm of Haupert. Note that the intensity interval of the resulting images differs from the original due to the post-subtraction LUTs.

Although the noise analysis is a good measure for noise, it is only one part of judging a noise reduction algorithm. Other features that are important in a proper judgement are e.g. contrast and detail preservation. Therefore, it is wrong to compare only some noise analysis plots.

If all judgement features are taken into account, it may be concluded that the algorithm using reconstruction performs slightly better than the algorithm of Haupert. In the remainder of this chapter it will be tested if more improvement is possible by filtering on different scales. First, in the next
section, it will become clear why a directional morphological filter has its structuring elements operating in parallel and not in series.

5.4 Filter with elements in series

Another idea that has been tried, is to perform the different structuring elements of a set in series instead of in parallel, such as in the previous discussed algorithms. The first idea can be seen in Figure 5.18. Here, the different structuring line elements are performed completely in series. After that, the result is eroded with the block elements of Figure 5.3.

The second idea is shown in Figure 5.19. The structuring line elements are again performed in series, but the dilated results of each line element are added and averaged. Next, this result is eroded with the block elements of Figure 5.3.

To show the effects of both ‘directional’ closings, a simple black and white picture (see Figure 5.20) is taken as input. In Figures 5.21 and 5.22 the results can be seen. These results are not satisfying at all: the size of the block has become smaller, the line detail is completely lost and there is a lot of blurring present. So, it may be concluded that in a proper noise reduction algorithm based on morphological operators, a set of structuring elements must not be used in series.

Figure 5.20: Original image.  
Figure 5.21: Result idea 1.  
Figure 5.22: Result idea 2.
5.5 MIC algorithm

Instead of testing some self-created ideas with morphological operators, also noise reduction algorithms based on morphological operators have been found in literature [2]. One of the most promising algorithms appeared to be the Morphological Image Cleaning algorithm (MIC), which will be explained in this section.

The idea of the Morphological Image Cleaning algorithm (MIC) originates from an article of Peters [13]. In this article a noise reduction algorithm has been described which splits an image into subbands and uses morphological operators to filter these subbands. MIC was primarily designed to enhance scanned images and still-video images. Therefore, the MIC algorithm has been modified for filtering DSA images.

5.5.1 Principle

The goal of MIC is the same as every other image cleaning algorithm: make the image look better. Although this is a subjective statement, there are two characteristics that are commonly regarded as good qualifiers:

1. Edges, thin lines, and small features are sharp and clear.
2. Areas between these features are smoothly varying.

Because linear filters always affect edges, morphological operators like closing and opening are used. First, a rough sketch of the MIC algorithm will be given:

Let $I$ be a greyscale image and $S$ its smoothing with openings and closings. Assume that $S$ is 'noise free'. Then the difference image, $D = I - S$, contains most of the noise in $I$. Because $S$ does not contain all the image features, due to the closings and openings, $D$ contains image features as well. If the noise in $I$ has a smaller dynamic range than the thin features, then $D$ will contain noise at lower amplitude levels and features at higher amplitudes. Now, if $D$ is thresholded at a value greater than the amplitude of the noise, the result is a mask of the thin features in the image. This mask image can be used to recombine the thin features in $D$ with $S$ while leaving the noise behind. The result is an image that is smoothly varying except for edges, thin lines and small spots.

Figure 5.23: Flow diagram of MIC algorithm.
The MIC algorithm described in the next section is based on the idea above, but is extended to different scales.

5.5.2 Original algorithm

A flow diagram of the MIC algorithm can be seen in Figure 5.23. It will be explained in this subsection. In the original MIC algorithm the smoothed images $S$ are obtained with the so-called OCCO filter ('filter 1') from Figure 5.24. Here, a disk-shaped structuring element is used.

$S_{j-1}$ → closing → opening → $S_j$

$S_{j-1}$ → opening → closing → $S_j$

**Figure 5.24: OCCO filter, 'filter 1' in original MIC algorithm.**

Now, the residual images are calculated in the following way:

$$D_j = S_{j-1} - S_j.$$  \hspace{1cm} (5.1)

Here, $S_j$ is the result of filtering $S_{j-1}$ with 'filter 1' and structuring elements of size $d_j$. Let $S_0 = I$ be the input image. Assume that $d_1 < d_2 < \ldots < d_j$ and $d_0 = 1$. Note that filtering $I$ with $d_0 = 1$ is $I$ itself.

Because on each level a structuring element of different size is used, the residual image $D_j$ has no features larger than $d_j$ and none smaller than $d_j$. These residual images are signed and therefore they are split up in a positive and negative part before filtering (see Figure 5.23). In this way, only positive signed residuals are filtered by 'filter 2'.

The filtering of the residual images in the original MIC algorithm is done in a rather complex way, therefore it will not be explained in full detail. First, a smart threshold, based on the histogram of the residual, is performed on the residual image to separate image details and noise. Next, some extra steps are done to remove isolated pixels and recover the details in a correct way.

Finally, when all levels have been passed, the output image is obtained by adding all filtered positive residual images (bright features) and subtracting all filtered negative residual images (dark features) from the most coarsely smoothed image $S_k$ with $k$ indicating the last level.

Because DSA images are too heavily polluted with noise, attempts to reduce noise with the original MIC algorithm on DSA images were not successful. Therefore a modified MIC algorithm has been designed, which will be explained in the next section.

5.5.3 Modified algorithm

The modifications that are made to the original MIC algorithm are restricted to 'filter 1' and 'filter 2' (see Figure 5.23). So, the main set-up remains the same, while the smoothing at the different scales is changed, as well as the filtering of the residual images.

**Modified filter 1**

The idea behind the modified MIC algorithm is to remove as much noise as possible with 'filter 1' in order to 'catch' all the noise in the residual images. In the residual images this noise can be separated from the image features.

Because from section 5.3 a good noise reduction filter is available already, this 'filter with reconstruction' will be used as 'filter 1' in the modified MIC algorithm. Only one modification is made: the size of the structuring elements is made variable (see Figure 5.25). Note that at each level in the MIC algorithm a larger structuring element is used in 'filter 1' to separate the different scales in
the image. Here, an interesting link can be made to an article of Wilkinson [16], where also reconstruction is used in the decomposition to different scales.

Modified filter 2
As a next step, both residual images are filtered with a simple morphological filter: a directional opening like in the filter of Haupert (see Figure 5.26). Such a simple filter is sufficient, because the residual images contain only one-signed noise (positive). The main goal is to extract these noise pixels (bright spots) from the residual images.

![Directional filter from Figure 5.11, with variable structuring elements (filter 1).](image1)

![Directional opening (filter 2).](image2)

A result of the modified MIC algorithm can be seen on the next page in Figure 5.30. For completeness the input DSA image is given in Figure 5.27, with its corresponding noise analysis in Figure 5.28.

![Original DSA image.](image3)

![Mean noise STD per bin, original image.](image4)
The filter result of the modified MIC algorithm is obtained using 3 levels with the following structuring elements: \( d_1 = 3 \), \( d_2 = 5 \) and \( d_3 = 9 \). If this result is compared with the filtering result of the reconstruction algorithm (section 5.3) in Figure 5.29, it can be noticed that the vessel walls look smoother. Both results contain more or less the same details.

From the filter results above, also the noise analysis results are compared.

As can be seen from the figures above, the noise level is less in the result of the modified MIC algorithm (Figure 5.32). So, again, a little progression is made at this point. Because the contrast and detail preservation is more or less the same in both results, it may be concluded that the modified MIC algorithm performs best up till now.
5.5.4 Some considerations

In this section some considerations are given about possible variations of the MIC algorithm. Instead of using the OCCO filter for smoothing the input image, also the mean of a dilation and an erosion can be taken (see Figure 5.33).

