

ISSN 0280-5316
ISRN LUTFD2/TFRT--5662--SE

Medical Image Registration

Helena Fridman
Telina Englund

Department of Automatic Control
Lund Institute of Technology
February 2001

Department of Automatic Control Lund Institute of Technology Box 118 SE-221 00 Lund Sweden	<i>Document name</i> MASTER THESIS	
	<i>Date of issue</i> February 2001	
	<i>Document Number</i> ISRN LUTFD2/TFRT-5662--SE	
<i>Author(s)</i> Helena Fridman Telina Englund	<i>Supervisor</i> Brett Ninness, Newcastle, Australia Björn Wittenmark, LTH	
	<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Medical Image Registration		
<i>Abstract</i> <p>Image registration of X-ray images is used to correctly align the patient before the cancer tumour is treated with sets of high dose X-ray radiation. Before the first treatment, a <i>simulated X-ray image</i> is taken showing the best possible localisation of the tumour. From then on, after each of the radiation treatments of the tumour, a low dose of the X-ray therapeutic radiation is used to take an image of the tumour and its surroundings. This so called <i>portal X-ray image</i> is compared to the simulated image to decide whether the patient should be moved before the next treatment in order to improve the accuracy of the therapeutic beam and, hence, prevent the high dose radiation from hitting other surrounding tissues close to the tumour.</p> <p>Due to differences in quality of the X-ray plates used for the recording of the simulated and the portal image, the images differ a lot in contrasts, noise level, and possibly even scale and size. The diverse quality of the images is the main problem of the image registration task.</p> <p>At present, the comparison of the X-ray images is carried out by hand. A software for automating the process would decrease the influence of human intuition on the treatment, decrease the treatment time, and enable less qualified personnel to carry out the treatment.</p> <p>Much research has already been done within the field of image registration, many with various results. The preceding pilot study suggests that a deeper study, that would go beyond the scope of this project, is needed for the task. This report strives for presenting fields of possible approaches to the problem and tests performed within these fields. Hopefully, the report will give some guidance on directions that a future research could take.</p>		
<i>Key words</i>		
<i>Classification system and/ or index terms (if any)</i>		
<i>Supplementary bibliographical information</i>		
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 94	<i>Recipient's notes</i>
<i>Security classification</i>		

The report may be ordered from the Department of Automatic Control or borrowed through:
University Library 2, Box 3, SE-221 00 Lund, Sweden
Fax +46 46 222 44 22 E-mail ub2@ub2.lu.se

Acknowledgements

We would like to acknowledge the guidance and support of our supervisors Brett Ninness, from the Department of Electrical and Computer Engineering at the University of Newcastle, and Björn Wittenmark, from the Department of Automatic Control at Lund Institute of Technology.

Also, we would like to thank Sarah Johnson and Tristan Perez for their kind help with the spelling and for their comments on style, that certainly contributed to the final form of this report. Our appreciation also goes to Harry Daniel, Aaron Hardy, and the technical staff from the Department of Electrical and Computer Engineering at the University of Newcastle, for their help with the computing environment. We are also thankful to Per-Henrik Rasmussen at Lund Institute of Technology, for having sent the writing report handbook to us and Leif Andersson from the Department of Automatic Control at Lund Institute of Technology for his help with the final publication of the report.

Finally, our gratitude and acknowledgement go to many people that made our experience in Australia very pleasant: Osvaldo, Adrian, Leif, Sarah, Tristan, Scott, Bronwin, Jess, and Harry (specially for lending the bicycles to us).

Contents

1	Introduction	9
I	Technical Research	13
2	Theoretical Background	15
2.1	Preliminaries	15
2.1.1	Image registration	16
2.2	Filtering	18
2.2.1	Characteristics of filter	18
2.2.2	Linear filters	19
2.2.3	Non-linear filters	20
2.2.4	Designing FIR filters	21
2.2.5	Filter Banks	22
2.3	Wavelets	24
2.3.1	Wavelet and Fourier Analysis	24
2.3.2	Wavelet transforms and multiresolution analysis	28
2.3.3	Three wavelets	30
2.4	Edge detection	31
2.4.1	Sobel's method	32
2.4.2	Robert's method	32
2.4.3	Laplacian operator	33
2.4.4	Gaussian operator	33
2.4.5	The Canny Edge Detector	34
2.4.6	Spatial Selective Noise Filtration Technique	35
2.5	The cost function	37
2.6	The minimisation algorithm	38
2.6.1	Line Search Methods	40
2.6.2	Step Length	41
2.6.3	Termination conditions	42
2.6.4	Local or global minimum?	42
2.6.5	Hessian Modifications	42
3	Software For Testing	44
3.1	Why a software for testing?	44
3.2	Functions	45
3.2.1	General functions	45
3.2.2	Creating a test image	46

3.2.3	Preprocessing	46
3.2.4	Edge detection	47
3.2.5	Calculating the translation and rotation	47
4	Results	48
4.1	Wavelet Based Approach	48
4.2	Filters	50
4.2.1	Linear filters	50
4.2.2	Nonlinear filters	51
4.3	Edge detection	56
4.3.1	Sobel's method	56
4.3.2	Robert's method	56
4.3.3	Laplacian of Gaussian method	57
4.3.4	The Canny Edge Detector	58
4.3.5	The Spatially Noise Filtration Technique	58
4.4	Minimisation of the cost function	60
5	Used software	62
5.1	EXcursion V2	62
5.2	Matlab 5.3	62
5.3	L ^A T _E X V2	64
5.4	Miscellaneous software	64
II	Analysis	65
6	Analysis of the research project	67
6.1	Introduction	67
6.2	Methods	68
6.2.1	Internal dividends of work	68
6.2.2	Learning from each other	68
6.2.3	Meetings with the supervisor	69
6.2.4	Setting up goals on the way	69
6.2.5	Who had the outermost responsibility?	69
6.3	SWOT-analysis	69
6.3.1	SWOT-analysis of the image registration software	70
6.3.2	SWOT-analysis of the project	71
6.4	Cost Savings and Quality Assurance	72
6.5	Time plan	72
6.6	Discussion	74
6.6.1	Communication	74
6.6.2	The importance of well defined goals	75
6.6.3	Working in Australia	75
6.6.4	Physical working environment	75
6.7	Conclusions	76

III Discussion and Conclusion	77
7 Discussion	79
7.1 Portal image	79
7.2 Wavelets and Filtering	79
7.3 Calculation of translation and rotation	80
8 Final conclusion	81
A Time plan	83
B Graphical User Interface in Matlab	85
B.1 Graphics Objects	85
B.1.1 Uicontrol	85
B.1.2 Uimenu	87
B.1.3 Axes	87
B.2 Programming the Callback function	87
B.2.1 Storing Handles as Global Variables	89

Chapter 1

Introduction

In this project we have developed new image processing techniques for automating X-ray registration, which is a central process within radiation oncology. X-ray registration involves the comparison of two X-ray images to estimate the relative displacement and rotation between the two pictures. The translation can be either horizontal (x) or vertical (y), and the one dimensional rotation is considered along the z -axis, which is perpendicular to the xy -plane.

Prior to the set of X-ray cancer treatments, a reference image of the patient and the best possible location of the tumor is simulated in an Odelft Simulix MC radiotherapy simulator¹. The film used for recording this image is of diagnostic quality and accordingly the image, which will be referred to as the *simulated image* in this report, is almost free from any disturbance. Still, as it can be seen in Figure 1.2, the image has prominent gridlines which may cause problems for the comparison of the two X-ray images later on. The images shown in Figure 1.2 and Figure 1.3 correspond to a cancer tumor in the pelvis region.

For the pelvis cancer treatment, a Varian 600C or a Varian Clinac 1800 linear accelerator² is used to produce 4MV or 6MV photons. Different energy level photons are used for different treatments. A collimator situated after the accelerator delimits the area where the X-ray radiation is let through, and the beam of X-rays is then aligned with the cancer tumour according to Figure 1.1. Some of the patients are treated on a 2 cm thin mattress, called hyper, which improves patient comfort. The hyper is then used on both simulator and treatment machines. Under the hyper or directly under the patient, the recording X-ray plate is found. The collimator is set to let only an as small as possible beam through, ideally only hitting the tumor. High-dose X-rays are used for therapeutic purposes and it is, therefore, vital that as little as possible of the surrounding tissues are hit. After each high-dose treatment, the collimator is expanded. Low-dose X-rays, covering also the surroundings of the tumor, are used to take the so called *portal image*, see Figure 1.3. The portal image is later used for the comparison to the simulated image, to check the patient alignment. As the portal image is taken with the same equipment as is used for the treatment, only a conventional X-ray film of lower quality than the diagnostic film is used. Also, the recording of megavoltage photons makes the image lack contrast resolution [22]

¹Nucletron Odelft, Delft, The Netherlands.

²Varian, Palo Alto, CA, USA.

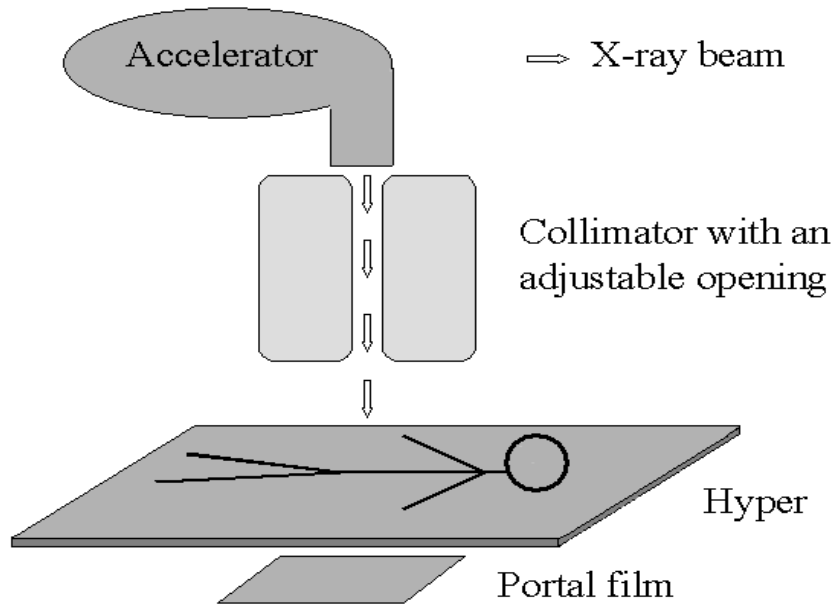


Figure 1.1: The set up.

and get very blurred. The comparison of the two X-rays is complicated since the two X-rays are of different quality, intensity, size and perhaps even scale.

As already mentioned, image registration becomes important to prevent the high-dose therapeutic X-rays from hitting too much of the tissues surrounding the cancer tumor. In both the simulated and the portal images, corresponding edges representing bony landmarks of the patient are sought. Once the edges are found, with the help of some edge detecting operator, only the simpler images obtained after the edge detection are used for the registration. In order to make these edge images as simple as possible, it is desirable to decrease the number of edges found by the edge detecting operator. This can be done by some sort of preprocessing of the original X-rays, for example filtering. By so doing, main features of the original X-rays can be strengthened before the edge detection. The problematic parts of the edge detection are the prominent gridlines in the simulated image and the pronounced rectangle, originating from the high-dose treatment area, which is in the middle of the portal image. These steep edges might have to be removed before the edge detection to distinguish the required edges of bonemarks. When the corresponding edges are found, a minimisation algorithm is used to detect possible translation and rotation. The minimisation algorithm uses a cost function (V), which is based on a least square algorithm between the pixel values; the two X-rays are put on top of each other and the square of the difference in grey value on each pixel position is summed up to a total cost:

$$V(\Theta) = \sum_{k=1}^N \sum_{j=1}^M [Z_{ref}(k, j) - Z(k, j)]^2. \quad (1.1)$$

The cost function refers to a $N \times M$ image matrix, where Z_{ref} and Z corresponds

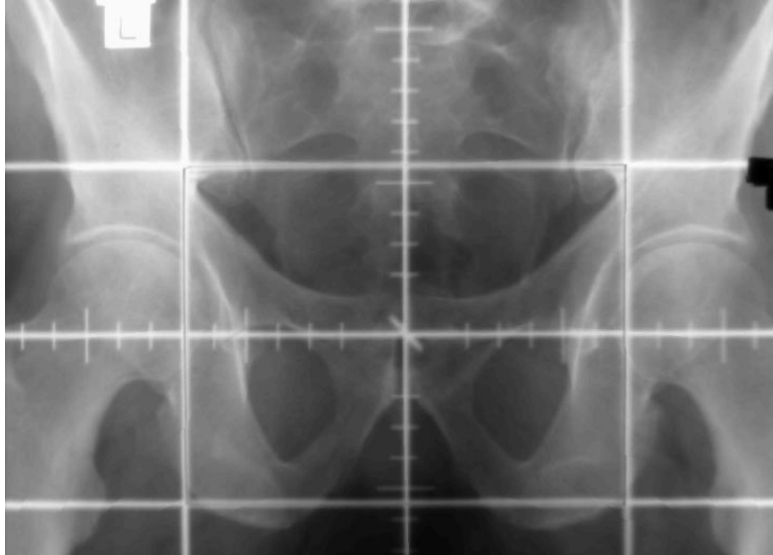


Figure 1.2: The simulated X-ray.

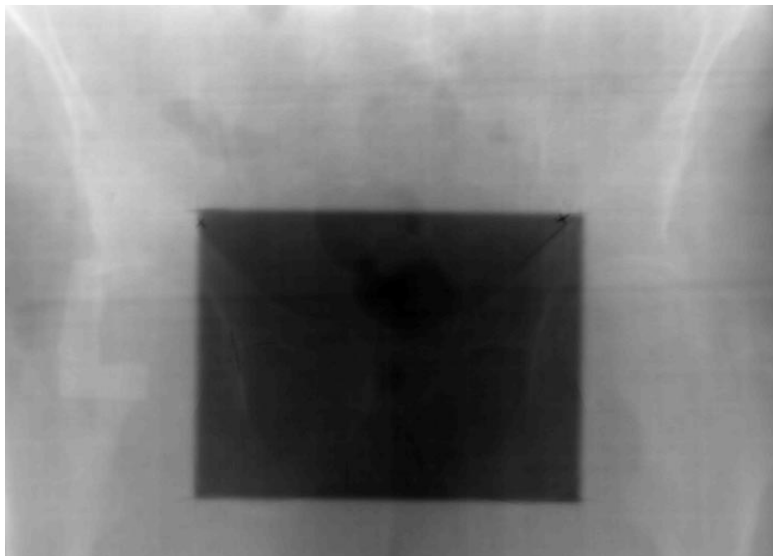


Figure 1.3: The portal X-ray.

to the simulated and the portal image respectively. $\Theta = [\theta_1\theta_2\theta_3]$ represents the translation (x, y) , and the rotation (z) . After having computed the cost at one position, the X-rays are moved with respect to each other in order to find the position that gives the lowest cost. The direction in which one of the X-rays, the portal, is moved is determined by gradient and hessian calculations. A pilot study has shown that the cost function has a clear global minimum for translation of very simple images, such as the images we obtain after the edge detection.

At present, all registration is done by hand. This motivates a software for automating the process, that would improve both accuracy and speed of the treatment. Using this software, decisions about possible alignment will no longer be based on intuition, consequently the registration could also be carried out by less qualified personnel.

More and more departments are beginning to use electronic portal imaging devices (EPID) to verify the positioning of the patient. These devices electronically acquire an image from the treatment field and are therefore easier to use than portal films as more than one image can be acquired per treatment and the image can be electronically enhanced immediately after acquisition [22]. Still, there are reports suggesting that portal images are superior to EPID [8]. In addition, new portal images with higher contrast qualities are developed [7]. Many former studies in the area have been carried out with various results. Some examples of methods can be found in [5], [15] and [16]. The preceding pilot study suggests that a deeper study, that would go beyond the scope of this project, is needed for the task. This report is intended to be a first approach to the problem and to give some guidance on directions that future research could take. Different approaches to the problem of image registration of X-rays are presented and tested.

In the first part of this report the technical research is presented. The second part consists, among others, of different aspects of group work and a SWOT analysis of the software and the project itself. This part is included in order to meet the requirements for a Master of Science in Electrical Engineering with alignment in Business Management at Lund Institute of Technology. In the third part, a final discussion and a conclusion of the two former parts are found. This project is the part of a joint effort between the Department of Electrical and Computer Engineering at the University of Newcastle, the Physics Department at the Mater Hospital and the Department of Radiation Oncology, also at the Mater Hospital, which are situated on the outskirts of Newcastle in NSW, Australia.

Part I

Technical Research

Chapter 2

Theoretical Background

2.1 Preliminaries

An image is a reproduction of an object. A radiant source, as for example a laser, an acoustic wave source as ultrasound, or, as in our case, X-ray radiation, is used to irradiate the object. The radiation, that has not been scattered away on its way through the object, or that has been reflected from the object, is then recorded by a sensor. There are a lot of different sensors, such as the retina in our eyes, the film in a camera, and CCD cameras, and some are more suitable than others for various types of radiation. Different radiation also causes varying disturbances to the recording of the images; for example diffraction, multiple reflection, back scattering, and surrounding radiation often make the image of the object blurred, distorted, or noisy. After being recorded by the sensor the output is sampled and quantised, hence it becomes digital.

A digital image is an image that has been discretised in both spatial coordinates and gray level values¹. Each digital image is represented by a matrix. In the matrix each entry (row and column position) has a value, $f(x, y)$, proportional to the brightness (or gray level) of the image at the spatial coordinates (x, y) , which correspond to the row and column position of the matrix, see Figure 2.1.

As a result of the enormous variety of both radiant sources and sensors, it is hard to build a model for an image recording system. A general block description is given in Figure 2.2, where f is the radiant energy distribution, reflected from the object. The optical system, often consisting of lenses, forms the radiant energy distribution, b .

In the case of this project, f is the radiant energy distribution of the X-ray beam after going through the body. No optical system is gathering the radiation before the recording by the sensor, i.e. $f = b$. The X-ray plates are the sensors. i represents the intensity image on the plates, which then are converted to bitmaps through digitisation. Even if X-ray radiation is the radiant source for both of the images, X-ray radiation with different intensity and films of different quality are used for the recording. As it can be seen in Figure 1.2 and Figure 1.3, it is much harder to distinguish different features on the portal image than on the simulated image.

¹Often the gray scale is expressed in values from 0 – 255 (2^8), where 0 is completely black and 255 is white. The intermediate values represent different darker or lighter gray tones.

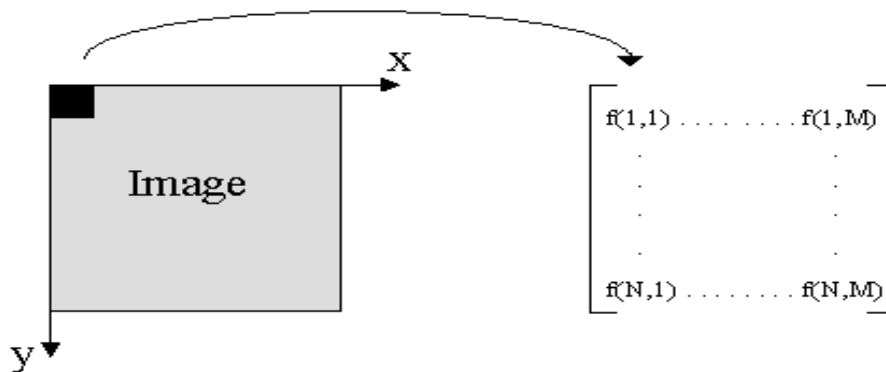


Figure 2.1: Representing a digitalised image with a matrix of size $N \times M$.

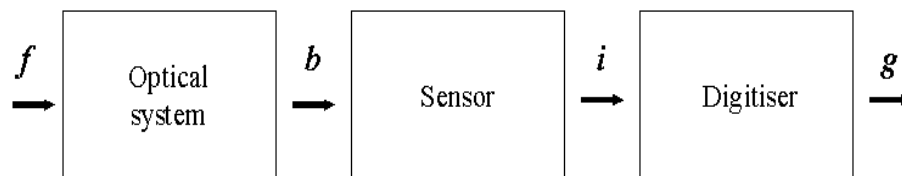


Figure 2.2: Model of a general digital image registration system.

