Faculty of Mathematics and Physical Sciences

Department of Computing Science

Multi-Modality Matching using Mutual Information

Josien P.W. Pluim

Advisors:
dr. J.B.T.M. Roerdink
Department of Computing Science
University of Groningen

Ir. J.B.A. Maintz
Imaging Center Utrecht
University Hospital Utrecht

November, 1996

RuG

Preface

This thesis is the final hurdle of my journey to a master's degree in computing science from the University of Groningen. I have conducted the research for this thesis at the Imaging Center Utrecht, University Hospital Utrecht.

I am indebted to a lot of people, but only a few will be mentioned separately.
Thank you, Max, for providing the opportunity to work among experts in the field of medical imaging and for agreeing to my visit last November in the first place.
Many thanks to Twan for being helpful and interested, for eating mints on Thursday afternoon and for restricting the use of 'you could try....' those last months. I have certainly learned a lot.
Thank you, Jos, my backbone high up in the north, for keeping a watchful eye on the whole process and for the critical notes that forced me to *really* understand the subject.
I would also like to thank Frederik Maes for answering my questions about MIRIT and for not breaking all contact when I mailed a small correction.
Finally, many thanks to anyone who has, over the past four years, contributed to my succeeding in my studies, in one form or another.

Contents

Abstract

For several medical treatments it is necessary to use information about the patient that is contained in different three-dimensional scan images. It is a hard task to mentally combine these images into one image and ways of letting a computer handle this task are being researched. Finding the transformation that brings two images of different scanner types into spatial correspondence, is called *multi-modality matching*. The difficulty of this task lies in the fact that an anatomical structure is not necessarily depicted in the same manner by each scanner: it can have a different intensity or even be completely absent.

One method of matching uses *mutual information*. This concept from information theory tries to find a relation between the intensities of the images. Since this relation does not depend on similarity of the intensities, the method is especially suited to multi-modality images. Software for the matching of images using mutual information has been developed by the Laboratory for Medical Imaging Research, Leuven.

This thesis researches the possibilities of accelerating the matching process, using the software mentioned. Acceleration is attempted by preprocessing the images with morphological operations and by matching hierarchies of scaled images. For the latter method, a further improvement is attempted by adapting the parameters of the optimisation methods used.

Samenvatting

Voor verscheidene medische behandelingen wordt gebruik gemaakt van driedimensionale beelden van de patiënt van verschillende scanners. Het is moeilijk om deze beelden in gedachten tot één beeld te combineren en er wordt gezocht naar methoden om dit door een computer te laten doen. Het vinden van de transformatie die twee beelden van verschillende scannertypes ruimtelijk met elkaar in overeenstemming brengt, heet *multi-modality matching*. De moeilijkheid is dat een anatomische structuur niet noodzakerlijkerwijs door elk type scanner op dezelfde manier wordt afgebeeld: het kan een andere intensiteit hebben of zelfs geheel afwezig zijn.

Eén methode van matching gebruikt *mutual information*. Dit concept uit de informatietheorie probeert een relatie te vinden tussen de intensiteiten van de beelden. Daar deze relatie niet afhankelijk is van de overeenkomst tussen de intensiteiten, is deze methode bijzonder geschikt voor beelden van verschillende scanners. Software voor het matchen van beelden met behulp van mutual information is ontwikkeld door het Laboratory for Medical Imaging Research, Leuven.

Deze scriptie onderzoekt de mogelijkheden om het matching proces te versnellen, uitgaande van de genoemde software. Er wordt getracht een versnelling te bereiken door de beelden vooraf te bewerken met morfologische operaties en door hiërarchieën van geschaalde beelden te matchen. Voor deze laatste methode wordt onderzocht of een verdere versnelling mogelijk is door de parameters van de gebruikte optimalisatiemethoden aan te passen.

Chapter 1

Introduction

Medical images are essential in the fields of neurosurgery, radiotherapy and minimally invasive surgery. They are used throughout the entire trajectory of planning, treatment and evaluation. Some types of medical images are *Magnetic Resonance (MR/MRI)* images, *Computed Tomography (CT)* images and *Single Photon Emission Computed Tomography (SPECT)* images. A few examples are shown in figure 1.1.

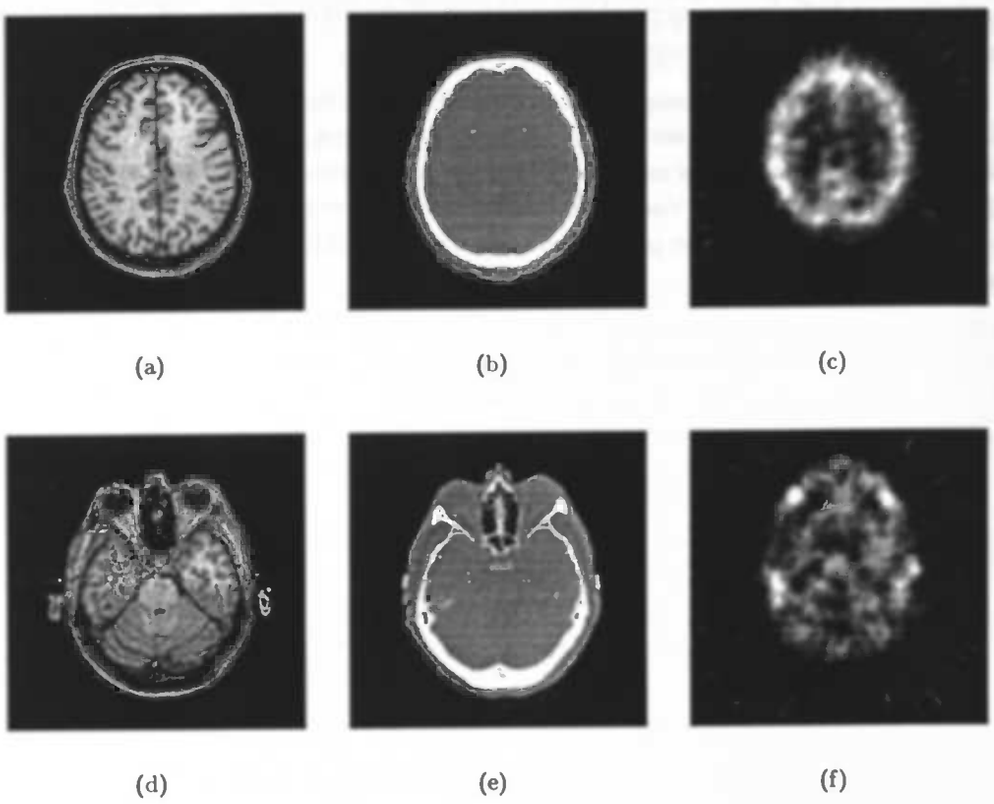


Figure 1.1: Examples of two slices from different three-dimensional scans: (a) and (d) are MR images, (b) and (e) CT images and (c) and (f) show SPECT images.

Registration (also: matching or alignment) of images is the process of finding the geometrical transformation that brings one image in a precise spatial relation with another image. When two images of an object are matched, every voxel in the first image shows the same

little piece of the object as the corresponding voxel in the other image. Registration can be performed on images of an equal or of a differing number of dimensions. Furthermore, the images involved can be obtained using the same type of scanner (mono-modal images) or different types of scanners (multi-modal images).

Mono-modal image matching is useful when comparing different patients or when evaluating one patient's illness over a period of time. A task more difficult than mono-modal image matching is the registration of multi-modal images, since there is far less similarity between multi-modal images than between mono-modal images. Dissimilarities can occur, amongst others, in the size of the region scanned. A computed tomography (CT) scan is always made of as small a region as possible, since the rays of the scanner are harmful. Therefore, a CT scan usually shows a smaller part of the patient than the corresponding magnetic resonance (MR) scan. There can also be differences in the grey values of the anatomies, because different scanners show different physical aspects of the imaged anatomy. The skin will have a different shade of grey in the MR scan than in the CT scan. Bone does not even show in an MR scan (although sometimes the marrow will be visible) and thus leaves a black space, while bone is almost white in CT scans. Lastly, the angle under which scanning takes place may differ, which results in different cross sections of the patient. All in all, an accurate registration is not easily obtained.

For many applications however, it is necessary to combine images of different modalities. These applications need information that cannot be obtained by a single scanner. Different scanning techniques produce different information in their resulting images. For example, although magnetic resonance and computed tomography images both contain anatomical features, brain structure is only visible in the former, while bone is much more prominent in the latter. Single Photon Emission Computed Tomography, on the other hand, depicts functional information of the scanned anatomy, as does *Positron Emission Tomography (PET)*. A combined visualisation of multi-modal images, which requires matching of these images, is very helpful when using information of different modalities for the treatment. In radiotherapy, for example, the dose calculation is based on a CT scan, but the outlining of the tumour on an MR scan.
A problem with many scanning techniques is that they are too slow to be used during an actual intervention, i.e. surgery or radiotherapy. Fluoroscopy and ultrasound are fast enough, but the images they produce are two-dimensional and thus lack the information that the three-dimensional aspect gives. Therefore, there is interest in the registration of 3D with 2D images. This would combine the information and high quality of a 3D pre-intervention scan with the speed of two-dimensional imaging.

The following are some examples of situations that would benefit from 3D-2D image registration.

Endovascular treatment of abdominal aortic aneurysms An aneurysm is a dilation of an artery due to weakness of the arterial wall. Some abdominal aortic aneurysms can be treated by placing a stent - a mould for keeping a skin graft in place - in the aorta. With the stent covering the aneurysm, the risk of the arterial wall rupturing is greatly reduced.
A CT scan is made to determine the required stent size and to plan the procedure. During the operation a delivery sheath, containing the stent, is guided to

the aneurysm by fluoroscopic images. Accurate placement of the stent is crucial: the stent must cover the whole aneurysm, yet not block arteries. This procedure will benefit from the registration of the CT scan and the fluoroscopic images. The three-dimensional information of the CT scan is then available during the procedure, which will contribute to the correct placement of the stent. Furthermore, determining the correct position is based on several anatomical features that do not show on fluoroscopic images, but are visible on the CT scan.

Verification of patient placement In external beam radiotherapy the radiation field is shaped like the tumour to be treated. This requires extremely high accuracy in the positioning of the patient. Such a requirement can only be met when using three-dimensional information. Before the treatment, a 3D scan (e.g. CT or MRI) should be made to obtain this information. If the 3D scan is matched with two-dimensional images, made during the treatment, the three-dimensional information can be used for the positioning of the patient.

Percutaneous laser treatment Tumours can also be treated in a minimally invasive manner using a laser. First, a needle is directed into the tumour. Down this needle an optical fibre is passed. A laser then delivers a high amount of energy through the fibre to vaporize the tumour. This is another case in which accurate placement is of great importance and thus three-dimensional information is helpful. Here, 2D CT images, made during the intervention, are matched with a pre-treatment 3D MR or CT scan.

Chapter 2

Problem definition

For the examination and treatment of a patient, it is not uncommon that several scans of different types (modalities) are made, like MRI or CT. Each type of scanner captures certain information about the patient. Often two or more scans of different modalities have to be made to obtain all the information necessary for the diagnosis or treatment. Combining these different scans into one image would greatly enhance the understanding of their contents. Visualising the scans together makes it easier to grasp the relations between the objects in the scans. In this way, for example, it can be immediately clear how close a tumour is to an artery or a vital organ.
In order to visualise two scans, combined into one image, these scans have to be registered. In other words, one of the images has to be brought into spatial correspondence with the other.

Registration of multi-modal images is a difficult task due to the intrinsic differences between the images. Extensive research has been done all over the world, trying to find the best method for a particular kind of multi-modal registration. Of all those techniques, *mutual information* is one of the most recently proposed. Mutual information is a concept from information theory, based on image grey (or colour) values. Unlike concepts such as correlation, mutual information does not depend on similarities between the grey values of the images to be registered. Mutual information depends on the existence of a certain relation between the grey values. So, if brain tissue is light grey in one image and dark grey in the other, these images can be matched with mutual information, whereas correlation would require the brain tissue to have similar grey values. Mutual information only needs some relation between colour values in one image and colour values in the other image. This property makes it very suitable for the registration of multi-modal images.

A group of researchers from the University Hospital Gasthuisberg in Leuven has developed a program for the registration of multi-modal images, MIRIT [17, 18]. MIRIT has been the basis for my research. Using this program, I have explored the possibilities of combining image registration based on mutual information with other methods. One method is to use morphology to simplify the images or to extract a certain feature. Another possible method is the use of different scales. The images are first registered on a coarse scale and, via ever finer scales, finally on a scale of high resolution. The results of registration on a certain scale are used as a starting point for matching on the next, finer scale.
It is hoped that the combination of mutual information and other methods will result in a registration technique that is significantly faster than mutual information alone, while retaining an acceptable accuracy.

The next chapter in this thesis is a literature survey on multi-modal image registration

methods. The emphasis is on the matching of 3D with 2D images, as it was originally the intention to research the possibilities of MIRIT in that area. This is followed by a description of mutual information and MIRIT. Then come the methods that are to be used in combination with mutual information. Finally, the experiments, their results and conclusions are presented.

Chapter 3

Literature Survey

3.1 Registration Techniques

Given two images to be matched, the essential idea behind image registration is finding a geometrical transformation that aligns one image with the other. A wide range of registration methods has been developed, which differ mainly in the type of properties that the match is based on. These properties can be divided into two categories, namely extrinsic and intrinsic properties. Extrinsic properties relate to artificial objects that have been placed in or on the patient, for example, frames or markers. Intrinsic properties address patient related properties, like voxel intensities or contours of objects.
Geometrical transformations range from rigid via affine and projective to nonlinear transformations. A rigid transformation preserves the distance between any two points in an image and an affine transformation maps any straight line onto a straight line, while parallelism is preserved. A transformation is called projective when any straight line is mapped onto a straight line, but the restriction of parallelism is dropped. Finally, with a nonlinear transformation a straight line can be mapped onto either a straight line or a curve.
In this chapter an overview is presented of registration methods for two 3D images. Nearly all methods find rigid transformations. Some methods for 3D-2D registration are also discussed. Important issues in evaluation of the methods are accuracy, patient-friendliness, robustness, generality and reproducibility (i.e. finding the same results for each attempt with the same input). Also important are the amount of user-interaction necessary and the possibility of retrospective matching, that is to say, the possibility of making the decision to register when the images have already been made. The emphasis is on intrinsic methods that require little user-interaction, since these are more desirable. These methods are patient-friendly, not very labour-intensive, can be applied in retrospect and their results can be more accurate than those of marker based methods [26].

3.1.1 Extrinsic Properties

Registration methods based on extrinsic properties all have in common that matching cannot be done retrospectively, i.e., once the scans have been made without the artificial matching points, extrinsic methods can no longer be applied. They are usually accurate and very general, as any type of image can be matched so long as the markers are visible in the image. Patient-friendliness is not one of these methods' strong points though and placing the markers can be labour-intensive.

Frames and moulds

Stereotactic frames are without doubt the least patient-friendly. A rigid frame is fixed to the patient's skull by screws. To this frame, markers are attached that act as matching points for the registration algorithm. Although the registration result when using a frame is accurate, the method clearly has many disadvantages: it cannot be applied retrospectively, attaching the frame can be very time-consuming and it is very uncomfortable for the patient. In some cases frames cannot even be used. For example, frames are too large for some imaging devices.
Individually shaped head or dental moulds are much more patient-friendly, but, apart from that, have the same disadvantages as frames. Furthermore, this method is less accurate.