![Diagram showing alternative smoothing for 'filter 1'.](image)

**Figure 5.33:** Alternative smoothing for 'filter 1'.

In this case the subband image will be equal to the morphological Laplacian of the input image. The definition of the morphological Laplacian is given in equation (5.2).

\[
\Psi_{\text{lap}}(I) = \frac{1}{2} [I \ominus B + I \ominus B - 2 \cdot I] \tag{5.2}
\]

Because the morphological Laplacian produces signed images (positive and negative), the subbands must again be divided into positive- and negative residuals in order to obtain only positive signed residual images. Note that this alternative smoothing is quite similar to the decomposition method used in the advanced ratio pyramid described by Metzler [12]. A main difference is, however, that the MIC algorithm is a multi-scale-SR pyramid, where all processing steps are performed on the same resolution level.

Note that multi-scale filtering can be done on one resolution level (single-resolution) or on several resolution levels (multi-resolution). In the single-resolution case, filter kernels of different size are used, while in the multi-resolution case, the same filter kernel is performed on different resolution levels. The effect of both multi-scale filters will be the same. The latter is, of course, computationally more attractive. Note that in this report the term ‘multi-scale-SR’ is used for a multi-scale filter working on one resolution level. When the extension SR is omitted, all multi-scale filters are meant; single- as well as multi-resolution ones.

In order to get a little more insight into the global structure of the MIC algorithm, the flow diagram in Figure 5.23 will be transformed into a structure which is commonly used for multi-resolution pyramids (see the figures in the last section of Chapter 4). The transformed diagram can be seen in Figure 5.34. From this figure it becomes clear that the MIC algorithm fits into the structure of a multi-resolution pyramid. The MIC algorithm can be turned into a multi-resolution pyramid if down-sampling is performed after each application of ‘filter 1’.
5. Noise reduction using mathematical morphology

Figure 5.34: Transformed flow diagram of MIC algorithm. Note the absence of a down-sampling step, as is commonly used in a multi-resolution pyramid.

5.6 Conclusions

From the morphological algorithms discussed in this chapter, the modified MIC algorithm seems to have the largest potential. Here, a good noise reduction is combined with contrast- and detail preservation.

However, the main disadvantage of the algorithms discussed in this chapter is that every pixel is filtered in exactly the same way. This is not preferable, because line structuring elements are used, which have a specific orientation. Therefore, it seems interesting to use structure information of the image to steer the filter. In the next chapter the design of a morphological filter that makes use of such a structure analysis will be explained.
6 Structure-steered directional morphological filtering

The directional morphological filters, discussed in the previous chapter, miss one important feature: the use of orientation information from the image. If such information can be obtained, the directional filters can be steered with this information and it is expected that more details in the image will be preserved.

In this chapter a way to obtain orientation information will be explained. Structure analysis will be performed by calculation of a so-called structure tensor. Next, this analysis will be combined with a directional morphological filter.

6.1 Structure analysis

The idea of combining a directional filter with structure analysis is based on the COSMIC filter [1]. This filter has several Gaussian smoothing kernels, which are steered by means of a structure analysis. Because this filter is very successful on ultrasound images, it is expected that quite an improvement can also be achieved when such a structure analysis is combined with a directional morphological filter.

The structure of an image can be analysed in many different ways. In this report, a method has been chosen which is based on the calculation of local gradients in an image. With this method a so-called structure tensor can be calculated for each pixel.

The structure tensor is a matrix that gives the orientation of a pixel in a given neighbourhood. The eigenvectors of this matrix give the main orientations in this neighbourhood and the corresponding eigenvalues are a measure for the pixel contrast. First, some theory of the structure tensor will be given and next it will be explained in which way this tensor can be used for filtering purposes.

6.1.1 Theory of the structure tensor

To detect an edge one can calculate local gradients of an image. First, the image is smoothed with a Gaussian filter with standard deviation $\sigma$ to eliminate high frequency noise and after that the gradient $\hat{g}(i,j)$ is calculated (see equation (6.1)). Here $u$ is the intensity of a pixel of the smoothed image located at $(i,j)$.

$$\begin{align*}
\hat{g}(i,j) & = \begin{pmatrix} g_x \\ g_y \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix}.
\end{align*}$$

Unfortunately, the gradient gives only a local direction in an image and not a local orientation. To overcome this problem the covariance matrix, also called structure tensor, of the gradient is calculated for each pixel (see equation (6.2)) [15].

Here, Cov is the covariance and $E$ is the statistical expectance of a stochastic variable.

$$\begin{align*}
C_g(i,j) = \text{Cov}[\hat{g}] = E[\hat{g} \cdot \hat{g}^T] &= E \begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix},
\end{align*}$$

© 2002 Philips Medical Systems Nederland B.V.
Philips Medical Systems
6. Structure-steered directional morphological filtering

\[
\begin{align*}
&= E \left[ \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial y} \end{bmatrix} \right] = E \left[ \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial x} \end{bmatrix} \right] E \left[ \begin{bmatrix} \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} \end{bmatrix} \right] \\
&= E \left[ \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} \right] E \left[ \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} \right].
\end{align*}
\]

(6.2b)

This structure tensor can also be obtained in a more fundamental and computationally attractive way by estimating the second derivative of the two-dimensional autocorrelation function \( \varphi_{uu} \) in a region around pixel \((i,j)\) (see [11]). Let \( u(x) \) be a grey value function, then its autocorrelation function \( \varphi_{uu} \) in the neighbourhood of pixel \( \bar{x} \) can be approximated by:

\[
\varphi_{uu}(\bar{x}) = \varphi_{uu}(\bar{0}) - \frac{1}{2} \bar{x}^T \cdot H \cdot \bar{x} + ... \tag{6.3}
\]

\( H \) is here the (negative) Hessian of \( \varphi_{uu}(\bar{x}) \) at \( \bar{x} = \bar{0} \):

\[
H = - \begin{bmatrix}
\frac{\partial^2 \varphi_{uu}}{\partial x^2} & \frac{\partial^2 \varphi_{uu}}{\partial x \partial y} \\
\frac{\partial^2 \varphi_{uu}}{\partial x \partial y} & \frac{\partial^2 \varphi_{uu}}{\partial y^2}
\end{bmatrix} \bigg|_{\bar{x}=\bar{0}}. \tag{6.4}
\]

Now, it can be shown via the differentiation and moment theorems of the Fourier transform that the elements of \( H \) are identical to (see [11]):

\[
h_1 = E \left[ \left( \frac{\partial u(\bar{x})}{\partial x} \right)^2 \right], \tag{6.5}
\]

\[
h_2 = h_{21} = E \left[ \left( \frac{\partial u(\bar{x})}{\partial x} \right) \left( \frac{\partial u(\bar{x})}{\partial y} \right) \right], \tag{6.6}
\]

\[
h_{22} = E \left[ \left( \frac{\partial u(\bar{x})}{\partial y} \right)^2 \right]. \tag{6.7}
\]

When the components above are compared with the matrix components in equation (6.2) the correspondence is clear. Therefore, we may conclude that the matrix \( C_\omega \) which contains the desired orientation information can be directly determined from the autocorrelation function \( \varphi_{uu}(\bar{x}) \).

In this report, however, the autocorrelation function will not be used to calculate the structure tensor. Instead, the first method based on gradients in (6.2), will be used. The mean of the components in (6.2) is calculated by taking the average over a local neighbourhood with a Gaussian filter with standard deviation \( \rho \). This neighbourhood must not be too large, otherwise details will be lost (in this report a 3x3 neighbourhood will be used).

