Project in Electrical Engineering, 15hp **October 12, 2014**
Department of Neuroscience, Uppsala University

Automatic Recognition of Regions of Interest in Synthetic MRI

- Application to the Measure of Relaxation Times in Patients with Normal Pressure Hydrocephalus

Ola Söderström
Ylva Winsnes

Abstract

The goal of this project has been to develop a program for automatic recognition of regions of interest with a graphical user interface to facilitate its use. The algorithms are design for research on normal pressure hydrocephalus and aimed at measurements on eight predefined areas of interest. The purpose with the program is to avoid co-registration of images that differs substantially from a standard brain due to the nature of the NPH symptoms. The data calculated by the algorithm is presented in two tables and allows for easy copy-and-paste into a spread sheet program for further analysis. A total of 25 patients with 7 different scans and the same number of control persons have been processed.

Contents

1 Introduction

Magnetic resonance imaging (MRI) is often used in the study of the brain since the technique allows for detailed study of soft tissue structure, which is difficult to achieve with for example a computer assisted tomography scan (CT scan) [4]. One of many fields where the MRI is used is in the research on idiopatic normal pressure hydrocephalus, iNPH. iNPH is a state where the ventricles of the brain are enlarged and the amount of cerebrospinal fluid (CSF) is increased without, as the name iNPH suggests, an increase in pressure in the ventricle system. This state causes, inter alia, issues with the balance and walking abilities of the patient. Commonly, a patient's state can be improved if the amount of CSF is decreased. Patients that can be helped in this manner would profit from surgery where a shunt implant is performed to allow for continuous tapping of the CSF.

At the Department of Neuroscience at Uppsala University research is conducted on different aspects of iNPH and on the causes for the symptoms. Specifically, the value of a CSF tap test as a preoperative diagnostic is examined together with the result of shunt implantation on the state of the patients [7]. This research involves brain scans with MRI of a number of patients at occasions prior to and after a tap test as well as after a potential operation. In total, this results in up to seven scans of each patient. In addition, healthy control persons are investigated corresponding to each patient in age and gender. From the data obtained by the MRI, particular regions of interests (ROIs), in the brain are measured and analyzed to document any trends and changes in the tissue. These ROIs can vary in shape depending on their position in the brain, but it is important that they are placed in the anatomically same place in each scan session and in every patient. The mean values of all voxels in each ROI, together with the standard deviations, will give comparative values for each patient.

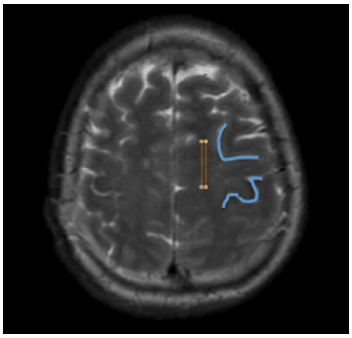
1.1 Purpose

The purpose of this project is to create and place different ROIs in parts of the brain that has been suggested by a neuroradiologist. These ROIs, see Figure 1, will have simple shapes that vary in size depending on the area where they are positioned. Since the task would require to place 14 ROIs in approximately 50 patients at different scanning occasions, resulting in a total of over 2600 ROIs, a program that can automatically place these ROIs is favorable. There might be arguments for using more complex shapes so that the areas covered with the ROIs more closely resemble the areas of interest, but because of the huge number of ROIs to be placed and to increase the repeatability the structures of the ROIs will remain simple. The automatic feature of the program also increases the repeatability of the positioning of the ROIs and reduces errors due to the human factor in the ROI's positioning. After the ROIs are placed in the correct regions, data are to be extracted in each region and presented in a table that can be transferred to e.g. a spreadsheet program for further analysis.

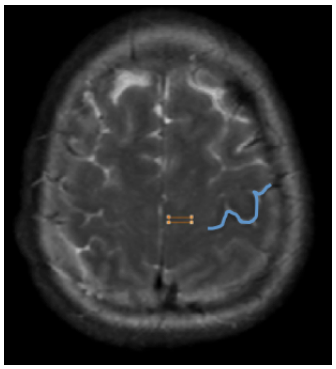
1.1.1 Description of the Regions of Interest

The ROIs are numbered and shown in Figure 1 $(a - h)$. ROI 1-3, 6,7 and 12 are all bilateral. All ROIs should contain only white matter, except ROI 6 that should contain only grey matter.

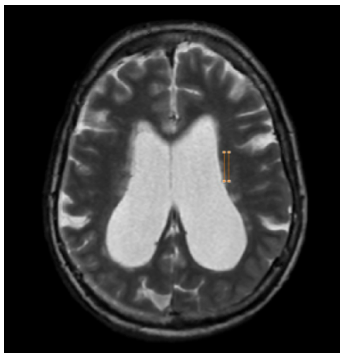
In the following text the first slice of a scan sequence will refer to the slice closest to the patients neck and the last slice will refer to the one at the top of the head. ROI 1, marked as an orange rectangle in Figure 1a, is supposed to lay in a slice above the last slice with visible side ventricles. Two points of reference are the L-shape and the upside down omega (Ω) shape as shown in blue and the aim is to measure superior frontal white matter. ROI 2 should be positioned one slice above ROI 1 close to the inter hemispheric fissure and located in the medial precentral gyrus, in level with the omega-shape, see Figure 1b. ROI 3 should be located in white matter adjacent to the mid portion of the lateral ventricles according to Figure 1c. In addition to the mean value, a horizontal gradient is also of interest for this ROI. ROI 6 should measure in the head of the caudate nucleus, see Figure 1d. In Figure 1e ROI 7 is shown in the white matter anterior to the frontal horns of the ventricles and similarly to the case with ROI 3 a vertical gradient should be calculated. ROI 10 is to be placed in the genu of the corpus callosum where the dark part is at its most prominent. The region is shown in Figure 1f. ROI 11 should be located in the splenium of the corpus callosum, that is, in the dark saddle shaped structure immediately outside the back of the ventricles as indicated in Figure 1g. Lastly, ROI 12 should be located in the topmost slice with clearly visible ventricles, in the body of the corpus callosum between the middle line and the separated ventricles, see Figure 1h. In all cases where the ROIs are situated close to the ventricles, care must be taken so as not to include any part of the ventricles in the ROI since this would distort the measurements.