Both the simulated and the portal images are already recorded and digitised into bitmaps on arrival from the hospital. Still, even though the task of this project begins after the image formation and recording, it is helpful to know details about the recording in order to continue the chain of blocks presented in Figure 2.2 in the best possible way for this particular application.

2.1.1 Image registration

The image registration process can be divided into three major blocks, presented in Figure 2.3. Within each field, different theories have been tried out searching for the best possible result. Information about the actual recording and digitisation of the images is important, and almost necessary, when selecting from the huge variety of filters and wavelets, which can be used for the filtering prior to edge detection. Both theory and trials of different already implemented filters and wavelets in Matlab have been used while seeking for good preprocessing methods. Attempts using different edge detection operators were carried out only on the original simulated image and the original portal image. The best operator on the original images should also have the best effect on an improved image, for example after filtering. To not get stuck in the first blocks, all the minimisation tests were performed on other test images, similar to those that can be obtained as a result of the edge detection.

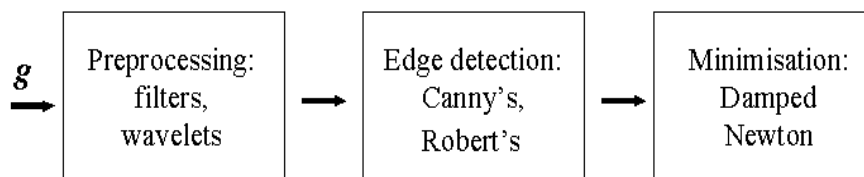


Figure 2.3: Model of the image registration part.

2.2 Filtering

A filter is a linear or nonlinear operator, $L(\cdot)$, that is acting on the input vector, $x(t)$ to create the output vector, $y(t)$.

$$y(t) = L(x(t)) \quad (2.1)$$

A unit impulse at time zero² as input causes an important output called the *impulse response* where $y(t) = h(t) = L(\delta(t))$. The filter coefficients are represented in the impulse response function, $h(t)$. For a digital filter, the filter coefficients are taken at discrete times, $t = n * k$, $h(n): \{ h(0), h(1), \dots \}$. Filters with a finite number of filter coefficients, a finite $h(n)$, are called Finite Impulse Response (FIR) filters. Alternatively, Infinite Impulse Response (IIR) filters have an infinite number of filter coefficients and therefore an ideal response, see Section 2.2.4. However, they are often impossible to construct. The output from a *casual filter* never comes before the input, i.e. $h(n) = 0$ for $n < 0$.

In the time domain, the use of linear filters is carried out through convolution of $h(t)$ and $x(t)$, as shown for a digital filter in Equation 2.2.

$$y(n) = h(k) * x(k) = \sum_k h(k)x(n - k) \quad (2.2)$$

The filter can also be represented in the frequency domain by applying a Fourier transform on $h(t)$. Convolution in the time corresponds to multiplication in the frequency domain and, as the latter is an easier action to carry out, it is often preferable to implement filtering in the frequency domain. $H(w)$, the impulse response function in the frequency domain, is frequently used to describe filters throughout this report.

2.2.1 Characteristics of filter

A filter has a *stop band*, a *pass band* and also, in the non-ideal case, a *transition band* as shown in Figure 2.4(b). $H(w)$, the impulse response function in the frequency domain, is ideally equal to zero in the stop band and therefore prevents the frequencies in this interval from passing through the filter. In the same way, $H(w) = 1$ (ideally) in the pass band. Realisable filters are often not ideal and hence do not have an instant change from stop band to pass band or vice versa. The interval between the two bands is the transition band.

The filter that constraints on what is allowed is called a *brick wall filter*, see Figure 2.4(a), because of the step function in its response [26]. This brick wall filter is an ideal low pass filter and its response, a square wave, is the ideal low pass response. Thus, the ideal coefficients for a low pass filter are samples from the sinc function described in Equation 2.3. Equation 2.4 shows the approximated impulse response function in the frequency domain.

$$h(n) = \frac{\text{sinc}(\frac{\pi n}{2})}{\pi n} = \begin{cases} \frac{1}{2}, & n = 0 \\ \pm \frac{1}{\pi n}, & n \text{ odd} \\ 0 & n \text{ even, } n \neq 0 \end{cases} \quad (2.3)$$

$$H(\omega) = \frac{1}{2} - \frac{e^{i\omega} + e^{-i\omega}}{\pi} + \frac{e^{i3\omega} + e^{-i3\omega}}{3\pi} - \frac{e^{i5\omega} + e^{-i5\omega}}{5\pi} + \dots \quad (2.4)$$

² $x(t) = \delta(t) = (\dots 0, 0, 1, 0, 0 \dots)$

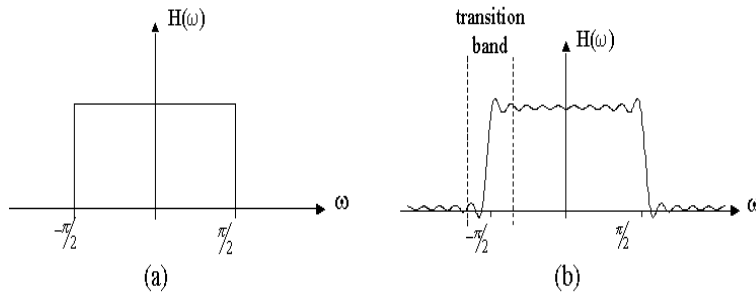


Figure 2.4: Ideal low pass filter and an approximation with 20 terms. The cut-off frequency is at $\omega = \frac{\pi}{2}$, where $H(\frac{\pi}{2}) = 0.5$.

The first term in the summation in Equation 2.4 sets the “DC term” as it remains when $\omega = 0$. The rest of the summation adds up all of the odd numbered frequencies, forming the desired square wave. Ideally, *all* odd numbered frequencies are included, i.e. the sum should be infinite. A FIR filter chops off the series in Equation 2.4 after N terms and produces as a result a N th approximation of the filter response. The errors due to the cut off are represented in form of *ripple* on the output, see Figure 2.4(b). The bigger the N , the narrower the ripple gets. Nevertheless, its amplitude approaches a constant, about 0.09. This is called the *Gibbs phenomenon* [26].

2.2.2 Linear filters

For a linear filter, the operator, $L(\cdot)$, in Equation 2.1, is linear, i.e. it satisfies the superposition and proportionality principles [19]. These two principles enable filtering through convolution with the impulse response function, $h(n)$. If $L(\cdot)$ also is shift-invariant, the filter is linear shift-invariant and it is better known as a *LSI filter*.

Two common linear filters are the high pass and the low pass filters. The simplest **low pass filter** takes the average of the input, $x(n)$, and the input $x(n - 1)$; it is a *moving average*. $x(n)$ is sometimes referred to as the “identity” and $x(n - 1)$ as the “delay”. As any linear operator working on an input vector, the low pass filter can be represented by a matrix. In this case, the matrix would consist of numbers only in the main diagonal and the sub diagonal on the left hand side of the main diagonal,

$$\begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{pmatrix}.$$

The number in the diagonals indicates the cut-off frequency of the filter using normalised frequency. Thus, 0.5 represents the middle of the frequency span, $\pm \frac{\pi}{2}$. A low pass filter smoothes the input, as it reduces all smaller and faster changes in the image. Ideally $H(0)=1$, but also $H(0) \approx 1$ can be called a low

pass filter³. However, the latter can not produce wavelets, see Section 2.3. The impulse response of this low pass filter is the coefficients in the main diagonal.

Instead of taking the sum of the signal and the delayed signal to make the average, the *high pass filter* takes the difference between the same samples. The matrix has coefficients in the same places, but the coefficients in the sub-diagonal are negative instead of positive. A high pass filter picks up the bumps in the signal. Also, worth mentioning, the Haar wavelet, presented in Section 2.3, is in fact a high pass filter with coefficients ± 1 . The impulse response is the coefficients in the main diagonal.

An example of a simple high pass filter:

$$\begin{pmatrix} 0.5 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 \\ 0 & -0.5 & 0.5 & 0 \\ 0 & 0 & -0.5 & 0.5 \end{pmatrix}.$$

A high pass and a low pass together form a *band pass filter*, where only frequencies between the two cut-off frequencies are let through. If instead the same frequencies are being blocked, the filter is called a *band stop filter*.

A low- and a high pass filter can be represented in a filter bank and then be called a *Quadrature Mirror Filter, QMF*⁴.

A FIR filter, low or high pass, is rarely invertible due to the shortened summation. Nevertheless, a filter bank, see Section 2.2.5, using only low and high pass filters can be invertible.

2.2.3 Non-linear filters

The operator, $L(\cdot)$, is in this case not linear; it does not satisfy either one or both of the superposition or the proportionally principles. Therefore, filtering can not be carried out through convolution between the input and the impulse response function.

A common family of non-linear filters are the *order statistic filters*, often used for noise removal [19]. One of the simplest filters in this family is the *median filter*⁵. The median value m of a population (i.e. a set of pixels forming a neighborhood) is that value in which half of the population has smaller and the other half has larger values than m . A median filter smoothens the data while keeping small and sharp details such as edges. However, there is a tradeoff between noise reduction and the sharpness of the image when the size of the population is increased. The filters, or algorithms, which arrange the pixels within this certain population in increasing order are described by a mask. One of the pixel values covered by the mask is set to the current median pixel value. Examples of masks can be seen in Figure 2.5. A basic median filter exchanges the middle value of the mask, but a filter can be implemented to exchange any of the pixel values. The pixel values involved can also be weighted. Weighted Median, WM, filters are of great interest as they inherent robustness and edge preserving characteristics from the median filter and resemble linear FIR filters in certain properties. They offer much greater flexibility in design

³ $\omega = 0$ means near very low frequencies as $\omega = 2\pi f$.

⁴First constructed by Croisier-Esteban-Galand in 1976.

⁵Implemented for the first time by Tukey in 1971.

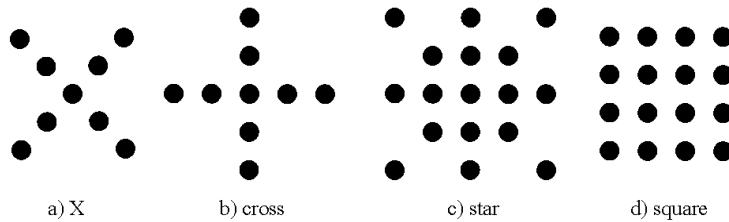


Figure 2.5: Different masks used in median filtering.

specifications than the median filter, as their filtering behavior can be controlled by a set of weights. A nonnegative integer is assigned to each input sample in the observation and the filter (or the algorithm) duplicates each sample to the number of the corresponding weight and chooses the median value from the new sequence. Also, nonnegative weights can be used, but the filter is then harder to implement. Other examples of order statistic filters are the *stack filters*, the *median hybrid filters*, the *α -trimmed mean filters*, and the *L-filters* [19].

Another example of non-linear filters are the ***morphological filters***. As in mathematical morphology, these filters search for geometric forms instead of analytic features in the images. Many of the members of this family have found more and more applications in biomedical image processing, shape recognition, and edge detection etc. [19].

One of the oldest groups of non-linear filter are the ***homomorphic filters***. These filters use non-linearities such as logarithms to transform convoluted or non-linear signals to additive signals and then process them with linear filters. The output from the linear filter is afterwards transformed with the inverse non-linearity, such as the exponential function for the logarithm. Noise on recorded images such as the X-rays in this project, is multiplicative. Accordingly, by using a homomorphic filter, the noise becomes additive and can thus be dealt with using linear filters such as a low pass filter or a wiener filter[19]. Homomorphic filters are also used for image enhancements as they follow the same principles as the human eye; they increase the intensity of a region of actual constant intensity, close to the edges of the region. This is called the *Mach band effect* and it shows how important edges are to our eyes and, hence, that our eyes react like high pass filters [19].

More complicated non-linear filters are for instance the polynomial filters based on Volterra series, and adaptive non-linear filters. Also, neural networks can be found within the field of non-linear filtering [19].

2.2.4 Designing FIR filters

There are different ways of truncating the ideal response of an IIR filter to make the best possible N th order FIR approximation. The easiest way is by *windowing*. As the name indicates, a window is applied to the function to form the pass and the stop band (and the enclosed transition band). The bigger the window size, the closer to the cut off frequency, ω_c the ripple gets in location. Still, the error amplitudes do not decrease⁶. Three well known windows are the

⁶Gibb's phenomenon.

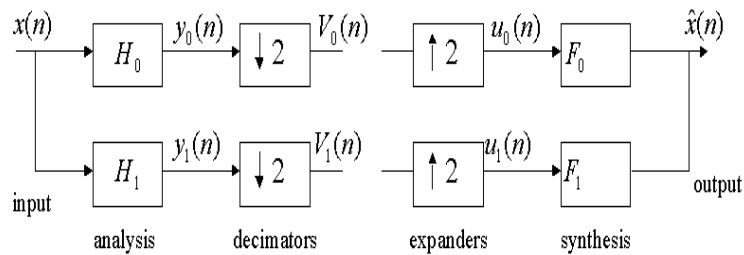


Figure 2.6: Structure of a filter bank.

Hanning, Hamming, and Kaiser windows [26].

There are various filters designed to meet different expectations for the ripple in the stop band and respectively in the pass band. Some examples are the equiripple filters with the smallest maximum error, the half band and M th band filters designed to handle changes in sampling rate, the maximum flat filters, and the eigenfilters [26].

2.2.5 Filter Banks

A filter bank is a set of filters. Often an *analysis bank* divides the input signal into frequency bands, before sending or storing. Afterwards, a *synthesis bank* recombines the signal. The filter banks often only consists of low pass and high pass filters, the output of which is downsampled (analysis) or upsampled (synthesis). As the signal is both high and low pass filtered in each step of the bank, down/upsampling is needed to prevent a signal of double size, see Figure 2.6. The downsampling often implies that only the even samples of the signal are kept after filtering.

Filters in the filter bank are represented as matrices. An orthogonal filter bank consists of transposes and inverses of the filter matrix, a biorthogonal of only inverses. The only total effect of a filter bank after analysis and synthesis is a delay. Wavelets are, in fact, a filter bank, dividing the original function into frequency bands. Yet, unlike the FIR filter banks, the frequency bands in wavelet based banks are logarithmic; the size of the frequency band gets twice as long in time for each step. In a FIR filter bank, all interval are of equal size. Sometimes the frequency bands can overlap each other, causing aliasing and non-orthogonality, see Figure 2.7.

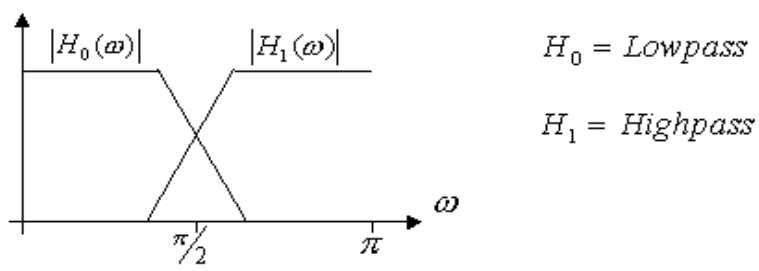


Figure 2.7: Overlapping responses cause aliasing.

2.3 Wavelets

A wavelet is a small wave, which is localised both in time and in frequency; the oscillating property makes it a wave, the localisation a wavelet. Originally, the word wavelet came from French literature on geophysics [26]. As the wavelet wave is localised in time, the French word for wave, *onde*, became *ondolette* meaning small wave and, accordingly, through direct translation to English the word wave became **wavelet**. The first wavelet was constructed by Alfred Haar in 1910 and had the shape of a square-wave. Still, Haar did not then call the function a wavelet. Nowadays, after a fast development in the 80's, a variety of different applications are using wavelets and a big diversity of different wavelets, with various properties, covers almost any need. It is, in most cases, unnecessary to create new wavelets. Wavelets are used for applications such as signal processing, image analysis, and efficient data compression.

Recent important names within the wavelet area are Stéphane Mallat, a French mathematician, who introduced multiresolution analysis, MRA, described in Section 2.3.2, in 1989. Ingrid Daubechis, a Belgian mathematician, has constructed several families of wavelets and has also shown that wavelets other than the Haar wavelet can have compact support⁷. She is also the author of a well known book on how to apply wavelets [3].

Wavelets can be deeply mathematical, but like Daubechis book, this report strives only for a very brief and more applied introduction to the immense field of wavelet theory. Perhaps “Notes on wavelet theory” would be a more correct title for the section.

2.3.1 Wavelet and Fourier Analysis

Like Fourier analysis, wavelets describe a certain function with the help of some basic functions. Wavelets were introduced to increase the speed of computations by making the theory of decomposition applicable also for non-periodic functions. **Fourier analysis**⁸ decomposes functions into components of the trigonometric functions $\{\sin(j), \cos(j), j = 1, 2, \dots\}$, i.e. it uses these trigonometric functions to construct the base onto which the original function is projected. The Fourier transform is a transformation from the time plane to the frequency plane and has the following general form:

$$f(t) = 0.5a_0 + \sum_{i=1}^{\infty} a_j * \cos(jx) + b_j * \sin(jx). \quad (2.5)$$

Trigonometric functions are perfectly localised in frequency (as can be seen when studying the frequency spectrum of a sin or cosine function), but are not at all limited in time. Only periodic functions, having the same frequency

⁷compact support \approx the time interval of the signal that the wavelet covers (unlike the cosine and sinus used in Fourier, the wavelets are localized in time). Stricter definition of support = the closed interval in continuous time outside which the wavelet, $W(t)$, equals zero. When we close the set where it is nonzero to include the jump location $t = 0.5$ as well as the end points, we have found the support of function $W(t)$. Compact support = this closed set is bounded.

⁸Jean-Baptiste Fourier, a French mathematician, discovered that it is possible to decompose any of a large class of functions into component function of only standard trigonometric functions while studying heat conduction in the early 19th century [18].

contents throughout the whole signal, can therefore be described by Fourier analysis. Each Fourier coefficient represents how much of a certain frequency the original function contains. Many functions can hence be well described, only by using the significantly large coefficients. This is something that we will find for wavelet analysis as well. As always, there is one exception to the rule of time localisation for Fourier analysis, the Short Time Fourier Transform.

The *Short Time Fourier Transform*, *STFT*, is also called the Gabor transform after its inventor. A sliding Gaussian window version is used to create a Fourier transform which is localised in both time and frequency. Stationarity is assumed inside the window and the window size is, unlike wavelet analysis, constant. The smaller the window, the better the time resolution. However, the smaller the window size, the fewer the number of discrete frequencies that can be represented in the frequency domain. The STFT transform has the following form:

$$STFT(\tau, \omega) = \int_{-\infty}^{\infty} s(t)g(t - \tau)e^{-j\omega t} dt \quad (2.6)$$

where $s(t)$ represent the signal, $g(t - \tau)$ the window around the time-point, τ , and the frequency, ω .

Wavelet analysis uses time and frequency located⁹ wavelets instead of the periodic trigonometric functions to construct the basis upon which the original function, $f(t)$, is to be projected. The wavelet transform divides the function into different frequency interval, or *scales*. Each scale has its own basis for its approximation of the original function and for its details, i.e. details containing frequencies within the frequency interval that this scale represents, see Section 2.3.2. The whole family of basis vectors comes from one single so called *mother wavelet*, covering the time interval $t = [0, N]$. The mother wavelet is then dilated and translated with the help of the *scaling function*, $\Theta_{j,k}$, (also called the *father wavelet*) throughout the transform. In Figure 2.8 it is shown how the time interval that the wavelet covers changes, due to dilation and translation. Notice that the number of periods inside the interval always stays the same; the carrier frequency changes. This is the fundamental difference between wavelets and the STFT, which is the closest related Fourier transform, see Figure 2.10. Thanks to the frequency localisation, wavelet analysis often needs fewer coefficients than Fourier analysis in order to reconstruct the original function. Sudden changes can be described using a high frequent wavelet only on the position of the change. The periodic trigonometric functions used for Fourier analysis are not localised in frequency and long summations (hence more coefficients), such as in Section 2.2.1, have to be used to obtain a good approximation of the function.