To illustrate the calculation of a structure tensor, an example calculation is performed on the phantom image of Figure 6.1, which contains structures with different orientations.
6. Structure-steered directional morphological filtering

First the phantom image is smoothed with a Gaussian filter \((\sigma = 0.7, 3\times3\) neighbourhood). Then the gradient of the smoothed phantom image is calculated (6.1), see Figure 6.2 and 6.3.

These results show what we expect: dark lines (= negative values) indicate a left-to-right or up-down transition in the phantom from white to black, light lines (= positive values) indicate such transitions from black to white. In the grey regions the gradient is more or less zero.

Next, the structure tensor \(C_s\) is calculated and averaged with a Gaussian filter \((\rho = 2\) on a \(3\times3\) neighbourhood). Its matrix elements are displayed in Figure 6.4, 6.5 and 6.6.
The results are quite obvious: the lines in Figure 6.4 correspond with the horizontal part of the gradient and the lines in Figure 6.5 correspond with the vertical part of the gradient. Information about the diagonal oriented structures is given by the element shown in Figure 6.6. In Figure 6.7 the trace of the structure tensor is shown; this seems to be a good measure for edges in an image. Note that the trace is also equal to the sum of both eigenvalues of the structure tensor. The determinant is shown in Figure 6.8 and is a good measure for corners.

### 6.1.2 Eigenvalues and vectors of the structure tensor

Now that the structure tensor has been defined for each pixel in the image, its eigenvectors and corresponding eigenvalues can be calculated. As already mentioned, the eigenvalues are a measure for the pixel contrast and the corresponding eigenvectors indicate the orientation of a pixel.

Eigenvector $v_1$ has its orientation perpendicular to a structure and $v_2$ has its orientation along this structure. Generally, the following equation must hold for each eigenvector and corresponding eigenvalue of $C_g$.

$$C_g \bar{v} = \mu \bar{v},$$  (6.8)
6. Structure-steered directional morphological filtering

where \( \vec{v} \) is one of the eigenvectors and \( \mu \) is its corresponding eigenvalue.

To solve equation (6.8), it can be rewritten as:

\[
(C_g - \mu I) \cdot \vec{v} = 0,
\]

with \( I \) = identity matrix.  \( \text{(6.9)} \)

For ease of notation the structure tensor is written as:

\[
C_g = \begin{pmatrix}
t_{11} & t_{12} \\ t_{12} & t_{22}
\end{pmatrix}.
\]  \( \text{(6.10)} \)

When \( \vec{v} \) is defined as \( \vec{v} = (v_x, v_y)^T \), the following two equations result from (6.9):

\[
(t_{11} - \mu)v_x + t_{12}v_y = 0, \quad \text{(6.11)}
\]
\[
t_{12}v_x + (t_{22} - \mu)v_y = 0. \quad \text{(6.12)}
\]

Rewriting (6.11) and (6.12) results in:

\[
v_y = \frac{\mu - t_{11}}{t_{12}}v_x, \quad \text{(6.13)}
\]
\[
v_y = \frac{-t_{12}}{\mu - t_{22}}v_x. \quad \text{(6.14)}
\]

Now combine (6.13) and (6.14):

\[
t_{12}v_x = (\mu - t_{11})(\mu - t_{22})v_x. \quad \text{(6.15)}
\]

From equation (6.15) the eigenvalues can then be calculated:

\[
\mu_1 = \frac{1}{2} \left( t_{11} + t_{22} + \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2} \right), \quad \text{(6.16)}
\]
\[
\mu_2 = \frac{1}{2} \left( t_{11} + t_{22} - \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2} \right). \quad \text{(6.17)}
\]

Note that, since \( C_g \) is symmetric positive semidefinite, the following relation is always valid:

\[
\mu_1 \geq \mu_2 \geq 0. \quad \text{(6.18)}
\]

The corresponding eigenvectors can be written in the following 'standard' expressions:

\[
\vec{v}_1 = \begin{pmatrix}
2t_{12} \\
t_{22} - t_{11} + \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2}
\end{pmatrix}, \quad \text{(6.19)}
\]
\[
\vec{v}_2 = \begin{pmatrix}
2t_{12} \\
t_{22} - t_{11} - \sqrt{(t_{11} - t_{22})^2 + 4t_{12}^2}
\end{pmatrix}. \quad \text{(6.20)}
\]
In the following figures the eigenvalues of the phantom image in Figure 6.1 are plotted. As one can see they indicate the regions where a lot of contrast in the image is present. Eigenvalues $\mu_1$ are especially high on edges, whereas eigenvalues $\mu_2$ are high at corners.

**Figure 6.9:** Eigenvalues $\mu_1$ of phantom image in Figure 6.1.

**Figure 6.10:** Eigenvalues $\mu_2$ of phantom image in Figure 6.1.

In literature (e.g. Weickert [15]) the eigenvalues are often used to split the image in three different regions:

- isotropic regions (not much contrast): $\mu_1 \approx \mu_2 \approx 0$
- edge regions: $\mu_1 \gg \mu_2 = 0$
- corner regions: $\mu_1 \geq \mu_2 \gg 0$

So with the known eigenvalues of the phantom image, it is not difficult to detect these three different regions. Below in Figures 6.12 - 6.14, one can see the detected regions with the corresponding boundaries of the phantom image in Figure 6.11. These boundaries are based on the so-called coherence in the image. In Weickert [15], the coherence is defined as $(\mu_1 - \mu_2)^2$. So, the coherence is high at edges- and corners regions and small in isotropic regions.

The thresholds for the chosen boundaries have been experimentally chosen. Of course, it is not convenient if for each image these thresholds must be determined in this experimental way. Therefore, in the filter design a so-called $\alpha$-map and $\beta$-map will be used. Both mappings will be explained in the next section.

**Figure 6.11:** Original phantom image, corrupted with noise.

**Figure 6.12:** Isotropic regions, $(\mu_1 - \mu_2) \leq 14000$ & $\mu_2 \leq 1100$. 
6. Structure-steered directional morphological filtering

6.2 The α- and β-map

The α- and β-map are both mappings that are based on the same principle: map the range of a certain input parameter to a scale from 0 to 1. A mapping can be done in many different ways. In our case we want to make a mapping to separate the isotropic regions, containing not much contrast, from the anisotropic regions, that contain a lot of contrast. This mapping is called an α-map.

Another useful mapping is one that isolates corner regions from the rest of the image. This will be done to protect the corners from filtering. In principle, corners must not be filtered, because this can affect the look of an image tremendously. This mapping is called a β-map.

As said before, both mappings are calculated in a similar way. To make a good separation, such a mapping must perform a thresholding on the input parameter. The following mapping function is used for this purpose:

\[
map(x) = \frac{1}{1 + e^{-c(x-T)}}.
\]

(6.21)

Here, \(x\) is the input parameter for which the mapping will be done. With the parameters \(C\) and \(T\), the mapping function can be tuned to obtain a desirable mapping. The mapping function is also visualised in Figure 6.15.

Figure 6.13: Edge regions, \((\mu_1 - \mu_2) > 14000 \& \mu_2 <= 1100\).

Figure 6.14: Corner regions, \(\mu_2 > 1100\).

Figure 6.15: Mapping function.
The slope of this curve at the threshold $x = T$ can be adjusted by $C$. A small calculation results in relation (6.22) between $C$ and the slope at $x = T$:

$$map(T)' = \frac{C}{4}.$$  

(6.22)

So if $C$ is given a large value, the slope of the mapping function will be very steep at $x = T$ and the mapping function will have the shape of a step function. For moderate values of $T$ the mapping will have an S-shape.