Skin markers

Skin markers are specially designed objects that are attached to the skin for registration purposes. Reproducibility is good for short periods if the positions of the markers are recorded on the patient's skin with ink. For longer periods the reproducibility is a problem. Another problem with these markers is the movement of the skin. Finally, the results deteriorate in case of thick slices or large interslice gaps.

3.1.2 Intrinsic Properties

All intrinsic methods can be used in retrospect, although some algorithms require the calibration parameters of the imaging device. They are also patient-friendly as they are based solely on patient related image data. As far as generality and the amount of user-interaction is concerned, the methods vary greatly.

Anatomical landmarks

Anatomical landmarks usually have to be pointed out interactively. The accuracy is reasonable, but the method relies on the landmarks being visible in both modalities. The reproducibility of the results depends on the steady hand of the user.

Structures and features

Surfaces and contours are examples of structures that can be the basis for registration. The accuracy of structure based methods can be good, but they often require a large amount of user interaction. Furthermore, the methods are application specific as they depend on the object(s).
Van den Elsen *et al.* [26] describe the use of differential operators to create ridge features. These were matched using cross correlation. Visualisation of the results showed an even better match than marker based results. Maintz *et al.* [20] evaluated several differential operators, but found no clear preference for a single one.

Voxel properties

Most methods based on voxel properties are automatic and require no preprocessing of the images. Concepts from information theory are the inspiration for these methods. One often used concept is correlation. The major disadvantage of this method however, is the need for similarity in intensities between the images. For example, an image and its inverse would never yield a good result using correlation. One solution to this problem is increasing the similarity of the images. A grey value scaling of CT images to make them resemble MR images, is proposed by van den Elsen *et al.* [27].
A more recent trend is the use of mutual information. This concept from information theory is defined as the amount of information that one variable contains about another [17]. The advantage of mutual information is that not a similarity in intensities is required, but *some* dependence between the intensities. Furthermore, the measure is robust with respect to occlusions of one of the images [30], to noise and to intensity inhomogeneity [17].
Wells *et al.* [31] tested the possibilities of mutual information for matching of MR and PET images, which is a difficult task owing to the relative blurriness of PET images. Visual inspection of the results was promising.
Finding affine transformations with mutual information is described by Studholme *et al.* [25]. Scaling parameters and skew angles are included in the optimisation process. This registration seemed more accurate than their registration with rigid transformations, but the computation time increased.
All authors reported good accuracy for mutual information registration.

3.2 3D-2D Registration Techniques

Lately, the 3D-3D registration methods as described in the previous section are being tried on the registration of a 3D and a 2D image. Usually, this involves projecting (a feature in) the 3D image. The solution of the registration is the transform of the 3D image that results in the best match of the projection and the 2D image.
The following is a survey of some attempts at registering 3D and 2D images. Not all of these methods use multi-modal images; not all of them are applied to clinical data. This survey is purely descriptive: all the testing described has been done by the authors of the respective papers.

3.2.1 Landmarks

Bijhold's method of registration [5] is used for the verification of patient placement during radiotherapy. Before the treatment a 3D CT scan is made. Also, a two-dimensional reference image is obtained during a simulation of the treatment, the simulator image. During the actual treatment, 2D images are acquired, the portal images.
A set of match points, S_i, is selected interactively from the CT scan. These points have projections in the simulator image. Similarly, the positions of the match points during the actual treatment (i.e. in the patient) have projections in the portal images. The assumption is made that the two sets of match points in three dimensions are related to

their respective projections in the same manner. Furthermore, the match points in the CT scan and in the patient are assumed to be related by a rigid transformation. Using this, a set of equations can be determined that relates the points in the scan, S_i, to the points in the portal images. These equations are then simplified, either under the assumption that all points S_i are in a plane perpendicular to the beam axis (plane method) or that they are in a volume (volume method). Finally, the parameters of the rigid transformation (the actual solution) are found by solving the equations.
Both methods were tested on a phantom of a pelvic bone. The resulting registrations were accurate, although the plane method showed errors when out-of-plane rotations were involved.

3.2.2 Edges/Ridges

Gilhuijs *et al.* [11] have extended the work of Bijhold [5] to a more automatic method of registration. The simulator and portal images are matched in two steps. First, the edges of the radiation fields are aligned. Next, selected anatomy edges, in this case bone edges, are aligned. In both cases the registration is achieved by chamfer matching [1, 6]. This method basically matches a drawing onto features in an image. A cost function, the measure for the goodness of fit, is minimised in order to align the drawing and the image. Some methods also include a mechanism for attracting the drawing to a structure in the image (e.g. distance coding: assigning to each pixel the distance to the nearest feature in the image). In the simulator images the field and anatomy edges are drawn before the treatment. These edges serve as the drawings in the chamfer matching algorithm. In the portal images the edges are extracted automatically, using a top-hat transform [24]. This results in the feature images of the matching methods.
Several combinations of cost functions and distance coding for the two matching phases have been tested. Considering the cost functions, it is important to note that in this particular application the features in the images have the lowest intensity values. For the alignment of field edges an optimal combination could not be found, because the best combination depends on the kind of dissimilarity between the portal and simulator images. However, it was shown that with the right strategy, registration would never have to be performed for more than three combinations. For the anatomy matching two combinations performed equally well for all situations. The best cost function was the arithmetic mean pixel value combined with either 8-neighbour distance coding or the negated inverse-squared 8-neighbour distance transformation (NIS-distance).
The conclusion for the field edge matching is that an average accuracy of 0.0 pixels (a rather dubious number) can be found for typical radiation fields. When deviations in the shape of the radiation fields occur, the results are still consistent, but not as accurate. For the alignment of anatomical structures the average accuracy is 1.8 mm.

In a more recent paper [10], a new method is described. The registration involves Digitally Reconstructed Radiographs (DRRs)[1] and either simulator or portal images. From these images the bone ridges are extracted, which can be done by hand or automatically. The automatic procedure is either the top-hat transform or a multi-scale medial axis enhancement [9]. A cost function is defined which denotes the negated average distance through

[1] A DRR is produced by casting rays through a CT scan and integrating (a function of) the Hounsfield numbers along the ray.

bone from the focus of the irradiation unit to the location of bone ridges in the images. The locations of the bone ridges coincide with the projective trajectories that encounter the longest distances of bone. Therefore, minimisation of the cost function should lead to a good alignment. Two methods of minimisation were investigated: the downhill simplex minimisation of Nelder and Mead [21] and Powell's minimisation [7]. For this application the method of Nelder and Mead was found to be the most efficient. The method was tested for both simulated setups and actual clinical examples. In both cases the deviations were within 1 *mm* and 1 degree. The computation time was about 2 minutes on a 90-MHz Pentium PC.
Although the matching results are promising, there are still problems when using smaller fields. Also, the lack of sufficiently many bony structures leads to a decrease in accuracy.

3.2.3 Cores

A possible feature for image registration is a core. The core of an object is defined by Liu *et al.* [15] as the loci of centres of circles that best fit a normalised, Gaussian blurred instance of the object, where the degree of blurring is proportional to the circle's radius. An advantage of cores is their insensitivity to disturbances smaller than the scale of the object, e.g. noise or blurry edges.
The input to the registration algorithm consists of a set of 3D curves and a set of 2D curves. The method is an iteration over two phases. The alignment phase extracts a list of common points for each pair of a 3D and a 2D curve. These curves are aligned using a measure for the distance between corresponding points. The gradient descent phase then uses these points to refine the current estimate of the registration. The method of gradient descent is similar to Lowe's algorithm [16]. The iteration stops once the estimate has converged to a satisfactory solution. In the first pass the user has to provide an approximate guess.
In the experiments two MR scans of a head were used, from which sets of respectively 204 and 109 vessels were extracted to serve as 3D curves for the registration algorithm. Applying a perspective projection to the segmented vessels resulted in simulated angiograms, from which the set of 2D curves could be obtained.
The resulting registration was good, with errors less than 1 *mm*. However, the experiments were conducted on *simulated* angiograms. No experiments on clinical data are reported. A method for approximating cores took 20 minutes on a DECStation 125 to extract more than 100 vessels. An indication of the registration time or the computational demands was not given.
Experiments on the choice of the basic set of curves showed that using more curves for matching produces more accurate results. Tests also revealed that the distance between the sets of curves in the initial approximation should be less than seven centimetres. Larger initial deviations may lead to errors of over 1 *mm*. With a user defined initialisation this restriction will most certainly be met, but a fully automatic approach may cause problems here.

3.2.4 Curves and Surfaces

Betting and Feldmar describe two methods of 3D-2D registration: for either curves [8] or surfaces [3]. The latter is explained for both a calibrated 2D sensor and the non-calibrated case.
The 3D-2D curve registration works on 3D (CT or MR) images and 2D images of vessels. In all images these vessels are represented by curves described by points and their associated tangents. The problem is to find a projective transformation that maps a 3D curve onto a 2D one. The method starts by finding an initial transformation estimate using bitangent lines (a detailed description can be found in [8]). Next, this estimate is iteratively updated. In each iteration a set of matching points from the sets of curves is chosen. For each point on the 3D curve a matching point on the 2D curve is found by searching on the 2D curve for the point that is closest to the projective transform of the 3D point. This *closest point* is determined by compromising between the distance and the tangent orientation. Also, for every pair of points a generalised Mahalanobis distance is computed as a measure of 'plausibility' of the match. Pairs whose Mahalanobis distance falls below a certain threshold are rejected. The second step in the iteration consists of calculating the transformation that minimises the difference between the pairs of matching points. This minimisation algorithm is an extension of the Iterative Closest Point algorithm [2].
Experiments on unspecified datasets resulted in an average distance of 0.83 pixels for the final registration. This calculation is based only on the matched points, though. If other points are used, the results may be less accurate. Computation time was 20 seconds on a DEC alpha workstation.

For the second method [3], a surface (represented by points and normals) of an object is extracted from the 3D image and the object's contour from the 2D image. The algorithm is basically the same except that now bitangent *planes* are also used in the calculation of the initial estimate and the minimisation measure is the difference between points on the surface and their matching points on the contour. The calibrated case uses a minimisation with respect to six parameters. When the 2D scanner is non-calibrated, the five calibration parameters are also unknown and the minimisation takes eleven parameters into account. The testing of the registration algorithm was done on a 3D CT scan and a 2D video image of a mannequin head. With known calibration parameters the average errors were 0.76 pixels and 0.17 degrees, while these were 0.79 pixels and 0.7 degrees for the non-calibrated case. It should be noted however, that the latter is more sensitive to the initialisation conditions.

3.2.5 Cross Correlation

Penney *et al.* [22] describe 'a first attempt' at the registration of a 3D CT volume and 2D images. The chosen registration measure is cross correlation, although I suspect some calculation is done on the correlation value as they are looking for a minimum, not a maximum.
A digitally reconstructed radiograph (DRR) is made of the CT volume. This DRR and the 2D image are the images of which the cross correlation is calculated. The search strategy considers all degrees of freedom of the rigid body transformation (i.e. three translations

and three rotations) and chooses the direction which achieves the lowest value of cross correlation. This process is repeated until a minimum is found. For an increase in speed a multi-resolution approach was used, halving the pixel dimensions at each scale until a minimum was found at a precision of 1 mm.
In the experiments either two or three out of the six degrees of freedom were held constant, as movements in these directions produced very little change in the DRR. The experiments are started from a manual registration of the images.
To allow registration of all six degrees of freedom, the CT volume was combined with two fluoroscopy images, taken at a sixty degree angle to each other. Exactly how these images are registered is not described clearly.
In both cases the experiments tested mainly the consistency and not the accuracy of the algorithm. This consistency is called 'reasonable'. Also, visual inspection of the results showed no large deviations.

3.2.6 Correlation/Gradients

For the confirmation of electrode positioning on the brain, Lemieux *et al.* [14] have devised a method of aligning a pre-intervention CT volume with Anterior-Posterior (AP) and lateral radiographs, taken during the procedure.
An AP and a lateral DRR are obtained from the CT set and registered with the radiographs. For this registration two cost functions were considered. The first is the cross correlation of the pixel values. The other cost function is the cross correlation of the gradient images of the radiographs and their corresponding DRRs. Although the differences in the registration error of these functions were very small, it was concluded that the best strategy would be to use the cross correlation cost function at the initial stages of registration and the gradient based function at the final stages.
As two DRRs have to be generated at every iteration, a fast convergence is crucial. Therefore, the initialisation is assumed to be close to the final solution. A standard minimisation routine can then be used to find the minimum solution instead of searching the entire parameter space, since the risk of converging to a local minimum is very small. To minimise the cost function, Powell's method [7] was employed.
In experiments the mean registration error (mean distance between validation points) was found to be 0.5 mm. However, this was only the case if a number of constraints was met. First of all, the method is not successful on datasets that do not contain distinguishable features. Also, the radiographic projection parameters need to be known precisely. Finally, the patient has to be specifically positioned. Loosening of these conditions is possible, but will mean an increase in computational time. The average time needed for the algorithm as described was 30 minutes on a SUN SPARCStation 10/30.

3.3 Conclusion

Making comparisons between the different methods as just described is not realistic. The tests are performed on different datasets, on different machines, the evaluation measures are incomparable (and sometimes dubious), often information is missing and the sets of evaluated properties are at best overlapping.

Most authors report accuracies that would be good enough for actual use of their methods. The evaluation of accuracy is not unambiguous, however, as there are no gold standards for registration. Marker based matches are considered the standard, although they now seem to be outperformed by intrinsic methods [26]. For artificial transformations the matching result can be compared to the known parameters. In practice, these parameters are exactly what the registration is supposed to solve and visual inspection of the match is one of the best options (although this does rely on the correctness of the visualisation). Another problem is the initialisation. Many methods depend on an initial match that is close to the final solution. Computation time is a matter of concern as well. None of the algorithms have the speed that is required for real-time applications. The relatively fast methods once again rely on a good initial guess.
The future appears to be lying in intrinsic methods, especially the ones based on feature images or voxel properties. They show promising results, are patient-friendly, can be applied in retrospect and do not require the time-consuming tasks of making moulds or attaching frames and markers. Currently, the solution is still a compromise between speed, accuracy and user-interaction. The search continues for a fast, accurate and fully automatic method.

Chapter 4

Mutual Information

4.1 Theory

Many concepts from information theory have already been tried for the registration of medical images, such as, for example, cross correlation. However, these methods use the difference between image intensities and therefore rely on similarities between the images. When matching multi-modal images, such similarities are few. Thus, for a correct alignment of multi-modal images with these methods, some preprocessing of the images is necessary. Either the required similarities are '*created*' by a grey value rescaling of one of the images, such that it resembles the other. Or one or more of the existing similarities are *extracted*.
A relatively new measure in the world of image registration is *mutual information*. This measure does not suffer from the restrictions described above since it is not based on similar appearances of the images, but on the fact that both images depict approximately the same object, albeit in a different manner. Mutual information defines *some* statistical relation between the intensities of two images. This does not necessarily imply an equivalence relation (which would rely on similarities). An image and its inverse are very dissimilar, but clearly related. All that is white in one image is black in the other and vice versa. Mutual information would perform well on these images, whereas measures like standard cross correlation (without preprocessing) would fail. As no assumptions are made concerning the nature of the relation between the image intensities, this measure is a very general one. It is also one that can be applied automatically: no preprocessing is required, no contours or landmarks have to be defined.