In other words, the resulting wavelet coefficients represent the frequency content at different positions and on different scales of the image, and their connection to the original function, $f(t)$, is described by the *wavelet transform*. The wavelet transform converts from the time plane to a scale/translation plane, (j, k) , and the scaling functions has the following form:

$$\Theta_{j,k} = 2^{0.5*j} * \Theta(2^j t). \quad (2.7)$$

⁹time located = the wavelet only approximates the part of the function being in the same time interval as the wavelets compact support, see footnote in Section 2.3.

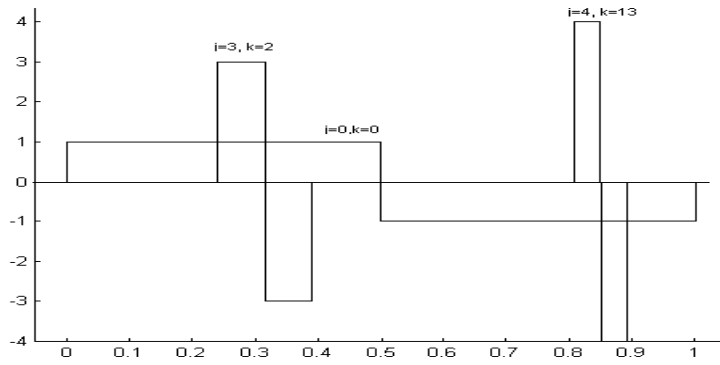


Figure 2.8: Mother wavelets and two dilations and translations.

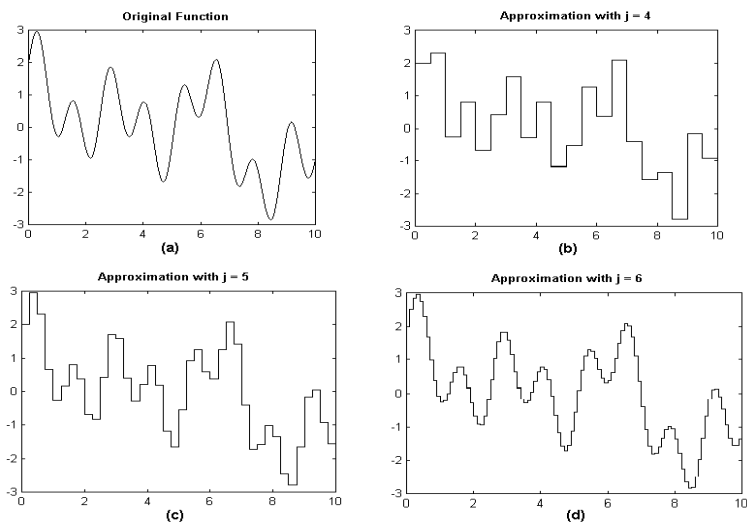


Figure 2.9: Different levels of approximation of the original function.

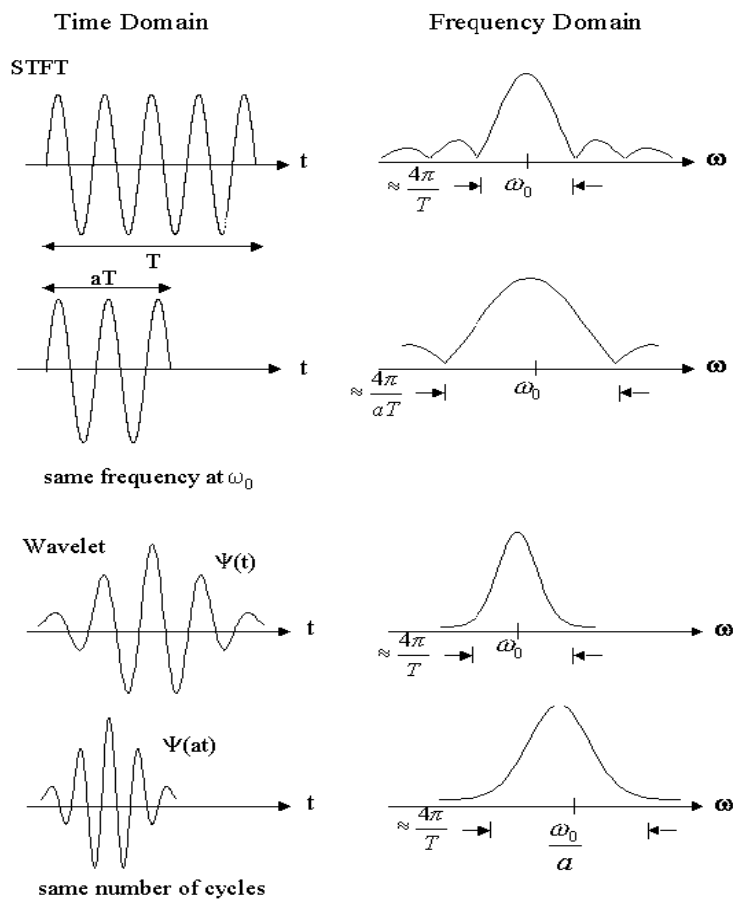


Figure 2.10: The difference between STFT and wavelets.

2.3.2 Wavelet transforms and multiresolution analysis

Wavelets are splitting functions into several scales of resolution, V_j . Multiresolution analysis (MRA) combines the details from different scales; it consists of studying the differences of approximations made at adjacent levels, the *detailed signals*. When it was discovered that the wavelet coefficients could be found recursively, i.e. that there is a connection between V_j and V_{j-1} , wavelet transforms became *fast* transforms; the same transform, or the same filters, are used over and over again in the decomposition of the original signal.

The transform moves towards larger and smoother subspaces, V_j , and complementing detail spaces, W_j , by taking *differences* (high pass filtering, see Section 2.2.2) and *averages* (low pass filtering, see Section 2.2.2). The differences between two adjacent samples stay at the finer scale as so called *details*, $d_{j,k}$. The detailed projection of the original function, f , on this scale plane, is expressed by the coefficients, $d_{j,k}$, using the wavelets, $w_{j,k}$, in order to span the detail space, $W_{j,k}$, see Figure 2.11.

Unlike the differences, the averages between the samples continue down (or up for a pyramidal representation) to the next scale, to become the coefficients, $c_{j,k}$, of this scale. At every scale, the scaling function, $\Theta_{j,k}$, forms a base for its approximating subspace, V_j , see Figure 2.9. The approximation (the projection) of $f(t)$ on V_j , is a combination of basis vectors spanned by the scaling functions (instead of the wavelets as in the case of the detail space) and expressed by the coefficients $c_{j,k}$.

Each scale plane can be looked upon as the result of band pass filtering applied to the original function (or image matrix), as it represents the original function in a certain frequency interval. As we re-scale and change levels, the frequency increases by 2^j and the time interval decreases by the same factor. Borders or features from the image, containing frequencies within the interval that the scale represents, should be visible in a graphical representation of the details of the plane. The approximation of the original function in a representation of the coefficients $c_{j,k}$ should be visible in V_j .

As mentioned earlier in Section 2.3.1, a longer wavelet (in time) represents lower frequencies of the image. Because of its longer length, less coefficients are needed to cover low frequency features of the whole image. Often, these low frequency structures in the image therefore are represented in a very smoothed and low sampled version of the image. This blurred image is complimented with coefficients representing finer and higher frequency details. Just as a human ear can not distinguish between two slightly different high frequencies with the same accuracy as for lower frequencies, our eyes can not detected too quick or too small changes in an image or a film. All the coefficients, representing the very finest details in an image or audio recording, are therefore usually not being used. Data can be compressed through saving only coefficients representing, for us, detectable changes or having significantly large values. Different accuracy is needed for different applications.

In Figure 2.11, the principles of multiresolution analysis will be shown by transforming a 4x4 matrix using the earlier mentioned Fast Wavelet Transform in two dimensions. Most of the unidimensional wavelets and transforms also have multidimensional counterparts.

For orthogonal transforms, see Section 2.3.3, each scale can be calculated from the former as seen in Equation 2.8. Hence, the original image can be

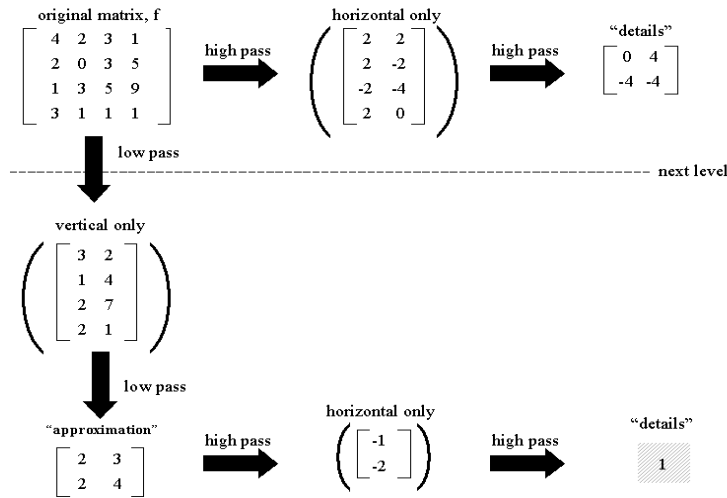


Figure 2.11: Multiresolution explained by an example using the Fast Wavelet Transform.

restored by the sum of all scales. For biorthogonal wavelets, the restoration is not a direct sum [26].

$$V_j = V_{j-1} + W_{j-1} \quad (2.8)$$

Instead of splitting the frequencies into uniform bands, w to $w+\Delta w$, multiresolution divides the frequencies into *octave bands*, with steps from w to $2w$. This is the same partitioning as for the music scale. Most commonly, the shortest note is a 32nd note, corresponding to the 5th level $2^{-5} = 1/32$. Often wavelets stop on that resolution as well, even though the theory goes on to infinity [26]. The only difference is that in music there are notes between for example two C:s separated by an octave, wavelets only have the end notes.

There are at least two different ways of introducing wavelets, the Continuous Wavelet Transform, CWT, and the Discrete Wavelet Transform, DWT.

CWT The *Continuous Wavelet Transform*. The input signal is correlated with an analysing continuous wavelet. In practice, some discrete version of the continuous transform will almost always be used, except for showing modules and phases for some transforms [10]. Leaving out the proofs, we just state that the CWT is a linear transform and it is covariant under translation and dilation. The definition of the continuous wavelet transform follows in Equation 2.10.

$$W(a, b) = \langle f, \Psi_{a,b} \rangle \quad (2.9)$$

where $\Psi_{a,b}$ represents the scaling function:

$$\Psi_{a,b} = \frac{1}{\sqrt{|a|}} \Psi\left(\frac{x-b}{a}\right). \quad (2.10)$$

The signs $\langle \rangle$ indicates the inner sum, a ($a > 0$) is the scale parameter and b indicates the position. f is the original function.

DWT *The Discrete Wavelet Transform.* A discrete version of the continuous transform, that generally does not have an exact analytical reconstruction formula. Two subclasses of the discrete transform exist: redundant versus non-redundant (e.g. pyramidal) transforms and orthogonal versus non-orthogonal bases of wavelets [24]. When it was shown that the wavelet coefficients on different levels could be found recursively through MRA, the DWT became the *Fast Wavelet Transform*, *FWT*, or as it also is called the Pyramid algorithm or Mallat algorithm.

2.3.3 Three wavelets

There are three different main kinds of wavelets: orthogonal, biorthogonal, and semi-orthogonal. Two properties are established for the bases they produce: *linear dependence* and *completeness*. Adding extra vectors to a subspace will destroy the independence and remove the completeness. Three properties for wavelets are symmetry, compact support and orthogonality. All three of the properties mentioned above can not be optimised for the same wavelet, with the Haar wavelet as the unique exception. Depending on the application, one of the three properties might be more important than the others and one particular kind of wavelets might be preferable.

Orthogonal wavelets

Orthogonal vectors are perpendicular to each other and their vector product, $x * y = 0$. Real functions are orthogonal when $\int X(w)Y(w)dw = 0$. Mutually perpendicular vectors form an orthogonal basis, which becomes orthonormal when it is normalised thorough division by $\sqrt{2}$.

Orthogonality of wavelets comes from orthogonality of filters [26]. Orthogonal wavelets have compact support, see footnote in Section 2.3. Thus, as the wavelet is zero outside a bounded interval, it represents a FIR filter¹⁰. Two wavelets with the same dilation index, j , but differing k can never have overlapping support and are therefore orthogonal [18].

For orthogonal wavelets, all wavelets, $w(2^j t)$, are orthogonal to the scaling functions $\Theta(t - k)$. Hence, the wavelets, $w(2^j t - k)$, and the scaling functions, $\Theta(t - k)$ are also mutually orthogonal. Moreover, the wavelets are orthogonal to all of its translations and dilations. Yet, orthogonality does not hold between scaling functions on different scales; $\Theta(t)$ is not orthogonal to $\Theta(2t)$.

Orthogonal wavelets give orthogonal matrices and unitary transforms [26]. An unitary transform is a lossless transform, which means that the rows of the matrix (the perpendicular axes) are correctly normalised. An orthonormal transform is a square matrix with orthonormal columns. Its inverse, X^{-1} , is its transpose, X^T , which means that the analysis bank matrix, see Section 2.2.5, is inverted by its transpose.

As mentioned before, in multiresolution analysis with orthogonal wavelets, each scale can be calculated from the former, see Equation 2.8, and the original image is the sum over all subspaces, V_j , and detail spaces, W_j .

¹⁰FIR= Finite Impulse Response. The filter has a finite number of nonzero coefficients.

Two signals, which are orthogonal in discrete time, have orthogonal transforms in continuous frequency.

There are, at least, two famous families of orthogonal wavelets, the Battle-Lemarie family including for example the Haar wavelet, and some of the Daubechis families such as symmlets and coiflets [18].

Linear independence is automatic for orthonormal vectors.

Biorthogonal wavelets

A biorthogonal wavelet is compact supported, symmetric, but not orthogonal (even though the name might seem to indicate so). For some applications, symmetry can be more important than orthogonality, hence the need for biorthogonal wavelets. Biorthogonal wavelets give invertible matrices and perfect reconstruction [26]. When the rows of the matrix, T , are a basis, the rows of T^{-1} are the biorthogonal basis, or the *dual basis*. When $T^t = T^{-1}$, the bases are self-dual and the same as an orthogonal basis.

As the name dual basis hints, biorthogonality refers to two separate bases belonging to two multiresolution analyses, one dual to the other. Both the mother wavelets to the corresponding MRAs are dual and so are the scaling functions. In the orthogonal wavelet case the two MRAs coincide and therefore the two scaling functions and the two mother wavelets are the same. A biorthogonal wavelet is orthogonal to the scaling functions.

An orthogonal wavelet forms an orthonormal basis (after normalisation), whereas a biorthogonal wavelet forms a so called *Riesz basis*, also called a stable or an unconditional basis (uniformly independent) [26]. The translations of the scaling function are also a Riesz basis. A MRA with a biorthogonal wavelet does not form an orthogonal basis (only a Riesz basis) for V_0 . The detail space W_0 , the wavelet, and the detailed spaces are no longer orthogonal to the approximation spaces. It is, however, possible to sum over the spaces, but the sum is no longer direct [10].

Semi-orthogonal wavelets

Semi-orthogonal wavelets are a special case of biorthogonal wavelets. They keep the orthogonality between detail levels, W_j , but still do not create an orthogonal basis for the different scales. These wavelets are formed from B-splines and not filters [26]. The semi-orthogonal wavelets were described by Chui and Wang in 1991 and are therefore also called *Chui-Wang wavelets* or *pre-wavelets* [18].

For an index m , the scaling function has support $[0, m]$ and the mother wavelet $[0, 2m - 1]$. $m = 1$ represents the Haar system. The semi-orthogonal wavelets are symmetric for even m and anti symmetric for odd m .

2.4 Edge detection

In the process of finding edges in an image, a filter operation that emphasises the changes in gray values and suppresses areas with constant gray values is necessary. It is found [6][11] that derivatives are suitable for this operation, and they are used in most of the edge detection techniques present today. By searching for local maxima in the first order derivative of the image, or zero-crossings in the second order derivative, an edge can be identified.

In the two dimensional case the derivative is calculated as the magnitude and the direction of the gradient vector. In the image $f(x, y)$ at location (x, y) the gradient vector is defined by Equation 2.11.

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}. \quad (2.11)$$

The vector \mathbf{G} points in the direction of the maximum rate of change of f at location (x, y) . The magnitude of the gradient vector is calculated in Equation 2.12, and the direction in Equation 2.13.

$$G[f(x, y)] = [G_x^2 + G_y^2]^{1/2} \quad (2.12)$$

$$\theta = \tan^{-1} \left[\frac{\Delta_y}{\Delta_x} \right] \quad (2.13)$$

In an image, the derivatives can only be approximated, and different edge detection techniques use different approximation methods. We will here present the edge detection techniques that we have been investigating. Most of them use the derivative, however, the last technique uses a wavelet transform in order to find the edges.

2.4.1 Sobel's method

One way of calculating the derivatives is to use the Sobel operators [6], which is the two 3x3 masks shown in Figure 2.12(b) and (c). A 3x3 size mask tends to make the derivative operations less sensitive to noise, without increasing the computational cost too much.

A part of an image is shown in Figure 2.12(a), x_5 is the pixel value at (x, y) and the other x_i represents the gray scale values of the eight neighbors of (x, y) . The component of the gradient vector in the x and y directions is defined in Equation 2.14 and Equation 2.15.

$$G_x = (x_7 + 2x_8 + x_9) - (x_1 + 2x_2 + x_3) \quad (2.14)$$

$$G_y = (x_3 + 2x_6 + x_9) - (x_1 + 2x_4 + x_7). \quad (2.15)$$

By convolving the masks with an image, $f(x, y)$, the vector gradient at all points in the image, is calculated. Equation 2.12 is then used to calculate the gradient, $\mathbf{G}[f(x, y)]$.

2.4.2 Robert's method

With reference to Figure 2.12(a) the gradient according to Robert's method [6] is calculated as

$$G[f(x, y)] = \sqrt{(x_5 - x_9)^2 + (x_8 - x_6)^2}, \quad (2.16)$$

and the direction of the gradient is given by

$$\alpha = -\frac{\pi}{4} + \tan^{-1} \left(\frac{x_8 - x_6}{x_5 - x_9} \right). \quad (2.17)$$

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

(a)

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

(b) (c)

Figure 2.12: (a) A part of an image. (b) The Sobel mask for the x-direction and (c) for the y-direction.

0	1	0
1	-4	1
0	1	0

Figure 2.13: The Laplacian mask.

2.4.3 Laplacian operator

The Laplacian operator [6] searches for zero crossings after filtering the image with a Laplacian filter. The Laplacian is a second order operator and is defined in Equation 2.18.

$$L[f(x, y)] = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad (2.18)$$

Equation 2.19 is used in order to calculate the digital Laplacian on the image in Figure 2.12 (a) and the mask in Figure 2.13 gives the coefficients for each pixel value.

$$L[f(x, y)] = x_2 + x_4 + x_6 + x_8 - 4x_5 \quad (2.19)$$

Since the Laplacian is a second order derivative it is seldom used for edge detection by itself since it makes the detection very sensitive to noise. However, it is often used in combination with other filters.

2.4.4 Gaussian operator

By using a Gaussian smoothing filter together with the Laplacian operator it is possible to detect edges [32]. The Gaussian distribution with two variables is given by Equation 2.20.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.20)$$

σ is the standard deviation representing the width of the Gaussian distribution.

Once the image is smoothed with the Gaussian operator a Laplacian filter can be applied to the image. The total operation then becomes

$$\Delta^2(f(x, y) * g(x, y)), \quad (2.21)$$

where $f(x, y)$ is the image pixel value at (x, y) and $g(x, y)$ is the gaussian operator. This operator is usually referred to as the *Laplacian of a Gaussian, LOG*. It is possible to convolve the original image with the operator in Equation 2.21 directly. By ignoring zero-crossings produced by small changes in image intensity, the LOG-operator can be made more resistant to noise.