In contrast to the COSMIC filter [1] not the trace, $\mu_1 + \mu_2$, is used as input parameter for the $\alpha$-map, but the square root of the coherence, $\mu_1 - \mu_2$, has been chosen (see (6.23)). From the previous section it was already clear that this parameter is a good measure for the contrast in an image. In fact, the main difference is that the trace takes more corner information into account. However, this is not necessary for composing the $\alpha$-map. For the $\beta$-map, $\mu_2$ is chosen as input parameter (see (6.24)). For both mappings finally a smoothing is done with a Gaussian filter.

With both mappings it is possible to make a differentiation between different regions in an image. By choosing the right parameters $C$ and $T$ this differentiation can have the effect of a hard thresholding or a very smooth transition. See for example the $\alpha$- and $\beta$-map of Figure 6.11, displayed in Figure 6.16 and 6.17. In section 6.4 it will become clear how the $\alpha$- and $\beta$-map are used in a structure-steered morphological filter.

$$\alpha = map(\mu_1 - \mu_2, C_\alpha, T_\alpha)$$  

(6.23)

$$\beta = map(\mu_2, C_\beta, T_\beta)$$  

(6.24)

Subsequently to (6.23) and (6.24) an extra smoothing is performed with a Gaussian filter.

Figure 6.16: $\alpha$-map of Figure 6.11, $C_\alpha = 30$ and $T_\alpha = 0.5$.

Figure 6.17: $\beta$-map of Figure 6.11, $C_\beta = 60$ and $T_\beta = 0.2$.
6. Structure-steered directional morphological filtering

6.3 Calculation of weight factors for directional morphological filters

In the directional filters, as described in the previous chapter, each structuring element is used with the same weight (see the directional closing in Figure 6.18). So, in each direction an equal filtering is performed. In an area where a lot of contrast (e.g., an edge) is present, this is undesirable, because it will cause much blurring. In such regions it is preferred to filter mainly in one direction.

![Diagram of directional dilation and erosion](image)

**Figure 6.18:** Directional closing with equal weight factors.

If e.g., structuring line elements are used, the question remains what the preferred direction is in which these elements must be placed. In fact, there are only two options: parallel or perpendicular to structure (edges) in an image. After some experiments it appeared that the structuring line elements in a morphological filter can be placed best parallel to a structure. See the experiment shown in Figures 6.19-6.21. Here a detail of a DSA image with a vertically oriented vessel is filtered with the directional closing from Figure 6.18, but now no equal weight factors are used. Figure 6.20 is filtered with only horizontal line elements: $w_x = 1$, $w_y = w_{dl} = w_{dr} = 0$. Figure 6.21 is filtered with only vertical line elements: $w_y = 1$, $w_x = w_{dl} = w_{dr} = 0$. It is clear that the vertically oriented vessel in Figure 6.21 looks much better; especially the vessel walls are smoother.

![Figure 6.19: Detail DSA image containing noise.](image)

![Figure 6.20: Image 6.19 filtered with a horizontal line element.](image)

![Figure 6.21: Image 6.20 filtered with a vertical line element.](image)

Now we know how to place the structuring line elements, a method must be designed to calculate the weight factors for each of the different structuring line elements. In our case there are line elements in 4 different orientations: horizontal, vertical and two diagonals. A more advanced method would take even more orientations into account. In such a case the size of the line elements can not be too small. Note that with line elements of 3 pixels, only 4 orientations can be taken into account.
To be able to place a line element along a structure, orientation information from the structure analysis will be used. More specifically, the orientation of the second eigenvector $\tilde{v}_2$ is taken, because this vector has its orientation along the structure.

Next, the angle between this eigenvector and each line element is calculated. Let us define this angle as:

$$\phi_{dir} = \angle(\tilde{v}_2, \tilde{s}_{dir}),$$

(6.25)

where $\tilde{s}_{dir}$ is the line element in the direction $dir$ with $dir =$ hor, vert, diag1 or diag2.

Dependently on the angle in (6.25), in each direction a weight factor must be defined. The direction with the smallest angle must get the largest weight factor. Experiments showed that it is not preferred to give all weight to just one direction, while making the weight factors in the other 3 directions zero. The angle in (6.25) can be used as an input parameter for a weight-function that makes a weight factor one when this angle is zero and which goes to zero when this angle increases. The following relation is defined between the weight factor $w_{dir}$ and angle $\phi_{dir}$ that satisfies the demands above:

$$w_{dir} = w(\phi_{dir}) = e^{-K \sin^2(\phi_{dir})}.$$  

(6.26)

Thus, if the angle between the eigenvector $\tilde{v}_2$ and a line element is zero or a multiple of $\pi$, the corresponding weight factor $w_{dir}$ will be one. The relation in (6.26) is visualized in Figure 6.22.

![Figure 6.22: Relation weight factor $w_{dir}$ and angle $\phi_{dir}$. Dashed line: $K=3$, Solid line: $K=20$.](image)

The relation in equation (6.26) can be tuned via parameter $K$. A small peak around $\phi_{dir} = 0$ is obtained when $K$ is large.

When all weight factors in each orientation are calculated, they are normalized so their summation is one in each pixel. This is necessary to keep the filtered image in the same greyscale. The definition of a normalized weight factor is given in the following equation:

$$\tilde{w}_{dir} = \frac{w_{dir}}{\sum_{dir} w_{dir}}.$$  

(6.27)
6.4 Combination of analysis with directional morphological filters

6.4.1 Global structure

Now that all the elements necessary to steer a directional morphological filter have been designed, they must be combined in a proper way. A global structure of the so-called structure-steered morphological filter can be seen in the flow diagram in Figure 6.23.

Although from Figure 6.23 it is not yet clear how a structure-steered morphological filter works exactly, it gives a global overview of how the structure analysis part is connected to the directional filter part. First, a structure analysis is performed on a noisy image. From this analysis the eigenvectors are used to calculate the weight factors. The eigenvalues are used to calculate the $\alpha$- and $\beta$-map. The combination of all the elements that are necessary to steer the filter takes place in the block ‘combine’, which will be discussed in the remainder of this section. Also the block ‘directional morphological filter’ will be explained in more detail.

6.4.2 Combination of the $\alpha$- and $\beta$-map

The $\alpha$- and $\beta$-map are combined in order to create a new $\alpha$-map that doesn’t contain corner regions. See the relation in the next equation:

$$\alpha_{\text{new}}(i, j) = \alpha(i, j) \cdot (1 - \beta(i, j)).$$

(6.28)

Here, $\alpha(i, j)$ indicates the $\alpha$-map and $\beta(i, j)$ the $\beta$-map at pixel $(i, j)$. When a corner is detected the $\beta$-map will be one (or near one) and therefore the new $\alpha$-map will be (almost) zero. So, with the use of relation (6.28), corner regions will not be filtered in the same way as other regions with high contrast, like edges.

6.4.3 Combination of weight factors and the new $\alpha$-map

The weight factors, as calculated in section 6.3, are not yet ready to be used in the directional morphological filter. The main reason is that these weight factors make no differentiation between
isotropic and anisotropic regions. To obtain this differentiation, the weight factors and the new $\alpha$-map (6.28) are combined, as can be seen in the following equation:

$$\tilde{w}_{\text{dir}}(i,j) = \alpha_{\text{new}}(i,j) \cdot \tilde{w}_{\text{dir}}(i,j) + \frac{1}{4} \cdot (1 - \alpha_{\text{new}}(i,j)).$$  

(6.29)

In this equation $\tilde{w}_{\text{dir}}(i,j)$ is a weight factor for direction $\text{dir}$ as calculated in (6.26) and (6.27) for pixel $(i,j)$. In our case 4 directions are used: horizontal, vertical and 2 diagonals. The new weight factor $\tilde{w}_{\text{dir}}(i,j)$ will be the same as the old one if the new $\alpha$-map is one and therefore at such pixels an anisotropic filtering will be performed. If the new $\alpha$-map is zero at pixel $(i,j)$, $\tilde{w}_{\text{dir}}(i,j)$ will be equal to $\frac{1}{4}$ and an isotropic filtering will be performed. Note that due to the relation in (6.29) the corner regions are filtered in the same way as isotropic regions. It is not possible, however, to prevent this filtering by means of changing the weight factors. Therefore, a correction is needed after filtering with the weight factors. This will be explained in the next section. Note that if more directions are used the constant $\frac{1}{4}$ must be changed to $1/(\text{no. of directions})$. In this way, the summation of all weight factors in each direction in a pixel $(i,j)$ will be equal to one.