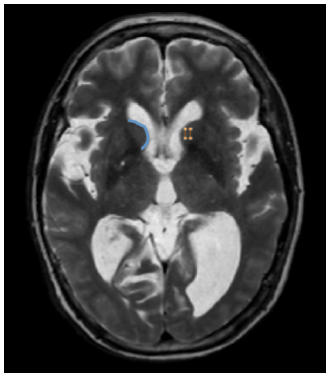
(a) ROI 1



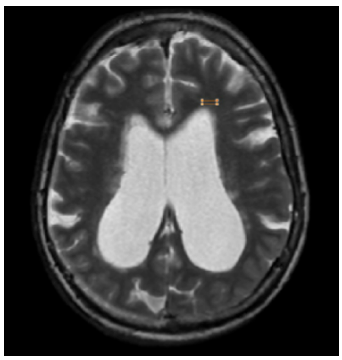
(b) ROI 2



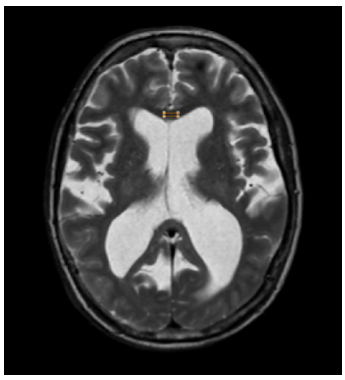
(c) ROI 3



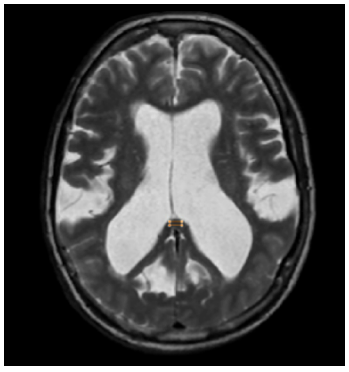
(d) ROI 6



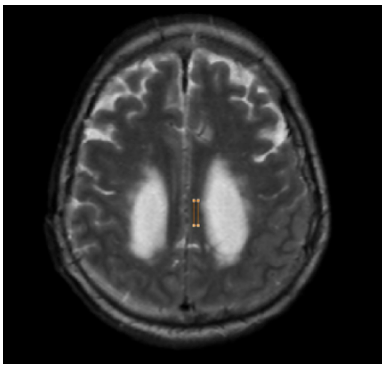
(e) ROI 7



(f) ROI 10



(g) ROI 11



(h) ROI 12

Figure 1: The regions of interest, *ROIs*, to be positioned in the images. The orange markers signifies the ROIs and the blue marks are guide lines to find the correct positions.

2 Theory

The theory section will briefly describe how MRI works, how the signal is created and how an image is obtained from the signal as well as which parameters are of interest when it comes to retrieving the data. Since our script uses several algorithms for image analysis, the most important of these will be described at the end of this section.

2.1 Magnetic Resonance Imaging

The MRI technique takes advantage of a phenomenon called *nuclear magnetic resonance* to create an image of an object and it is normally the density and spatial distribution of protons that is measured to visualize anatomic structures and functioning. The MRI technology was developed in the 1970' following Raymond Damadian's discovery that tumorous tissue exhibits longer relaxation times than does normal tissue. Damadian was also the one who built the first MRI scan and produced the first image by means of nuclear magnetic resonance. Following this discovery, Paul Lauterbur proposed to use magnetic field gradients to select the targeted area of imaging, i.e the frequency encoding, and Peter Mansfield developed the phase encoding. Lauterbur and Mansfield were awarded with the Nobel Prize in Physiology or Medicine in 2003 for their contributions. [2]

The MRI technique provides a mean of imaging that is safe for the organism in the way that it does not make use of any hazardous radiation or instable nuclei, but exploits the properties of the stable hydrogen nuclei. [4]

2.1.1 Physical Background

All elementary particles, hadrons and atomic nuclei have a quantum mechanical property called *spin*. Spin can be seen as an intrinsic angular momentum and gives rise to the property of *magnetic moment*. Spin is a quantized property and for protons it takes the half integer value $\pm 1/2$. Because of Pauli's exclusion principle [5] protons with positive and negative spin will couple. This leads, in the case of an atomic nucleus, to that an odd number of protons and neutrons sum up to a nonzero spin whereas the spin of an even number sums to zero. Particles with nonzero spin therefore interact with an external magnetic field, and it is these interactions that are exploited in the MRI.

Body tissue contain large quantities of water and therefore an abundance of protons. Because of the charge of the proton and its movement it creates a circular current that produces a magnetic dipole. In the free state, the spin vector of every proton in a sample point in a random direction so that the net magnetization is zero. If a proton is placed in an external static magnetic field $\mathbf{B}_z$, the magnetic moment, $\boldsymbol{\mu}_p$, of the proton tilts in the plane due to the torque $\boldsymbol{\tau} = \boldsymbol{\mu_p} \times \mathbf{B_z}$ and creates an angle to the z-axis. This is called *spin precession* and the angle is *the precession angle*, θ. The spin now has a component $\boldsymbol{\mu}_{xy}$ in the x and y directions rotating around the axis of the external magnetic field with the frequency ω_L called *the Larmor frequency*. This is illustrated in Figure 2. [4]

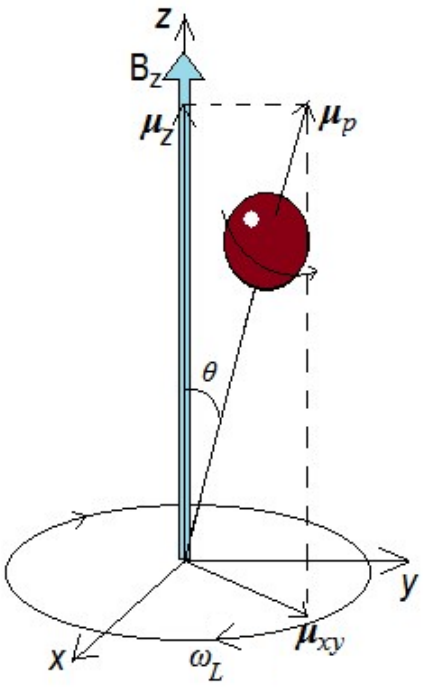


Figure 2: A proton with magnetic moment $\boldsymbol{\mu}_p$ in a static magnetic field $\mathbf{B}_z$ precesses at an angle θ and with the frequency ω_L.

The interaction energy due to $\mathbf{B_z}$ is the difference compared to the situation without magnetic field, that is $E = -\boldsymbol{\mu}_p \cdot \mathbf{B_z}$. This interaction causes the proton to change into one of its two possible energy states $\pm\mu_p B_z$. Because of the Boltzmann's distribution the two states are unequally populated with a slight favor for the $+\mu_p B_z$-state [9]. The magnetic moment gets two components, $\boldsymbol{\mu}_z$ and $\boldsymbol{\mu}_{xy}$ respectively parallel and perpendicular to the static field, see Figure 2. Because of the rotation of the protons, the $\boldsymbol{\mu}_{xy}$ component of each proton cancels out. In a sample with many protons this will give a resultant magnetization $\mathbf{M}_z$, but $\mathbf{M}_z$ will still be difficult to measure because it is parallel to $\mathbf{B}_z$. The Larmor frequency depends on the magnetic field strength $B_z = |\mathbf{B_z}|$ and the gyromagnetic ratio, γ, according to

$$\omega_L = -\gamma B_z. \tag{1}$$

If a linearly polarized electromagnetic wave, *radio frequency* (RF), with the same frequency as the Larmor frequency is applied to the sample in the xy-plane, some or all protons, depending on the energy of the pulse, will absorb energy corresponding to $\mu_p B_z$ and become excited. What happens can be described if the observer places herself in a rotating coordinate system R' with z' axis parallel to the fixed reference system R. R' rotates with the Larmor frequency. From the point of view of R', the linearly polarized electromagnetic wave can be described as two circularly polarized fields one rotating clockwise and one rotating counterclockwise around z'. This creates a time-independent component $\mathbf{B}_{x'}$ in the direction of the x'-axis, rotating in phase with the pre-