However, the LOG-operator can become very large, and as an approximation the Difference of Gaussian (DOG) can be used instead as shown in Equation 2.22.

$$g(x, y) = \frac{1}{2\pi\sigma_e^2} e^{-(x^2+y^2)/2\sigma_e^2} - \frac{1}{2\pi\sigma_i^2} e^{-(x^2+y^2)/2\sigma_i^2} \quad (2.22)$$

2.4.5 The Canny Edge Detector

In 1983, John Canny introduced the Canny edge detection scheme [1] and it has since then been the standard edge detection scheme used around the world [33].

Canny introduced a function that was the sum of four exponential terms. However, the function resembled the first derivative of a Gaussian and it was the Gaussian function that ended up being used.

An image, $f(x, y)$, is convolved with an operator $g'_n(x, y)$ which is the first derivative of the two dimensional Gaussian, in some direction n .

$$g'_n(x, y) = \frac{\delta g(x, y)}{\delta n} \quad (2.23)$$

n should ideally be oriented normal to the direction of the edge to be detected. However, this direction is not known and an estimation, shown in Equation 2.24 must be used. This turns out to be a very good estimation for edge normal directions since a smoothed step has a strong gradient normal to the edge.

$$\mathbf{n} = \frac{\nabla(g(x, y) * f(x, y))}{|\nabla(g(x, y) * f(x, y))|} \quad (2.24)$$

An edge point is defined to be a local maximum of the operator $g'_n(x, y)$ applied to the image $f(x, y)$, i.e

$$\frac{\delta}{\delta \mathbf{n}} g'_n(x, y) * f(x, y) = 0 \quad (2.25)$$

and when substituting Equation 2.23 into Equation 2.25 the above becomes

$$\frac{\delta^2}{\delta \mathbf{n}^2} g(x, y) * f(x, y) = 0. \quad (2.26)$$

The edge strength at such an edge point will be the magnitude of

$$|g'_n(x, y) * f(x, y)| = |\nabla(g(x, y) * f(x, y))|. \quad (2.27)$$

When the Canny edge detection scheme is implemented, the image is usually convolved with a symmetric Gaussian and thereafter the second order derivative

is computed and put equal to zero in order to locate edges. This can be done because of the associativity convolution. The magnitude is used to estimate the edge strength. The method can detect strong and weak edges by using two different threshold values, but there is a trade-off between good edge detection and good edge localisation. Weak edges are only included in the output if they are connected to a strong edge and the method is therefore less likely to be fooled by noise.

2.4.6 Spatial Selective Noise Filtration Technique

In order to suppress noise in signals and images Xu, Weaver, Healy and Lu presented a filtration technique [28] that reduces the noise content by more than 80% while maintaining at least 80% of the value of the gradient at most edges. Since this filtration technique works by identifying edges we have been studying it to see whether it is applicable to our image registration problem.

Wavelet transforms are very attractive in image filtering because the basis function, $\psi_{m,n_x,n_y}(x,y)$, used are locally supported; they are nonzero only over the domain represented, see Section 2.3. This means that sharp transitions in images are preserved and depicted extremely well. The developed technique aims at finding edges, identified as features that have signal peaks across many wavelet scales. The filter extracted with this technique can be seen as a low pass filter that passes selected high frequency data. By computing the correlations between adjacent scales, edges are detected from noise. Small scale data is passed at positions where the correlation is large and suppressed if the correlation is small. The direct multiplication of wavelet transform data at adjacent scales are used to compute the correlation, which is given in Equation 2.28.

$$Corr_l(m, n_x, n_y) = \prod_{i=0}^{l-1} W(m+i, n_x, n_y) \quad 1 \leq n_x \leq N_x, 1 \leq n_y \leq N_y \quad (2.28)$$

l is the the number of resolution scales involved in the direct multiplication, and m is the intended processing scale. The sharp edges have large coefficients over many wavelet scales and therefore, the direct spatial correlation, $Corr_l(m, n_x, n_y)$, between wavelet coefficients at different adjacent scales of the wavelet transform is used to accurately detect the locations of edges or other significant features.

The algorithm

An algorithm was developed in order to calculate a spatial filter that selects the part of the data to keep and the part of the data to eliminate. The algorithm for the one-dimensional case is outlined below.

For each resolution scale m , the power of $Corr2(m, n)$ and $W(m, n)$ are computed and the power of each pixel in $Corr2(m, n)$ is re-scaled to that of $W(m, n)$. The pixel values in the new $Corr2(m, n)$ are then compared with $W(m, n)$ and an edge is identified at the position n if $|Corr2(m, n)| > |W(m, n)|$. The edge position and its corresponding value are stored. Finally, all the edges identified in this way are extracted from $Corr2(m, n)$ and $W(m, n)$ by resetting the values of these signals to 0's. The remainder of the data points in

$W(m, n)$ and $Corr_2(m, n)$ after the first round of edge extraction are referred to as $Corr'_2(m, n)$ and $W'(m, n)$. By re-scaling the power of $Corr'_2(m, n)$ to that of $W'(m, n)$ and comparing their absolute values, the next most significant edges are extracted. This process can be iterated many times until the power of the unextracted data points in $W(m, n)$ is nearly equal to some reference noise power at the wavelet scale m .

```

First, save a copy of  $W(m, n)$  to  $WW(m, n)$ 
Initialize the "spatial filter mask":  $mask(m, n)$  to 0's
Loop for each wavelet scale  $m$ 
{
  Loop for the iteration process
  {
    Compute the power of  $Corr_2(m, n)$  and  $W(m, n)$ :
       $PCorr(m) = \sum_n Corr_2(m, n)^2$ 
       $PW(m) = \sum_n W(m, n)^2$ 
    Re-scale the power of  $Corr_2(m, n)$  to that of  $W(m, n)$ :
    Loop for each pixel point  $n$ 
    {
      new  $Corr_2(m, n) = Corr_2(m, n) * \sqrt{\frac{PW(m)}{PCorr(m)}}$ 
    } end loop  $n$ 
    Loop for each pixel point  $n$ 
    {
      Compare pixel values in new  $Corr_2(m, n)$  and  $W(m, n)$ :
      if  $|Corr_2(m, n)| > |W(m, n)|$ 
      {
        Extract edge information from  $W(m, n)$  and  $Corr_2(m, n)$ ,
        and save it in the "spatial filter mask":
           $Corr_2(m, n) = 0.0$ 
           $W(m, n) = 0.0$ 
           $mask(m, n) = 1$ 
      } end if
    } end loop  $n$ 
    iterate until  $PW(m) \leq$  the noise threshold at scale  $m$ 
    Apply the "spatial filter mask" to the saved copy,  $WW(m, n)$ ,
    at scale  $m$ . Save the filtered data to  $Wnew(m, n)$ :
    Loop for each pixel point  $n$ 
    {
       $Wnew(m, n) = mask(m, n) * WW(m, n)$ 
    } end loop  $n$ 
  } end loop  $m$ 

```

The code is easily transformed into the two-dimensional case by substituting (m, n) with (m, n_x, n_y) and looping for every pixel pair instead of every pixel. It is an easy algorithm to implement and that is its advantage over more complicated algorithms [13][27] which detect edges directly on the wavelet transform. The disadvantage is that it may be less accurate than the more sophisticated techniques.

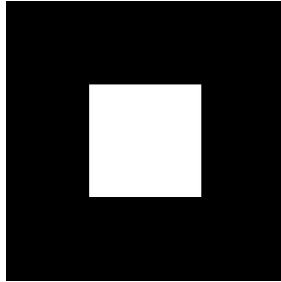


Figure 2.14: A simple image.

Results

When testing the algorithm Xu, Weaver, Healy and Lu found that the wavelet filtering technique was superior to the Wiener filtering algorithm that was used for comparison. They used a 256 by 256 large digital image as a test image and added a certain amount of synthetic Gaussian distributed white noise. In the implementation of the filter, only two adjacent scales ($l = 2$) were used in calculating the direct spatial correlation and the wavelet transform data was filtered at six scales ($m = 1 \dots 6$).

With this method the average SNR was enhanced by more than 25 dB which corresponded to a 90% noise reduction, however, the lowest contrast parts of the image could not be fully recovered. The Wiener filtered image had less noise reduction, the average SNR was 22.5dB. In addition, the image sharpness was lost and ringing at the edges was noticeable.

The wavelet domain filter is said to be superior to the Wiener filter and it is therefore an approach worth trying. The results of our investigation of this algorithm are presented in Section 4.3.5.

2.5 The cost function

To find the true translation and rotation between two images, a cost function, which penalises the disparity between pixel values, is introduced, see Equation 2.29.

$$V(\Theta) = \sum_{k=1}^N \sum_{l=1}^M |Z_{ref}(k, j) - Z(k, j)|^2. \quad (2.29)$$

$Z_{ref}(k, j)$ is the matrix of the reference image, and $Z(k, j)$ is the translated and rotated image. By penalising the disparity and trying to minimise the cost function, the true translation and rotation can be found.

The cost function can look very differently depending on the reference image and the translated and rotated image. If the cost function has several local minima it can be hard to find the global minimum. By translating the image in Figure 2.14, the cost function in Figure 2.15 is obtained. This is an extremely simple image but it can be used to test a minimisation algorithm since there are no local minima.

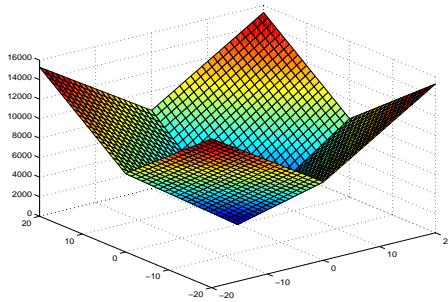


Figure 2.15: The cost function associated with the translation of a rectangle.

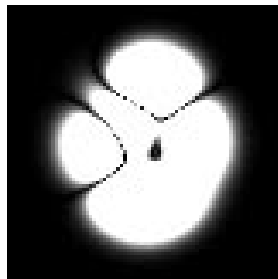


Figure 2.16: A simple image.

Another example of a cost function is the one in Figure 2.17, which is obtained from translating the image in Figure 2.16. It is also a rather simple image but a small local minimum in one of the corners is noticeable.

So far, the global minimum has been quite sharp and therefore easy to detect. When noise is added, as in Figure 2.18, the cost function becomes much flatter as can be seen in Figure 2.19.

The images described in this chapter have been of the original image translated into different positions, rotation has been left out in order to show the cost function graphically. In reality, rotation is added to the translation.

2.6 The minimisation algorithm

Once the cost function is defined, see Section 2.5, a way to find the θ that minimises the function needs to be implemented. This section aims at describing the theory behind a number of line search methods and considerations that need to be taken into account when using a line search method.

The minimisation problem is solved by using a line search method, which is an iterative search routine that aims at finding a θ that minimise the function f , that is $\min_{\theta} f(\theta)$. If the function f is an objective function that depends on real variables, with no restrictions at all on the values of these variables, the optimisation problem becomes unconstrained [17]. The function, f , is quite often complex and can not be described with an equation. In addition, the

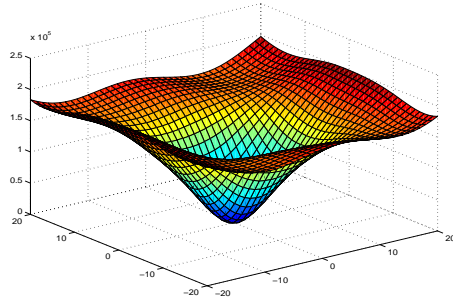


Figure 2.17: The cost function associated with the translation of the peaks.

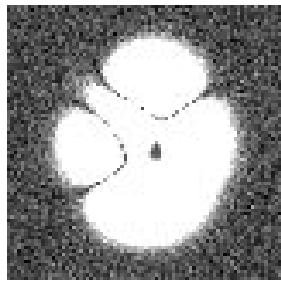


Figure 2.18: An image with added noise.

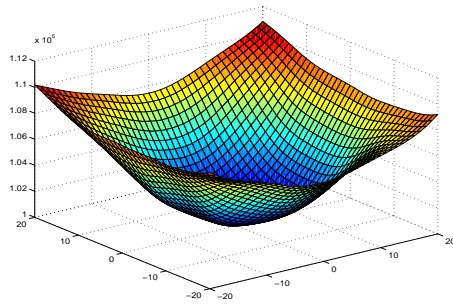


Figure 2.19: The cost function associated with the translation of the peaks with added noise.

knowledge of f is usually local and since the computation is often demanding, we only want to visit a few points in the iterative search for the minimum.

All algorithms for unconstrained minimisation require the user to supply a starting point, θ_0 , and then the algorithm generates a sequence of $\{\theta_k\}_{k=0}^{\infty}$ through iteration. To decide how to move from one θ_k to another the algorithms use information about the function f at θ_k and sometimes also from earlier iterations $\theta_0, \theta_1, \dots, \theta_{k-1}$. The information is used to find a new θ_{k+1} with lower function value than θ_k .

2.6.1 Line Search Methods

Each iteration of a line search method computes a fixed search direction, p_k , and then decides how far to move along that direction by choosing an appropriate step length, α_k . The iteration is given by:

$$\theta_{k+1} = \theta_k + \alpha_k p_k. \quad (2.30)$$

The success of a line search method depends on the choice of direction and step length. Most algorithms require p_k to be a descent direction, meaning that $p_k^T \nabla f_k < 0$. This property guarantees that f can be reduced along the chosen direction. The search direction often has the form:

$$p_k = -B_k^{-1} \nabla f_k, \quad (2.31)$$

where B_k is a symmetric and non singular matrix. The line search method chooses a direction, p_k , and searches along this direction from the current iteration θ_k for a θ_{k+1} that has a lower function value, i.e. $f(\theta_{k+1}) < f(\theta_k)$. The distance to move along p_k is decided by the step length which is usually large at the beginning and then divided it into smaller steps until it finds a new θ that loosely approximates the minimum. From there a new search is performed. There are several ways to choose the search direction and the most used methods are described in the coming sections.

Steepest descent

The most obvious choice of direction in which to search for the minimum would be in the direction where f most rapidly decreases. This approach is called the steepest descent method and it moves along the direction

$$p = -\frac{\nabla f_k}{\|\nabla f_k\|}. \quad (2.32)$$

The iteration is given by substituting Equation 2.32 into Equation 2.30.

$$\theta_{k+1} = \theta_k - \alpha_k \frac{\nabla f_k}{\|\nabla f_k\|}. \quad (2.33)$$

Newton direction

Another way of choosing the direction in which to search for the minimum is the Newton direction search. It is derived from the second-order Taylor series approximation:

$$f(\theta_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \stackrel{\text{def}}{=} m_k(p). \quad (2.34)$$

The step length is naturally 1 and if $\nabla^2 f_k$ is positive definite we obtain the Newton direction by finding the vector p that minimises $m_k(p)$. By setting the derivative of $m_k(p)$ to zero, the search direction, p_k , becomes

$$p_k = -\frac{\nabla f_k}{\|\nabla^2 f_k\|}. \quad (2.35)$$

The Newton direction search is reliable when the difference between the true function $f(\theta_k + p)$ and its quadratic model $m_k(p)$ is not too large.

The main drawback of the Newton direction search is the use of a Hessian which demands a lot of computations. The **Quasi-Newton** search does not use the Hessian. Instead it uses an approximation, b_k , which is updated after each step to take into account the additional knowledge obtained.

To obtain global convergence, we require the search direction in Equation 2.35 to be a descent direction, which is true if the Hessian $\nabla^2 f(x_k)$ is positive definite. If the Hessian is not positive definite, or close to being singular, p_k may be an ascent direction. A strategy to ensure that the step is of good quality is presented in Section 2.6.5.

Nonlinear conjugate gradient methods

The nonlinear conjugate gradient method is one of the best methods for solving large linear systems of equations and can be adapted to solve nonlinear optimisation problems. The search direction is given by:

$$p_k = -\nabla f(x_k)\beta_k p_{k-1}, \quad (2.36)$$

where β_k is a scalar which ensures that p_k and p_{k-1} are conjugate, which means that $p_k * p_{k-1} = 0$. This method is usually more effective than the steepest descent method and almost as easy to compute. It does not attain the fast convergence rates as the Newton method but, on the other hand, it does not require much memory storage either.

2.6.2 Step Length

There are a number of methods available to choose the step length and also some trade offs to be considered. On one hand, we would like to choose a step length that gives a substantial reduction of f , on the other hand, we would not like to spend too much time deciding the step length. The simplest approach is to choose α_k as a constant.

For Newton and quasi-Newton methods, the step $\alpha_0 = 1$ is the best initial choice. This choice ensures that unit step lengths are taken whenever they satisfy the termination conditions, see Section 2.6.3, and allows the rapid rate-of-convergence properties of these methods to take effect.

A method that works well in practical systems [30] is the so called **damped method** in which α_k is initialised as $\alpha_k = 1$ and then

$$f(\theta_k + \alpha_k p_k(\theta_k)) < f(\theta_k) \quad (2.37)$$

is tested. If true, then a new prediction $\theta_{k+1} = \theta_k + \alpha_k p_k(\theta_k)$ is made and the iteration repeats. If not true, then α_k is reduced by half, $\alpha_k := \alpha_k/2$, and Equation 2.37 is re-tested. This procedure continues until the test in Equation 2.37

is successful. This method can be developed further by using cubic modeling when reducing the α_k and it is then known as the 'backtracking approach'.

Regarding the steepest descent and the conjugent gradient methods, it is important to take into account the current information about the problem and the algorithm to make a initial guess [17] about the step length.

2.6.3 Termination conditions

When the function f is complex, it can be difficult or even impossible to find the exact minimum without an immense computational effort. Therefore, it is important to decide upon a number of termination conditions. It is found [30] that the iteration routine should be terminated in any of three ways:

1. If the gradient Euclidean norm, $\|\nabla f(\theta_k)\|$, falls below a set threshold, then it may be concluded that a minimum has been reached and the search is terminated.
2. If the number of iterations of updating θ_k is greater than a predefined number, the search is terminated.
3. If the cost can not be decreased, the search is terminated.

2.6.4 Local or global minimum?

Another aspect of the minimisation problem is to determine whether the minimum found is a local or global minimum. This can be done by examining the gradient, $\nabla f(\theta_*)$, and the Hessian, $\nabla^2 f(\theta_*)$. A necessary condition for a vector, θ_* , to be a *local minimum* is that $\nabla f(\theta_*) = 0$ and that $\nabla^2 f(\theta_*) > 0$. A vector θ_* is a *global minimum* of $f(\theta)$ if it gives the *lowest* value of $f(\theta)$, which means that there exists a θ_* so that

$$f(\theta_*) \leq f(\theta), \forall \theta \in \mathbf{R}^n. \quad (2.38)$$

It might seem like the only way to determine if a minimum is global or local is to examine all the points to make sure that none of them have a smaller function value. Luckily this is not the case; [17] gives a couple of solutions to the problem.

2.6.5 Hessian Modifications

Line search in the Newton direction produces the fastest possible convergence [30] to a local minimum. However, in practice, the Hessian is often ill conditioned due to two principal reasons:

1. The function $f(\theta)$ is locally linear at $\theta = \theta_k$ which means that $\nabla^2 f(\theta_k) \approx 0$.
2. A ridge of θ exists for which the function $f(\theta)$ has a constant value.

To avoid an ill conditioned Hessian, a regularisation method is needed. Still, the modification should be small in order to preserve the second order information in the Hessian as far as possible. The 'Levenberg-Marquardt' regularisation

method calculates the relative condition number¹¹ of the Hessian approximation, $\nabla^2 f(\theta_k)$, and if the condition number falls below a threshold κ , the Hessian approximation is substituted with a *regularised* version

$$\nabla^2 f(\theta_k) := \nabla^2 f(\theta_k) + \delta I. \quad (2.39)$$

Typical values are $\delta = 10^{-8}$ and $\kappa = 10^{-18}$.