As an example, new weight factors are calculated for the image in Figure 6.24 and the results can be seen in Figure 6.25 - 6.28.

**Figure 6.24:** Phantom image, containing noise.

**Figure 6.25:** Horizontal weight factors, $\tilde{w}_{\text{hor}}$.

**Figure 6.26:** Vertical weight factors, $\tilde{w}_{\text{ver}}$.

**Figure 6.27:** Diagonal1 weight factors, $\tilde{w}_{d1}$.

**Figure 6.28:** Diagonal2 weight factors, $\tilde{w}_{d2}$.

In the white regions in the figures above, the weight factors are approximately one, whereas the black regions indicate that the weight factors are zero. The grey background corresponds with weight factors of $\frac{1}{4}$ (isotropic filtering). Note that the corner regions in Figure 6.24 are filtered in the same way as isotropic regions.
For completeness a flow diagram of the block 'combine' in Figure 6.23 is given in Figure 6.29.

![Flow diagram of the 'combine' block in Figure 6.23.](image)

6.4.4 Steered directional morphological filter

When all elements of the structure analysis are combined, the directional morphological filter can be steered in a proper way. As an example, in Figure 6.30 a flow diagram is shown of a steered directional (closing, opening) filter with structuring line elements.

![Structure-steered directional (closing, opening) filter with corner correction.](image)

Here the weight factor \( \hat{w}_{\text{hor}} \) corresponds with the horizontal element, \( \hat{w}_{\text{ver}} \) with the vertical element, \( \hat{w}_{\text{d1}} \) with the diagonal from left-up to right-down and \( \hat{w}_{\text{d2}} \) with the diagonal from left-down to right-up. As one can see, the \( \beta \)-map is used to correct the isotropic filtering of the corners. The filtered corner regions are (partly) replaced with the original pixels of the input image.

At this point the structure information is completely integrated with a directional morphological filter. To show the performance of such a filter, the directional (closing, opening) filter of Figure 6.30 is performed on a noisy phantom image (see Figure 6.31 and 6.32).
As can be seen from the result in Figure 6.32 quite a lot of noise has been reduced, while corners are preserved. Also the edges in the filtered image are not blurred due to the structure analysis.

### 6.4.5 Considerations

As can be noticed, the actual concept of the structure-steered filter in Figure 6.30 is quite different from the original filter concept of Haupert. Both filter concepts differ mainly in two ways:

1. In the structure-steered directional morphological filters, the structuring block elements are not used. The main reason is that image features are distorted when block elements are used in combination with line elements and different weight factors. To prevent distortions, it is necessary to use the same structuring elements with the same weight factors, when a combination like (closing, opening) or (dilation, erosion) is made. See the example in Figure 6.30. Here, the closing and opening are steered with the same weight factors.

2. The filter of Haupert makes use of a bothat and tophat operator. This concept can also be used in combination with a structure analysis. However, simulations did not result in better images than the filter concept of Figure 6.30 (see Chapter 7).

The example shown in Figure 6.30 is, however, not the only possible option with the filter structure presented in this chapter. There are several possibilities and some of them are listed below:

- **Opening, closing.** The closing and opening in Figure 6.30 can be swapped, i.e. first an opening, then a closing is performed. The difference with the filter of Figure 6.30 is not very large. To stick to the idea of Haupert [7], i.e. first removing dark noise pixels, it seems most logical to begin with a closing.

- **Dilation, erosion, erosion, dilation.** Another possibility is to split the closing and opening in a dilation, erosion and an erosion, dilation respectively. After each morphological operator the β-map must be used to protect the corner regions from filtering.

- **Combinations.** Of course, also combinations of the options given above can be made.

- **Reconstruction.** It is also possible to combine one of the options above with reconstruction. An example is a structure-steered version of the reconstruction filter from Chapter 5.

The list above is, of course, not complete; a lot of other variations can be made. There is only one thing that should be respected in every filter: after each performance of a morphological operator, the β-map must be used to protect the corner regions from filtering.
The structure-steered filter concept can also be used in a multi-scale setting. In the next chapter a simulation will be performed with a structure-steered version of the MIC algorithm, as discussed in the previous chapter. Although it is a rather small step to integrate a structure-steered filter in a multi-scale setting, no other algorithms that make use of such a setting will be discussed in this report (unfortunately this was not possible, due to a lack of time). Here some possibilities will be given that are interesting for future research.

- With a structure-steered morphological filter the subbands of a morphological pyramid (e.g. Figure 4.17) can be filtered. The structure analysis can be done on the original image or on the subband image itself. Probably, the filter must be changed to filter the subbands in the best way.
- Instead of using structuring elements of different length in the MIC algorithm, the residual images can also be obtained in a multi-resolution setting. See Figure 5.34, but now with down- and up-sampling. In theory the MIC algorithm will have the same effect in this setting.
- Another interesting morphological decomposition method that has been found in literature and seems quite promising, is one that performs a decomposition in two steps that are combined to get a subband [12].

6.5 Conclusions

From this chapter it becomes clear that it is possible to steer a directional morphological filter by means of structure analysis of the original image. There are quite a lot of parameters in this analysis that can be changed and influence the filter result. On a phantom image a structure-steered directional morphological filter performs quite well. In the next chapter simulations are also done on DSA images and the results will be compared with the ones of the directional morphological filters of the previous chapter. Although even better results seem to be possible in a multi-resolution environment, this could not be discussed in this report due to a lack of time.
7 Simulation results

In this chapter some simulation results will be given for the morphological filters designed in the previous chapters. Also the results will be compared with each other. In Table 7.1 the different simulations are summarized. The number of each filter is related with the corresponding section number in this chapter. For each filter also the figure number of its flow diagram is given and the parameter settings that are used for the simulation.

The input image on which all filters are tested, is a DSA image of a knee (exam002) and is shown in Figure 7.2. Due to the new DSA process, which deals with the ‘vessel over bone’ problem, it contains a lot of noise. When we look at the original X-ray image in Figure 7.1, the ‘vessel over bone’ problem is quite obvious.

Unfortunately, it is not possible to give a full proper judgement on paper of the results presented in this chapter. In the real clinical practice, image quality specialists observe the DSA images on a special medical greyscale monitor, which gives the images a quite different impression. Therefore, the simulation results in this report are shown with another purpose: mainly to compare the results with each other.

Table 7.1: Summary of simulations.