Mutual information is defined in terms of probability distributions. Given two images A and B, given the marginal probability distributions $p_A(a)$ and $p_B(b)$ and the joint probability distribution $p_{AB}(a,b)$ of their grey values, the mutual information $I(A,B)$ is computed as follows:

$$I(A,B) = \sum_{a,b} p_{AB}(a,b) \log_2 \frac{p_{AB}(a,b)}{p_A(a) \cdot p_B(b)} \tag{4.1}$$

This definition results in a mutual information value of zero when the images are statistically independent. For two maximally dependent images, there exists a bijective transformation T such that $p_A(a) = p_B(T(a)) = p_{AB}(a, T(a))$, while the images are independent when $p_{AB}(a,b) = p_A(a) \cdot p_B(b)$. Therefore, equation 4.1 defines mutual information as a measure for the difference between the joint distribution $p_{AB}(a,b)$ and the distribution in case of complete independence, $p_A(a) \cdot p_B(b)$. The idea behind the concept is that two

images that are aligned are maximally dependent. Since mutual information is a measure for the degree of dependence, the search is for the transformation T that maximises $I(A, B)$.

Mutual information is related to another concept from information theory, *entropy*. The most widely used measure for entropy is the Shannon-Wiener entropy, which is defined by

$$H(A) = -\sum_{k=1}^{n} p_A(k) \log_2 p_A(k).$$

This is interpreted as the average information of image A. An image with similar amounts of all intensities is said to contain more information than an image in which the majority of voxels have the same intensity. The joint entropy $H(A, B)$ is the Shannon-Wiener measure of the joint probability distribution $p_{AB}(a, b)$, defined by

$$H(A, B) = -\sum_{a,b} p_{AB}(a, b) \log_2 p_{AB}(a, b).$$

The relation between mutual information and entropy is

$$I(A, B) = H(A) + H(B) - H(A, B).$$

The changes in the information in the combined image are related to the information provided by the images separately. The information content of the combined image when correctly aligned will be lower than when misaligned. Thus, this measure will indeed be maximum at registration and decrease with misregistration.

Although at first sight the definition of mutual information does not seem to take spatial information into account, registration with mutual information performs much better when the image contains areas of reasonably homogeneous intensities. This 'spatial homogeneity' of the intensities causes the joint probability density function $p_{AB}(a, b)$ to change smoothly with the registration parameter α, which is a vector consisting of three rotation angles (in degrees) and three translations (in millimetres). Then, the function of misregistration (i.e. the behaviour of the mutual information value for α deviating from the optimal solution) will also be smooth. Since the optimisation of the mutual information criterion depends on the smoothness of the misregistration function, spatially homogeneous images are registered better.
This can be more clearly explained with an example. Imagine two chess boards with a pattern of tiny squares, lying on top of each other. These boards are matched. When the board on top is moved forward until one row of squares from the bottom board is visible, the tiny squares are aligned, but the whole boards are misaligned by one row. The mutual information value of this position would be rather high, at least higher than when the top board had been moved a little less or a little more. In other words, this is a local maximum. Several local maxima are possible, making it hard to find the absolute maximum. If the pattern of the boards consists of only four huge squares (i.e. high spatial homogeneity), there are no local maxima. Registration is much easier, as any small translation of the top board decreases the mutual information value.

4.2 MIRIT: the algorithm

F. Maes *et al.* [17] of the University Hospital Gasthuisberg in Belgium have created a software package for 3D-3D multi-modal image registration, using mutual information. MIRIT (Multi-modality Image Registration using Information Theory) will be the basis for my research. A brief manual for MIRIT can be found in appendix A.
For the alignment of image A with image B, A is defined as the *floating image* F and B as the *reference image* R. Samples s are taken from the floating image and to these a transformation T_α is applied. Here α denotes the registration parameter of a rigid body transformation (a vector). Only S_α, the set of samples for which $T_\alpha(s)$ falls inside the volume of the reference image R, is considered.

Histograms of the image grey values are created to estimate the probability density functions of the images. For the joint probability density function, the entries of the histogram are pairs of a sampled point s and the transformed point $T_\alpha(s)$. In general, $T_\alpha(s)$ will not coincide with a grid point of R and interpolation is necessary. The authors have experimented with several interpolation methods, namely *nearest neighbour interpolation*, *trilinear interpolation* and *trilinear partial volume interpolation*. Given a point p in image A and the transformed position p', the first interpolation method finds the nearest neighbour of p' (on the grid of image B) that falls inside the volume of B. The intensity of this nearest neighbour is assigned to $B(p')$ and the histogram is updated by adding 1 to the entry $(A(p), B(p'))$. Trilinear interpolation finds the eight nearest neighbours of p'. The intensities of all those neighbours that are inside the volume of B are trilinearly interpolated, this value is assigned to $B(p')$ and the histogram entry $(A(p), B(p'))$ is increased by 1. Finally, partial volume interpolation is identical to trilinear interpolation, except that in this case, for all neighbours $i = 1, \ldots, 8$, the histogram entry $(A(p), B(i))$ is increased by w_i, with w_i the weight associated to neighbour i as used in trilinear interpolation.
Of these methods, the authors of MIRIT prefer partial volume interpolation. Nearest neighbour interpolation is insensitive to translations smaller than the voxel dimension and in those cases sub-voxel accuracy can only occur by accident. Trilinear interpolation can introduce intensities that were originally not present in the reference image.

The probability distributions of the image intensities are estimated by binning intensity pairs $(f(s), r(T_\alpha(s)))$ for all $s \in S_\alpha$. Here, $f(s)$ denotes the intensity of F at position s and $r(T_\alpha(s))$ the intensity of R at the transformed position. The resulting joint image intensity histogram $h_\alpha(f, r)$ is used to obtain estimations for the marginal intensity distributions $p_{F,\alpha}(f)$ and $p_{R,\alpha}(r)$ and the joint intensity distribution $p_{FR,\alpha}(f, r)$ in the following manner:

$$p_{FR,\alpha}(f, r) = \frac{h_\alpha(f, r)}{\sum_{f', r'} h_\alpha(f', r')} \tag{4.2}$$

$$p_{F,\alpha}(f) = \sum_r p_{FR,\alpha}(f, r) \tag{4.3}$$

$$p_{R,\alpha}(r) = \sum_f p_{FR,\alpha}(f, r). \tag{4.4}$$

The mutual information registration criterion $I(\alpha)$ is then evaluated by

$$I(\alpha) = \sum_{f,r} p_{FR,\alpha}(f,r) \log_2 \frac{p_{FR,\alpha}(f,r)}{p_{F,\alpha}(f) \cdot p_{R,\alpha}(r)} \tag{4.5}$$

and the optimal registration parameter α^* is found from

$$\alpha^* = \arg\max_{\alpha} I(\alpha) \tag{4.6}$$

Here $\arg(x)$ denotes the arguments that resulted in x.

The initial value of each transformation parameter is zero, unless otherwise defined in the *starting position* in the file MIRITparams (see appendix A). *Powell's direction set method* is then used to maximise $I(\alpha)$, using *Brent's one-dimensional optimisation algorithm* for the line maximisations (see section 4.3).
The order in which the parameters are optimised may be of influence to the optimisation, as some translations and rotations are likely to be larger than others due to the shape of the object. F. Maes *et al.* suggest optimising the in-plane parameters (t_x, t_y, ϕ_z) before the out-of-plane parameters (ϕ_x, ϕ_y, t_z), but they have not yet investigated this thoroughly.

4.3 MIRIT: the optimisation methods

The following are descriptions of the methods used by MIRIT to find the optimal value of mutual information. The descriptions are based on [23] and the MIRIT source code.

4.3.1 Powell's direction set method

The problem of matching two three-dimensional images by a rigid transformation is equivalent to finding the global optimum of the registration function in a six-dimensional space. It suffices to describe the method of finding a maximum, since the minimum of a function f is the maximum of $-f$. The basic idea of Powell's method is to define the space by six (initially orthogonal) directions and to perform one-dimensional maximisations in each direction consecutively. Move along the first direction to its maximum, from there along the second direction to *its* maximum and so forth. This process is repeated until a maximum in the six-dimensional space is reached.
The question is how to find a set of directions that takes you to the desired maximum *quickly*. It is very likely that an arbitrary set of directions will converge slowly, taking only a tiny step towards the maximum in most (or perhaps even all) directions. Powell's direction set method is a good technique to avoid tiptoeing through parameter space.

The method starts by initialising the set of directions u_i to the standard basic vectors e_i, for $i = 1, \ldots, N$ (N being 6 in this particular case). Next, the following sequence of steps is repeated until there is no more increase in the registration function:

1. Call the starting position P_0.
2. For $i = 1, \ldots, N$, move from P_{i-1} to the maximum along direction u_i and call this point P_i.

3. Define a new direction $u_{new} := P_N - P_0$.
4. Replace the direction u_i along which the largest increase was achieved by u_{new}.
5. Move from P_N along direction u_{new} to its maximum.

The new direction defined is the average direction moved. It may seem strange to let this new direction replace the direction along which the largest increase was achieved, since that direction was the best one in the current iteration. However, it was also one that largely influenced the determination of the new direction. The danger of keeping both directions is that after several iterations all directions can become almost identical.
In certain cases the new direction is rejected and the old set of directions is kept. This happens, for example, when a little further along the new direction the function values are already decreasing again, i.e. there is nothing more to be gained from this direction. It also happens when the increase along the average direction was not primarily due to any single direction's increase.
The process of iterations is stopped when a maximum number of iterations is reached. Alternatively, it is terminated when the difference between the function values at the starting position of an iteration and at its final position is less than a fraction of their average value. Both the maximum number of iterations and the fraction are defined by the user (see MIRITparams, appendix A).
Now that the method for finding a maximum in a multi-dimensional space is defined, the one-dimensional optimisation can be specified.

4.3.2 Routine for initially bracketing a maximum

When searching for a maximum in a certain direction, it is useful to first determine an interval that contains a maximum. Given three points a, b and c, with $a \leq b \leq c$, a maximum of the function f is 'bracketed' by a and c if both $f(a)$ and $f(c)$ are less than $f(b)$.
In the registration algorithm of MIRIT, point a is set to the *starting position* (see appendix A). The second point, b, is initialised by a + *step size*. The variable *step size* is defined in the input file MIRITparams, see page 51. Both the starting position and the step size are in millimetres and degrees. When a maximum is searched for, the function f should go uphill from a to b. If necessary, the points are interchanged. Finally, point c is initialised by $b + 1.61803 \times (b - a)$. The multiplication factor is deduced from the *golden section search* (see section 4.3.3).
If $f(c) \leq f(b)$, a maximum is bracketed by a and c. Otherwise, a parabola is fitted through the three points. The position of the parabola's maximum is called u. Depending on $f(u)$ and on the position of u with respect to the other points, the points a, b and c are moved in order either to bracket a maximum or to move closer towards one. When moving the points, the invariants $a \leq b \leq c$ and $f(a) \leq f(b)$ must not be violated. In cases that the function is not parabolic on the interval (a, c), a default magnification step is taken $(u = c + 1.61803 \times (c - b))$.
The process of determining a new point u and moving the points a, b and c is repeated until $f(c) \leq f(b)$. A maximum is then contained in the interval (a, c).

4.3.3 Brent's method for one-dimensional maximisation

Brent's method assumes that a maximum has already been bracketed. To pinpoint the location of the maximum, the method keeps track of six points: a, b, x, v, w and u. The maximum is bracketed by a and b, x is the point with the highest function value found so far, w is the point with the second highest function value, v is the previous value of w and u is the point at which the function was evaluated most recently.
Again, parabolic interpolation is attempted, this time fitting through the points x, v and w. To be acceptable, the parabolic step must (i) fall within the bounding interval (a, b) and (ii) imply a movement from the best current value x that is *less* than half the movement of the *step before last*. The demand on the decrease in step ensures convergence. The comparison of the current step to the *step before last* gives the algorithm a chance to make up for a mis-step. The function should also never be evaluated too close to an already evaluated point, since the difference in function values could fall below machine precision. If the step is not accepted, a new point is determined by *golden section search*. This method uses the fractions 0.38197 and 0.61803 of the *golden section*. Given the interval (a, b) and the point x inside this interval, the next point to be tried is that which is a fraction 0.38197 into the larger of the two intervals (a, x) and (x, b) (measuring from point x).
Once a new point u has been found (by either method), the function is evaluated at this point and, depending on u and its value, all points are updated (see figure 4.1). This process is repeated until the interval containing the maximum is sufficiently small.

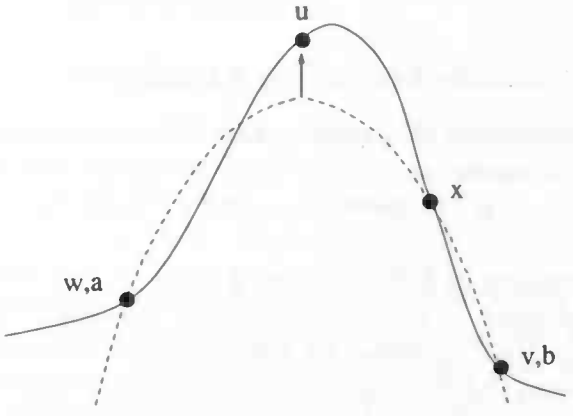


Figure 4.1: Convergence to a maximum by a parabolic fit. A parabola (dashed line) is fitted through the three initial points w, x and v. The function is evaluated at the parabola's maximum, resulting in point u. Point v is discarded and point b is set to point x. Then point v is changed to point w, w to x, x to u, after which the fitting can be repeated with the new points v, x and w.

The implementation of Brent's method in MIRIT starts by initialising $a = a_p, b = c_p$ and $v = w = x = b_p$, with (a_p, b_p, c_p) the points found by the routine for initially bracketing a maximum. The process of convergence to a maximum is continued until either a user defined number of iterations is reached or the width of the bracketing interval (a, b) is less than a certain tolerance. This tolerance is a fraction (defined by the user) of the distance between the point x and the starting position a of the routine when initially bracketing a maximum.

4.4 MIRIT: the results

Experiments were conducted by the authors on four datasets: a pair of high resolution CT and MR images, a lower resolution pair (obtained by smoothing and subsampling the previous pair), stereotactically acquired MR, CT and PET images, from which the markers have been removed, and an MR image.
All datasets were registered using MIRIT. The parameters were optimised in the order $(t_x, t_y, \phi_z, \phi_x, \phi_y, t_z)$, all parameters being zero initially. Different interpolation methods were tried and the floating and reference images were interchanged. The registration solutions, as found by the providers of the datasets, were used as standards for measuring the accuracy of registration.