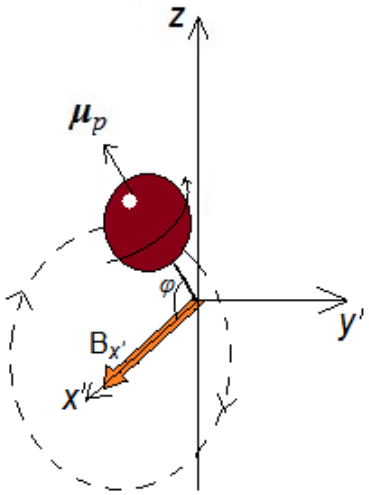


Figure 3: The precession of the proton during the application of a transverse magnetic field in the x'-direction seen from a reference system that rotates counterclockwise around the z-axis of the lab frame. Because of the torque from the additional magnetic field the proton now precesses around the x'-axis with the angle φ in addition to the precession in z-direction shown in Figure 2.

cession of the proton around the z-axis. As an effect, a torque $\boldsymbol{\tau}_{R'} = \boldsymbol{\mu_p} \times \mathbf{B}_{x'}$ causes the proton to precess around the x'-axis as well. An attempt to illustrate this is shown in Figure 3.

The total magnetic field acting on the protons in *the rotating frame* is then

$$\mathbf{B}_{tot} = (B_{x'}, 0, B_z - \omega/\gamma). \tag{2}$$

The radio frequency can be engineered to excite the protons to 'flip' their spins through a predefined angle, for example π radians, making them antiparallel to their initial direction. After a period of time, the protons will flip back to their initial state and re-emit electromagnetic radiation. The time it takes until the proton de-excites is called the *relaxation time* and will be described in detail in Section 2.1.2. If the pulse has enough energy to excite all protons with an angle π, the resultant magnetization $\mathbf{M}_z$ is completely inverted and becomes $-\mathbf{M}_z$. This kind of pulse is usually called a π-pulse because of the effect is has on *all* the spin angular momentum vectors. On the other hand, if the pulse only carried enough energy to excite *exactly half* of the protons, the magnetic moment $\mathbf{M}_z$ becomes equal to zero and a component $\mathbf{M}_{xy}$ perpendicular to the static magnetic field will occur because of the phase coherence between the protons. A pulse of this energy is called a $\pi/2$-pulse.

Immediately after the protons have been excited in this fashion, interactions between protons and the surroundings cause them to loose energy and fall back,

or relax, to their original state. This is called *free inductive decay* (FID) and generates a signal in a receiver coil through the emitted electromagnetic field. The FID is initially at its maximum value and consists of only one phase. The FID is generated by two interactions, the *spin-spin relaxation* and the *spin-lattice relaxation*. The spin-lattice relaxation is caused by energy loss to the surrounding molecules in the lattice and spin-spin relaxation is due to a loss of phase coherence originating from interactions between protons.

The physical phenomenon of nuclear magnetic resonance describes this ability of nuclei to absorb and re-emit electromagnetic radiation and the emitted energy is a measure of the proton density in a volume unit.

2.1.2 Properties of the Relaxation Times

When the RF pulse is switched off the protons emit a DC signal when $\mathbf{M}_z$ returns to its equilibrium value and a AC signal from the change in $\mathbf{M}_{xy}$. The time it takes for $\mathbf{M}_z$ to return to a factor of its previous value is referred to as the T_1 *recovery* (7) while the time for the $\mathbf{M}_{xy}$ to return to a factor of its previous value (that is, zero) is called T_2 *decay* (8). The T_1 recovery is the spin-lattice relaxation mentioned above and is a function depending on the initial $\mathbf{B}_0$ field, here chosen to be in the z-direction. The T_1 value varies for different tissues and is the time it takes for $\mathbf{M}_z$ to return to $1 - 1/e$, i.e 63 %, of its maximum value.

The magnetization on a macroscopic scale in a material is described by the Bloch equation [3]. If we let the nuclear magnetization be
$\mathbf{M}(t) = (M_x(t), M_y(t), M_z(t))$ and the total magnetic field when the RF pulse is applied be $\mathbf{B}_{tot} = (B_{RF}cos(\omega_0 t), B_{RF}sin(\omega_0 t), B_0)$ in the lab-frame, the Bloch equation is written

$$\frac{d\mathbf{M}}{dt} = \gamma \mathbf{M} \times \mathbf{B}_{tot} - \frac{M_x + M_y}{T_2} + \frac{M_0 - M_z}{T_1} \tag{3}$$

with the effects of the relaxation times taken into account. M_0 is the magnetization in z-direction when B_0 is applied. This can be written in component form as

$$\begin{aligned}
\frac{dM_x}{dt} &= \gamma \left[M_y B_0 + M_z B_{RF} sin(\omega_0 t) \right] - \frac{M_x}{T_2} \\
\frac{dM_y}{dt} &= -\gamma \left[M_x B_0 - M_z B_{RF} cos(\omega_0 t) \right] - \frac{M_y}{T_2} \\
\frac{dM_z}{dt} &= -\gamma \left[M_x B_{RF} sin(\omega_0 t) - M_y B_{RF} cos(\omega_0 t) \right] + \frac{M_0 - M_z}{T_1}
\end{aligned} \tag{4}$$

If solved with the initial condition that $B_{RF} = 0$ immediately after the RF pulse is turned of, the solutions to (4) are [2]

$$
\begin{aligned}
M_x(t) &= \left[M_x(0)cos\omega_0 t + M_y(0)sin\omega_0 t\right] e^{\frac{-t}{T_2}} \\
M_y(t) &= \left[M_y(0)cos\omega_0 t + M_x(0)sin\omega_0 t\right] e^{\frac{-t}{T_2}} \\
M_z(t) &= M_z(0)e^{\frac{-t}{T1}} + M_0\left[1 - e^{\frac{-t}{T_1}}\right].
\end{aligned} \tag{5}
$$

Adding the initial condition of a $\pi/2$-pulse along the x' axis we get $M_x(0) = 0$, $M_y(0) = M_0$ and $M_z(0) = 0$ and

$$
\begin{aligned}
M_x(t) &= M_0 sin(\omega_0 t)e^{\frac{-t}{T_2}} \\
M_y(t) &= M_0 cos(\omega_0 t)e^{\frac{-t}{T_2}} \\
M_z(t) &= M_0\left[1 - e^{\frac{-t}{T_1}}\right].
\end{aligned} \tag{6}
$$

When solved for T_1 and T_2 we get

$$
T_1 = -t \cdot ln\left(1 - \frac{M_z(t)}{M_0}\right)^{-1} \tag{7}
$$

and

$$
T_2 = -t \cdot ln\left(\frac{M_{xy}(t)}{M_0}\right)^{-1}. \tag{8}
$$

In (7) and (8) the absolute value of the magnetization vectors are used and M_{xy} represents the magnitude of the component in the xy-plane. The value of T_2 is always shorter than T_1. Both T_1 and T_2 are properties that are better distinguishable between different tissues than the proton density (PD) in itself and because of this it is usually the T_1 and T_2 values that form the MR image [8]. The T_1 and T_2 relaxation times show how quickly the spin regrows to its "equilibrium" value. The inverse relaxation times $R_1 = 1/T_1$ and $R_2 = 1/T_2$ are often used in applications.