<table>
<thead>
<tr>
<th>no.</th>
<th>Filter</th>
<th>Flow diagram</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Haupert’s algorithm</td>
<td>Figure 5.5</td>
<td>$c_1 = 1$, $c_2 = 0$</td>
</tr>
<tr>
<td>2</td>
<td>Reconstruction filter, based on Haupert’s algorithm</td>
<td>Figure 5.11</td>
<td></td>
</tr>
</tbody>
</table>
| 3   | Modified MIC algorithm | Figure 5.23, 5.25 and 5.26 | Filter 1: $d_1 = 3$, $d_2 = 5$, $d_3 = 9$
Filter 2: size struct. elements: 3 |
| 4   | Structure-steered (closing, opening) filter | Figure 6.30 | Structure tensor: $\alpha = 0.7$ and $\rho = 0.2$  
$\alpha$-map: $C_{\alpha} = 300$ and $T_{\alpha} = 0.01$  
$\beta$-map: $C_{\beta} = 60$ and $T_{\beta} = 0.2$  
weight factors: $K = 20$  
size line elements: 3 |
| 5   | Structure-steered (closing, opening) filter | Figure 6.30 | Same settings as simulation 4, except $K = 5$. |
| 6   | Structure-steered (dilation, erosion, erosion, dilation) filter | Based on Figure 6.30 | Same settings as simulation 4. |
| 7   | Structure-steered (dilation, erosion, erosion, dilation) filter | Based on Figure 6.30 | Same settings as simulation 5. |
| 8   | Structure-steered (dilation, erosion + reconstruction) filter | Figure 7.10 | Same settings as simulation 5. |
| 9   | Structure-steered variant of Haupert’s algorithm | Figure 7.12 | $c_1 = 1$, $c_2 = 0$
Same settings as simulation 5. |
| 10  | Structure-steered variant of modified MIC algorithm | Figure 5.23, 5.25 and 7.14 | Filter 1: $d_1 = 3$, $d_2 = 5$, $d_3 = 9$
Filter 2: size line elements: 3 |
7. Simulation results

Figure 7.1: Original X-ray image of a knee, exam002.

Figure 7.2: Noisy DSA image of a knee, exam002. (right: detail image)
7. Simulation results

7.1 Haupert's algorithm

The first simulation result given in this chapter, is a result obtained with Haupert's algorithm [7]. This already quite satisfactory result can be seen in Figure 7.3 and is, actually, the main reference for further obtained results.

![Figure 7.3: Figure 7.2 filtered with Haupert's algorithm ($c1 = 1, c2 = 0$). (after additional post-subtraction LUTs)](image)

7.2 Reconstruction filter, based on Haupert's algorithm

Also the result with the reconstruction algorithm from Figure 5.11 is shown again in Figure 7.4. From Chapter 5 it was already clear that the reconstruction filter performs slightly better than Haupert's algorithm. The vessel walls are smoother and the noise level is a bit smaller.

![Figure 7.4: Figure 7.2 filtered with reconstruction filter, based on Haupert's algorithm. (after additional post-subtraction LUTs)](image)
7.3 Modified MIC algorithm

The simulation result of the modified MIC algorithm (see section 5.5) is shown in Figure 7.5. From that section it was already clear that the result of the modified MIC algorithm was the best so far. In this result, the vessel walls have the smoothest impression.

![Figure 7.5: Figure 7.2 filtered with modified MIC algorithm, d_1 = 3, d_2 = 5, d_3 = 9. (after additional post-subtraction LUTs)](image)

The following simulations are done with some different combinations of the structure-steered filter from Figure 6.30. It will be investigated if this will give even better results, with less noise and a higher contrast.

7.4 Structure-steered (closing, opening) filter, K = 20

![Figure 7.6: Figure 7.2 filtered with a structure-steered filter: closing followed by opening (K = 20). (after additional post-subtraction LUTs)](image)
First, a simulation is done with the structure-steered filter from Figure 6.30. The parameter settings that are used for the structure analysis are listed in table 7.1. This result is not satisfactory: the vessel walls are not smooth and the contrast is unnaturally high. In order to improve the smoothness of the vessel walls, the parameter $K$ in relation (6.26) is changed to 5.

### 7.5 Structure-steered (closing, opening) filter, $K = 5$

![Figure 7.7: Figure 7.2 filtered with a structure-steered filter: closing followed by opening ($K = 5$). (after additional post-subtraction LUTs)](image)

If the result in Figure 7.7 is compared with the result in Figure 7.6, it is hard to see any differences. In Figure 7.7 the transitions at the vessel walls are a little bit smoother, but the overall result is still not satisfactory.

### 7.6 Structure-steered (dilation, erosion, erosion, dilation) filter, $K = 20$

![Figure 7.8: Figure 7.2 filtered with a structure steered filter: dilation, erosion followed by erosion, dilation. ($K = 10$) (after additional post-subtraction LUTs)](image)
Because the previous two simulations were not successful, another set-up is chosen for the structure-steered filter from Figure 6.30. The closing is split up in a directional dilation and a directional erosion and the opening is split up in a directional erosion and a directional dilation. After each step a β-map is performed to protect the corners from filtering.

For this simulation the same parameter settings are used as for simulation no. 4. The simulation result can be seen in Figure 7.8 and is quite different from the previous two simulation results. The image looks much smoother and there are no sharp contrasts. However, this smooth result has also an unnatural look. Therefore it is not an acceptable filter result.

### 7.7 Structure-steered (dilation, erosion, erosion, dilation) filter, \( K = 5 \)

The same simulation as in the previous section has been done, but with a changed parameter setting: \( K = 5 \). The result is almost the same as from the previous simulation and can be seen in the figure below. Here, the image impression is slightly smoother.

![Figure 7.9: Figure 7.2 filtered with structure steered filter: dilation, erosion followed by erosion, dilation. (\( K = 5 \)) (after additional post-subtraction LUTs)](image)

From Figure 7.8 and 7.9 it is clear that due to the smooth look, the image has a slightly unnatural impression. To overcome this problem, also a simulation with a structure-steered filter has been done where reconstruction has been used.
7.8 **Structure-steered (dilation, erosion + reconstruction) filter**

In this simulation a structure-steered filter that uses reconstruction has been tested. First a directional dilation followed by a directional erosion is performed. Next, this result is eroded with equal weighted line elements and the result is reconstructed. See the flow diagram in Figure 7.10.

![Flow diagram of structure-steered filter with reconstruction](image)

The simulation result can be seen in Figure 7.11 and has much more a natural look than the previous result in Figure 7.9. Also this result has more contrast. However, if a close look at the vessel walls is taken, it seems that there is still a quite 'hard' transition between the vessel walls and the isotropic regions. A more smooth result, like from the modified MIC algorithm, is preferred.

![Figure 7.2 filtered with structure steered filter: dilation, erosion followed by a reconstruction. (after additional post-subtraction LUTs)](image)
7.9 Structure-steered variant of Haupert's algorithm

Also a structure analysis is combined with Haupert's algorithm. Here, the directional closing is steered with weight factors, obtained after structure analysis. See Figure 7.12 for the flow diagram of this filter. Note that the directional tophat is not structure-steered. This choice has been made, because the bothat image doesn't contain much structure information.

![Flow diagram of structure-steered variant of Haupert's algorithm](image)

**Figure 7.12:** Flow diagram of structure-steered variant of Haupert's algorithm.

The simulation result can be seen in Figure 7.13. It looks quite similar with the simulation result of Haupert's algorithm in Figure 7.3. The vessel walls have a slightly smoother impression and therefore this result is preferred.

![Simulation results](image)

**Figure 7.13:** Figure 7.2 filtered with a structure-steered filter variant of Haupert's algorithm. (after additional post-subtraction LUTs)
7.10 Structure-steered variant of modified MIC algorithm

The last simulation that has been done, is a combination of structure analysis with the modified MIC algorithm. The modified MIC algorithm performed best up till now and it will be investigated if still some progression can be made. Filter 1 of the modified MIC algorithm is kept the same, while filter 2 has been changed (see Figure 7.14). This directional opening is steered with a structure analysis of the original input image and not one of the residual image on which the filter is performed. In this way only one analysis has to be performed. The parameter settings are listed in table 7.1.