¹¹The relative condition number is the ratio of the smallest to the largest eigenvalue.

Chapter 3

Software For Testing

3.1 Why a software for testing?

The software was implemented in order to simplify the required tests, both during the project and in any future research that may have use of the results of this project. Compared to, for example, creating a file system, which would be harder to have a quick glance of, the software is a practical way of presenting the work accomplished throughout this thesis. All the m-files implemented to carry out various operations on the images are gathered and connected to a graphical user interface (GUI), which is saved on a compact disk. Figure 3.1 shows how the files implemented during this project relate to each other and enables an easy check up if all the functions are obtained after a possible download of all the software from the compact disk. The graphical user interface enables an easy execution of the functions, one by one or one after the other; only a simple click on a button is needed. The result is shown in one of the windows on the left-hand side of the graphical user interface, see Figure 3.2.

After the theory behind the different fields and before presenting the performed tests and results, an introduction to the implemented software is appropriate.

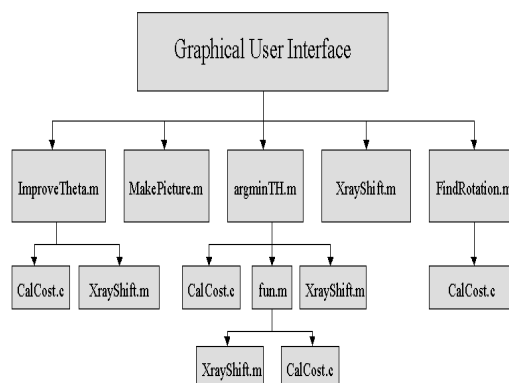


Figure 3.1: The structure of the files connected to the Graphical User Interface.

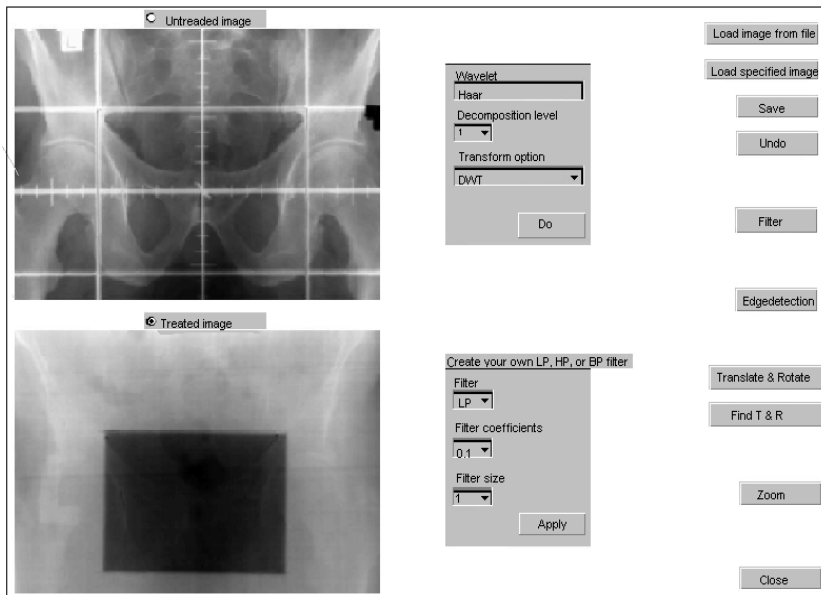


Figure 3.2: The implemented graphical interface (GUI).

3.2 Functions

The presentation of functions strives for following the same structure as in Section 2.1. General functions are first described in Section 3.2.1, followed by functions in order to create a test image in Section 3.2.2. Thereafter the three main parts of the image registration task, see Section 2.1, are presented in Sections 3.2.3-3.2.5.

3.2.1 General functions

All these functions can be found as push buttons to the very right of the graphical interface. All functions, except for “Close”, act on the marked image window on the left-hand side of the graphical user interface. The user can choose which image to process by clicking on the radio buttons, see Appendix B, located above the image windows.

Load image from file Load a bitmap image to the window indicated by the radio buttons above the image windows on the left hand side of the graphical user interface. The file system can be browsed to find the file.

Load specified image Load a special image designed for testing the algorithm to find the translation and rotation between the images.

Save Saves the chosen image as a bitmap file.

Undo Resets the image as it was before the last action was carried out.

Zoom Enables to zoom in on a particular part of the image. Choose which image to zoom with the radio buttons. When the marker turns into a

cross, mark the interesting area. The result will appear in a new resizable window.

Close Closes the graphical user interface.

3.2.2 Creating a test image

Enables translation and/or rotation of any image similar to the image expected to be obtained after edgedetection on the simulated or the portal image. Thus, it is possible to test the minimisation program that is detecting the translation and rotation, see Section 3.2.5, even if the preprocessing of the image is unsatisfying. A substitute for the portal image is created by the translation and rotation and can thereafter be compared to the original image, see Section 3.2.5.

Translate & Rotate Translates and rotates the selected image by the pixels and degrees given in the dialog box which appears when pressing the push button.

3.2.3 Preprocessing

As mentioned in Section 2.1, the portal image is very blurred and lacks contrasts. The preprocessing is an attempt to strengthen required edges of certain bonemarks and to distinguish them from other edges. One good way of getting an idea of which, out of a hugh variety of wavelets to use or to try different parameters for filters is to test how they work on the actual X-ray images.

Wavelets

The idea behind the wavelet based approach is more thouroughly explained in Section 4.1. A scroll list contains options of different wavelets such as the Haar wavelet, Symlets, a Daubechis wavelet and a biorthogonal (spline¹) wavelet. Decomposition levels from 1-8 (normally 5 is the lowest level used) can be choosen from the scroll list beneath. Either the Discrete or two versions of the Fast Wavelet Transform, see Section 2.3.2, can be used.

The wavelet functions require Matlab Wavelet Toolbox and has not been tested on the final GUI since we did not have the required toolbox at the time.

Fix filters

The push button on the left marked “Filter” uses already implemented filter routines from Matlab. The name of the Matlab m-file is written in parentheses after each filter options mentioned below. Default settings for the filters are used and no paramaters can be changed.

Filter A scroll list with different filters to choose from appears. The filter options are a medianfilter (medfilt2), a wienerfilter (wiener2), a second order filter (ordfilt2) and high, low, and bandpass filters (FIR1).

¹For more information about the three different methods of wavelet construction, refere to [25].

Adjustable filters

In the frame underneath the wavelet frame in the middle of the graphical user interface, different options for implementing filters are gathered. Choose between a low or a high pass filter, which are implemented as the matrices in Section 2.2.2. For creating a bandpass filtering, a low and a high pass filtering have to be carried out irrespectively of the order. The filter coefficient, i.e. the matrix entries, and the filter size, i.e. the size of the matrix, can also be chosen.

3.2.4 Edge detection

The edge detection is mainly used to see which pair of preprocessing and edgedetection that reveals few but required edges of useful bonemarks for the detection of the translation and/or rotation. This is to make the images used in the minimisation as simple as possible.

Edge detection A pop-up menu, see Appendix B, with different operators to use appears. Options are the following operators: Sobel's, Prewitt's, Robert's, Canny's, Laplacian, and Zero crossing. Descriptions of the different operators can be found in Section 2.4.

3.2.5 Calculating the translation and rotation

The algorithm for finding the translation and rotation is implemented and executed when pressing the push button described below. Note that a good preprocessing of the image is necessary in order for the algorithm to work. The best results is obtained when using the images found when pushing the **Load specified images** button. Use the **Translate & Rotate** button to define a translation and rotation and then push the **Find T & R** button. The result will have opposite sign to the defined translation and rotation.

Find T & R The translation and rotation between the upper and lower images in the graphical user interface is calculated when pressing the push button. The result is presented in a dialog box.

Chapter 4

Results

4.1 Wavelet Based Approach

The wavelet transform enables examination of details in the image having frequencies within a certain range separately from others with different frequency contents. On each level, a projection of the original function onto the approximating space, V_j , and another on the detail space, W_j , are to be found. As mentioned in the theory of wavelets, the approximating space is spanned by the scaling functions and the detail space by the wavelets. Approximations of a noisy test function in different scales is shown in Figure 4.1 and the details of each scale are found in Figure 4.2. Similar figures are found in [18].

Edges often contain higher frequencies than other features in the images, where the frequency depends on how steep the edge is (e.g. how big the difference in gray scale value is from one side to the other). Moreover, the edge frequencies are mostly lower than the frequencies of noise. Consequently, a reasonable guess would be to expect to find the edge frequencies among the higher of the lower frequencies in the images, that would be under the noise scale but over other bigger features of the images, i.e. in one of the lower levels of a decomposition like the one in Figure 4.2. The primary goal was to examine whether a certain scale, where most of the frequencies representing the edge of interest for the comparison between the two images, could be found. If so, it might be possible to use only the wavelet coefficients representing these frequencies, instead of the whole images, in the minimisation algorithm to find the translation and the rotation.

To be able to use this detection, the wavelet transform, and therefore the coefficients, should preferably be translation and rotation invariant. Also, the two images might have different scales, the edges in the images should preferably be reinforced, and at least the portal image might have to be noise filtered. Searching for the perfect transform became a time-consuming matter and instead, different transforms from the different wavelet toolboxes, see Section 5.2, were tested on the simulated X-ray image.

Several sources pointed towards non-orthogonal wavelets in order to solve similar problems. As orthogonality is one of the biggest advantages of the use of wavelets, they might not be what we are searching for. A frequency analysis of the images would be helpful in order to understand what could be expected from

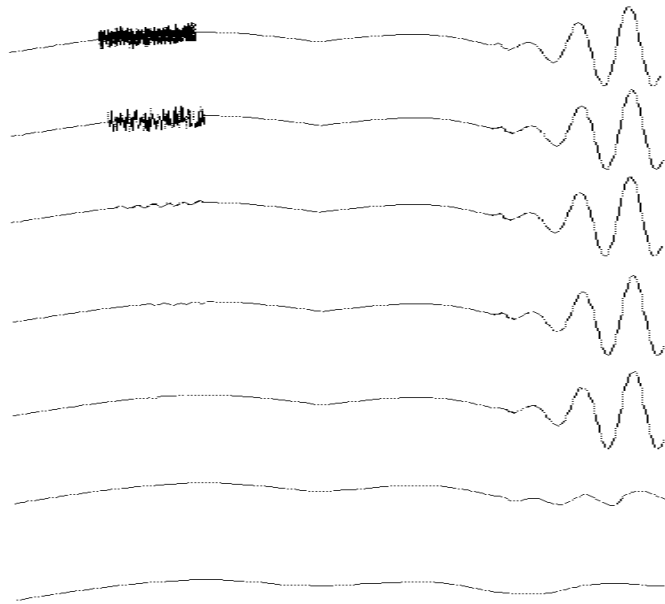


Figure 4.1: Multiresolution approximation of a noisy test function.

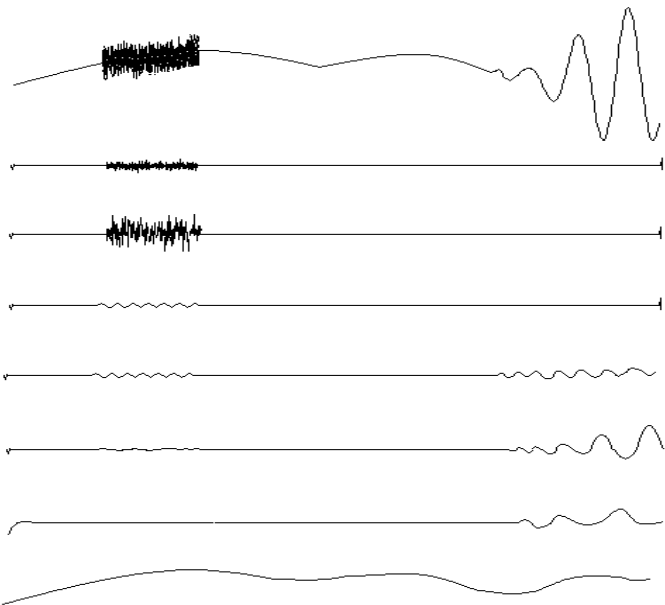


Figure 4.2: Multiresolution decomposition of a noisy test function. Details of Figure /reffig:mra approx.

the result. A wavelet transform is in fact a filter bank, yet with a logarithmic partition, and therefore the next field to try became filters and filter banks, see Section 4.2. Band pass filters could hopefully be used both for the preprocessing and for the frequency analysis of the images.

4.2 Filters

The golden rule of striving for simple solutions became clear when even the simplest of high pass filters, see Section 2.2.2, gave a better result for the following edge detection than the wavelet transforms. Both linear and non-linear filters were tested on the portal image to get rid of the pronounced edges of the treatment region in the images, i.e. the dark rectangle in the middle. In addition, an already edge detected image was filtered to hopefully pick out only the necessary edges needed for the minimisation algorithm. When detecting edges, we have been using the Canny edge detection algorithm because of its superiority over other algorithms as shown in Section 4.3.

The linear high pass and low pass filters were tested based on the advice from a number of different sources within the field of image analysis and processing. For the frequency analysis of the images, the high pass and low pass filters were combined to form band pass filters. Yet, only the simple filters described in Section 2.2.2 have been used to represent the two filters in the convolutions. Since the result of the filtering could not be seen directly on the images, an edge detection was carried out to examine the results. Edge detection is more thoroughly explained in Section /refsec:tests:edgedetection. The same edge detection algorithm is used in all Figures. The result of the linear filtering is shown in Figures 4.3 - 4.9.

4.2.1 Linear filters

By changing the size of the filters within the interval 1x1 to 6x6, the size influence of the low pass and high pass filters was examined. All sizes were tried out with normalised cut off frequencies from 0.1 to 0.9. Both the simulated and the portal images were used for the filtering.

Low pass filtering

A low pass filtering was carried out on both the simulated and the portal image, using a 3x3 filter. In Figure 4.3 the results of the edge detection after the low pass filtering on the simulated images are shown. The normalised frequency of the three different filters was 0.1, 0.5, and 0.9. The same operation, but carried out on the portal image, is shown in Figure 4.4.

High pass filtering

The same tests used for the low pass filter were carried out with the high pass filter. The results on the simulated images is shown in Figure 4.5 and on the portal images in Figure 4.6.

Having seen both results, it is clear that neither individual low pass nor high pass filtering would give a better result than an edge detection directly on the original image. The result is even clearer for the portal image than for the

simulated image. As edges have mostly high frequencies, a high pass filter should be more appropriate for this application. The tests confirm this theory, even if the result of high pass filtering is not overwhelming. A comparison between edge detection after 3x3 high pass and low pass filtering, both with filter coefficient 0.5, is shown in Figure 4.7. The influence of the filter size is shown in Figure 4.8. As seen, there is not a big difference due to choice of filter size. Yet, for filter coefficients larger than and equal to 0.5, we noticed some blurring effects. Also, black areas appeared on the borders of the images, probably because the filter was taking into account too many pixels from the outside of the actual image matrix, when filtering pixels closer to the borders.

The conclusion drawn after these tests is that a band pass filter would better meet our expectations of filtering out certain edges and that the size of the filters is of less importance than the filter coefficients, i.e. the cut off frequency. From now on, primarily filters of size 3x3 will be used.

Band pass filtering

A band pass filter was implemented by using a low pass filter followed by a high pass filter. The size of the pass band, between the cut off frequencies of the two filters, was changed from 0.4 to 0.1 in normalised frequency. Thereafter, the pass band was drawn over the image to cover all different intervals. When performing the tests, both the simulated and the portal images were used. However, we have chosen to only include the results from the band pass filtering of the portal image with the interval 0.3 between the two cut off frequencies. In Figure 4.9 the result of the band pass filtering of the portal image is shown. We find the result a bit surprising and disappointing, as we expected the filter to better filter out frequencies in the interval specified by the band pass filter. The expected outcome would be images where different edges were detected in different images. Yet, this is not the case and, perhaps, a more sophisticated filter is needed. These results encouraged us to investigate non-linear filters.

4.2.2 Nonlinear filters

According to the theory, a non-linear filter should give better result on multiplicative noise in images, shape recognition, edge detection etc. As multiplicative noise appears on “recorded” images, such as the X-ray images in this case, non-linear filters should give better result than the linear filters presented in the earlier sections of this chapter.

Median filtering

From the beginning the size of the mask, determining the number of neighborhood pixels taken into account, was held constant on 3x3. Various shapes of the mask were tested to see if a difference in the edge detection could be seen. Different shapes of the mask were for instance a 'X', a cross, a star and a square. The patterns can be seen in Figure 4.10.

Better results than with the linear filters were obtained accordingly to the theory. Still, too few tests were performed to prove the advantages of median filtering over other filters.

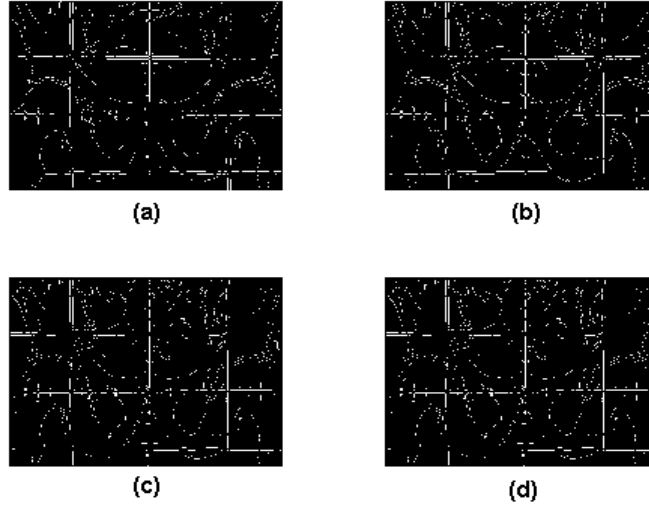


Figure 4.3: (a) Edge detection on the simulated image, (b)-(d) edge detection after low pass filtering, 3×3 . Filter coefficients are 0.1, 0.5, and 0.9 respectively.

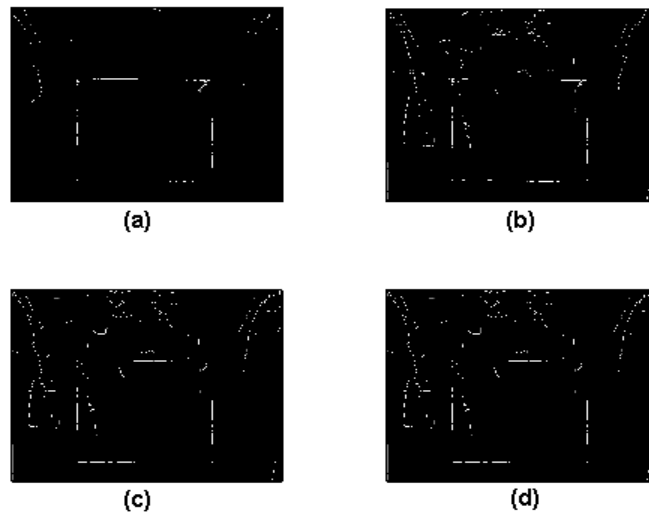


Figure 4.4: (a) Edge detection on the portal image, (b)-(d) edge detection after low pass filtering, 3×3 . Filter coefficients are 0.1, 0.5, and 0.9 respectively.

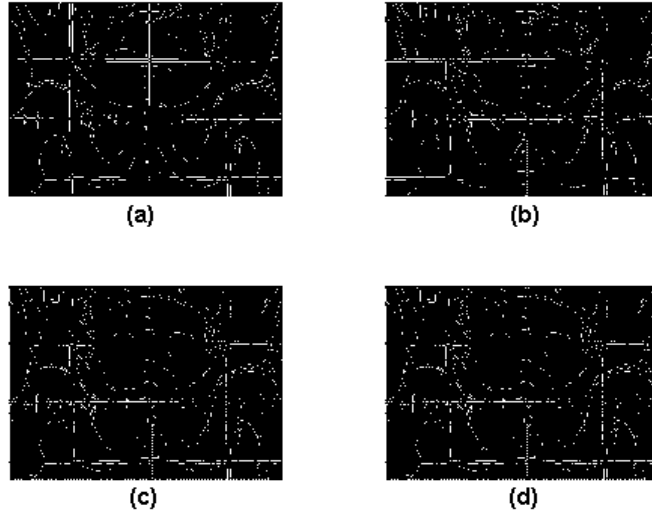


Figure 4.5: (a) Edge detection on the simulated image, (b)-(d) edge detection after high pass filtering, 3x3. Filter coefficients are 0.1, 0.5, and 0.9 respectively.