2.1.3 Spatial Encoding

The FID signal from the proton de-excitation constitutes the MR-signal. To deduce from what region in a scanned subject (for example the patient) a signal originates, there need to be spatial information contained in the signal. This spatial encoding is often realized in two steps, firstly by a slice selection to delimit the space in the z-direction and secondly an application of two consecutive magnetic field gradients that gives a spatial frequency in the xy-plane.

To describe the spatial encoding, we introduce a coordinate system with z-axis along the length of the patient laying horizontally in the scanner. In order to restrict the MR signal to a two dimensional plane, or perform a so-called *slice selection*, a magnetic field gradient $\mathbf{g}_z$ is applied simultaneously with the

static field $\mathbf{B}_z$ which has the effect that the total magnetic field $\mathbf{B}_{z,tot}$ varies along the length of the patient,

$$\mathbf{B}_{z,tot} = \mathbf{B}_z + z\mathbf{g}_z. \tag{9}$$

The situation is illustrated in Figure 4a. Thus, the force acting on the protons varies along the z-axis and according to (1) the Larmor frequencies will vary. To identify the signal in the xy-plane, a magnetic gradient $\mathbf{g}_y$ is applied in the y-direction. This gradient is applied only for a limited period of time, entailing a phase shift between the protons when it is switched off. In this manner, a *phase encoding* is performed that permits a localization of the signal origin in the y-direction, see Figure 4b. Finally, a constant gradient $\mathbf{g}_x$ is applied similarly to $\mathbf{g}_z$ and with the same result on ω_L but with the variation along the x-axis. This *frequency encoding* is shown in Figure 4c. Equations similar to (9) can be set up for the x and y directions. With this combination of gradients a unique frequency and phase can be assigned to each volume element in the slice.

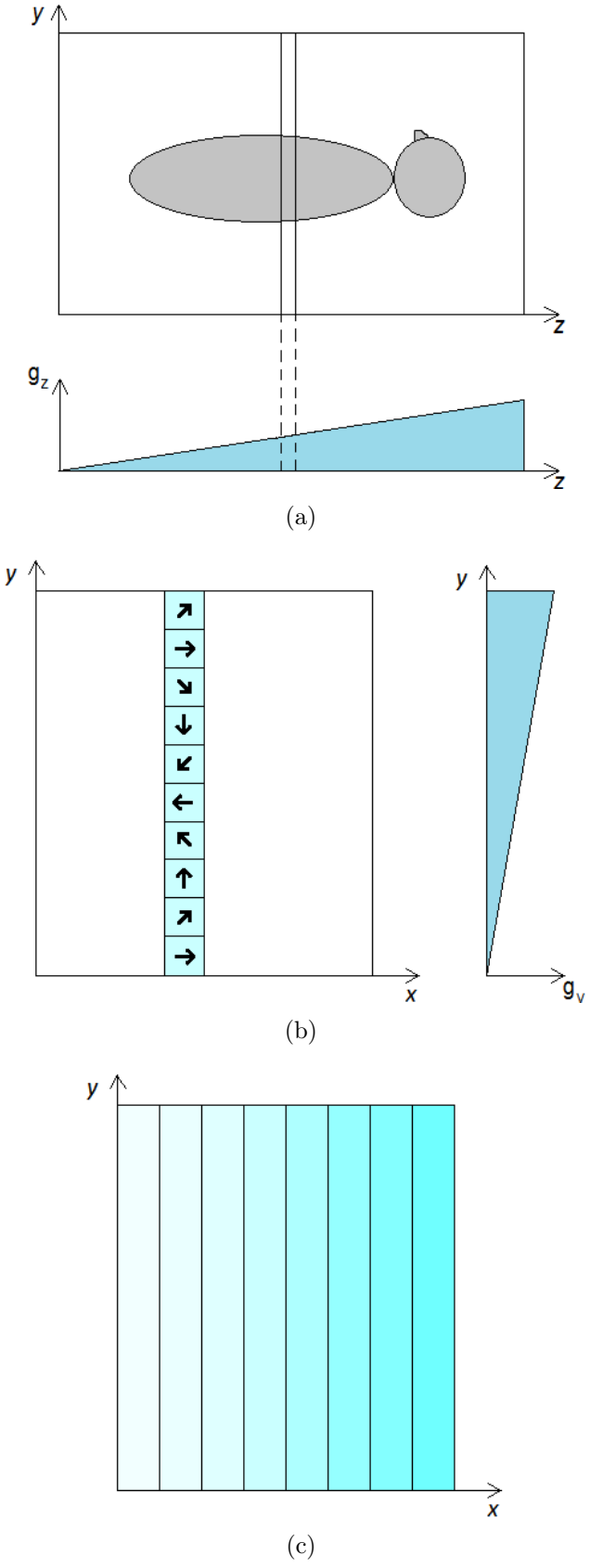


Figure 4: The spatial encoding is obtained in three steps. a) Slice selection by gradient in the z-direction. b) Phase encoding with a gradient in the y-direction. c) Frequency encoding with a gradient in the x-direction.

2.1.4 Generating the Signal

For reasons described in Section 2.1.1 the protons in the body tissue need to be excited by means of a RF pulse with a carefully selected bandwidth corresponding to the Larmor frequency in the particular slice of interest in order to produce a signal in the MR scan. In fact, a sequence of pulses are performed and this sequence is repeated several times. The procedure is shown in Figure 5. Firstly, a $\pi/2$-pulse is applied to excite half of the protons and produce a magnetization in the xy-direction, $\mathbf{M}_{xy}$. Immediately after the pulse a milliseconds long $T_2{}^*$ decay makes the FID signal too brief to be readily measured. After a time $2t$ when the protons are out of phase and $\mathbf{M}_{xy}$ is small a π-pulse is applied. This causes the faster moving leading protons to become laggers behind the slower moving ones. A phase coherence of the protons will soon be in place and $\mathbf{M}_{xy}$ has regrown to its instantaneous maximum value. The signal is detected at this moment, just before the dephasing T_2 decay causes the FID signal to disappear again. After $t = T_1$, $\mathbf{M}_z$ is restored and the cycle is repeated with a different phase encoding gradient in order to obtain a high resolution in the final image. For a resolution of 512×512 pixels, 2×512 repetitions with increasing phase encoding gradient need to be performed [1]. With each gradient value, a particular spatial frequency of the phases will be detected.