![Flow diagram of 'filter 2' of structure-steered variant of modified MIC algorithm.](image)

The simulation result of the filter presented in this section can be seen in Figure 7.15. It looks slightly better than the result of the non-steered MIC algorithm in Figure 7.5, however the differences are hard to see. In the author's opinion this result is the best one from the simulations done in this chapter. Here, the amount of noise in combination with the contrast and the 'natural' look of the image is the best. In Figure 7.16 and 7.17 a comparison of the noise statistics is given of the results from Figure 7.5 and Figure 7.15.

![Figure 7.15: Figure 7.2 filtered with a structure-steered filter variant of the modified MIC algorithm. (after additional post-subtraction LUTs)](image)
As can be seen, the noise statistics don’t differ very much. The result in Figure 7.15 seems to have a little bit more noise for higher grey values. However, the result from Figure 7.15 is preferred.

![Figure 7.16: Mean noise STD per bin, corresponding with Figure 7.5.](image1)

![Figure 7.17: Mean noise STD per bin, corresponding with Figure 7.15.](image2)

The filter discussed in this last section performs best and therefore it has been tested also on another DSA image. This DSA image is of a hip and shown in Figure 7.18.

![Figure 7.18: Noisy DSA image of a hip, exam006. (right: detail image)](image3)

The filtered result of the DSA image in Figure 7.18 can be seen on the next page in Figure 7.19.
7. Simulation results

Figure 7.19: Figure 7.18 filtered with a structure-steered filter variant of the modified MIC algorithm. (after additional post-subtraction LUTs)

More simulation results, to show the performance of the structure-steered variant of the modified MIC algorithm in comparison with Haupert's algorithm, can be seen in appendix C.
8 Conclusions and recommendations

In order to solve the ‘vessel-over-bone’ problem, a previous trainee at XRD predevelopment, Haupert [7], has designed a new DSA process, which makes use of a logarithmic LUT. As a consequence, however, a lot of noise will be introduced in the dark regions of DSA images. It was found out by Haupert that the present noise could be removed quite successfully by using a morphological noise reduction filter. However, it was reasoned that even better results could possibly be obtained by improving the designed noise reduction algorithm.

In this chapter the conclusions will be given of research that has been done to develop a better noise reduction algorithm. Also some recommendations will be given, which could be a guide for future research on this subject.

8.1 Conclusions

Noise reduction by using basic concepts from mathematical morphology, with specialisation to grey-scale images, appears to be a powerful tool in DSA imaging. In this report several different noise reduction methods have been proposed, some of which appeared to be successful.

*Structure-steered directional morphological filters perform better than their non-steered directional variants.* So it may be concluded that the use of structure analysis of the image improves the performance of directional morphological filters. Simulation results of some structure-steered filters showed that especially the vessel walls appear sharper (more contrast) in comparison with results of their non-steered variants.

However, some caution must be exercised in order to prevent unnatural results, as can be seen in Figure 7.6 and 7.7. From these results it may be concluded that it is not preferred to perform a structure-steered directional (closing, opening) filter. This gives very ‘hard’ and unnatural effects at edges, such as vessel walls. A better alternative is to split the directional closing in a directional dilation and a directional erosion and the directional opening in a directional erosion and a directional dilation. See Figure 7.14 for an example of such a two-step directional opening. The results of such filters have a more natural impression, but look, in contrary, too smooth (see Figure 7.8 and 7.9).

When *reconstruction* is done after e.g. a two-step directional closing (see Figure 7.10), more information from the original image is kept and thus the result looks more natural (see Figure 7.11). Still, due to some ‘hard’ edge effects at vessel walls, this result is not yet completely satisfactory.

From the filters tested in this report, the *structure-steered variant of the MIC filter performs best*, although only a small improvement was obtained with structure analysis. Comparing this result with Haupert’s result, the vessels have a smoother look, while the contrast is more or less the same. Also the standard deviation of the noise is smaller according to a noise analysis computation.

Because the MIC algorithm is the only multi-scale filter tested in this report, we have not enough evidence to say that multi-scale morphological filters perform better than single-scale morphological filters. In the context of this report it can only be concluded that the multi-scale principle in the MIC algorithm works quite well on DSA images and that it can probably still be improved.

Generally speaking, due to a lack of knowledge at this moment, it is not possible to say which kind of morphological filter for DSA images is optimal. Therefore more research is needed.
In the calculation of the weight factors of a structure-steered morphological filter, several parameters can be varied. For example, the α-map can be adjusted to take more or less structure into account. With the parameter $K$, in equation (6.24), the structuring elements can be steered more or less in a specific direction. Therefore, it is possible to tailor the parameters to create an optimal set-up for a certain type of images. Although the structure-steered filters in this report are only tested on DSA images, it is expected that with some adjustments they will also perform successfully on other image types.

From section 5.4 it may be concluded that a set of directional structuring elements must be used in parallel and not in series. Otherwise features in the image will be distorted. When non-symmetric weight factors are used, the same set of directional structuring elements must be used when consecutive filter operations are performed, like a dilation followed by an erosion.

### 8.2 Recommendations

As could be read from the conclusions, the most promising filter concept seems one with a multi-scale approach. Although some multi-scale concepts have been explained in Chapter 4, only one multi-scale filter has been implemented and tested so far: the modified MIC algorithm. Therefore, in this multi-scale area, more experiments should be performed in order to obtain even better results.

To validate these filter results, more objective criteria must be found (e.g., smoothness and contrast of a blood vessel). In this report all the filter results are validated in a subjective way by the author, which is, of course, not preferred.

Especially multi-scale filters that use multi-resolution decomposition are interesting to examine. Because filtering is done on lower resolution levels, the computational time will decrease, while the filtering is performed on different scales. In the modified MIC algorithm however, the different scales are obtained by adjusting the size of the structuring elements. A promising idea is therefore to implement the MIC algorithm in a multi-resolution environment.

Another more advanced morphological multi-resolution decomposition, that could be tried, is shown in Figure 4.20. This idea is based on an article of Metzler [12]. In this morphological pyramid, on each level two subbands are present; a tophat subband as well as a bottomat subband.

Alternatively, directional structuring elements in more different orientations could be tried. In this report only 4 orientations were chosen, because mainly elements of size 3 were used. When larger elements are used it becomes possible to filter in more than 4 orientations. Another interesting idea in this context, which has not been implemented, is to steer not only the orientation, but also the size of the structuring elements.
### List of abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>AF</td>
<td>Application Factor</td>
</tr>
<tr>
<td>CIS</td>
<td>Common Imaging Subsystems</td>
</tr>
<tr>
<td>COSMIC</td>
<td>Noise reduction algorithm for Ultra Sound images</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Subtraction Angiography</td>
</tr>
<tr>
<td>FDXD</td>
<td>Flat Dynamic X-ray Detector</td>
</tr>
<tr>
<td>Inv LUT</td>
<td>Inverse Look-Up Table</td>
</tr>
<tr>
<td>IP</td>
<td>Image Processing</td>
</tr>
<tr>
<td>Log LUT</td>
<td>Logarithmic Look-Up Table</td>
</tr>
<tr>
<td>LP</td>
<td>Low-Pass filter</td>
</tr>
<tr>
<td>LUT</td>
<td>Look-Up Table</td>
</tr>
<tr>
<td>MIC</td>
<td>Morphological Image Cleaning algorithm</td>
</tr>
<tr>
<td>NPS</td>
<td>Noise Power Spectrum</td>
</tr>
<tr>
<td>NS LUT</td>
<td>Non-Subtraction Look-Up Table</td>
</tr>
<tr>
<td>OCCO</td>
<td>mean of (Opening,Closing) and (Closing,Opening)</td>
</tr>
<tr>
<td>PMS</td>
<td>Philips Medical Systems</td>
</tr>
<tr>
<td>S LUT</td>
<td>Subtraction Look-Up Table</td>
</tr>
<tr>
<td>SNC</td>
<td>Sensor Noise Curve</td>
</tr>
<tr>
<td>SR</td>
<td>Single-Resolution</td>
</tr>
<tr>
<td>STD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>VISUB</td>
<td>Viewing SUBsystem</td>
</tr>
<tr>
<td>XRD</td>
<td>X-Ray Diagnostics</td>
</tr>
<tr>
<td>XTV</td>
<td>X-ray TV</td>
</tr>
</tbody>
</table>
References

[1] Berenschot, G.
SPATIAL ANISOTROPIC COSMIC FILTER.