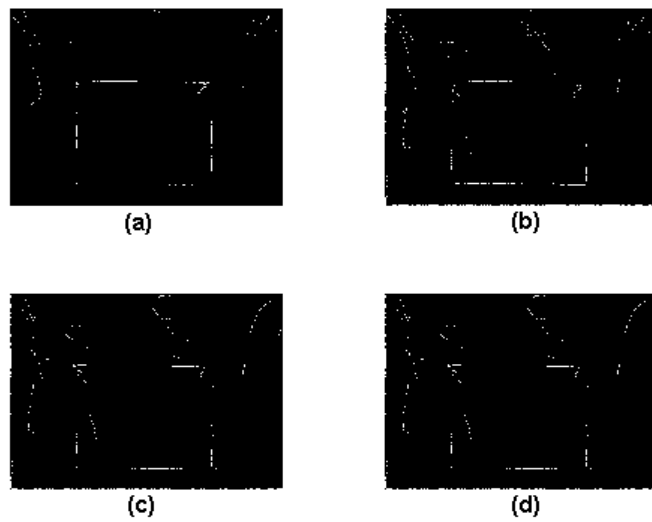


Figure 4.6: (a) Edge detection on the portal image, (b)-(d) edge detection after high pass filtering, 3x3. Filter coefficients are 0.1, 0.5, and 0.9 respectively.

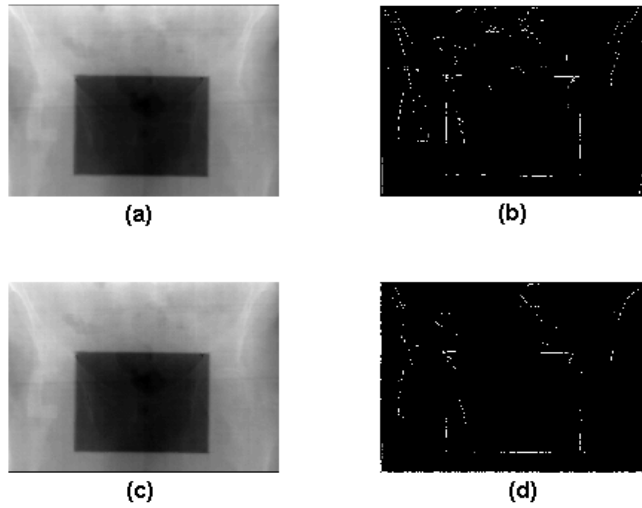


Figure 4.7: Comparison between high pass and low pass filtering with filter coefficient 0.5 and filter size 3x3. (a) Low pass filtered image, (b) edge detection on (a), (c) high pass filtered image, (d) edge detection on (c).



Figure 4.8: Filter size comparison. (a) 3x3, (b) 6x6.

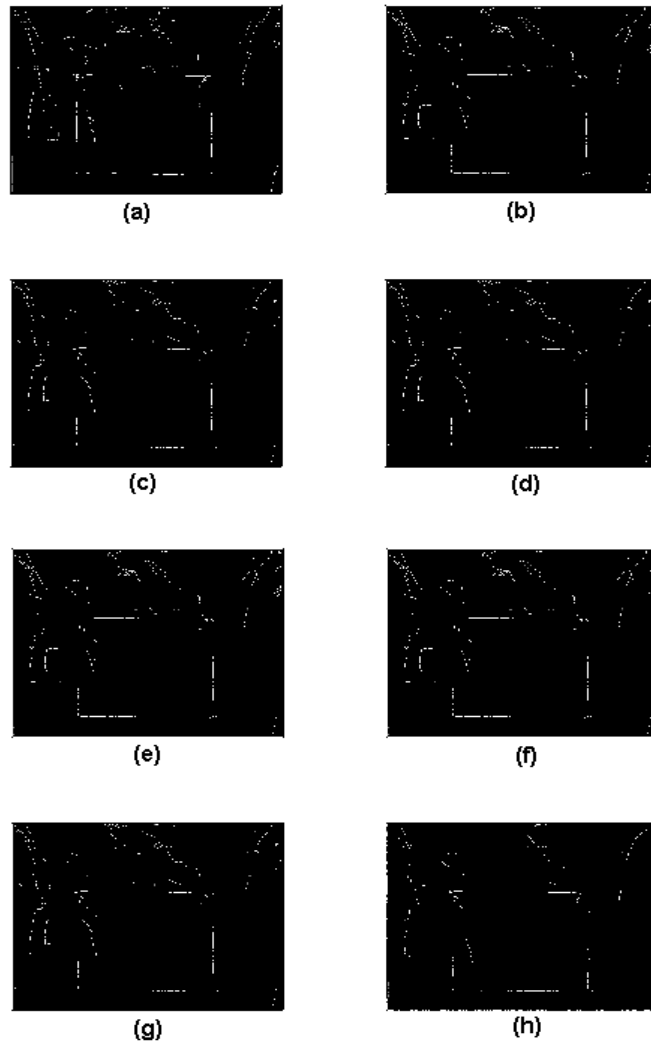


Figure 4.9: Bandpass filtering with passband 0.3 on the portal image. (a) 0.0-0.3, (b) 0.1-0.4, (c) 0.2-0.5, (d) 0.3-0.6, (e) 0.4-0.7, (f) 0.5-0.8, (g) 0.6-0.9, (h) 0.7-1.0 in normalised frequency.

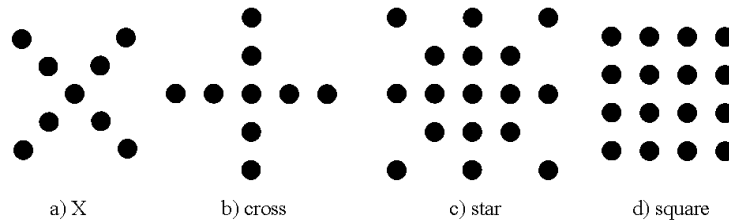


Figure 4.10: The different shapes of the mask used in the median filtering.

Homomorphic and morphological filtering

The edges seem to preserve too little of the structure to be used in the minimisation algorithm and are also sometimes too fragile. A possible idea would be to search for segments of the images instead of only edges. This is how mathematical morphology works on images; hence, morphological filters are perhaps a likely solution to the preprocessing task.

No documented tests of these two families of non-linear filters were carried out. Still, their existence is well worth being mentioned once more.

4.3 Edge detection

The filtering of the image is performed with the purpose of detecting frequencies which contain the important edges. If this approach is successful, it will most likely help the edge detection algorithm to detect the preferred edges. To find the best edge detection method, a number of tests have been performed on the two test images, i.e. the simulated and the portal X-ray. The results are shown in the following sections and the theory behind each edge detection method is presented in Section 2.4.

4.3.1 Sobel's method

Because of its simplicity, the Sobel method does not demand much computation power. On the other hand, it is often too simple to handle images distorted by noise. When applying the Sobel mask to the simulated and the portal images, see Figure 4.11, hardly any edges from the interesting bone structure are detected. The grid pattern is the only feature that the Sobel method can detect. It is clear that the Sobel method is too simple to be used on our images.

4.3.2 Robert's method

Robert's method uses Robert's approximation of the derivative and return edges where the gradient is maximum. The results on the simulated and portal images are shown in Figure 4.12. As in the Sobel case, only the grid features are present and nothing from the bone structure is detected.



Figure 4.11: The Sobel mask applied to the simulated (a) and the portal (b) images.

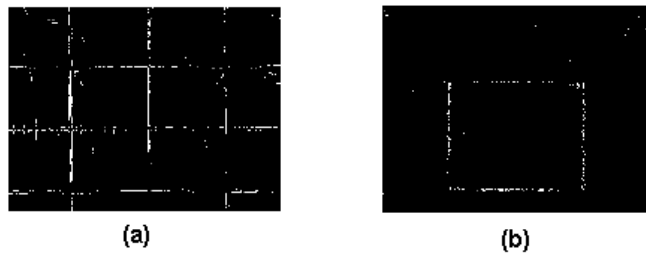


Figure 4.12: The simulated (a) and the portal (b) images with Robert's method.

4.3.3 Laplacian of Gaussian method

The result of the simulated and portal images after applying the Laplacian of Gaussian method is shown in Figure 4.13. The method fails to detect any bone structure and has a more of distortion than the images where Sobel's and Robert's methods were used. The Laplacian of Gaussian method by itself is not good enough for our special application.

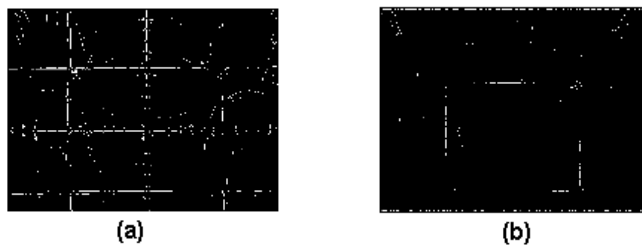


Figure 4.13: The simulated (a) and the portal (b) images with the Laplacian of Gaussian method.



Figure 4.14: The simulated (a) and the portable (b) images with the Canny edge detector method.

4.3.4 The Canny Edge Detector

The Canny edge detector is a more sophisticated method, using more complex calculations and therefore the computational cost is increased. Even so, it is the only method using derivatives so far that has been able to detect something but the grid features. In Figure 4.14 the results are shown and some of the bone structure is visible. Still, the result is not good enough for our application and additional treatment of the images before applying the edge detector method is going to be necessary.

4.3.5 The Spatially Noise Filtration Technique

Unlike all the other edge detection methods we have been investigating, the spatially noise filtration technique uses the wavelet transform to produce a mask that is used to detect edges in an image. The theory and algorithm is described in Section 2.4.6. In order to test the method, we implemented the algorithm in Matlab.

As test images, we used the simulated and portal X-rays, on which a wavelet transform was applied, using the Haar wavelet as a mother wavelet. As seen in Figure 4.15, the correlations between adjacent scales detects quite well some edges in the simulated image when using the correlation between scales one and two. However, as soon as we use lower scales, the resulting correlation image becomes useless. The portal image is even worse, no correlation can be detected even when using scales one and two.

The correlation images are then used to produce a mask that will filter the original images, Figure 4.16. Since the calculation of the masks relies on good correlation images, the resulting masks also become very distorted in all cases but the first. It is only when calculating the mask for the simulated image, using the first and the second wavelet scales, that some patterns from the original image could be detected. The remaining three images show no interesting features.

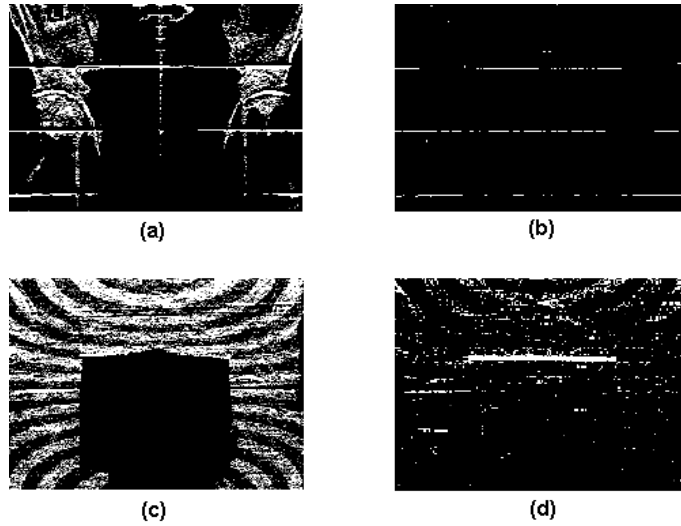


Figure 4.15: (a) The correlation between scales one and two using the simulated X-ray. (b) The correlation between scales two and three using the simulated X-ray. (c) The correlation between scales one and two using the portal X-ray. (d) The correlation between scales two and three using the portal X-ray.

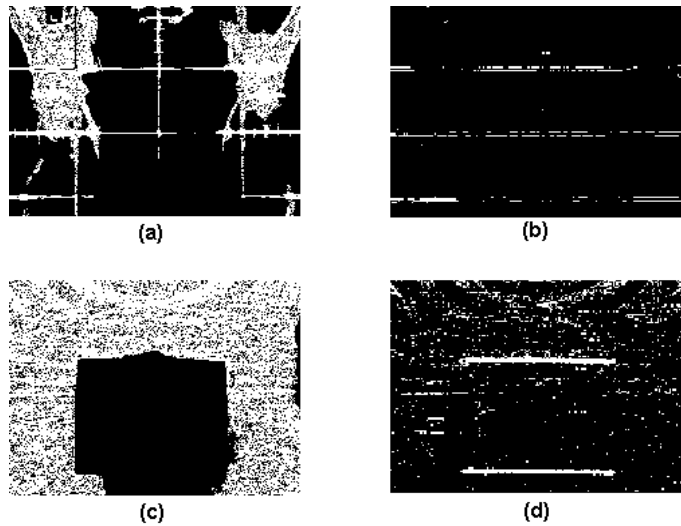


Figure 4.16: The masks for the simulated and portal images.

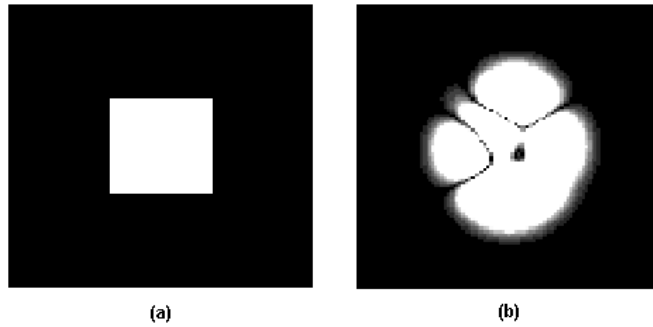


Figure 4.17: The two images used to test the minimisation algorithm.

4.4 Minimisation of the cost function

The image is filtered and an edge detection algorithm is applied to the image in order to enhance the features that will help us determine the true translation and rotation between the portal and the simulated image. Calculating and trying to minimise the cost function is the third and final block in our chain, see Section 2.1.

The theory behind the minimisation algorithm can be found in Section 2.6. The implemented program first searches in a *Damped Newton direction*. If a decrease in cost is not possible, the program switches to a *Gradient line search*. In addition, if the Hessian is ill-conditioned, Levenberg-Marquardt regularisation is added. The program returns a rough answer and which is tested to see if additional improvements can be made.

Since the real images are complex and difficult to work with, we have produced simpler images in order to test the minimisation program. The pictures used are shown in Figure 4.17 (a) and (b). The image is rotated and translated and is given as an input to the minimisation program together with the original and unaltered image. The translation and rotation is later compared with the minimisation program result. These images are quite different from the real images this project deals with, nevertheless, they are important as a test to make sure that the implemented algorithm is working as intended.

Several trials have carried out using these images and the minimisation algorithm works well for a number of translations and rotations. Table 4.1 and Table 4.2 show some input and the corresponding output when testing the program on the images in Figure 4.17.

As Table 4.1 and 4.2 imply, the program returns the translation and rotation which need to be performed in order to make the two images identical again. When developing the software further, considerations about the computational cost need to be taken into account since the real pictures often are quite large.

Input	Output
$\theta = 2, 2, 2$	$\theta = -2, -2, -2$
$\theta = 0, 6, 8$	$\theta = 0, -6, 8$
$\theta = -6, 0, -7$	$\theta = 6, 0, 7$
$\theta = 4, -7, 5$	$\theta = -4, 7, 5$

Table 4.1: Test results from the minimisation program using Figure 4.17 (a). The output is the action to perform to align the images.

Input	Output
$\theta = 6, 6, 6$	$\theta = -6, -6, -6$
$\theta = -10, 5, -4$	$\theta = 10, -5, 4$
$\theta = -3, 10, -9$	$\theta = 3, -10, 9$
$\theta = -7, -4, -1$	$\theta = 7, 4, 1$

Table 4.2: Test results from the minimisation program using Figure 4.17 (b). The output is the action to perform to align the images.

Chapter 5

Used software

In this chapter, the main software used throughout the project are presented. Needless to say, the Internet has been an almost irreplaceable source of information. All the software has been licensed for the Department of Electrical and Computer Engineering or found as free software on the internet (or it has just been used under a free trail period of time).

5.1 EXcursion V2

EXcursion V2 is a XRemote Technology provided by NCD Inc. It is a X11R6-based, 32-bit X Windows System Server. Digitally developed, the eXcursion family of display server products provide interoperability between desktop personal computers (PCs) running the Microsoft Windows operating system and remote hosts running the X Window System operating system under the UNIX or OpenVMS operating systems. Through an xterm window, the unix environment becomes as easy to use as all other software on the PC. For more information about eXcursion, refer to [34].

5.2 Matlab 5.3

Matlab 5.3 from The MathWorks, Inc., is an easy-to-use computing environment, that simplifies the calculation of various numerical problems. Additional toolboxes, in for example signal processing, optimisation, wavelets and images, increase the functionality and enable the use of Matlab also for more field specific problems. Neither the wavelet, nor the image toolboxes were accessible in the PC environment and there was only one license for the image toolbox under unix. The wavelet and image toolboxes were therefore only used under the 30 days of free trail period obtained when downloading the toolboxes from the Internet. Substituting toolboxes, see Section 5.2, were also found on the Internet.

An introduction on the implementation of graphical interfaces in Matlab can be found in Appendix B.

Additional Toolboxes

As no license for the Matlab Wavelet Toolbox was available, similar toolboxes had to be found on the Internet. All three of them are free software. The descriptions and more information can be found on [35].

Wavekit The Wavekit-toolbox is a collection of functions for Matlab that implement the following wavelets and wavelet packet algorithms: One- and two-dimensional (periodic) fast wavelet and wavelet packet transforms and the best basis algorithm for wavelet packets. An implementation of the fast matrix multiplication algorithm of Beylkin, Coifman, and Rokhlin for both wavelets and wavelet packets. Various demonstrations on visualizing wavelets, signal analysis, and the multiplication algorithm. The author of the toolbox is Harri Ojanen.

WaveLab802 Library of Matlab routines for wavelet analysis, wavelet-packet analysis, cosine-packet analysis and matching pursuit. The WaveLab library is implemented at the Stanford University, USA.

Uniwave3 Advanced toolbox from the University of Vigo. Among its functions are: Discrete Wavelet Transform, Scale Function and Wavelet Function, Multiresolution analysis.

Mex-files

It is possible to call C or Fortran subroutines from Matlab as if they were normal Matlab functions. These files are referred to as Mex-files and they are dynamically linked subroutines that the Matlab C/Fortran compiler can load and execute. On the PC, the Borland C++ Builder Compiler has been used. This compiler can be downloaded for free from [36].

The advantage with Mex-files is that highly computational parts of a program, usually for-loops, are computed much more efficiently if they are written in C++ or Fortran than in Matlab code. The cost computation would have been too slow to use, if it had not been for Mex-files. The Mex-files also makes it possible to use large, already written Fortran and C programs without rewriting them as m-files. Only Mex-files written in C have been used in the project. The source code of a C Mex-file consists of two parts, the computational routine and the gateway routine. The former routine contains the C code for performing the computations that you want to implement in the Mex-file, the latter is the interface between the computational routine and Matlab. Input and output arguments for the computations are defined and space is allocated in memory. The gateway routine calls the computational routine as a subroutine.

For a more complete introduction to Fortran and C Mex-files and also to the rest of the Matlab Application Program Interface, API, refer to [29]. The application program interface does not only support Mex-files, but also *Mat-files*, providing the possibility to import and export data to and from the Matlab environment, and a set of routines that allows you to call Matlab from your own programs, only using Matlab as a computation engine. These routines are called engine programs or *engine applications* and they are written in C or Fortran.