4.4.1 Accuracy

Neither interchanging the floating and reference images nor using different interpolation methods led to large variations in the solutions of registration for the first two datasets. The differences with the standard solution were sub-voxel. For the second dataset the differences were slightly larger, but these may be (partially) caused by the creation of these images.
There was one exception to the rule that the choice of floating and reference image did not matter. For the stereotactically acquired datasets, the registration of CT to MR using trilinear interpolation did not converge to the standard solution, while that of MR to CT did.
With MR to CT matching, all 'errors' are smaller than the voxel dimensions (1.25 x 1.25 x 4.0 mm), the maximum error being 1.2 mm in the y direction using partial volume interpolation. When registering MR to PET as well as PET to MR images, however, partial volume interpolation showed the smallest differences.

4.4.2 Robustness

The performance of MIRIT was tested for partially overlapping volumes. The high resolution MR image was registered with three different 50-slice slabs of the CT image. Most errors were subvoxel with respect to CT voxel size (1.55 mm) and the maximum error was 1.5 CT voxel.
The influence of reduced quality of the images on the performance of MIRIT was also tested. The results of registering the MR image with itself and with a reduced quality version of itself were compared by evaluating their *traces* for all transformation parameters. A trace shows the behaviour of the mutual information value for deviations from the solution found in one or more parameters. Adding noise to the image (zero-mean Gaussian noise with a variance of either 50, 100 or 500 grey values) did not influence the result of registration as the position of the maximum mutual information value did not change. This was also the case for intensity inhomogeneities, i.e. uneven brightness of the image. Geometric distortions, however, did alter the registration results. In the trace of the translation parameter corresponding to the dimension in which the image was distorted, the position of the maximum shifted. This shift was proportional to the average distortion.

A more extensive description of the experiments can be found in [18].

Chapter 5

Preprocessing and other methods

5.1 3D Morphology

I have examined the performance of MIRIT on images preprocessed with morphological operations. Some were standard operations. More complex operations were used to extract features.

5.1.1 Some standard morphological operations

Standard operations result in images that can be used directly as input for MIRIT. These operations can also be combined into feature-extracting operations. Programs for the standard operations are available. Examples of the results of these morphological operations on an MR image are shown in figure 5.1.

Erosion

The *erosion* ε of a binary set A with a structuring element B that is symmetrical with respect to reflection, is defined by

$$\varepsilon(A) = \{x \mid \forall b \in B, x + b \in A\}.$$

In less technical terms: $\varepsilon(A)$ consists of those points of A for which, when the origin of the structuring element B coincides with that point, B fits completely inside A. The result of eroding a binary image is an image in which all the objects have 'slimmed down'. Along the outer edges of the objects a strip is removed. The width of this strip depends on the size of the structuring element: the larger the structuring element, the wider the strip. Objects smaller than the structuring element will disappear completely.
In case of grey value images, a function $g(x)$ exists, denoting the grey value of pixel x. The erosion ε with a flat structuring element B is now defined by

$$\varepsilon(g)(x) = \inf_{b \in B} (g(x + b)).$$

For the erosion of 3D grey value images with a cubic structuring element, a program *Erode* is available. The half width t of the structuring element can be specified, resulting in a total width of $2 * t + 1$.

Dilation

Closely related to the erosion is the *dilation*. The dilation δ of a binary set A with a symmetrical structuring element B is defined by

$$\delta(A) = \{x + b \mid x \in A \wedge b \in B\}.$$

The dilation can be interpreted as all translations of B for which B has a non-empty intersection with A. Dilating a binary image will have the opposite effect of an erosion: all objects will be enlarged and small holes may disappear.
For grey value images, the definition of a dilation is

$$\delta(g(x)) = \sup_{b \in B} (g(x + b)).$$

This is implemented in the program *Dilate*. Again, the structuring element is a cube with a total width of $2 * t + 1$, with t the specified half width.

Opening

The opening of an image is a combination of the previous two operations. First an erosion is applied to the image, followed by a dilation. The erosion removes small or thin objects and it removes strips along the edges of the larger objects. The following dilation will (partially) restore the strips, but not those objects that have disappeared completely. An opening thus rids an image of small objects and thin protuberances. Holes that are close to the edge or to another hole will be opened up, as the thin dividing structure disappears. A program *Open* is available.

Closing

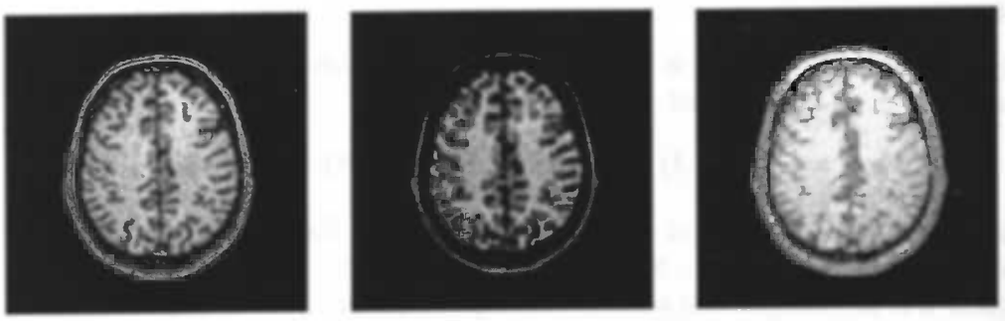
The combination of first a dilation and then an erosion is called the *closing*. This operation can fill holes and connect objects.
This can be achieved with *Close*.

Morphological gradient

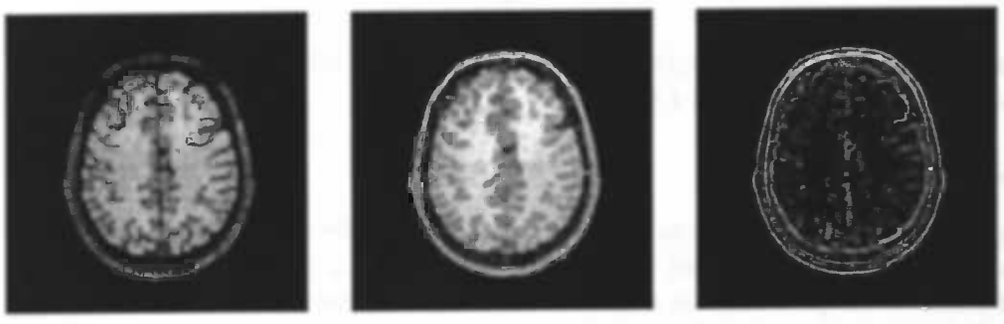
The *morphological gradient* of an image is defined as the difference between the dilation and the erosion of that image. It can be used to acquire information about the edges in an image. Thinner edges are shown when the *inner* or *outer gradient* is computed. The inner gradient is found by subtracting the eroded image from the original, while the outer gradient is the dilation of the image minus the original.
The program *morphgrad* calculates the morphological gradient. It has options for computation of the inner and outer gradients.

5.1.2 Some complex morphological operations

Features can be extracted from images by means of morphological operations. For the registration of images it is important that these features are not subject to change in time. That is to say, they are not removed, they do not expand, shrink or disappear completely.



(a) MR image (b) An erosion of (a) (c) A dilation of (a)



(d) An opening of (a) (e) A closing of (a) (f) A morphological gradient of (a)

Figure 5.1: Examples of simple morphological operations on an MR image.

When considering scans of the head, suitable features are the skull, the edge of the skin and the edge between the skin and the skull. I have attempted to extract these features from a set consisting of an MR and a CT image, using combinations of the morphological operations described above. In the description of these combinations, the names of the morphological operations are followed by a number in brackets, denoting the half width of the structuring element in voxels.

The skull

On the CT scan a close(5) was used to fill 'holes', like the frontal sinus. This was followed by an open(9) to remove the skull. When the result is subtracted from the original, an image of only the skull is left.
For the MR image, a close(1) was first applied to smooth the brain structure. A close(6) was computed separately. This filled the gap where the skull should have been. By subtracting the first result from the second, an image of the skull was created.
Examples of extracted skulls are shown in figures 5.2(a) and (d).

The edge of the skin

A method for extracting the edge of the skin was proposed by J.B.A. Maintz [19]. For the CT image it is a combination of a close(4), followed by an open(8), followed by an inner gradient(1). The closing fills the holes and the opening removes the bones. This results in a quite homogeneous image of the head. The edge of the skin is found with the inner gradient.
On an MR scan, a close(4) followed by an inner gradient(1) suffices. The closing creates an image that is homogeneous enough to find the edge of the skin.
Results of the method just described are given in figures 5.2(b) and (e).

The edge between the skin and the skull

For the CT scan, the skull (obtained by the method described above) is subtracted from the original. To this result an open(6) is applied, which leaves the brain, approximately. Next, the skull is added again and an outer gradient(1) then finds the edge between the skin and the skull.
With the MR image it is necessary to first obtain an image of the skin only. This is done by a close(1) to fill small gaps in the skin, followed by an open(5) to remove the skin. To get the image of the skin, the result of the opening has to be subtracted from the result of the closing. Now, the edge between the skin and the skull can be computed. First, a close(9) is applied to the original to smoothen the image. From this, the image of the skin is subtracted. An outer gradient(1) on this last result will return the desired edge.
In figures 5.2(c) and (f) examples are shown of images of this edge.

I have defined the complex morphological operations while working with one pair of an MR and a CT image (dataset A, described in section 6.1). The operation proposed by J.B.A. Maintz was also tested on this set. The results of the operations are not perfect: there are often some faint edges left of anatomical structures that were not supposed to be extracted. However, the desired feature is dominant. Most problems arise with the lower part of the brain as the structure of the skull is more complex there.

To evaluate the general applicability of the complex morphological operations, these were tried on another pair of an MR and a CT image (dataset B in section 6.1). On the CT image the operations work fine the way they are. CT images are rather homogeneous with respect to intensities. They differ mainly in the shape of the anatomical structures and the size of holes, like the frontal sinus and the orbits.
The results were not as hopeful for the MR image. MR images of two persons will not be similar, since their brain structures can be very different and this structure is visible in MR scans. Even two MR images of the same person will probably not be identical, because this scanning technique is very sensitive to parameter settings.
Applying the operation for extracting the skull, resulted in an image that showed far too much other tissue. Better results are possible by adjusting the widths of the structuring elements. The results of the other two operations were reasonable.

Preprocessing of the datasets cannot be done automatically with the operations described. Especially for MR images, the results should be checked. Slight adjustments of the operations may be necessary.

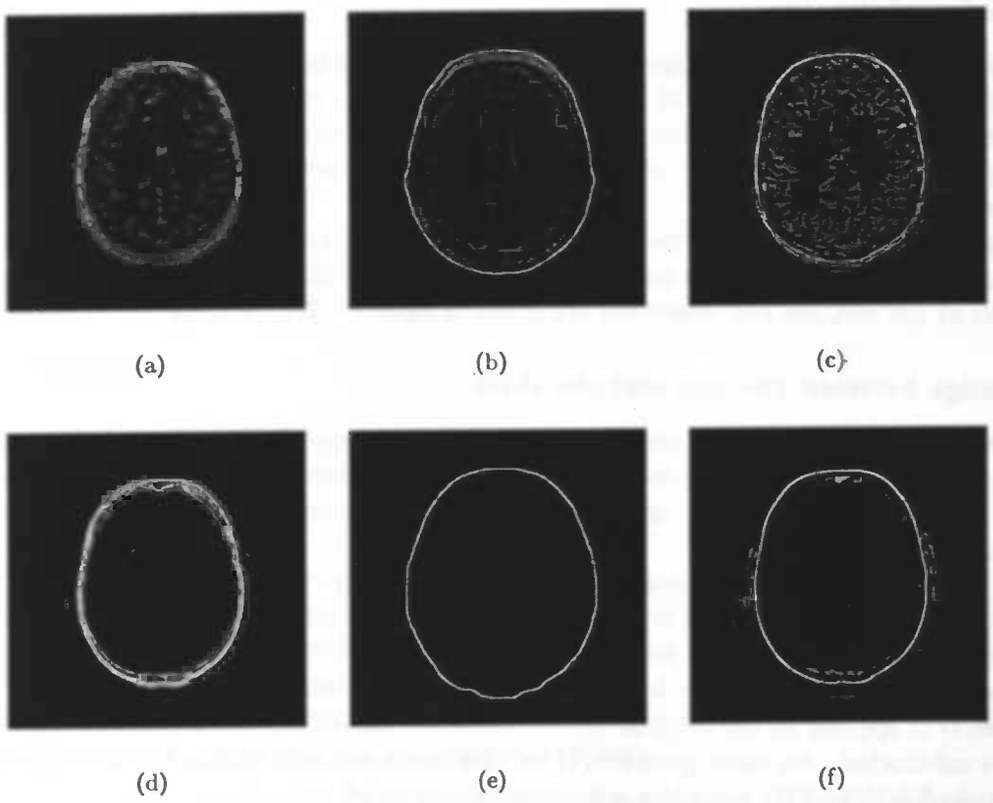


Figure 5.2: Examples of complex morphological operations on an MR image (top row) and a CT image (bottom row). Each row shows the results of the operations for extracting the skull, the edge of the skin and the edge between the skin and the skull, respectively.

5.2 Programs

To facilitate my research, I have used several programs. Some of these had already been written by members of the research group, others I have implemented myself. The existing programs were sometimes altered to obtain suitable output. All programs are written in C++, unless stated otherwise.

5.2.1 Existing programs

Mat2par

Mat2par is a program that calculates transformation parameters given a matrix that defines a transformation in voxel dimensions. The input of the program *Mat2par* consists of a 4x4 transformation matrix and two image info files. The matrix describes the transformation between two images and the image info files contain information about the images, like the dimensions, the pixel size, the slice thickness and the interslice gap. *Mat2par* returns the transformation parameters, in millimetres and degrees.

Par2mat

Par2mat constructs a transformation matrix on the basis of the transformation parameters.

Maxdist2

The maximum distance between the results of two transformations of an image is computed by *maxdist2*. The maximum distance is determined by sampling points on a sphere of radius r around the centre of the image. For each sample point, the distance between the two transformed points is calculated. The maximum of these distances is returned.

5.2.2 Developed programs

Prima_reduce

Prima_reduce downsamples a 3D image in *PRIMA* format (Public Research, IMage Archive). For each dimension the reduction factor can be set separately. The downsampling method used is computing either the mean or the maximum of the voxels' grey values.

Compute_entropy

This program computes the entropy of a 3D *PRIMA* image. The entropy measure is the Shannon entropy, see section 4.1.

getTransform

A program to extract any of the four transformation matrices from a MIRIT registration output file (cf. appendix A).

getMI

The program *getMI* was made to create a file of trace values. As these values are written to standard output by MIRIT, the output has to be redirected to a file when MIRIT is in tracing mode. The mutual information values can then be extracted from this file by *getMI*.

PlotTrace.m

PlotTrace.m is a *Mathematica* module. Given the file of trace values, as created by *getMI*, and the number of steps and the step size of the trace, the module returns a plot of the trace.

Chapter 6

Experiments

6.1 Setup

Several facts about the setup of the experiments should be mentioned before any results are shown.

Datasets

Dataset A contains a CT and an MR scan of a 35 year old male. The CT scan consists of 100 slices of 256 by 256 pixels each and there is no gap between the slices. The slice thickness is 1.55 mm and the pixel size is 0.94 mm.
The MR scan contains 180 contiguous slices of 1.0 mm thickness. The dimensions of a slice are 256 by 256 pixels and the pixel size is 0.98 mm.