The measured T_2 is therefore the result of a spin-echo pulse sequence, which basically makes it possible to measure the amplitude of T_2 after $3t$, which corresponds to the amplitude of $T_2{}^*$ at the time t. The resulting signal is essentially two FID signals back to back and appears as an "echo" of the original signal. The signal will be reduced in amplitude by T_2 decay, and the *time of echo* (TE) is what determines the weighting of the T_2 value in MRI. After $3t$, or TE, the $\pi/2$ RF signal is repeated and the time for such a cycle is referred to as the *repetition time* (TR). Since TR determines how far M_z will recover between signals, this is the parameter controlling the T_1 weighting in MRI. To create a PD weighted signal, a large TR is used (reducing T_1 effect on contrast) together with an as short as possible TE (reducing the T_2 effect on the contrast).

The three-dimensional MR image is a three-dimensional Fourier transform of the spatial spin density function. (In an image, the three-dimensionality will only be present by a slice thickness but not actually spread out along the z-axis). We denote the spatial spin density $\rho(\mathbf{r}) = \rho(x, y, z)$. A single measured resonance signal is $s(\mathbf{r}, \mathbf{t}) = s(x, y, z, t_x, t_y, t_z)$ where t_x, t_y and t_z are the times of the respective gradients and can be obtained by solving the relevant Bloch equations [1]

$$s(\mathbf{r}, \mathbf{t}) = M_0 cos\left[(\Delta + \eta_x x)t_x + (\Delta + \eta_y y)t_y + (\Delta + \eta_z z)t_z\right] e^{-(t_x+t_y+t_z)/T2}. \tag{10}$$

where $\eta_k = -\gamma g_k$ and $\Delta = -(\gamma B_0 + \omega_1)$. ω_1 is the frequency of the phase sensitive detection by which the signal is detected. The signal from the entire slice is

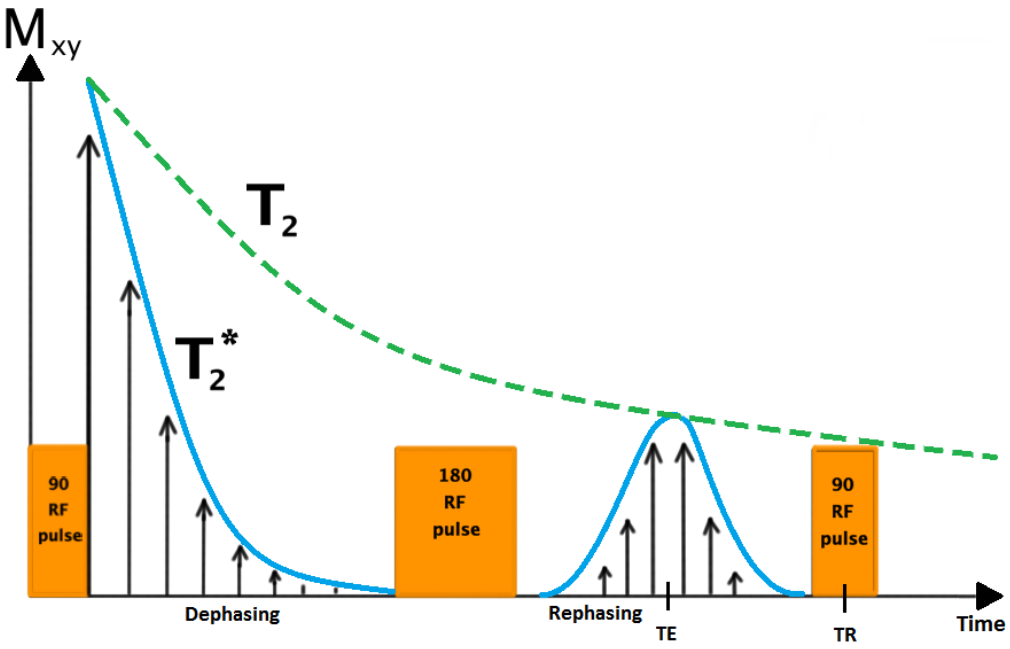


Figure 5: The magnetization in the xy-plane over time for a spin-echo sequence.

$$s(\mathbf{t}) = \iiint \rho(\mathbf{r}) s(\mathbf{r}, \mathbf{t}) dv \tag{11}$$

and by means of the Fourier transform it can be shown [1] that the final result is a filtered spin density function,

$$\bar{\rho}(\mathbf{r}') = \frac{1}{2} \iiint \rho(\mathbf{r}) G(\eta_x(x - x')) G(\eta_y(y - y')) G(\eta_z(z - z')) dv \tag{12}$$

where $G(w)$ is the complex line shape function

$$G(\omega) = \frac{M_0/T_2}{(1/T_2)^2 + \omega^2} + i \frac{M_0 \omega}{(1/T_2)^2 + \omega^2}. \tag{13}$$

2.2 Algorithms

In image analysis, the first task often comes down to segmenting the image to allow for a partition into constituting objects and regions [10]. There are several different methods and approaches available to achieve this image segmentation. We have used both morphological operations and spatial filtering in our program. In this subsection some of the more frequent operations and algorithms are described.

2.2.1 Morphological operations

Morphological operations can be used to extract components from an image that are useful in the description of a region. These operations can help to find shapes such as boundaries, convex hulls and skeletons. Morphological processes are based on the movement of so called *structural elements* (SEs) over the predefined set of points in a binary image. The SEs are binary matrixes that probes the whole image and compares every pixel with neighboring pixels and, depending on the method, checks if some predefined criteria are fulfilled. A new binary image is then created with 1s at the positions where these conditions are fulfilled and 0s where they are not.

Two of the most basic morphological operations are *erosion* and *dilation* which are used to erode and dilate image components respectively. The mathematical description of the erosion of the image A by the SE B, here defined as $A \ominus B$, can be written

$$A \ominus B = \{z|(B)_z \subseteq A\} \tag{14}$$

where both A and B are sets contained in the 2-D integer space Z^2 containing the points z. In the same way, the dilation of A by B, here defined $A \oplus B$, can be written as

$$A \oplus B = \{z|(\hat{B})_z \cap A \neq \emptyset\} \tag{15}$$

where $\hat{B}$ denotes B reflected about its origin and $\emptyset$ is the empty set. As can be understood from (14) and (15), the shape of B is what determines what the resulting image will looks like.

Closing and *opening* are two other basic morphological processes that consists of a combination of erosion and dilation. Opening is erosion followed by dilation and closing is dilation followed by erosion. They are often used to smooth contours of objects to make structures either connect or disconnect. Closing may also be used to eliminate small holes and fill gaps. The mathematical description of closing, $A \bullet B$, and opening, $A \circ B$, are defined as

$$A \bullet B = (A \oplus B) \ominus B. \tag{16}$$

and

$$A \circ B = (A \ominus B) \oplus B \tag{17}$$

Boundary extraction is another very common and useful tool when analyzing images and as the name implies it is used to find boundaries of objects in a binary image. To find the boundary of the set A one can make use of (14),

$$\beta(A) = A - (A \ominus B) \tag{18}$$

where $\beta(A)$ denotes the boundary of A. That is, (18) describes the subtraction of the eroded set from the original set.