[2] Eekeren, A.W.M.
LITERATURE SEARCH ON MATHEMATICAL MORPHOLOGICAL NOISE REDUCTION METHODS.

NOISE ESTIMATION IN FLUOROSCOPY.

MORPHOLOGICAL METHODS FOR BIOMEDICAL IMAGE ANALYSIS.

MULTIRESOLUTION SIGNAL DECOMPOSITION SCHEMES - PART I: LINEAR AND MORPHOLOGICAL PYRAMIDS.

NONLINEAR MULTIRESOLUTION SIGNAL DECOMPOSITION SCHEMES - PART I: MORPHOLOGICAL PYRAMIDS.

[7] Haupert, S.
IMPROVING THE VESSEL-TO-BONE CONTRAST IN DIGITAL SUBTRACTION ANGIOGRAPHY (DSA) IMAGES.

IMPROVING MEDICAL IMAGE QUALITY USING GENETIC ALGORITHMS.

RELEVANT X-RAY PHYSICS FOR NOISE REDUCTION IN FLUOROSCOPY IMAGING.

[10] Matheron, G.
RANDOM SETS AND INTEGRAL GEOMETRY.
References

ORIENTATION ESTIMATION: CONVENTIONAL TECHNIQUES AND A NEW NON-DIFFERENTIAL APPROACH.
Ed. by M. Gabbouj and P. Kuosmanen.

DENOISING OF ULTRASOUND SECTOR SCANS BY NONLINEAR FILTERING OF A MORPHOLOGICAL AND LINEAR RATIO PYRAMID.

A NEW ALGORITHM FOR IMAGE NOISE REDUCTION USING MATHEMATICAL MORPHOLOGY.

[14] Serra, J.
IMAGE ANALYSIS AND MATHEMATICAL MORPHOLOGY.

NON-LINEAR DIFFUSION FILTERING: THEORY, ALGORITHMS, AND APPLICATIONS.
Dept. of Computer Science, University of Copenhagen, Denmark (1998).

SHAPE PRESERVING FILAMENT ENHANCEMENT FILTERING.
Appendix A: Matlab code noise analysis

function [noiseLUT,count] = noise_lut_spatial(x, threshold_factor, WindowSize_row, WindowSize_col, step)

% Calculate per image pixel the local mean and standard deviation
% on a given neighbourhood, e.g. 4x4
% meanx = colfilt(x,[WindowSize_row WindowSize_col],'sliding','mean');
% sigmax = colfilt(x,[WindowSize_row WindowSize_col],'sliding','std');

% Downsmaple meanx and sigmax to reduce the total amount
% of data to an acceptable level (optional)

for k=1:2
    meanx = reduce(meanx,[1 2 1]);
    sigmax = reduce(sigmax,[1 2 1]);
end;

% Remove negative values of meanx
% meanx = max(meanx,0);
% mx = meanx(:);
% sx = sigmax(:);

% Omit all values of sigmax which are larger than a certain
% threshold, e.g. all values where sigmax > 15.
% smax = 15;
% smin = 0;
% ix = (sx > smax) | (sx < smin);
% mx(ix)=[];
% sx(ix)=[];

% calculate the noise LUT size
% min_value = min(mx) % The minimum gray value of the image
% max_value = max(mx) % The maximum gray value of the image
% LUTsize = floor( (max_value-min_value)/step );

% If there are too few pixels, the estimation result can be unreliable.
% Select threshold used to check whether an estimation result for noise
% standard deviation is reliable.
% One calculates at first the number of occurrence that the pixel gray
% value is within an intensity interval. As result, a histogram is set
% up. The median value of this histogram is selected as the threshold.

histo = zeros(LUTsize,1);
for i = 1:LUTsize
    GrayLevel = min_value+(i-1)*step;
    p = find( (mx>=GrayLevel) & (mx<(GrayLevel+step)) ) ;
    histo(i) = length(p);
end;

threshold=floor(median(histo)/threshold_factor);

© 2002 Philips Medical Systems Nederland B.V.
Philips Medical Systems
% calculate noise LUT and set up the histogram
%--------------------------------------------------
j = 1;
GrayLevel = min_value;
for i=1:LUTsize % Loop over the noise LUT size
  % find the pixels whose value is within an intensity interval
  p = find( (mx>GrayLevel) & (mx<(GrayLevel+step)) );
p_length = length(p);
  if p_length > threshold
    % count(1,j) = local mean image intensity
    count(1,j) = GrayLevel;
    % count(2,j) = number of occurrences that local mean value
    % is within a defined intensity interval
    count(2,j) = p_length;
    tmp = mean(sx(p)); % mean of local std
    % noiseLUT(1,j) = local mean image intensity
    noiseLUT(1,j) = GrayLevel;
    % noiseLUT(2,j) = local variance
    noiseLUT(2,j) = tmp*tmp;
    j = j+1;
  end;
  % another mean intensity interval
  GrayLevel=min_value+i*step;
end;

% Plot noise STD curve
%--------------------------------------------------
plot(noiseLUT(1,:), abs(sqrt(noiseLUT(2,:))),'k*');
grid on;
title('Mean noise STD');
xlabel('Mean intensity in bin');
ylabel('Mean noise standard deviation in bin');

% Plot histogram of observations
%--------------------------------------------------
figure;
bar(count(1,:),count(2,:), 'r');
title('Histogram of selected observations');
xlabel('Mean intensity in bin');
ylabel('Number of observations in bin');

% Plot relation mean grey level and noise std
%--------------------------------------------------
figure;
plot(mx(:,),sx(:,),'r');
grid on;
title('Relation mean grey level and noise std');
xlabel('Mean intensity');
ylabel('Noise standard deviation');
drawnow;
Appendix B: Flow diagram noise analysis

noisy image

window size

calculate mean & std

optional: filter and down-sample data

define threshold
to separate noise and features

plot 'relation mean grey level and noise std'

bin size

divide greyscale in bins and count observations

median

threshold factor

plot 'histogram of observations'

plot 'noise STD curve'

calculate mean std in each bin

© 2002 Philips Medical Systems Nederland B.V.
Philips Medical Systems
Appendix C: Extra simulation results

1. Exam002: knee
2. Exam006: hip
3. Exam011: knee
4. Exam013: hand
5. Exam016: head
Figure C.1: a) Noisy DSA image of a hand. b) DSA image filtered with Haupert's algorithm. c) DSA image filtered with structure-steered variant of MIC algorithm.
Figure C.2: a) Noisy DSA image of a hand. b) DSA image filtered with Haupert's algorithm. c) DSA image filtered with structure-steered variant of MIC algorithm.
Exam011: Knee

Figure C.3: a) Noisy DSA image of a knee. b) DSA image filtered with Haupert's algorithm. c) DSA image filtered with structure-steered variant of MIC algorithm.
Figure C.4: a) Noisy DSA image of a hand. b) DSA image filtered with Haupert's algorithm. c) DSA image filtered with structure-steered variant of MIC algorithm.
Exam016: Head

Figure C.5: a) Noisy DSA image of a hand. b) DSA image filtered with Haupert’s algorithm. c) DSA image filtered with structure-steered variant of MIC algorithm.