5.3 L^AT_EX V2

L^AT_EX V2 is a popular software system, especially for the writing of technical reports, books, articles etc. L^AT_EX is more like a programming language than

other software such as Microsoft Words; it uses an editor, for example emacs (unix) or WinEDT (PC) for the writing, commands for creating features such as titles, tables, figures and changings in font, and also a \LaTeX compiler in order to check the document for errors and to convert the text document to a more readable format. \LaTeX makes it easier to concentrate more on the text during the writing and leave the layout for later.

Xdvi

Xdvi is used by \LaTeX to show how the actual document looks like. Even though we had some problems with the setup and only had light green text on a dark green background, Xdvi was a great help to get an impression of how the document would look like. The text in the document is unfortunately a little bit too small in Xdvi to actually be read.

WinEDT and MiKTeX

The PC version of \LaTeX is called MiKTeX and it uses WinEDT as its editor. MiKTeX is a very easy-to-use graphical interface, that writes much of the code for you. For this project the \LaTeX under unix has been used for the writing, but WinEDT (MiKTeX) has been used for spelling.

5.4 Miscellaneous software

Most of the figures in the report are made in **Microsoft Power Point 97** and thereafter cut to proper formats in **Paint** on the PC. Afterwards, the images were transferred to the unix system using ftp and converted to postscript format in **Xv**.

Part II
Analysis

Chapter 6

Analysis of the research project

6.1 Introduction

Adapting yourself to new surroundings and to the way local people live when going abroad, is a well known goal. Yet, it can be harder than expected to actually reach this goal. Expectations and demands on similar tasks can be completely different. How could you, for example, even expect that something as defined as an eight hour working day could be considered in various ways? Inviting beaches, a shining sun, friendly people and great surroundings, all lovely things, but they sometimes do seem to be against your determined will to keep up with your own “standards” and expectations from back home. Different university degree systems also make it hard to explain to people around you what you actually try to achieve with your work. Consequently, it is hard for them to know how to classify you in their system. A shared working space shows how different attitudes to “work” can be, new friendships and acquaintances lighten up your day but also demand a lot of your energy¹. Without a car, in a country in which having a car is a standard, makes the smallest tasks very time consuming. Indeed, everything becomes a smaller excursion, even buying milk.

However, this is in a way what you are actually searching for, when choosing to go abroad. The chance to set up a new life and do something different for a shorter period of time: to try something new. But how do you deal with such a situation, once there? How do you make it effect your work as little as possible?

As always, communication and open discussions have helped us to avoid many misunderstandings and taught us a lot about ourselves. This period of time has also been a giving experience and a great opportunity to learn more about different mentalities, nationalities, and personalities.

In this chapter, we analyse some of the situations that you may encounter when going abroad to do your Master thesis based on our own experiences in Australia. Also, our general ideas, subjects of discussions, and thoughts about our personal cooperation as well as the project and the developed software in general, are presented in this section. Our goal is to obtain or to increase the understanding of how important the dynamic of a group is in order to work

¹Especially with the dry and ironic Australian humour.



Figure 6.1: Australia.

together irrespective of the number of people in it. One thing is for sure: the technical aspect of the problem is not the only challenge.

6.2 Methods

When doing your final thesis, the most important challenge and motivation is the amount of freedom you are given in order to solve a problem or task the best way you can. It is easy to lose sight of the overall goal and put time and energy on parts that will not contribute to the final result. As this is the case, a method for how the work should be performed and how goals should be met is necessary to develop. We have not followed any particular theory when developing a working method; instead, we have had discussions about how we would like to work, and used the outcome of the discussions to identify and decide on some key aspects.

6.2.1 Internal dividends of work

To get the widest possible knowledge about the theory behind image processing and to perform all the tests needed, the work had to be divided between the two of us. This was done by first giving both of us the opportunity to learn a little about every related topic and then decide on which subjects to focus on. The idea was that, as the project evolved, the dividends of work would come naturally. This worked out, since we have quite different interests. Should we both had had interest in the same issues, we would have solved it through discussions.

6.2.2 Learning from each other

As we gained knowledge about different theories and test results, the need for educating each other increased. The updating was done on a regular basis so that information and knowledge were not forgotten or lost. The education did not aim at teaching the other person everything about the subject, but rather to give a brief introduction, to enable discussions of its application to our project.

We make a great team

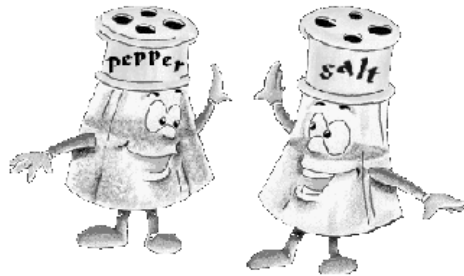


Figure 6.2: The importance of teamwork.

6.2.3 Meetings with the supervisor

The meetings with our supervisor were regarded as a very important part of the project. The aim was to have meetings every two weeks to discuss what had been done, what needed to be done and also encouraging a discussion about different theories and approaches to the problem. E-mails were used for smaller questions and also for the communication with our supervisor in Sweden.

6.2.4 Setting up goals on the way

Achieving the main goal in one step is both hard and undesirable, as you lose sight on what is important. One of our main ambitions was to have relevant and realistic smaller goals throughout the project to ensure that the project developed as it had been intended. Goals such as finishing parts of the report, having early versions of software running, and reading about different theories were set up and later checked to know whether they had been achieved in time. If not, an analysis was performed to find out why the goal had not been met and what effect the delay would have on the overall project.

6.2.5 Who had the outermost responsibility?

We had shared responsibility for the quality and the outcome of the project. The shared responsibility put high demands on communication within the group to make sure that no tasks fell between chairs. This was accomplished by making sure that everyone was allowed to have their opinion and that the working environment encouraged discussion about problems that might have disturbed the communication opportunities. No regular meeting were set up between us, instead, we had a discussion whenever we needed it.

6.3 SWOT-analysis

The SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis is a useful tool for the evaluation of any collaboration or project. It requires an internal survey of strengths and weaknesses and an external survey of threats and

opportunities. The internal survey of this analysis covers personal backgrounds and individual preparedness for the work, whereas the external looks more into the environment at work.

SWOT analysis can be performed individually or in groups. Group techniques are particularly efficient in providing structure, objectivity, clarity, and focus to discussions, which might otherwise be strongly influenced by personality. This section starts with a SWOT-analysis of the image registration software and thereafter continues with an analysis of the project itself.

6.3.1 SWOT-analysis of the image registration software

Making a SWOT-analysis of the developed product, helps you to collect your thoughts and to present the advantages and the disadvantages for yourself and others in a very predictable manner. It might even help you to get some new ideas about how to improve the product. Many factors can, and should, be placed in several of the four categories (strengths, weaknesses, opportunities, and threats).

Strengths

By decreasing the influence of the human factor, the image registration software will improve the accuracy of the treatment. Thus, neither a doctor having a bad day nor his former experiences in the field, will affect the treatment due to differences in decision making. Also, the software will simplify the gathering of statistical data and enables different statistical analysis, which might have been hard to perform before. Finally, the software does not require extreme computer power and runs on standard operative systems.

Weaknesses

Unfortunately, no strengths come without weaknesses. Should we really strive for computerising everything and totally neglect the importance of the human intuition and the personal contact between doctor and patient? As we are still treating human beings, the latter is an essential question in many cases. Hopefully, this test product will develop with time and be a more flexible tool. Unfortunately, 20 weeks of time is far too short to develop more than a test version of the final software.

Opportunities

This project is the result of a cooperation between the Department of Electrical and Computer Engineering at the University of Newcastle and the Mater Hospital in Newcastle. As the field of image analysis surely can be used for various applications within medical technology, we hope to see this project only as the start of an increasing collaboration from both sides in the future. Moreover, it could strengthen the links between the Department of Electrical and Computer Engineering (The University of Newcastle) in Australia and Lund's Institute of Technology (The University of Lund) in Sweden.

Threats

The problem of comparing images of different quality, size, and even origin have long caused many difficulties. Many different angles of approach have been tried, with different results. Each case seems to have its own optimal solution, depending on various factors. Thus, an extensive study of different fields, many of which for us were completely new, was necessary. Different theories were continuously tested, and the result of the tests pointed us towards new suggestions of solutions to our problem. Little contact with the hospital in general, rather few meetings with our supervisor at the University of Newcastle, and the fact that the project is on the very border of the field of our Swedish supervisor at Lund's Institute of Technology, may have increased the hours of self studies and therefore decreased the results obtained in the given time frame. It also made it easier to occasionally get on to the wrong track.

6.3.2 SWOT-analysis of the project

Unlike the more objective analysis of the software in the earlier section, this investigation strives for treating our individual backgrounds, different personalities, and their effect on the project and indirectly on the software itself.

Strengths

With different alignments inside the vast field of electrical engineering, working together had the advantage of having greater basic skills and also different ways of looking and understanding new things thanks to former experiences. Moreover, a big motivation for making the best project possible in order to finish the Master of Science degree in time and in a good manner, our goal consciousness, and persistency often helped us to keep up the motivation in harder periods and to still keep on developing the project. Furthermore, good communication between the two of us helped unraveling small problems occurring in the every day life and preventing them from getting more complicated than necessary.

A similar attitude towards the problem and very competent, yet sparse, tutoring from our supervisor at the University of Newcastle were two very important strengths of this project. Also the habit of learning new things, received from our home university, has been a valuable support.

Weaknesses

To go abroad is a very instructive personal experience, but perhaps not the most efficient way of doing a 20-weeks-project. Personal concerns such as finding out about transportation, food stores, and bank accounts easily dominates the first weeks and a different social order keeps on giving you surprises and compete about your precious and limited time. As mentioned in the introduction, many new and interesting things tend to draw your attention from work towards leisure.

Besides, the surroundings are not the only new thing. You work very close to another person and the process of getting to understand each other is a time consuming but important matter. To clear the air is sometimes a necessity.

Everything sums up by the words “lack of time”. In addition, it was our first small research project and therefore a completely new situation for the both of us.

Opportunities

Thanks to this experience of working in Australia and earlier similar experiences of other countries, we hope to have increased the possibilities of getting a work abroad, for a longer or shorter period in the future. Furthermore, it will hopefully help us to better understand possible foreign customers. It has been an excellent chance to learn the language and to get to know the Australian culture. Also new experiences of cooperation, structuring of your work, independent working, and handling of computers, might hopefully lead to a more interesting work, involving greater responsibility.

Threats

Fairly poor computer access in general resulted in the sharing of the workspace with persons having rather different attitudes to our and their work. Sometimes this put us all in disagreeable situations. Also temperatures over 30° Celsius without air conditioning, dry air, and poor lighting decreased the efficiency. To have the complete responsibility for your work and your result over such a long period was another sometimes hard, but yet very instructive experience.

6.4 Cost Savings and Quality Assurance

The SWOT analysis shows a number of possible improvements that can be achieved when implementing the image registration software. The main goal of the software is to improve treatment accuracy. However, it is interesting to investigate conceivable improvements that can be measured in economical terms.

Time savings Today, the image registration is done by the doctor at typically seven occasions. Each time, the doctor spends around three minutes with the image registration. With the image registration software, the time saving would be approximately half of this.

Farming out the image registration Another possible economical benefit could be to let less qualified staff handle the image registration. The obvious cost saving would be salary expenses.

Another important aspect is the quality assurance and as we have understood it, there is no such thing today. We highly recommend the implementation of a quality assurance scheme in order to evaluate and prove the improvements gained by the image registration software.

6.5 Time plan

Time is always something you cannot get enough of. No matter how hard you try planning your time you often end up with a heavy amount of workload at

the end. Nevertheless, a carefully considered time plan can make the workload bearable and it becomes a vital tool when planning your work. Timeplanning is one of the best ways to determine if the project is going to reach its goals on time and should give an early warning if there are any problems. A time plan is not a static document; things happen under the way and affect the project. Therefore, the time plan also has to be changed. Yet, it is always important to consider why a time plan needs to be altered, for instance, has something happened or did we fail to give an accurate time estimation from the beginning?

At the start of the project, a timeplan was outlined and it is enclosed in Appendix A. Different tasks were defined and described and have served as a guide throughout the project. In the following section, a brief description of the different tasks is presented as we defined them in the beginning of the project.

Initial literature studies An important part of the project, which should be allowed to take time, is the search for and the study of literature within the area. This initial literature study serves to give an introduction to the area of image processing and to give us different ideas on how to approach the problem.

Testing of theory and implementation of tests When the initial literature studies have been performed, the next step is to test the theory on real applications. This is done by planning and implementing algorithms, mainly in Matlab. Thereafter, a number of tests are performed to make sure the algorithms are implemented accurately and that the theory is applicable to the reality, which is often not the case.

Writing the theoretical parts of the report After a severe amount of studies, the importance of writing it all down and organising the knowledge is increasing. This time is set aside at an early stage in order to make the final writing of the report easier and also giving us a chance to review the theory, with the results of the tests in mind, before continuing the development of software.

Economical aspects The inclusion of the economical aspects in the final report have two reasons. First, it is important to consider a project from more than the technical point of view and we feel that a good technical solution cannot replace the possibility that there is no need for the product at all. Second, is one of the team members alignment towards Production Management and the need to include the economical aspects in order to fulfill her undertaking as a student specialised in Production Management.

Further development of software When writing the theoretical parts of the report, a number of ideas are surely raised on how to improve the algorithms and also some new theories might be found which would be interesting to implement and test. This part of the project is devoted to combining the existing software with the new theories to improve the algorithms.

Optimisation of software In this part, it is our goal to have working software, and then make it as efficient as possible. In the case were the different parts of the software is not working together, see Section 2.1, the aim is to optimise each individual block. Time consuming calculation loops in the

software were implemented in C and included in Mex-files to be recognised by Matlab. This significantly increased the speed of the calculations.

Test of software As a quality assurance, testing the software is an important part of the project. During this period, the software is tested on real images and the aim is to make it as user-friendly as possible with no errors and bugs.

Finishing the report and preparing the presentation The report is your chance to show people the results and the work we have been doing for the last 20 weeks and it is a compulsory part of the thesis. We also feel that it is an important part of the project since communicating your results is helpful for further research within the area.

Reviewing another project As a part of our final year project, the reviewing of another project has to be undertaken. This is done in Australia to simplify matters and also give us a opportunity to read an English written report.

Learning L^AT_EX The best tool we could find to write a report was L^AT_EX and since none of us ever used it before, some time had to be set aside to learn the basic features.

Presentation Communicating your results is an important part of the project and this is our chance to show the outcome of the project to interested people. Moreover, it is an compulsory part of the final year project at Lund Institute of Technology and we hope it will be a successful ending to our project.

6.6 Discussion

Moving to a foreign country, even if it just for 6 months, is exiting and provides you with new experiences and knowledge about yourself and the world around you. However, is also something that demands a lot of energy and commitment; finding a place to live and the way to university and the food store are just some obstacles that you might encounter at the beginning. Some may sound easy to solve, but without a car in a country prepared for cars, every task become an adventure itself. This chapter is written to bring some, for us, important aspects of the working environment to the surface. It is not supported by any theory; it is just based on our own thoughts about the matter.

6.6.1 Communication

Why should such a natural thing like communication cause any problems? It is not the communication itself that is a source of problem, it is the *lack of* communication that can destroy every opportunity to successful results. Even though good communication is something vital, it is not always something easy to achieve. Nevertheless, you can do your best to encourage good communication and that will in many cases be enough for a healthy, openminded working environment. The fact that your fellow-worker is also your friend, with whom you spend your free time, could be considered a disadvantage as things tend

to be personal instead of work related. However, we have chosen to see this as an advantage and regard it as a possibility to improve the communication even more. One of the most important aspects of communication is *listening* to each other which may sound like an easy task, and it is, if you just think about it.

In order to make sure that no one is feeling overlooked, regular conversations about the projects development and our contributions are important. However, no special time is set aside for this. Instead, whenever a problem occurs, you bring it up to discussion. Since we both are outgoing persons, ready to speak our minds, the chance of a problem not being discussed is minimal. Furthermore, good knowledge of each other personalities is a good help in order to understand each other and solve problems.

6.6.2 The importance of well defined goals

Even though the honours thesis is not a particular large project, it is still large enough for losing sight of the over all goal. To only work towards one main goal is not something that motivates you to perform your best, you easily get lost along the way. A more realistic approach is to set up smaller goals and achieve them one at the time, which is something we have been trying to do. It has not always been easy, and often we have been feeling that we have been tumbling around in the dark. Sitting down, and trying to analyse the situation has helped, even though it also sometimes has taken a lot of time and effort to do so. The problem, as we see it, has been to determine realistic goals. In addition, time estimation is a difficult skill to learn. Tasks that are planned to take a day can easily end up taking weeks.

6.6.3 Working in Australia

As mentioned in the introduction to this chapter, working in another country can be far from easy. It is easy to believe that, just because Australians, who usually call themselves Aussies, speak English, they would be English. It is not even true that they resemble the Americans and mentioning anything about possible similarities to an Australian would often be taken as a great offence. Business men coming to Australia should be very aware of the fact that Australians are not Europeans in order to be successful [23]. The main thing we have found is that as soon as the temperature reaches 30° Celsius, work is forgotten and people head for the beach and light their BBQ's. If the temperature is raising above 40° Celsius, you would be considered a madman if working.

It is an laid-back attitude and people do not seem to be stressed over little things. During this autumn, there have been several articles in the Swedish media regarding people being ill or even dying due to heavy workload. Those kind of articles is seldom encountered here, and the reason for this, is probably because of the different attitude towards work. Stress at your working place does not seem to be as common here as in Europe.

6.6.4 Physical working environment

It became quite clear already from the beginning, that this project were going to evolve towards research rather than building something. When doing research or studying, we found it very important to have a calm and relaxing working

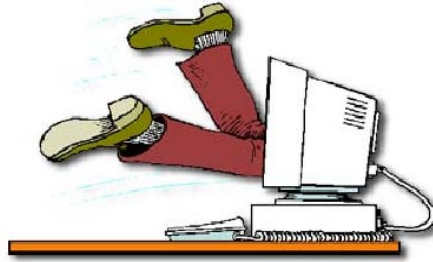


Figure 6.3: Life with a computer.

environment, inspiring you to learning. For a number of reasons, this could not be arranged, and this fact has sometimes put a limit on our ability to study. We think that feeling comfortable in your working environment is of great importance in order to achieve good results, and we wish the situation regarding our working environment could have been different.

Another aspect on working environment is the never ending hassle with computers, which we are sure everybody have experienced. When hours becomes days and you do not seem to be any closer to a solution it is easy to give up. The technical staff at the Electrical and Computer Engineering department, Newcastle, have been a great help, trying to solve all the problems, and without them we would still be sitting there.

Moreover, we found that music is an invaluable source of inspiration and has helped us a lot when trying to focus. However, a shared working space sets limits on your possibilities to play music. It sometimes becomes difficult to concentrate, when two or three different songs are playing at the same time in the same room.

6.7 Conclusions

As stated in the introduction, this chapter has the purpose of bringing a different aspect to the project, apart from the technical part dealt with in the previous chapters. We hope we have given the reader an understanding of how our work have been performed and also our opinions on the project as a whole. As the SWOT-analyse shows, the project has a great potential if fully developed. Better and more accurate treatments, cost savings, and patient security are just some benefits. We will probably not reach as far as to have a complete software for the image registration problem running. This is a due to a number of reasons; this project covers an immense area from the field of numerical optimisation to a number of different image processing techniques. Nevertheless, we hope that someone else will continue where we finished and hopefully find our research valuable and interesting. In addition to an insight in image analysis, we have learned a lot about ourselves, each other and the Australian culture throughout the project. It has been an interesting experience which surely has given us useful knowledge for our coming working life.

Part III

Discussion and Conclusion

Chapter 7

Discussion

Looking back on the project, there are many approaches that would have been interesting to continue working on. As always, the more you know the more interesting it gets. Hopefully this chapter will give some guidance in which direction future research within this highly interesting field could take.