A morphological process that makes use of (15) is *hole filling*. Holes are defined as pixels in binary image with value 0 surrounded by pixels with value 1. As the name suggests, the hole filling algorithm fills the holes giving the pixels values 1 instead of 0. When using this algorithm one first have to create an empty array X_0 with the same size as the array containing A. Locations in X_0 corresponding to a given point in each hole in A is set to 1 and the following algorithm can then be used to fill all the holes

$$X_k = (X_{k-1} \oplus B) \cap A^c, \qquad k = 1, 2, 3, ... \tag{19}$$

The set X_k now contains all the filled holes and together with A they contain both the holes and the boundaries. A^c denotes the complement of A and limits the process to fill only the holes and not the entire image.

In a similar way as in (19) dilation can also be used to find *connected components* in a binary image. This time a location on X_0 is set to 1 if it corresponds to any pixel in the connected component in A and the following expression is then used to find all connected components.

$$X_k = (X_{k-1} \oplus B) \cap A, \qquad k = 1, 2, 3, ... \tag{20}$$

Note that one pixel in the connected component must be predefined for this algorithm to work.

Finding convex parts of images are often useful in object descriptions and has proven useful in our program as well. The finding of convex parts is based on something called the *hit-or-miss transformation* which is a tool for shape recognition. Consider the set A with three subsets C, D and E so that $A = C \cup D \cup E$. If we want to find the location of one of these subsets, say E, we will have to erode A by E and then erode A^c by $Y = W - E$ where W is a small window enclosing E. This will leave only the center of gravity of E left and the location of E has been found. These steps can be written with one single equation where B is a set composed of E and its background,

$$A \circledast B = (A \ominus E) \cap (A^c \ominus Y). \tag{21}$$

If we let $B = (B_1, B_2)$ be SEs where B_1 is associated with the objects and B_2 be associated with the background (in the example above $B_1 = E$ and $B_2 = Y$) (21) can be rewritten in a more general way as

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2). \tag{22}$$

(22) is referred to as the *hit-or-miss transformation*. If we want to find the convex part of A, we will have to use four different SEs, B^i where $i = 1, 2, 3, 4$. The hit-or-miss transformation is iteratively applied to A with B^1, and when no further changes occur the union with A is performed and the result is called D^1. Then the procedure is repeated with B^2 and so on. The union of all four Ds are the convex hull of the set A. The equation for this procedure can be written, with $X_0^i = A$ as

$$X_k^i = (X_{k-1} \circledast B^i) \cup A, \qquad i = 1, 2, 3, 4 \qquad k = 1, 2, 3, ...$$

When $X_k^i = X_{k-1}^i$ we let $D^i = X_k^i$ and the convex hull, $C(A)$, is then

$$C(A) = \bigcup_{i=1}^{4} D^i. \tag{23}$$

For a more in depth description of the hit-or-miss transformation and the finding of convex hulls, see [10].

The last of the morphological methods that is explained in this section is the finding of the *skeleton* of an object. Skeletons are binary images that mostly preserves the connectivity and extent of an original object in an image but removes most of its original foreground. One can look at it as a kind of extreme thinning algorithm. The procedure for obtaining the skeleton of an image is also quite simple and uses the relations in (14) and (17). It can be shown [6] that the skeleton of A, $S(A)$, can be written as

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \tag{24}$$

where K is the last iterative step before A becomes an empty set and

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B.$$

When this procedure has been carried out, only a skeleton image of the original structures is left. The skeleton algorithm is well explained in [10].

2.2.2 Filtering

Filtering in image processing is of two main types. Spatial filters are based on pixel-by-pixel operations, such as an averaging filter. Frequency filtering on the other hand uses the Fourier transform of the image to enhance or suppress certain image components. We will focus on spatial filtering here since that is what we have used in our scripts.

For point, line and edge detections different filters are used but they all share the common goal: to calculate spatial derivatives. One can either use first or second order derivatives ((25) and (26) respectively) depending on the desired output.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{25}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x). \tag{26}$$

For example, if the spatial second derivate around the point x is used to check for changes in the pixel values it would give a large outputs if the change is sudden and small outputs if the change is "smooth".To calculate the derivatives a filter mask is created with element values w_k which depend on the method

used, in what spatial direction the derivative is interesting, and what type of structures the filter is supposed to find. The idea is to sum the products of the filter mask coefficients and the intensity values in the mask's region. The *response* of the mask, R, can thus be calculated as

$$R = \sum_{k=1}^{9} w_k z_k \tag{27}$$

where z_k is the intensity value of the pixel corresponding to the kth element of the mask. When finding the edge of an object, both the edge strength and direction in each point is of importance and a handy tool for this is to use the gradient ∇f:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \tag{28}$$

∇f will point in the direction of the greatest rate of change of f at the point (x, y). The direction can be calculated by the angle $\alpha(x, y)$ (with respect to the x-axis)

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right) \tag{29}$$

and the magnitude $M(x, y)$ of the vector ∇f, i.e. the value of the rate of change, is then calculated with

$$M(x, y) = \sqrt{g_x^2 + g_y^2} \tag{30}$$

When the goal is to find a pout, the Laplacian is a good tool since it is an isotropic derivative operator, making the filter mask rotation invariant.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{31}$$

When the mask operates on the image, points are said to be detected if the absolute value of R exceeds a specified threshold and a binary image is then created where detected points are labeled 1 and otherwise 0.

2.2.3 Hough transform

The *Hough transform* is used with unstructured environments when there are little to no knowledge of where objects of interest might be located, which means that in these cases points have to be accepted or denied depending on predefined global properties. The Hough transform is a way of finding points that lie on curves of specified shapes. Say that we want to find subsets of points in an image that lie on straight lines. First consider a point (x_i, y_i) that lies in the xy-plane. The equation of a straight line can be applied to this point

$$y_i = kx_i + m \tag{32}$$

with an infinite number of lines passing through (x_i, y_i) for different values of k and m. We can rewrite this in parameter space which yields the following equation for a single line for the fixed pair (x_i, y_i)

$$m = y_i - kx_i. \tag{33}$$

If another point (x_j, y_j) is also considered in parameter space, the lines associated with (x_i, y_i) and (x_j, y_j) will intersect at some point (k', m') unless they are parallel. k' is then the slope and m' the intersection of the line containing both (x_i, y_i) and (x_j, y_j) and it can be understood that all points on this line (in the xy-plane) have lines in parameter space that intersect at (k', m'). Thus, if all points in the xy-plane are examined, one could find those lying on a straight line by identifying points in parameter space where a large number of parameter-space lines intersect. A problem would be if k approaches infinity (a vertical line). To avoid this problem the normal representation of a line is used

$$\rho = x\cos\theta + y\sin\theta \tag{34}$$

where ρ is the perpendicular distance between the origin and the line and θ represents the angle between the x-axis and ρ, positive in counterclockwise direction. Writing (34) in parameter space, that is $\rho\theta$-space, the lines going through (x_i, y_i) and (x_j, y_j) will have a sinusoidal form and the point where they intersect (ρ', θ') will correspond to a line that passes through both points. This method is quite useful because of the relatively low number of computations needed. The number of computations is linear to the number of non-background points, n, in the xy-plane. Other methods involves finding all lines determined by every pair of points, involving $n(n-1)/2 \sim n^2$ operations, and then find all subset points that are close to the particular lines, involving another $(n)(n(n-1))/2 \sim n^3$ operations. [10]

Another benefit of the Hough transform is that it is theoretically applicable to any function of the form $g(\mathbf{v}, \mathbf{c}) = 0$ where $\mathbf{v}$ is a vector vector of coordinates and $\mathbf{c}$ is a vector of coefficients. Thus, for points lying on the circle

$$c_3^2 = (x - c_2)^2 + (y - c_1)^2 \tag{35}$$

the same approach as the one for finding a line can be used, the major difference is presence of the three parameters in $\mathbf{c}$ resulting in a 3-D parameter space.