Dataset B is also a pair of a CT and an MR scan of one patient. The CT scan is made up of 145 contiguous slices that are 320 by 320 pixels each. The slices are 1.5 mm thick and the size of the pixels is 0.71 mm.
The MR scan consists of slices of 256 by 256 pixels. There are 100 slices in total and there is no interslice gap. The thickness of the slices is 1.5 mm and the pixel size is 0.9 mm.

MIRIT input

The input for MIRIT is identical to the examples given in appendix A, unless stated otherwise.

Standard of performance

The images of the datasets were registered with MIRIT at full scale. After visual inspection of the results, the obtained accuracy was chosen as a standard of performance for the experiments, where either the scale is changed or some operation is applied to the images.

Hardware

All experiments were conducted on a HP 9000/755 system.

6.2 Matching of morphologically preprocessed images

Applying morphological operations like an opening, a closing or one of the feature-extracting operations of section 5.1.2 to an image, reduces the information content of the image. The intention of the following experiments is to find out whether MIRIT converges faster to a solution when applied to images with a lower information content and whether the solution found is acceptable.
The images resulting from four simple morphological operations were used in the experiments: an opening of half width 1, an opening of half width 2, a closing of half width 1 and a closing of half width 2. Of the feature-extracting operations only the edge of the skin was considered, since this one was most extensively investigated in the past.

Since the registration of two images at full scale takes a long time (about 1.5 hours), I decided to reduce the images in all dimensions by a factor of two. Reduction can be achieved in two ways: by using *prima_reduce* or by setting the *sampling factors* in the file MIRITimages to 2.0 (which means every second voxel is sampled). There are several possibilities for the combination of reduction and morphological operation. These are:

1. performing the morphological operations on the original set and using sampling factors 2.0 in the registration,

2. reducing the dataset by a factor of two in all directions, performing the morphological operations on this reduced set and using sampling factors 1.0 in the registration,

3. performing the morphological operations on the original set, reducing the resulting set by a factor of two in all dimensions and using sampling factors 1.0 in the registration.

I have tested all combinations for an opening of half width 1 and a closing of half width 1 on both datasets. The outcome of *maxdist2* served as the criterion for selecting the best combination. The standard of performance was chosen to be the result of registering the images at full scale (without morphological operations or added transformations). Combinations 1 and 3 gave the best results. I have chosen combination 1, because I had noticed in some mono-modal experiments that setting the sampling factors to 1.5 resulted in less accurate solutions than when the sampling factors were 2.0 or even 3.0. This made me wary of using the sampling factors and I wanted to investigate further before applying them.
Once the morphological operation had been performed, the dataset was reduced by a factor of 2 in all dimensions using *prima_reduce*, taking the mean value. The CT image was the reference image and the MR image served as the floating image.

Table 6.1 shows the results of registering the MR to the CT image of dataset A, after different morphological operations had been applied to the images. The results for dataset B can be found in table 6.2. 'Operation' denotes the morphological operation used. 'NrEvals' is the total number of evaluations of Brent's maximisation method and 'NrIts' shows the number of iterations of Powell's algorithm. 'Timing' is the required time in seconds. 'Transformation' refers to the transformation parameters as calculated by *mat2par* using the matrix *FloatToRefImageTransform* (see appendix A, page 52), in millimetres and degrees. This matrix defines the transformation that aligns the floating image with the

reference image (in voxel dimensions) and it can be found in the output file. 'Maxdist' is the outcome of *maxdist2* in *mm* (with radius 70 *mm*), calculated on the solution of the current experiment and the solution of the registration at full scale, without any morphological operations. Both the transformation parameters and maxdist have been rounded. The calculation of maxdist, however, has been done on the original results.
The first entry refers to the registration of the images at full scale, without any morphological operations. The resulting transformation is used as the standard for *maxdist2*. The second entry refers to the registration of the images reduced by a factor of 2 in all dimensions. The timing of this registration is a standard to compare the timing of the morphological experiments to. Its maxdist shows the smallest attainable deviation for the two images at half scale with this particular method and parameter setting.

Nr	Operation	NrEvals	NrIts	Timing
1	None, full scale	252	4	6027
2	None, half scale	269	4	550
3	Open(1)	262	4	537
4	Open(2)	244	4	507
5	Close(1)	272	4	551
6	Close(2)	255	4	515
7	Skin-edge	380	5	766

Nr	Transformation $(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$						Maxdist
1	-6.9	-1.2	-18.3	10.5	2.7	2.6	0.0
2	-6.9	-1.1	-18.4	10.5	2.7	2.6	0.2
3	-6.9	-1.2	-18.5	10.8	2.7	2.6	0.4
4	-7.0	-0.9	-18.7	11.1	2.8	2.6	1.1
5	-6.9	-1.1	-18.3	10.6	2.7	2.6	0.3
6	-6.9	-0.9	-18.0	10.8	2.7	2.4	0.9
7	-6.9	-0.6	-15.8	11.4	3.1	2.3	3.2

Table 6.1: Results of MR to CT morphological matching of dataset A. For explanation, see text.

As is clear from these results, almost none of the morphological operations can make MIRIT converge significantly faster. The solutions found are less accurate than the solutions of registration at half scale without morphology. This was to be expected as information is lost when applying morphological operations. Some of the solutions, for example those for an opening of half width 1, are acceptable, but the acceleration of the convergence is so marginal that the combination of MIRIT and morphological operations can hardly be called an improvement of the original method.
For the images with very little information content, i.e. the skin-edge images, the results were quite surprising: MIRIT converged not much faster (dataset B) or even much slower (dataset A). The loss of information resulted in a loss of smoothness of the misregistration function. This has an adverse effect on the convergence to the maximum, especially close to the maximum. This is clear from the traces in figure 6.3. In addition to the slow convergence, the solutions found were, respectively, a bad match and a mismatch.

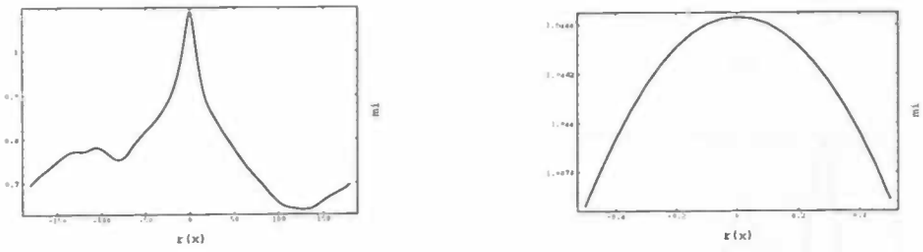
Traces were computed to examine the behaviour of the mutual information value around the solution found. A trace plots the value of mutual information against (one of) the

Nr	Operation	NrEvals	NrIts	Timing
1	None, full scale	361	6	5816
2	None, half scale	450	7	612
3	Open(1)	383	6	520
4	Open(2)	393	6	522
5	Close(1)	386	6	531
6	Close(2)	372	6	504
7	Skin-edge	418	5	545

Nr	Transformation $(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$						Maxdist
1	2.5	-8.6	40.4	-17.3	1.4	1.7	0.0
2	2.6	-8.2	40.3	-17.3	1.5	1.6	0.5
3	2.6	-8.1	40.2	-17.4	1.4	1.6	0.6
4	2.6	-8.1	40.5	-17.3	1.6	1.6	0.7
5	2.7	-8.1	40.2	-17.5	1.5	1.6	0.8
6	2.7	-8.2	40.1	-17.9	1.5	1.7	1.2
7	4.7	-16.3	1.4	-1.3	-0.0	1.8	49.6

Table 6.2: Results of MR to CT morphological matching of dataset B. For explanation, see text.

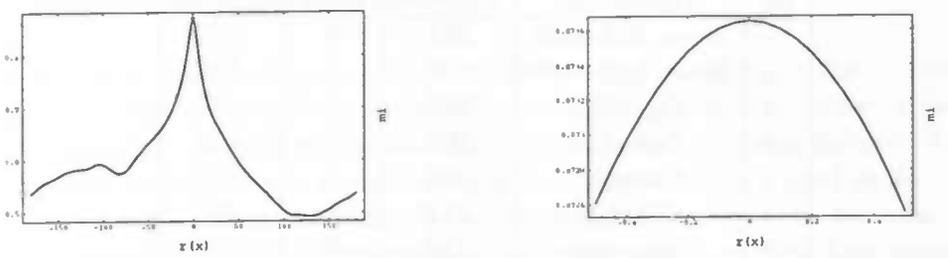
transformation parameters. If the solution found is the global maximum, MIRIT has performed well.
Figure 6.1 shows traces for rotation around the x-axis, computed on the result of an Open(1) on dataset A. For comparison, the same traces for the registration of the images at full scale without morphological operations are shown in figure 6.2.



(a) (b)

Figure 6.1: Traces of an opening of half width 1 on dataset A, for ϕ_x. The mutual information value (mi) is plotted against deviations from the solution found for ϕ_x. On the left, these deviations range from -180 to 180 degrees. On the right is a 'close-up' of the solution found, i.e. of the peak. Here, the deviations range from -0.5 to 0.5 degrees.

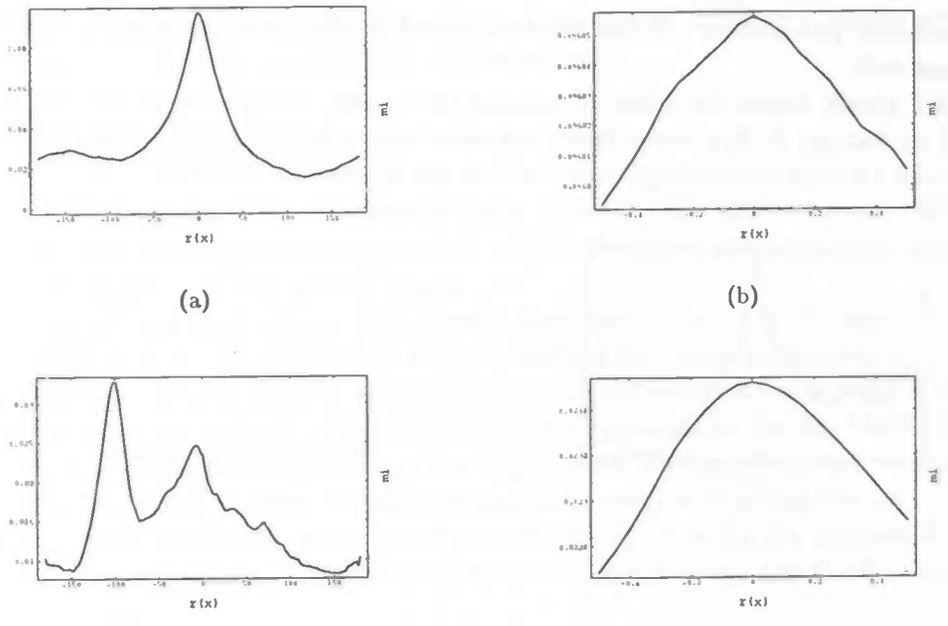
For all the other transformation parameters the same two traces were computed. These were all very similar to the ones shown. The same was true for the traces of the other morphological operations on dataset A and also for the traces of the images of dataset B. Only the tracing of the 'skin-edge' images gave different results. These traces are shown



(a) (b)

Figure 6.2: Traces of the images of dataset A at full scale and without morphological preprocessing, for ϕ_x. For explanation, see Table 6.1.

in figure 6.3. The registration results for dataset A were poor. For dataset B, these were *very* poor, which expresses the fact that the global maximum was not found. MIRIT converged to a local maximum, as is clear from figure 6.3(c).



(a) (b)

(c) (d)

Figure 6.3: Traces of the skin-edge images, for ϕ_x: (a) and (b) are on dataset A, (c) and (d) on dataset B. For explanation, see Table 6.1.

All in all, the combination of morphological operations and MIRIT does not seem a viable method. Matching slightly altered images gives satisfactory solutions, but hardly

any increase in calculation speed. When the information content of the images is more radically reduced, the results are both slow and inaccurate.

6.3 Multi-scale matching

The idea behind a multi-scale approach is to find a rough estimate of the solution on a coarse scale and then to refine the solution using images of a finer scale. First, a hierarchy of scales is created by downsampling the original images by increasing factors. A pair of images on a low, coarse scale is registered. The resulting solution is used as the starting position for the registration of a pair of images on a finer scale. The outcome of this registration can again be the starting position of matching on an even finer scale. This process can be repeated until the finest scale desired has been reached.
It is possible that registration on a hierarchy of a few scales will require less time in total than matching directly on the finest scale. Registering on a low scale is fast, because the images are smaller. Matching on a fine scale with a good starting position should also be fast due to the chosen optimisation technique. For every direction in parameter space, this technique starts by finding an interval that includes a maximum. This is done by initialising an interval around the defined starting position and then iteratively moving and enlarging this interval until a maximum is included. With a good starting position a satisfactory interval will be found (almost) immediately.
In the following experiments no efforts have been made to adapt the optimisation method for different scales. Further decreases in the time demands of the registration process could possibly be achieved, for example, by loosening the tolerance boundaries at coarse scales or by setting a more narrow initial interval around the starting position when this is already close to the final solution (i.e. at fine scales).

6.3.1 Using *prima_reduce*

To create the scaled images, the datasets were reduced with *prima_reduce*, taking the mean value. The reduction factors were 2, 3 and 4. Reducing the images by a factor of more than 4, left too few features to match.
Scale $1/x$ means that both images of a dataset have been reduced in all dimensions by a factor x. Scale $1/x \rightarrow 1/y$ means that registration has first been performed at scale $1/x$ and the results have been used as a starting position for registration at scale $1/y$. This initial position can be defined by setting the *starting position* in the file MIRITparams, see appendix A. From the output file of the registration at scale $1/x$, the *FloatToRefParameters* (see appendix A, page 52) were extracted to serve as starting position.
The results of the multi-scale experiments are shown in table 6.3 for dataset A and in table 6.4 for dataset B. The CT scan served as the reference image and the floating image was the MR image.