7.1 Portal image

The main problem of the image registration task is the very blur and contrast lacking portal image. Still, the question remains if the white layer that covers the portal image, see Figure 1.3, really is noise? Concentrating more on the removal of noise from an image, a deeper study of the portal film itself and known problems of its recording was overlooked in the beginning of the project. Later on, informative literature on portal imaging was found and it led us into the study of, for example, non-linear filters. More information on for instance how the X-ray images are digitalised (rectangular/hexagonal and with which distances), the frequency contents of the images and even more knowledge about the X-ray radiation and typical disturbances for the recording of the radiation, are matters that should have been studied in an earlier stage of the project.

7.2 Wavelets and Filtering

The wavelet based approach was the first main idea for a solution to the problem. If wavelet coefficients corresponding to edges representing important bonemarks could have been found on adjacent scales of the transform, these coefficients might be the only necessary representation of the image for the edgedetection and perhaps even for the translation and rotation detection. The problem of slow computations due to big image matrices would then have been solved. Why the wavelet transform did not give satisfactory result is hard to say; either we used wavelets that were not suitable for the task or the frequency contents of the images was limited in a too narrow band to be partitioned by the used wavelet transforms or to be displayed using the already implemented routine for displaying the result in Matlab. A frequency analysis of the images would have been very informative, but unfortunately the attempts of performing such an analysis were not successful. Instead of getting deeper into the complicated

field of mathematics behind wavelets in order to find more suitable wavelets, we choosed to return to the mathematically much simpler linear filters. This on the advice of people working in the field of image processing and analysis. Linear filter gave, as already mentioned, better results than wavelets, but the most promising results came from the non-linear filters. Also the study of portal images suggested non-linear filters, as the disturbances on the X-ray images have the structure of multiplicative noise and therefore can not be treated with linear filters. Linear filters can only deal with additive noise. If time would had allowed it, interesting filters to study would have been the morphological filters. These filters are already frequently used for biomedical applications. Also, the idea behind mathematical morphology to search for sets of similar functions, i.e. sets of pixel values when it comes to images, instead of only edges, could be a more suitable way of finding the required bonemarks in the X-rays. A very instructive introduction to different families of non-linear filters and also some theory about image recording and noise is found in [19].

7.3 Calculation of translation and rotation

In this thesis we have used a minimisation program in order to detect the translation and rotation. Some kind of optimisation method surely has to be used for this task, but perhaps the cost function (V) to be minimised could be calculated in another way than using a sum over the least-square difference between two samples.

The implemented minimisation algorithm terminates when the decrease in cost between two iterations is too small and this has shown to be a problem. The cost between the two X-ray images is rather large and the cost function around the minimum is pretty flat. The result is an algorithm which terminates before the minimum is found. Moreover, the problem cannot be avoided even if the termination conditions are altered. In any future development of the minimisations algorithm, this is an important aspect which needs to be considered.

Another suggestion was to not do the comparison between the images pixel by pixel, but to compare the equilibriums of both images. This was never tested, but it could be a very interesting approach.

Chapter 8

Final conclusion

Even though we used different test results and solutions for the two first blocks, no attempt gave enough satisfactory results to link the fields of preprocessing, edge detection, and minimisation together. For the preprocessing, a continued wavelet based approach would demand deeper understanding of the rather complicated mathematics behind the wavelet theory. This is likely a more complicated way than needed of solving something that probably can be solved in an easier manner, for instance with non-linear filters. The edge detection is using the Canny operator, which already is widely known for its good performance. As for the minimisation, the prestudy preceding this thesis only showed a clear minimum of the cost-function (V) when translation and no rotation is involved. Still, the minimisation of this cost-function would probably work, even with rotation involved, once the images obtained after the edgedetection only contain edges that are relevant for the comparision. Computations using these edge images are not too slow to carry out if the loops are implemented in C and used in Matlab as Mex-files, as it has been done in this project. It would be interesting, though, to compare the accuracy and the speed of computations, when using other ways of computing the gradient and the Hessian.

A future research in the area would be guided towards finding out more about portal images and to keep on testing different non-linear filters as preprocessing of the portal image. A very interesting family is the morphological filters, which are already frequently used in the field.

To conclude, this projects deals with a complicated, yet highly interesting task, comparing X-ray images differing a lot in quality. Still, the very useful application makes it worth fighting for; it is a technical challenge in order to faciliate the treatment of two of our dearest possessions, our body and our health.

Appendix A

Time plan

	2000				2001																			
	w.34	w.35	w.36	w.37	w.38	w.39	w.40	w.41	w.42	w.43	w.44	w.45	w.46	w.47	w.48	w.49	w.50	w.51	w.52	w.1	w.2	w.3	w.4	
Start date																								
Finish date																								
Project duration																								1/2
Christmas																								
Initial literature studies																								
Testing of theory																								
Implementation of tests																								
Writing the theoretical parts of the report																								
Economical aspects																								
Further development of software																								
Optimisation of software																								
Tests of the software																								
Finishing the report and preparing the presentation																								
Reviewing another project																								
Learning LaTeX																								
Reading "Elements of Style" and "Teknisk rapportskrivning"																								
Presentation (Date not yet set)																								

Appendix B

Graphical User Interface in Matlab

A test platform for testing the different theories and algorithms developed in this project has been created by using the Matlab graphics toolbox. This appendix is included in order to give a brief introduction to graphic user interfaces, a complete coverage can be found in [9][14].

B.1 Graphics Objects

The computer monitor screen is called the root and is the parent of all figure objects. There can be several figure objects present at the same time and they are identified with an integer number. By writing the command **figure**(figure_number) the specified figure becomes active.

The figure objects have three different types of children, namely, uicontrol, uimenu and axes objects, see Figure B.1. A figure can have multiple children and the children do not need to be of the same type. Since also the root can have several children (figures), there can exist several “trees” with different branch structures. The following sections describes the children and their basic features.

B.1.1 Uicontrol

The uicontrol objects are used to generate graphical user interface controls and they can be implemented in a number of ways. They are created with the purpose of asking the user for an action or setting up options for a future action. The user often uses the mouse pointer to select the desired option.

Check Box This is a useful uicontrol when you like to give the user two options, often referred to as “on” or “off”. In its off state the square next to the describing text will consist of an empty square and in the on state, the square will contain an “X”.

Editable Text The editable text style is used in situations where the user is required to supply a string of characters or numbers. The editable text box is often used together with the static text box in order to make the user aware of what she is suppose to provide to the application.

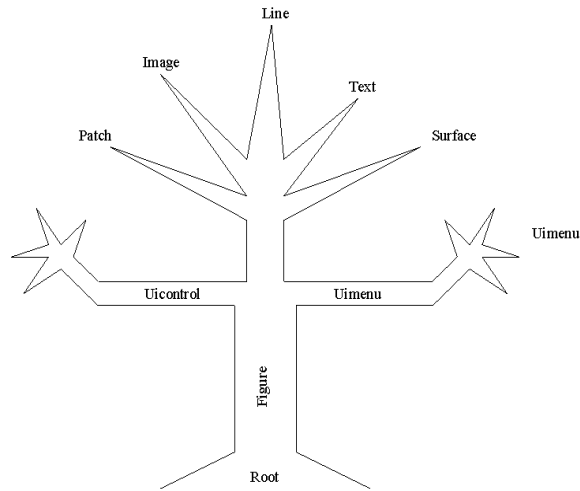


Figure B.1: The tree showing the parent-children relationship.

Frames The frame object does not demand any action from the user. However, it is used as a visual aid when grouping related objects and organising the GUI in a logical and intuitive fashion. The importance of a user friendly GUI can not be stressed too much and the frame object is an invaluable aid.

Pop-up Menus A pop-up menu is usually used in situations where multiple choices need to be available to the user. The current selection is displayed in an unopened pop-up menu. However, when the user activates the menu, a list of choices appears.

Push Buttons The push button is used when an action should take place immediately. They are often labeled with a verb describing the action taken if the push button is activated.

Radio Buttons The radio buttons are similar to the check boxes in that there are two states associated with each button, i.e “on” or “off”. However, radio buttons are usually mutually excluded which means that you can only activate one radio button within a group of linked buttons.

Sliders When the user has a fixed range of values to choose from, sliders is a useful uicontrol object. It is often comprised of a trough, an indicator bar, and a set of arrows. The trough represents the range of values to choose from, the indicator bar is used to represent the current value. By clicking on the arrows or moving the indicator bar, the values are changed.

Static Text Static text do not perform any action if the user clicks on any part of the object. It is used to create labels, status messages, or other information to the user.

Dialog boxes

There are four predefined dialog boxes generating functions that uses some of the uicontrol objects described above.

Error dialog box The error dialog box will appear in a new figure displaying some text about the error and a button labeled “Ok”. When pushing the button the dialog box will disappear.

Help dialog box This dialog box works in the same way as the error dialog box, displaying a message that will help the user. The dialog box disappear when the user pushes the “Ok” button.

Warning dialog box Works as described above except for the message type given to the user.

Question dialog box This dialog box differ somewhat from those described above. Instead of just being able to press the Ok-button, this dialog box gives the user two or three option, often “Yes”, “No” and “Cancel”.

B.1.2 Uimenu

In addition to the various uicontrol objects, you can also add pull down menus to your GUI. If a user clicks and holds down the mouse button when the pointer is located on the top of a title, a list of menu items will appear. The menu can also contain submenus and are then indicated as an arrow at the menu item. In the GUI designed for this project, the uimenu were not used and are therefore not described any further. However, a full description can be found in [14].

B.1.3 Axes

The child axes contains multiple types of objects, namely, images, lines, patches, surfaces and texts. The axes objects have a wide range of properties that can be set to satisfy every need of a GUI designer. However, they will not be discussed in detail in this appendix. In Figure B.2, the objects are shown graphically.

B.2 Programming the Callback function

In order to make your GUI do something, apart from looking nice, the Callback function needs to be implemented. The problem that needs to be considered, is that whenever a function A is executed in Matlab’s base workspace, a temporary workspace is created. This temporary workspace contains all the local variables and information associated with function A and after the function has been executed, the temporary workspace is cleared and the local variables are lost. The only variables left are those specified as global or generated as output from the function. Function A, in turn, can execute another function B, which also creates a temporary workspace for its local variables. Figure B.3 shows how the execution chain might look like.

The issue of temporary workspaces is important when designing a GUI. If you create graphic objects during the execution of a function, all of the locally stored information, such as graphic handles which might have been available in the

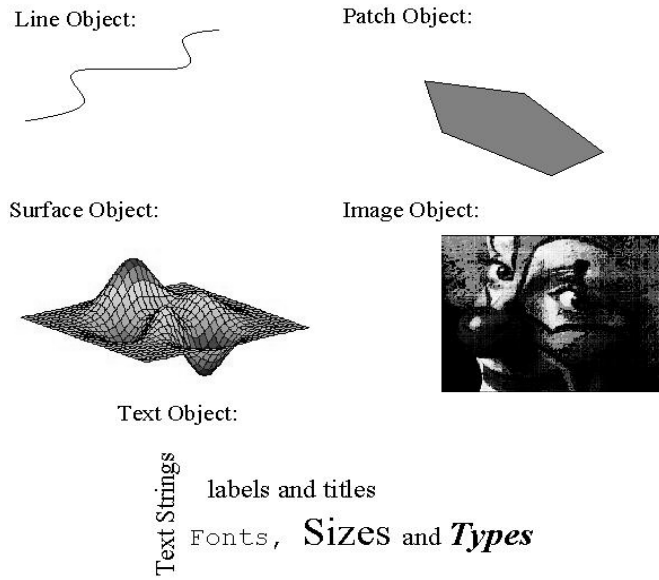


Figure B.2: The different axes objects.

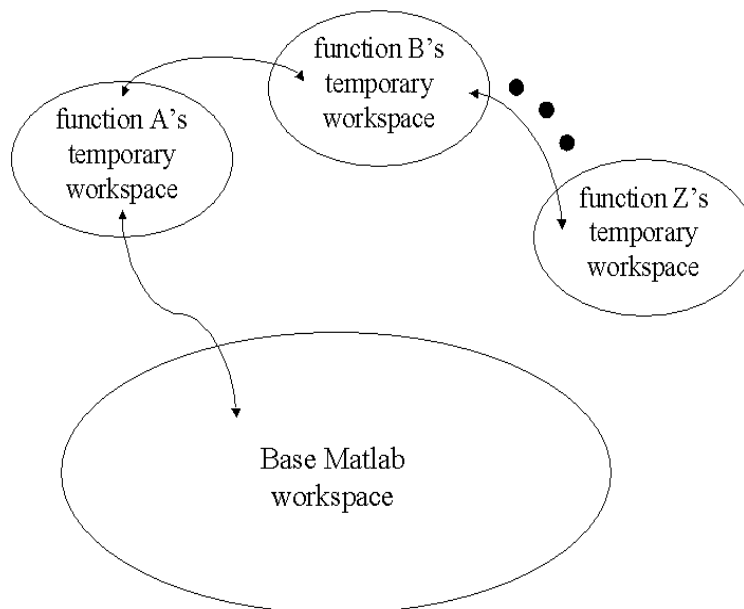


Figure B.3: Matlab's workspace during the execution of several functions.

functions local workspace, will be lost when the function has finished executing. This happens unless the information is either globally available, passed back and stored in the base workspace, or stored in the UserData property of a graphics object. Since there are so many way of storing and retrieving information, there are several structures that can be used to program a GUI. We will here discuss the method used in our GUI design.

B.2.1 Storing Handles as Global Variables

The idea in this method is to put all the code into one function and make the handles of the graphics objects global variables. When calling the function for the first time, the GUI is initialised and all the graphic objects and handles are created. The program then waits for an action by the user and when that happens the function is called again with another input variable defining the action that should be performed.

The Callback functions are separated from the initialisation of the GUI, which makes the code easier to read and modify. A short example is included below to clarify the idea.

```
function example(command_str)

if nargin == 0
    command_str = 'initialize';
end;
global h_1 h_2 ...
if strcmp(command_str,'initialize')
    %INITIALIZE THE GUI SECTION.
    %Make sure that the GUI has not been already
    %initialized in another existing figure.
elseif strcmp(command_str,'action1')
    %CALLBACK FOR action 1
elseif strcmp(command_str,'action2')
    %CALLBACK FOR action 2
end; %END command_str comparison checks.
```

There are a couple of problems which are not solved with this method. One is that an error will occur if the user issues the “clear all” or “clear global” command. Another is that the code for making sure that the GUI does not already exist is absolutely necessary. Otherwise, the globally store variables will contain the graphic handles to the most recently created GUI, that GUI will control the application, and the older object handles will no longer be stored.

Bibliography

References

- [1] Canny, John. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, November 1986.
- [2] Chan, Y.T.. *Wavelet Basics*. Kluwer Academic Publishers. 1995.
- [3] Daubechies, Ingrid. *Ten Lectures on Wavelets*. CBMS-NSF Series in Applied Mathematics, no 65, Society for Industrial and Applied Mathematics (SIAM), Philadelphia. 1992.
- [4] Dougherty, Edward R.. *Mathematical morphology in image processing*. Marcel Dekker, 270 Madison Avenue, New York, New York 10016. 1993.
- [5] Gilhuijs, K., van Herk, M.. *Automatic on-line inspection of patient set-up in radiation therapy using portal images* . Med Phys 1993, nr 20, pp 667-677.
- [6] Gonzalez, Rafael C. and Wintz, Paul.. *Digital Image Processing*. 2nd ed. Addison-Wesley. 1987.
- [7] Haus, A., Dickerson, R., Huff, K. et al.. *Evaluation of a cassette-screen-film combination for radiation therapy portal localization imaging with improved contrasts*. Med Phys 1997, nr 24, pp 1605-1608.
- [8] Hammond, R., Suchowerska, N., Manderson, C. and Drew, J.. *Electronic images vs film*. Radiat Therapist 1998, nr 7, pp 98-102.
- [9] Hanselman, Duane and Littlefield, Bruce. *Mastering MATLAB 5. A Comprehensive Tutorial and Reference*. Int ed. Prentice Hall. 1998.
- [10] Jawerth, Björn, Sweldens, Wim. *An overview of wavelet based multiresolution analyses*. SIAM Review, vol 36, no 3, pp 377-412, September 1994.
- [11] Jähne, Bernd. *Digital Image Processing*. 3rd ed. Springer. 1995.
- [12] Lamport, Leslie. *L^AT_EX A Document Preparation System*. 2nd ed. Addison-Wesley. 1997.
- [13] Mallat, S. and Hwang, W.L.. *Singularity detection and processing with wavelets*. IEEE Transactions on Inform. Theory, vol. 38, no 2, pp. 617-643, Mars 1992.

- [14] Marchand, Patrick. *Graphics and GUIs with MATLAB*. CRC Press, 2000 Corporate Blvd., N.W., Boca Raton, Florida 33431. 1996.
- [15] Meertens, H., Bijhold, J., Strackee, J.. *A method for measurement of field placements errors in digital portal images*. Phys Med Biol 1990, nr 35, pp 445-469.
- [16] Michalski, J., Wong, J., Bosch, W. et al.. *An evaluation of two methods of anatomical alignment of radiotherapy portal images*. Int J Radiat Oncol Biol Phys 1993, nr 27, pp 1199-1206.
- [17] Nocedal, Jorge and Wright, Stephen J.. *Numerical Optimization*. Springer. 1999.
- [18] Ogden, Todd R.. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser Boston. 1997.
- [19] Pitas, I and Venetsanopoulos, A.N.. *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academics Publishers. 1990.
- [20] Resnikoff, Howard L. and Wells, Jr., Raymond O.. *Wavelet analysis. The Scalable Structure of Information*. Springer. 1998.
- [21] Serra, Jean. *Image Analysis and Mathematical Morphology*. 1st Vol.. Academic Press. 1982.
- [22] Shalev, S.. *Megacoltage portal imaging* In: Teletherapy: present and future (Palta, J., Mackie, T.R., eds.), Proceeding of the 1996 Summer School, University of British Columbia, Vancouver. Madison: Advanced Medical Publishing, 1996, pp 445-469.
- [23] Sharp, Ilsa. *Culture Shock! Australia*. Kuperard, No 7 Spectrum House, 32-34 Gordon House Road, London NW5 1LP. 1992.
- [24] Starck, J.L. and Murtagh, F. and Bijaoui, A.. *Image Processing and Data Analysis*. Cambridge University Press. 1998.
- [25] Strang, G.. *Wavelets and Dilations Equations: A Brief Introduction*. Society for Industrial and Applied Mathematics Review, vol 31, No 4, pp.614-627, December 1989.
- [26] Strang, G. and Nguyen, T. *Wavelets and Filter Banks*. Wellesley-Cambridge Press. 1996.
- [27] Witkin, A.. *Scale space filtering*. Proc. 8th int. Joint Conf. Artificial Intell., 1983.
- [28] Xu, Y., Weaver, J. B., Healy, D. M. Jr., and Lu, J.. *Wavelet Transform Domain Filters: A Spatially Selective Noise Filtration Technique*. IEEE Transaction on Anal. Machine Intell. 1994.

Manuals

- [29] *MATLAB The Language of Technical Computing*. The MathWorks. 1998.
- [30] *A manifesto for Hammerstein-Wiener System Identification*.

Internet pages

- [31] The Matlab HTML helpdesk on function *edge*.
- [32] http://www.cm.cf.ac.uk/Dave/Vision_lecture/node29.html. Date:2000-12-14.
- [33] <http://www.dai.ed.ac.uk/CVonline>. Date:2001-01-23.
- [34] <http://www.digital.com/info/DTJL03/>. Date: 2000-12-05.
- [35] <http://www.mathtools.net/MATLAB/Wavelets/index.html>. Date: 2000-12-05.
- [36] <http://www.borland.com/bcppbuilder/freecompiler/>. Date: 2000-12-06.
- [37] <http://vivaldi.ece.ucsb.edu/projects/registration/registration.html>. Date: 2000-12-08.