2.2.4 Bresenham's line algorithm

Bresenham's line algorithm is an algorithm that approximates a straight line in a lattice where the starting and end points are known. Consider the starting point (x_0, y_0) and the end point (x_1, y_1). A general equation for the line through these points can be written

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}$$

$$\Rightarrow y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0 \tag{36}$$

The ideal y-value for successive integer values of x can be computed since the slope, according to (36), only depends on the end points' coordinates (x_0, y_0) and (x_1, y_1). Starting at y_0, the next y-value is computed and the process is repeated until the entire line between the end points are found. The algorithm tracks a small error value, defined as the vertical distance between the exact y-values and the rounded values at the current x-value. For every step when x is increased by 1 the error is increased by the slope and if it exceeds $1/2$ the y-value is increased by 1. The error is then decreased by 1 and the process starts over again with the next x.

3 Method

When brain images from different MR scanning sequences are compared, co-registration is often used to ensure that the measurements are made in the same regions in all of the image material [7]. The co-registration is a normalization procedure of each scan to a sample standard brain and this includes rotation and translation of the images which distorts the pixel values to various extents. In the case of iNPH, the co-registration procedure can give unreliable results because the anatomy differs substantially from a healthy brain. Our approach in this project will therefore be to rotate and adapt the regions of measurement, i.e the ROIs, to account for movements between consecutive scans.

To be able to define the ROIs and extract data from the MR images a MATLAB program is designed. The data used for analyzation comes from synthetic MRI scans and is acquired from the program SyMRI (developed by SyntheticMR AB). Q-maps (data representing T_1, T_2 and PD images) and an intracranial mask is extracted from SyMRI and used in our program. The images acquired is in DICOM format which works well in MATLAB. A GUI is created so that it is easy to keep track of all ROIs, extract the data and save all progress. With the GUI it is easy for someone not familiar with the code to operate the program and place the ROIs.

3.1 Placing of ROIs

The positions of the ROIs are of great importance since it is the pixel values acquired from these ROIs that will form the result when the program is used. The program will be used to compare the mean values of each ROI between scan sequences at different occasions and it is thus essential that the ROIs are placed at the same position based on the anatomy of each individual brain. The idea is to let the program place one of the predefined ROIs in a chosen slice in the first scan of each patient. This is done by the function *getROI* and is explained more thoroughly below. To get information about translation and rotation between scan sequences, the positions of the eyes are used (see the function *eyeFind*) as reference points and the ROIs placed in the first scan is then moved so that they correspond to the anatomy of the current scan. Below follows a brief explanation of functions mainly used in *getROI* to determine the positions of the ROIs. Later subsections will describe how the position of each ROI is found in more detail.

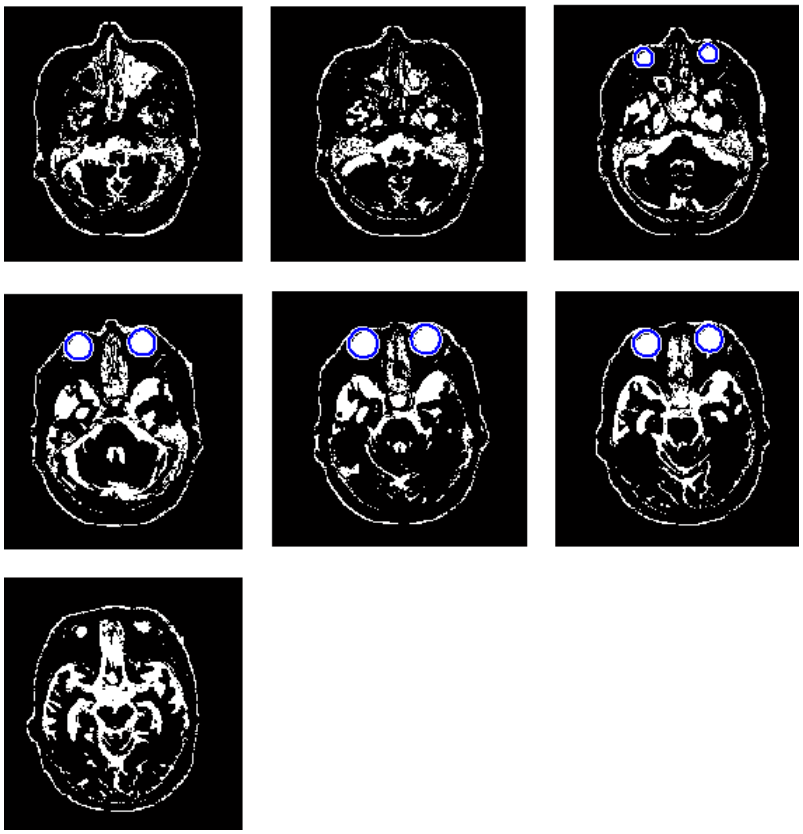


Figure 6: The seven first slices of a scan set is used to determine rotation and translation of the brain by comparing the positions of the eyes. The blue circles are found using the Hough transform.

The function *eyeFind* uses the Hough transform to localize circles corresponding to the eyes. For the seven lowest slices *eyeFind* tries to localize eyes in the upper part of the images. The mean position of the left and the right eyes respectively are then used as a reference to know how the brain is located in the scan sequence. From this procedure it is also possible to decide the rotation and translation between scans. To make sure that the algorithm finds the eyes, a sequence of dilation, erosion and other morphological processes and segmentations is carried out. A typical result of the algorithm can be seen in Figure 6 where the blue circles marks the eyes that the algorithm has found.

The function *ventricleSegmentation* is used when the shape and/or the position of the ventricles are of importance. The desired slice is first morphed with the processes dilation and majority so that the ventricles are pronounced and consistent. The function then finds all connected components to the center pixel in the image and these pixels are then assumed to together form the ventricles in the selected slice.