As was to be expected, registration at coarser scales resulted in larger deviations from the standard solution (i.e. maxdist increased). With a hierarchy of scales, an accuracy can be achieved that is similar to the accuracy of the finest scale in the hierarchy. This is also the case when a bad match or a mismatch is used as a starting position. However, one pays for this by slower convergence. For example, in table 6.3 it can be seen that a match at scale 1/3 does not give a good result (maxdist is 1.7 mm). When this result is used as the starting position for a match at scale 1/1, the solution has a maxdist of 3e-3 mm, which means it is almost as accurate as a direct match at scale 1/1. However, convergence takes

Nr	Scale	NrEvals	NrIts	Timing	Total Time	Transformation $(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$						Maxdist
1	1/1	252	4	6027	6027	-6.9	-1.2	-18.3	10.5	2.7	2.6	0.0
2	1/2	269	4	550	550	6.9	-1.1	-18.4	10.5	2.7	2.6	0.2
3	1/3	254	4	176	176	-7.0	-1.3	-19.8	10.8	2.5	2.6	1.7
4	1/4	300	4	108	108	-6.7	-0.5	-17.6	11.0	2.5	2.7	1.7
5	1/3 →1/2	189	3	383	559	-6.9	-1.1	-18.4	10.5	2.7	2.6	0.2
6	1/4 →1/3	202	3	138	246	-7.0	-1.3	-19.8	10.7	2.5	2.6	1.7
7	1/4 →1/2	115	2	234	342	-7.0	-1.1	-18.4	10.5	2.7	2.6	0.2
8	1/4 →1/3 →1/2	187	3	376	622	-7.0	-1.1	-18.4	10.5	2.7	2.6	0.2
9	1/3 →1/1	182	3	4914	5090	-6.9	-1.2	-18.3	10.5	2.7	2.6	3e-3
10	1/2 →1/1	132	2	2013	2563	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-4
11	1/3 →1/2 →1/1	130	2	1989	2548	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-3
12	1/4 →1/1	121	2	1854	1962	-6.9	-1.2	-18.3	10.5	2.7	2.6	2e-2
13	1/4 →1/3 →1/1	176	3	4830	5076	-6.9	-1.2	-18.3	10.5	2.7	2.6	3e-3
14	1/4 →1/2 →1/1	128	2	1970	2312	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-3
15	1/4 →1/3 →1/2 →1/1	132	2	2016	2638	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-3

Table 6.3: Results of MR to CT multi-scale matching for dataset A. 'Timing' is the required time in seconds. In case of multi-scale registration, timing denotes only the time required for the last step in the registration. The total time demand can be found in the column 'Total Time'. 'NrEvals' is the total number of evaluations of Brent's maximisation method and 'NrIts' shows the number of iterations of Powell's algorithm. The column 'Transformation' shows the transformation parameters in millimetres and degrees. 'Maxdist' is the outcome of *maxdist2* in *mm* (with radius 70 *mm*), calculated on the solution of the current experiment and the solution of the registration at full scale. Both the transformation parameters and maxdist have been rounded. The calculation of maxdist, however, has been done on the original results.

Nr	Scale	NrEvals	NrIts	Timing	Total Time	Transformation($t_x, t_y, t_z, \phi_x, \phi_y, \phi_z$)						Maxdist
1	1/1	361	6	5816	5816	2.5	-8.6	40.4	-17.3	1.4	1.7	0.0
2	1/2	450	7	612	612	2.6	-8.1	40.3	-17.3	1.5	1.7	0.5
3	1/3	427	6	212	212	4.3	-6.2	41.8	-17.5	1.4	1.6	3.8
4	1/4	256	2	67	67	3.9	0.8	0.0	0.0	0.0	-0.5	50.3
5	1/3 →1/2	195	3	266	478	2.6	-8.1	40.3	-17.3	1.5	1.7	0.5
6	1/4 →1/3	428	7	211	278	4.3	-6.3	41.8	-17.5	1.4	1.6	3.5
7	1/4 →1/2	368	6	500	567	2.6	-8.2	40.2	-17.4	1.5	1.7	0.5
8	1/4 →1/3 →1/2	217	3	290	568	2.6	-8.1	40.3	-17.3	1.5	1.7	0.5
9	1/3 →1/1	181	3	1792	2004	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
10	1/2 →1/1	181	3	1797	2409	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
11	1/3 →1/2 →1/1	173	3	1714	2192	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
12	1/4 →1/1	357	6	5779	5846	2.5	-8.6	40.4	-17.3	1.4	1.7	7e-3
13	1/4 →1/3 →1/1	177	3	1774	2052	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
14	1/4 →1/2 →1/1	179	3	1767	2334	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
15	1/4 →1/3 →1/2 →1/1	180	3	1779	2347	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2

Table 6.4: Results of MR to CT multi-scale matching for dataset B. For explanation, see Table 6.3.

4914 seconds, while a registration initialised with the result of the match at scale 1/2 (which has a maxdist of 0.2 *mm*) is done in 2013 seconds. Continuing with even finer scales can diminish the effect of a bad starting position. The bad initialisation is turned into a reasonable starting position by a match on a relatively coarse scale. The extra cost of convergence at this scale is marginal compared to the time requirements at the finer scales. For dataset B, for example, a match at scale 1/4 results in a mismatch. A registration at scale 1/1 using this mismatch converges slowly and the total time demand is 5846 seconds. When this registration is done via scale 1/3, the mismatch is compensated for at a coarse scale and the total registration takes only 2052 seconds.

Something that puzzled me occurred in the experiments on dataset A. Both the registration at scale 1/4 and at scale 1/3 resulted in bad solutions (maxdist was 1.7 *mm*). When the solution of scale 1/3 was used as the starting position for a match at scale 1/1, convergence took a long time (4914 seconds), as expected. However, doing the same with the solution of scale 1/4 as the initial position, was extremely fast (1854 seconds). I could not find a satisfactory answer to this puzzle. I could not see any difference in the traces of the solutions. I viewed the intermediate results of the registration process and did not find any major difference in convergence. In fact, the solution for the hierarchy $1/3 \rightarrow 1/1$ after two iterations, its final solution and the final solution of the hierarchy $1/4 \rightarrow 1/1$ were all very similar. I suspect that the final solution of the hierarchy $1/4 \rightarrow 1/1$ was just *within* the tolerance boundaries and the other hierarchy's solution after two iterations was just *outside* those boundaries, forcing the process to a third iteration.

In conclusion, an increase in convergence by a factor of nearly three can be achieved when using hierarchies including the full scale. The accuracy will be similar to that of a direct match at the full scale. The registration process can be accelerated by a factor of ten compared to the full scale timing when the images are only matched at scale 1/2. This gives the very acceptable maxdist of 0.2 *mm* for dataset A. For dataset B, however, the maxdist is 0.5 *mm*. Although this is still sub-voxel (the smallest voxel dimension is 0.71 *mm*), it is possible that the maxdist will be larger for other datasets. Further experiments are necessary before accepting this method.

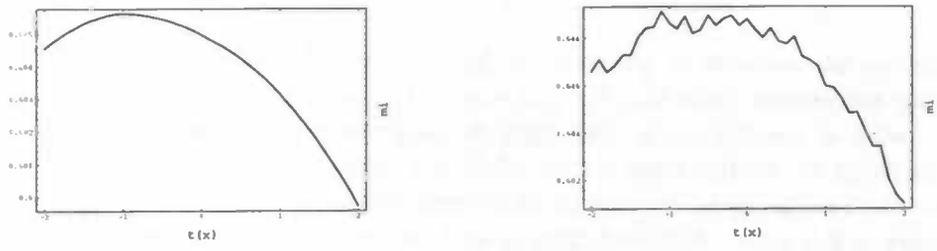
6.3.2 Using *sampling factors*

Instead of actually creating images at different scales, a multi-scale approach can also be realised by setting of the *sampling factors* (cf. appendix A). If the *sampling factor* for a certain dimension is set to x, one out of every x voxels in that dimension is used for registration. Thus, setting all sampling factors to 3 will result in registration at scale 1/3. This is not identical to reducing the images with *prima_reduce*, though. With *prima_reduce* the mean or maximum of a number of voxels is taken, while with sampling factors only the grey value of the sampled voxel is used.
Unless stated otherwise, the sampling factors are equal for all dimensions. In the experiments, the CT scan was the reference image and the MR scan the floating image. Maxdist denotes the maximal deviation in *mm* from the solution found when registering the images at full scale (i.e. sampling factors 1.0). Sampling factors of 1.0, 2.0, 3.0 and 4.0 were considered. Registration was performed both directly on those scales and for every

possible hierarchy of the four factors.

The results for multi-scale matching using sampling factors are shown in table 6.5 for dataset A and in table 6.6 for dataset B. Basically, the same conclusions as for the previous experiments can be drawn from these tables. An increase in convergence can be achieved by using a hierarchy of scales. This increase is influenced by the suitability of a solution to serve as starting position. When including the full scale, the factor of increase ranges from three to five. Unfortunately, this factor is not the same for identical experiments done on different datasets. There is not one optimal hierarchy. Again, registration can be done ten times faster when the full scale is not included. Although maxdist is then only 0.1 mm for both sets, I would still advise to test more datasets before accepting this method.

The registration of dataset A with sampling factors 4.0 results in a mismatch. Some of the parameters approach the correct solution, while others are completely wrong. The former is due to Brent's algorithm, the latter is the result of incorrect bracketing.
The solution found for parameter t_x (-5.2 mm), for example, approaches the correct solution (-6.9 mm). The bracketing routine finds an interval enclosing the global optimum, but Brent's algorithm does not converge to the precise optimum. Figure 6.4 shows an extreme close-up of the registration function near the global maximum. The function has many local extrema that Brent's method is sensitive to.



(a) sampling factors 1.0 (b) sampling factors 4.0

Figure 6.4: Close-up of traces for t_x near optimum, on dataset A.

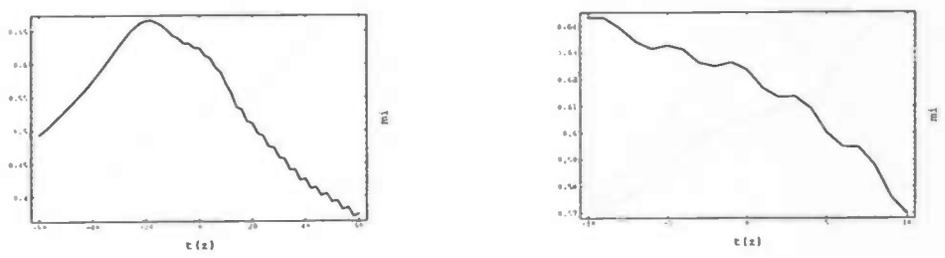
Parameter t_z of this match is an example of a solution that is completely wrong. The correct solution is -18.3 mm, the solution found is -0.8 mm. Traces of this parameter around the zero-position can be found in figure 6.5. That the optimum is near -18.3 mm is clear. The routine for initially bracketing an optimum finds the interval (-1.6,0.0), enclosing the local maximum at -0.8 mm. The sampling of the image has decreased the smoothness of the registration function and the routine for bracketing a maximum is sensitive to the tiny optima that occur.
A difference with the previous experiments, is that the final accuracy is not always equal to the accuracy of the finest scale. When the misregistration of dataset A with sampling factors 4.0 is used as the starting position for registration with sampling factors 3.0, i.e. hierarchy 4.0 $\rightarrow$ 3.0, the same mismatch is the result. A direct match with sampling

Nr	Sampling factors	NrEvals	NrIts	Timing	Total Time	Transformation($t_x, t_y, t_z, \phi_x, \phi_y, \phi_z$)						Maxdist
1	1.0	252	4	6027	6027	-6.9	-1.2	-18.3	10.5	2.7	2.6	0.0
2	2.0	240	4	537	537	-7.0	-1.2	-18.3	10.5	2.7	2.6	0.1
3	3.0	377	5	308	308	-6.9	-1.2	-18.3	10.5	2.6	2.7	0.1
4	4.0	342	4	148	148	-5.2	5.4	-0.8	0.3	0.0	2.9	24.6
5	3.0 →2.0	149	2	335	643	-7.0	-1.2	-18.3	10.5	2.7	2.6	0.1
6	4.0 →3.0	251	3	210	358	-5.0	5.8	-0.9	0.3	0.2	1.4	24.7
7	4.0 →2.0	244	4	550	698	-7.0	-1.2	-18.3	10.5	2.7	2.6	0.1
8	4.0 →3.0 →2.0	231	4	522	880	-7.0	-1.2	-18.3	10.5	2.7	2.6	0.1
9	3.0 →1.0	57	1	867	1175	-6.9	-1.2	-18.3	10.5	2.7	2.6	2e-2
10	2.0 →1.0	62	1	948	1485	-6.9	-1.2	-18.3	10.5	2.7	2.6	7e-3
11	3.0 →2.0 →1.0	73	1	1119	1762	-6.9	-1.2	-18.3	10.5	2.7	2.6	8e-3
12	4.0 →1.0	241	4	5862	6010	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-4
13	4.0 →3.0 →1.0	235	4	5764	6122	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-3
14	4.0 →2.0 →1.0	63	1	960	1658	-6.9	-1.2	-18.3	10.5	2.7	2.6	9e-3
15	4.0 →3.0 →2.0 →1.0	71	1	1086	1966	-6.9	-1.2	-18.3	10.5	2.7	2.6	9e-3

Table 6.5: Results for multi-scale matching with sampling factors for dataset A. For explanation, see Table 6.3.

Nr	Sampling factors	NrEvals	NrIts	Timing	Total Time	Transformation($t_x, t_y, t_z, \phi_x, \phi_y, \phi_z$)						Maxdist
1	1.0	361	6	5816	5816	2.5	-8.6	40.4	-17.3	1.4	1.7	0.0
2	2.0	418	6	659	659	2.5	-8.6	40.4	-17.3	1.4	1.7	0.1
3	3.0	460	6	267	267	2.6	-8.6	40.4	-17.4	1.4	1.6	0.2
4	4.0	554	7	168	168	2.5	-8.5	40.4	-17.2	1.4	1.6	0.2
5	3.0 →2.0	146	2	224	491	2.5	-8.5	40.4	-17.3	1.4	1.7	0.1
6	4.0 →3.0	162	2	93	261	2.5	-8.6	40.4	-17.3	1.4	1.6	0.1
7	4.0 →2.0	154	2	247	415	2.5	-8.6	40.4	-17.2	1.4	1.7	0.1
8	4.0 →3.0 →2.0	147	2	228	489	2.5	-8.5	40.4	-17.2	1.4	1.7	0.1
9	3.0 →1.0	132	2	1321	1588	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
10	2.0 →1.0	68	1	686	1345	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
11	3.0 →2.0 →1.0	57	1	570	1061	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
12	4.0 →1.0	126	2	1262	1430	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
13	4.0 →3.0 →1.0	134	2	1348	1609	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
14	4.0 →2.0 →1.0	65	1	651	1066	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
15	4.0 →3.0 →2.0 →1.0	62	1	621	1110	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2

Table 6.6: Results for multi-scale matching with sampling factors for dataset B. For explanation, see Table 6.3.



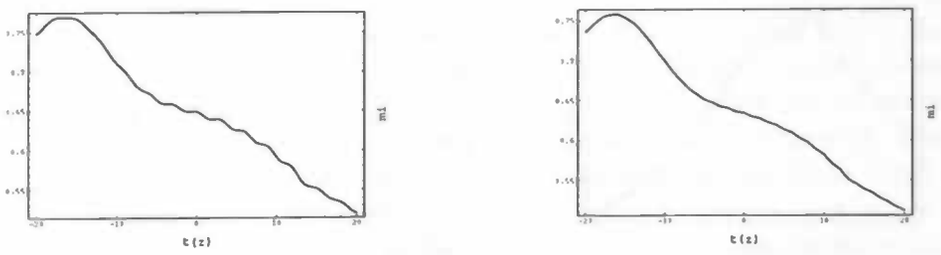
(a) -60 to 60 (b) -10 to 10

Figure 6.5: Traces for t_z around starting position on dataset A, sampling factors 4.0

factors 3.0, however, gives a fine solution with a maxdist of only 0.1 *mm*. Although the mismatch will be corrected at finer scales, the lengthy convergence at such scales will slow down the total registration process considerably. The hierarchy $4.0 \rightarrow 3.0 \rightarrow 1.0$, for example, takes even longer than the direct match with sampling factors 1.0. To avoid this carrying through of misregistrations to finer scales, it would help if mismatched scales could be excluded from hierarchies.

Interestingly enough, the hierarchy $4.0 \rightarrow 2.0$ *does* find a good solution. This difference can easily be explained by looking more closely at the registration function. As mentioned before, a mismatch occurs with sampling factors 4.0 as both the routine for initially bracketing a maximum and Brent's method end up in local maxima. If these results are used as a starting position for a match with sampling factors 3.0, the routine for initially bracketing a maximum starts looking for a maximum around the position found with sampling factors 4.0. As the registration function still has local optima here (see figure 6.6(a)), the routine cannot escape from the local maximum. Again, a mismatch is found.
When the results of sampling factors 4.0 are used for a match with sampling factors 2.0, the mismatch is corrected. The registration function at this scale is smooth (see figure 6.6(b)). There is no longer a local optimum near the starting position and the routine will look for another maximum. This time it finds the correct one.

When comparing the results of these experiments to those of the previous ones, the first thing that catches the eye are the differences in maxdist. In almost every experiment, the maxdist of the registration with sampling factors is smaller than that of matching with *prima_reduce* (with the exception of the total misregistration with sampling factors 4.0 for dataset A). Of the experiments where the performance with *prima_reduce* is better than with the sampling factors, the maxdists are so small that the differences are negligible. Some of the other differences are considerable, like the matches with sampling factors 3.0. For dataset A this results in a maxdist of 1.7 *mm* with *prima_reduce* and one of 0.1 *mm* when the sampling factors are used. For dataset B the maxdist is 3.8 *mm* as opposed to 0.2 *mm*. These differences in maxdist can be explained by the fact that *prima_reduce* blurs the images whereas registration with sampling factors uses the original images.



(a) sampling factors 4.0 to 3.0

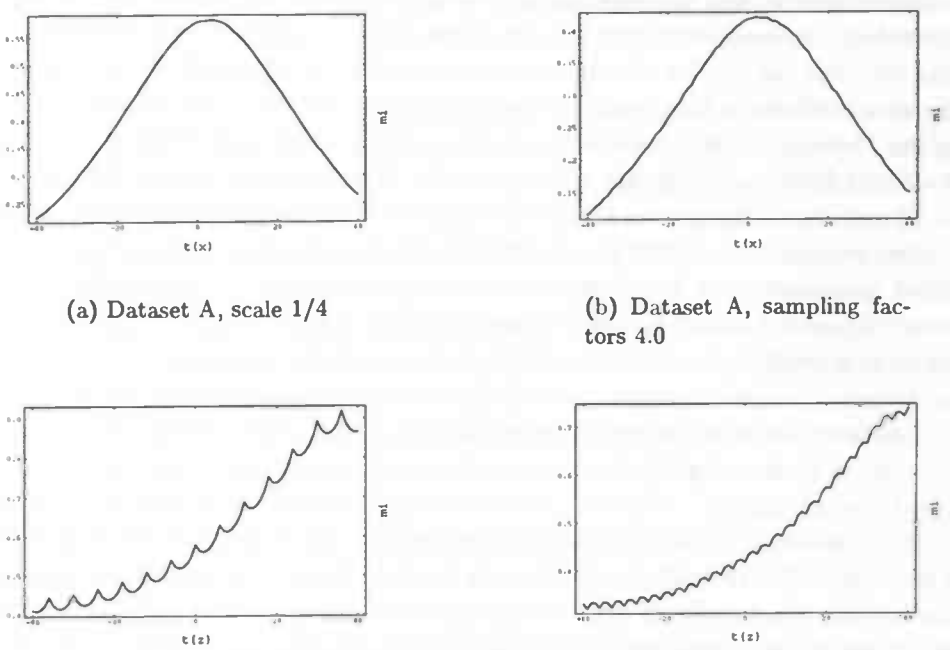
(b) sampling factors 4.0 to 2.0

Figure 6.6: Traces of t_z around starting position on dataset A.

Also striking is the fact that complete misregistrations occur with *prima_reduce* for the match at scale 1/4 for dataset A and with the sampling factors for the match with sampling factors 4.0 for dataset B. Obviously, it is not the case that one method always outperforms the other when registering at coarse scales.
This led me to taking a closer look at the registration function at different scales and with different reduction methods. Generally, there was little difference between the functions for the two reduction methods. At coarse scales local maxima and minima appeared - a sort of wave pattern - due to the rescaling. See, for example, figure 6.5. Increasing the reduction factor led to wave patterns with a lower frequency and a larger amplitude. For *prima_reduce* the functions were usually smoother than when sampling factors had been used to rescale the images, see figures 6.7(a) and (b). However, exceptions were not uncommon, as can also be seen in figure 6.7. Figures (c) and (d) show the same trace for different reduction methods. Clearly, this time the use of sampling factors resulted in less prominent optima. I have not found an explanation for this phenomenon .

Finally, there is the issue of speed. As the calculation of the scaled images by *prima_reduce* is not taken into account for the results, the experiments must be compared by the required number of evaluations of Brent's algorithm (NrEvals). From these numbers it would appear that MIRIT converges faster at fine scales when the sampling factors are set, while *prima_reduce* requires fewer evaluations at coarser scales. For the hierarchies, the performances cannot be compared by calculating the total number of evaluations as an evaluation at a fine scale requires more time than one at a coarse scale. By their time demands, however, it is clear that all the well performing hierarchies registering with sampling factors are faster than those using *prima_reduce*, even without increasing the timing results of the latter by the several hundred seconds needed for the computation of the scaled images. This suggests that the faster convergence at the finer scales with the sampling factors compensates for the extra time required at the coarser scales.

The differences in maxdist and timing as just described, make me draw the conclusion that registration with sampling factors is a faster and more accurate method.



(a) Dataset A, scale 1/4

(b) Dataset A, sampling factors 4.0

(c) Dataset B, scale 1/4

(d) Dataset B, sampling factors 4.0

Figure 6.7: Examples of the differences in registration function for different reduction methods.

6.3.3 Influences of the optimisation methods

Changing the parameters of the optimisation routines could influence the performance of MIRIT. Since there is only a small number of parameters that can be varied, the options are limited. However, there are three hypotheses for improving multi-scale matching.

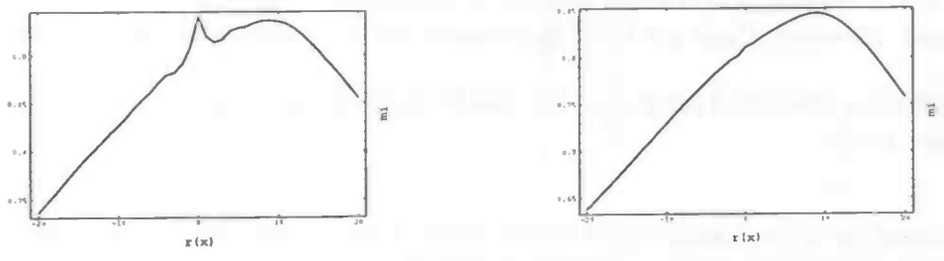
1. Enlarging the initial interval of the bracketing routine at coarse scales may prevent mismatches.

2. Aiming for a lower accuracy at coarse scales, thus leaving the precision work to finer scales, may speed up the registration process.

3. Given a good starting position, the initial interval of the bracketing routine can be small. When you are close to the correct solution, there is no point in searching in a large interval. Again, this should speed up the registration process.

To test the validity of these hypotheses, I have carried out several new experiments, varying the parameters of the optimisation methods. All the results can be found in Tables 6.7 and 6.8. For comparison, some previous results are included as well.

The first hypothesis is that mismatches at coarse scales can be avoided by enlarging the initial interval of the bracketing routine. At these scales the registration function contains many local extrema, as can be clearly seen in figure 6.5. A mismatch occurs, because the bracketing routine finds a maximum in the interval (-1.6,0.0). This can be overcome by enlarging the *step size*. The *step size* is used to calculate the initial interval. The size of this interval is 2.618×step size (see section 4.3.2). The step size should be large enough to pass local extrema. Then, even if the three points of the bracketing routine are in local extrema, their relation is conform the global behaviour of the registration function. When looking for a maximum, the function values of the points should be ascending.
Judging from figure 6.5, a step size of 2.0 should suffice. The corresponding experiment did indeed result in a good registration (see Table 6.7). I have further tried a registration with sampling factors 3.0 and step size 2.0 and also the same two experiments on dataset B. This to check whether a larger step size would not ruin a good match. From the results can be concluded that a slightly larger step size does not significantly alter the outcome of an already good match. However, determining a suitable step size without the aid of traces, is a problem. In fact, I doubt whether it is at all possible to choose a single step size per scale. For the mismatches I have tried to correct, some parameters required a step size of 6.0, whilst others were corrected with a step size of just 2.0. It depends on the steepness of the registration function and on the frequency and amplitude of the wave pattern. The question to be answered is how far apart the three points of the initial interval should be such that, even if the middle point is in a local maximum and the two outer points are in local minima, the function values are ascending.
Unfortunately, a mismatch cannot always be avoided by enlarging the step size. The misregistration of dataset B at scale 1/4, for example, could not be completely corrected. For several parameters the solutions continued to deviate, no matter how large the step size. The explanation can be found in the trace of one of these parameters, in figure 6.8. A new global optimum has appeared in the registration function, which is impossible to correct with a larger initial interval.



(a) Dataset B, scale 1/4

(b) Dataset B, scale 1/2

Figure 6.8: On the left is a trace for ϕ_x. It shows the situation in the second iteration, after the three parameters preceding ϕ_x (i.e. t_x, t_y, ϕ_z) have been optimised using a step size of 6.0. The corresponding trace for the images at scale 1/2 is shown on the right.

The second hypothesis states that the registration process can be accelerated by accepting a lower accuracy at coarse scales. This can be achieved by increasing the fractional

tolerance of Powell's method. This method is terminated when the difference between the function values at the beginning and end of an iteration is less than a fraction of their average value. The default value for this fraction is 1e-5.
I have tried several values for the fraction, matching dataset B (because no mismatches occurred with this set). I have also examined the performance of hierarchies that included less accurate coarse scale matches. The accuracy of a match is measured by the value of maxdist.
An acceleration of approximately 4% for the hierarchy $4.0 \rightarrow 1.0$ can be achieved by increasing the fractional tolerance to 1e-3 when registering with sampling factors 4.0. This yields a good match. Further increasing the fraction to 5e-3 will lead to a speed up of 7.5%. However, this fraction is dangerously close to 1e-2, which resulted in a registration of the hierarchy that was over three times *slower* than the original match. I would therefore conclude that no significant accelerations of the hierarchical registration process can be achieved by accepting lower accuracy at coarse scales.

The final hypothesis expresses the hope that the time demand of matching will decrease if the solution is sought in a small interval around the starting position, given that this starting position is close to the correct solution. I have tried to verify this assumption by registering both datasets with sampling factors 1.0, starting from the outcome of the match with sampling factors 2.0 (i.e. the hierarchy $2.0 \rightarrow 1.0$). Both experiments showed virtually no acceleration whatsoever. The number of evaluations was 61 versus 62 and 66 versus 68, respectively. By fitting a parabola, a large interval is quickly decreased because the steps taken are large. With a smaller interval, small steps are taken and the net result is the same.

The conclusion that can be drawn from these experiments is that only the first hypothesis seems to hold. By using a step size that is large enough to overcome local extrema in the registration function, misregistration is avoided. However, automatic determination of a suitable step size is not yet possible. Nor does this work in all cases.
Acceleration of multi-scale matching by adapting the parameters of Powell's optimisation method does not seem viable. It may be accomplished by other optimisation techniques.

Nr	Sampling factors	NrEvals	NrIts	Timing	Total Time	Transformation $(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$						Maxdist
1	1.0	252	4	6027	6027	-6.9	-1.2	-18.3	10.5	2.7	2.6	0.0
2	3.0	377	5	308	308	-6.9	-1.2	-18.3	10.5	2.6	2.7	0.1
3	4.0	342	4	148	148	-5.2	5.4	-0.8	0.3	0.0	2.9	24.6
4	4.0 →2.0	244	4	550	698	-7.0	-1.2	-18.3	10.5	2.7	2.6	0.1
5	4.0 →1.0	241	4	5862	6010	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-4
6	4.0 →2.0 →1.0	63	1	960	1658	-6.9	-1.2	-18.3	10.5	2.7	2.6	9e-3
7	4.0 using step size 2.0	383	5	155	155	-7.0	-1.3	-18.3	10.7	2.7	2.7	0.3
8	3.0 using step size 2.0	280	4	225	225	-6.9	-1.2	-18.3	10.4	2.6	2.7	0.2
9	exp 7 →2.0	149	2	329	484	-7.0	-1.2	-18.3	10.5	2.8	2.6	0.1
10	exp 7 →1.0	124	2	1885	2040	-6.9	-1.2	-18.3	10.5	2.7	2.6	2e-3
11	exp 9 →1.0	72	1	1095	1579	-6.9	-1.2	-18.3	10.5	2.7	2.6	1e-2
12	2.0 →1.0, step size 0.1	61	1	930	1467	-6.9	-1.2	-18.3	10.5	2.7	2.6	9e-3

Table 6.7: More experiments for multi-scale matching with sampling factors, dataset A. Above the dividing line are previous results (see Table 6.5), below the line are new experiments. If a parameter is mentioned, this parameter is only set to the given value in the last step of the hierarchy. For further explanation, see Table 6.3.

Nr	Sampling factors	NrEvals	NrIts	Timing	Total Time	Transformation $(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$						Maxdist
1	1.0	361	6	5816	5816	2.5	-8.6	40.4	-17.3	1.4	1.7	0.0
2	3.0	460	6	267	267	2.6	-8.6	40.4	-17.4	1.4	1.6	0.2
3	4.0	554	7	168	168	2.5	-8.5	40.4	-17.2	1.4	1.6	0.2
4	4.0 →2.0	154	2	247	415	2.5	-8.6	40.4	-17.2	1.4	1.7	0.1
5	4.0 →1.0	126	2	1262	1430	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
6	2.0 →1.0	68	1	686	1345	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
7	4.0 using step size 2.0	559	7	167	167	2.5	-8.5	40.4	-17.3	1.4	1.6	0.2
8	3.0 using step size 2.0	450	6	271	271	2.6	-8.6	40.4	-17.3	1.4	1.6	0.2
9	4.0, tolerance: 1e-3	392	5	116	116	2.4	-8.6	40.5	-17.3	1.5	1.7	0.2
10	4.0, tolerance: 5e-3	323	4	97	97	2.4	-8.6	40.4	-17.1	1.7	1.7	0.5
11	4.0, tolerance: 1e-2	163	2	49	49	3.0	10.3	39.6	11.2	1.2	-0.0	9.5
12	exp 9 →2.0	138	2	209	325	2.5	-8.6	40.4	-17.2	1.4	1.7	8e-2
13	exp 9 →1.0	127	2	1252	1368	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
14	exp 10 →1.0	123	2	1223	1320	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
15	exp 11 →1.0	234	4	4502	4551	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2
16	2.0 →1.0, step size 0.1	66	1	659	1318	2.5	-8.6	40.4	-17.3	1.4	1.7	3e-2

Table 6.8: More experiments for multi-scale matching with sampling factors, dataset B. Above the dividing line are previous results (see Table 6.6), below the line are new experiments. If a parameter is mentioned, this parameter is only set to the given value in the last step of the hierarchy. For further explanation, see Table 6.3.

Chapter 7

Discussion

7.1 Conclusions

Bearing in mind that all the experiments have been conducted on only two datasets, it is nonetheless possible to draw conclusions from their results. Especially about what is *not* feasible.

The experiments on morphologically preprocessed images showed no improvement of the registration process. A slight decrease of the information content of the images did not accelerate the registration significantly. When the images contained very little information, matching could even become slower and mismatches occurred. Furthermore, these methods are not yet fully automatic since the morphological operations still require user interaction.

Better results were obtained with the multi-scale methods. An acceleration of the registration by a factor ranging from three to five was achieved when the full scale was included in the hierarchy. When the finest scale consisted of images that were halved in size in all directions, the registration time was reduced by a factor of ten. In that case, the accuracy of the solutions was also slightly reduced, but still acceptable. That is to say, the deviation from the standard solution is small compared to the voxel sizes and thus, if the standard solution is acceptable, these solutions are as well. In general, the accuracy of hierarchical registration was equal to the accuracy of a direct match at the finest scale used. Exceptions occurred when the finest scale used was rather coarse (e.g. a reduction by a factor of three in all dimensions) and the previous scale in the hierarchy yielded a mismatch.
Mismatches arose at coarse scales due to the appearance of local optima in the registration function. The optimisation methods used are sensitive to local extrema. Such mismatches should be avoided, since they will slow down the registration at the next scale and can even cause more mismatches.
Of the two methods for scaling the images, *prima_reduce* and *sampling factors*, the registration using the sampling factors seems to yield the best results. The solutions were closer to the standard solution and the matching required less time.
Unfortunately, there is not one optimal hierarchy. Further experiments are necessary to find the best series of scales.

Adapting the parameters of the optimisation methods did not lead to significant improvements.

First of all, aiming for lower accuracy at coarse scales did not accelerate registration. Matching at fine scales requires so much more time compared to matching at coarse scales, that it seems preferable to approximate a correct solution as closely as possible at coarse scales.
Furthermore, the time demands of registration were not improved by searching the solution in a smaller interval, given a good starting position.
Finally, by enlarging the initial interval of the routine for bracketing a maximum, it is possible to avoid mismatches at coarse scales, but only in certain cases.

All in all, accelerating the matching process only seems to succeed by using a hierarchy of scaled images. However, the optimal hierarchy is left undecided and a way of avoiding mismatches is needed.

7.2 Further Research

To start with, there were two unexpected results that I came across while getting acquainted with MIRIT. The first thing I noticed was that trilinear interpolation sometimes clearly outperforms partial volume interpolation. This was the case with mono-modal matching of CT or MR images. When setting the sampling factors to 3.0, partial volume interpolation produced a mismatch while the result of trilinear interpolation was correct. For higher sampling factors the results of trilinear interpolation slowly deteriorated until, finally, the result of either method was equally wrong. I have witnessed this on many an occasion and it might be interesting to research.

The other intriguing phenomenon also involves the sampling factors. Setting these to fractions yielded considerably less accurate results than when whole numbers were used, both for mono- and multi-modal matching. For example, registration with sampling factors 1.5 resulted in a solution that was inferior to that of matching with sampling factors 2.0 and even with sampling factors 3.0.

On the subject of accelerating the registration process, there are several points that need more attention. Using the morphological operations described does not seem feasible, but multi-scale matching shows promise.

From the experiments it is evident that there is not one hierarchy that yields the best results in all cases. More experiments are required to determine a hierarchy that performs well on average.

The problem of mismatches at coarse scales can be partially solved by enlarging the step size, which determines the size of the initial bracketing interval. It should be further researched whether a suitable step size per scale is possible and, if so, what this step size is.
Another method of overcoming misregistration, is by implementing other optimisation techniques. MIRIT was not developed for multi-scale matching and the optimisation methods chosen are not resistant to local extrema. Possible choices are *simulated annealing* and *genetic algorithms*. These methods have a better chance of finding the global optimum, but the downside is a probable increase in computation time.

Simulated annealing assigns a probability to the move from the current position x to another, say x'. This probability is approximately defined by

$$p(x \to x') \approx e^{(f(x')-f(x))},$$

with f the registration function. If $f(x') \geq f(x)$, this probability is assigned the value 1 and if $f(x') \leq f(x)$, it is in the range (0,1). In other words, a step towards a higher function value is *always* taken, but *sometimes* the algorithm also moves in the direction of a lower value [12, 23].
A genetic algorithm starts with a *set* of positions, instead of just one. These are encoded to (binary) strings. In every iteration, a new generation of positions is created by mutation (changing the value of a string element) and crossover (creating new strings by exchanging parts of two strings). From this generation, a number of positions is selected by their fitness (in this case, their function value) and these are allowed to reproduce [13].

The only problem is the instance of scaling with *prima_reduce*, when a new global maximum was introduced. This will lead to a misregistration with any of the methods just described. It is therefore advisable to examine why this new global optimum appeared and whether there is a chance of such occurrences when using the sampling factors.

Appendix A

MIRIT: the manual

MIRIT is a program that registers two images using mutual information (as described in section 4.2). This manual gives a short overview of how to work with MIRIT, showing the possibilities of the program and the parameters that can be controlled. Not all of the input parameters are explained. These parameters are either self-explanatory or they are never changed. The emphasis is on the use of MIRIT for registration and tracing. A full explanation of the MIRIT software can be found in the official manual pages [18].

Running MIRIT

The program is run by calling the executable *Mirit* with an input file. This file, *MIRIT-input*, specifies all of the input to the program. Output is written both to a file and to the screen.

The input specification: MIRITinput

In this file the user defines the file containing the image data, the file containing the parameters, the mode of operation and the file in which to place the output. The following is an example of an input file.

```
MIRITimages              (* image data *)
MIRITparams              (* parameter specification *)
1                        (* the mode of operation *)
MIRIToutput              (* output *)
```

The mode of operation can be any of the numbers 1, 2, 3 or 4. These numbers denote:

1. Registration.
   The images in the file MIRITimages are registered by mutual information, using the parameters as specified in MIRITparams.

2. Tracing.
   A trace is computed, showing the mutual information value of the registered images as a function of changes in the transformation parameters.

3. Histogram.
   The joint intensity histogram of the images is computed.

4. Resampling.
   The reference image is resampled by the sampling factors as specified in MIRITimages. The resampled image is written to MIRIToutput.

The image specification: MIRITimages

The file MIRITimages contains the names of the image files, information about those images and several parameters relating to the images. The contents of this file are:

```
2                       (* number of images to be registered *)
1 2 3                   (* reference image orientation *)
256 256 10  0           (* reference image dimensions *)
0.9375 0.9375 1.55      (* reference image voxel sizes *)
0.0                     (* reference image gantry *)
CTimage.im              (* reference image filename *)
0                       (* reference image fileheader *)
1 2 3                   (* floating image orientation *)
256 256 180             (* floating image dimensions *)
0.9765 0.9765 1.0       (* floating image voxel sizes *)
0.0                     (* floating image gantry *)
MRimage.im              (* floating image filename *)
0                       (* floating image fileheader *)
0 255 0 255 0 179       (* floating image ROI *)
1.0 1.0 1.0             (* floating image sampling factors *)
```

The numbers denoting the *image orientation* are a code for the orientation of the image axes. The *image fileheader* specifies the number of bytes preceding the actual image data in the image file. The *ROI (Region Of Interest)* defines the part of the floating image that is taken into account when registering the images. The numbers denote the first and last coordinates of the region in the three dimensions. The sampling factors determine the sampling of the floating image. A sampling factor of n for a certain dimension means that one out of every n voxels in that dimension is sampled. For non-grid points nearest neighbour interpolation is used. Supersampling (i.e. factors ≤ 1) is also possible.

The parameter specification: MIRITparams

The parameters for MIRIT are specified in MIRITparams. This is a collection of the parameters for all the different operation modes. Not all parameters will be used in every operation mode, but they do always have to be defined. The contents of the file are:

```
0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0     (* starting position *)
4 5 3 1 2 6 0 0 0                       (* parameter order *)
5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0   (* number of steps
                                           and step sizes *)
4 256 256 3                             (* criterion parameters *)
```

```
100.0 1e-2 1e-5 50 50 1                (* optimisation
                                          parameters *)
```

The *starting position* denotes the initial value for each of the registration parameters, in the order $\phi_x, \phi_y, \phi_z, t_x, t_y, t_z, s_x, s_y, s_z$ (rotation, translation, scaling). The rotation parameters are in degrees and the translation parameters in millimetres. In registration mode, these parameters define the starting position of the calculations. When in tracing mode, they equal the centre point of the trace.
The *parameter order* is for the registration mode. It determines the order in which the parameters are optimised in Powell's algorithm. An entry of zero for a parameter means that it will not be optimised, but kept constant. In the example above, therefore, the scaling parameters are not changed and the others will be optimised in the order $t_x, t_y, \phi_z, \phi_x, \phi_y, t_z$.
The third line contains parameters for both tracing and registration. When tracing, the centre of the trace is the initial position. The mutual information value is computed at this position and at intervals of a *step size* around it. Each parameter has its own step size (in either millimetres or degrees). The *number of steps* (in this case 5) determines the length of the trace: $2 *$ the number of steps $+ 1$. For registration, only the step sizes are used (see section 4.3).
Of the *criterion parameters*, the first one is the registration criterion and the value 4 states that mutual information is to be used. The second and third parameters are the number of histogram bins for the reference and floating image, respectively. The final parameter selects the type of interpolation in the reference image. This is used when the transformation of a point in the floating image does not coincide with a point in the reference image. The possible values for this parameter are 1, 2 and 3, denoting nearest neighbour, trilinear and partial volume interpolation, respectively.
The *optimisation parameters* are part of either Brent's or Powell's algorithm, used in the registration (see section 4.3).

The output

MIRIT generates output both on the screen and in the specified output file. In both cases, the program first mirrors the input parameters and choices.
For the registration mode, this is followed by intermediate results of the optimisation algorithms on the screen. In the output file, however, the final results of the registration problem can be found, namely the *FloatToRefTransform* and the *RefToFloatTransform*. The former is the transform to be applied to the floating image to align it with the reference image and the latter registers the reference image to the floating image. Both transforms are represented in three forms: by the transformation parameters, by a transformation matrix in world coordinates and by a transformation matrix in image coordinates.
When in tracing mode, the output file contains the number of points in the overlapping volume, the marginal entropies, the joint entropy and the values of the registration criterion along the trace. On the screen, the current step along the trace and the mutual information value are shown, amongst others. These last two are needed to visualise the trace.

Bibliography

[1] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching. In *Proceedings of 5th Int. Joint Conference on Artificial Intelligence*, pages 659–663, 1977.

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *PAMI*, 14(2):239–256, Feb 1992.

[3] F. Betting and J. Feldmar. 3D-2D projective registration of anatomical surfaces with their projections. In *Information Processing in Medical Imaging*, pages 275–286. Kluwer, 1995.

[4] F. Betting, J. Feldmar, N. Ayache, and F. Devernay. A new framework for fusing stereo images with volumetric medical images. In *Computer Vision, Virtual Reality, and Robotics in Medicine*, volume 905 of *Lecture Notes in Computer Science*, pages 30–39. Springer-Verlag, 1995.

[5] J. Bijhold. Three-dimensional verification of patient placement during radiotherapy using portal images. *Medical Physics*, 20(2):347–356, 1993.

[6] G. Borgefors. An improved version of the chamfer matching algorithm. In *Proceedings of 7th Int. Conference on Pattern Recognition*, pages 1175–1177, 1984.

[7] R.P. Brent. *Algorithms for minimization without derivatives.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[8] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. In *International Conference on Computer Vision*, pages 549–556. IEEE Computer Society Press, 1995.

[9] D.S. Fritsch, S.M. Pizer, B.S. Morse, D.H. Eberly, and A. Liu. The multiscale medial axis and its applications in image registration. *Pattern Recognition Letters*, 15:445–452, 1994.

[10] K.G.A. Gilhuijs, P.J.H. van den Ven, and M. van Herk. Automatic three-dimensional inspection of patient setup in radiation therapy using portal images, simulator images, and computed tomography data. *Medical Physics*, 23(3):389–399, 1996.

[11] K.G.A. Gilhuijs and M. van Herk. Automatic on-line inspection of patient setup in radiation therapy using digital portal images. *Medical Physics*, 20(3):667–677, 1993.

[12] P.J.M. Laarhoven and E.H.L. Aarts. *Simulated annealing: theory and applications.* D. Reidel Publishing Company, Dordrecht, 1987.

[13] M.M. Lankhorst. *Genetic algorithms in data analysis.* PhD thesis, University of Groningen, 1996.

[14] L. Lemieux, R. Jagoe, D.R. Fish, N.D. Kitchen, and D.G.T. Thomas. A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs. *Medical Physics*, 21(11):1749–1760, 1994.

[15] A. Liu, E. Bullit, and S.M Pizer. 3D/2D registration using cores of tubular anatomical structures as a basis. Unpublished.

[16] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[17] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multi-modality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 1996. In press.

[18] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. *Multi-modality Image Registration using Information Theory. MIRIT Manual Pages.* University Hospital Gasthuisberg, Herestraat 49, 3000 Leuven, Belgium, April 1996.

[19] J.B.A. Maintz. *Retrospective registration of tomographic brain images.* PhD thesis, Utrecht University, November 1996.

[20] J.B.A. Maintz, P.A. van den Elsen, and M.A. Viergever. Evaluation of ridge seeking operators for multimodality medical image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):353–365, 1996.

[21] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[22] G.P. Penney, J.A. Little, and D.J. Hawkes. 2D-3D image registration for use in image guided interventions. Unpublished.

[23] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[24] J. Serra. *Image analysis and mathematical morphology*, volume 1. Academic, San Diego, CA, second edition, 1988.

[25] C. Studholme, D.L.G. Hill, and D.J. Hawkes. 3D registration by multi-resolution optimisation of mutual information. Unpublished.

[26] P.A. van den Elsen, J.B.A. Maintz, E.J.D. Pol, and M.A. Viergever. Automatic registration of CT and MR brain images using correlation of geometrical features. *IEEE Transactions on Medical Images*, 14(2):384–398, 1995.

[27] P.A. van den Elsen, E.J.D. Pol, T.S. Sumanaweera, P.F. Hemler, S. Napel, and J. Adler. Grey value correlation techniques used for automatic matching of CT and MR brain and spine images. In *Visualization in Biomedical Computing*, volume 2359 of *Proc. SPIE*, pages 227–237. SPIE Press, Bellingham, WA, 1994.

[28] P.A. van den Elsen, E.J.D. Pol, and M.A. Viergever. Medical image matching– a review with classification. *IEEE Engineering in Medicine and Biology*, 12(1):26–39, 1993.

[29] M.A. Viergever, J.B.A. Maintz, R. Stokking, P.A. van den Elsen, and K.J. Zuiderveld. Matching and integrated display of brain images from multiple modalities. In *Medical Imaging*, volume 2434, pages 2–13, SPIE, Bellingham, 1995.

[30] W.M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1, 1996.

[31] W.M. Wells III, P. Viola, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. In *Medical Robotics and Computer Assisted Surgery*, pages 55–62. Wiley, 1995.