The *xyVertical* function is designed to find the longitudinal cerebral fissure. It uses the Hough transform to find the line in the desired slice after the image has been segmented with different filters and morphed to create a clear skeleton representation of the image. The output from the function is the x and y-coordinates of the endpoints of the line found with the Hough transforma-

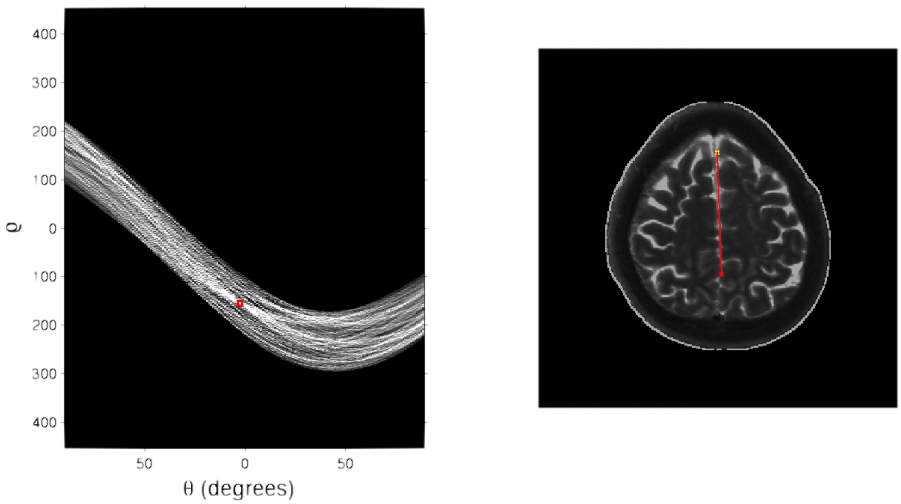


Figure 7: Left: The result of the Hough transform to parameter space with the strongest point marked with a red square. Right: The two endpoints of the strongest line marked with crosses and connected with a line in the MR image.

tion. To the left in Figure 7 the parameter space used in the Hough transform is shown with the strongest point marked with a square and to the right in the same figure these points, marked with crosses, have been connected with a line in the MR image.

drawLineVert is a function that extends and creates the line found in *xyVerticle*. To do this it uses Bresenham's line algorithm which is executed in the function *bresenham*. In the same sense, *drawLineHorz* extends and creates a "horizontal" line based on the position of the eyes. It uses *bresenham* to create the line much like *drawLineVert*. The horizontal line is then placed at the first pixel (in y-direction) where the vertical line coincide with the ventricles. The line is then systematically moved downwards until the vertical line and the ventricles are not overlapping. In each step the horizontal line counts the number of pixels in the ventricles that it coincides with and when finished, the horizontal line is placed at the level with fewest overlapping pixels. These steps are carried out because it is often necessary to place the horizontal line where the ventricles are thin, see for example ROI 3 where the ROIs are positioned adjacent to the mid portion of the lateral ventricles. This function has a special case when used in ROI 12 where it finds and places the line at a height of the bilateral ventricles shown in a slice at the top of the head.

3.1.1 ROI 1

These ROIs are supposed to be placed bilaterally above the ventricles in the superior frontal white matter. The algorithm that finds and places the ROIs uses image segmentation of the ventricles in a slice where the ventricles are prominent and finds the center of the ventricles at both sides of the brain. Each ROI is about 40x3 pixels and is placed with its center at the same coordinates in the xy-plane as the center of the ventricles found in some slice below. The algorithm then checks so that no CSF is contained inside the ROI and moves

the ROI towards the center of the brain one pixel at a time if the criteria is right. When checking for CSF the binary image of the slice is used and CSF is supposed to have value 1 in these images.

3.1.2 ROI 2

ROI 2 are to be placed in the medial precentral gyrus. There are no ventricles present in the slices where these ROIs are placed which makes it difficult to create a fail-safe algorithm. The algorithm finds where the skull is broadest and places the ROIs bilaterally one ROI width away from the central line which is found with the help of *xyVertical* and *drawLineVert*. The ROIs are about 3x13 pixels and no considerations are taken for CSF because of the difficulty of placing the ROIs in the first place.

3.1.3 ROI 3

ROI 3 are primarily meant to measure the spatial gradient of the relaxation times and PD in a radial direction in the white matter adjacent to the mid portion of the lateral ventricles. The ROIs are placed just outside the thinnest part of the ventricles in a slice below where they divide. To find this place the ventricles are segmented and a horizontal line is created by use of *eyeFind* and *drawLineHorz* so that the rotation of the patient's head is taken into consideration and the line is placed where the ventricles are thin. The ROIs, each 5x9 pixels, are then placed at the center of the horizontal line and stepped outwards in both directions until they do not contain any CSF in a way similar to the one described for ROI 1. The gradient is calculated in the function *getData* and is further described in Section 3.2.

3.1.4 ROI 6

ROI 6 differs from the other ROIs in several ways, e.g. the ROI has a shape to resemble a circle instead of a square and it does not rotate between scans. The reasons for this is that the position of the ROI is at the head of the caudate nucleus which is quite narrow and the circular shape allows for a better fitting of the ROI to the anatomic structure than a square shaped ROI would have. To find where to place the ROIs the algorithm first divides the MR slice into four frames based on the vertical and horizontal lines found with the functions *drawLineVert* and *drawLineHorz* and then uses the two uppermost frames as areas of interest. The ventricles are found with *ventricleSegmentation* and the convex part is found using the MATLAB function *regionprops*. The ROI is created to resemble a circle with a total of 25 pixels and the function places the ROI in the middle of the convex area found in the algorithm.

3.1.5 ROI 7

ROI 7 is aimed at investigating the spatial gradient in the white matter anterior to the frontal horns of the ventricles. The ventricles are segmented in the same slice as in ROI 3 and the uppermost pixels are then found bilaterally. The ROIs are 9x5 pixels and placed at the top of the ventricles so that no CSF is present inside the ROI. The gradient is then calculated in the function *getData* which is further described in the Section 3.2.

3.1.6 ROI 10

ROI 10 is designed to be placed in the genu of corpus callosum. The ROI is created to be the 3x11 pixels and is first placed in the centre of the vertical line found with *drawLineVert*. It then paces upwards along the vertical line (negative y-direction) one pixel at a time until it does not contain CSF.

3.1.7 ROI 11

ROI 11 is placed in the splenium of corpus callosum and this algorithm works in the exact same way as ROI 10, but is moved downwards along the vertical line (positive y-direction) instead of upwards. The ROI is 3x9 pixels.

3.1.8 ROI 12

ROI 12 should be place in the body of corpus callosum, that is, between the ventricles and the interhemispheric fissure in a slice below the slice where the ventricles disappear. The ventricles are found in a similar manner as in ROI 1 and the ROIs are then placed with their center between the ventricles and the central line. *drawLineHorz* have a special case when the position of ROI 12 is to be found. The horizontal line is in this case placed in level with the visible ventricles in the selected slice, see the function *drawLineHorz* for more details. The dimension of the ROIs are 21x3 pixels.

3.2 Extraction of data

The extraction of data is done by the function *getData*. The mean value of all pixels are calculated with the MATLAB function *mean* and the number of pixels used is also calculated. The standard deviation, σ, is then calculated using the MATLAB function *std* which uses the standard formula

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where x is the values of each pixel in the ROI and n is the number of pixels. The mean value and σ is calculated for T_1, T_2, PD, R_1 and R_2 and the data is presented in the GUI and updated when a ROI is moved. The data is also stored in a structure called *subjectStats* that can be used to present the data for all ROIs at the same time. The table where this is presented is created in the function *saveTo_ tableData* (headers created in *create_ tableData*) and the

idea is that the user can export this table to a spreadsheet program for further analyzations.

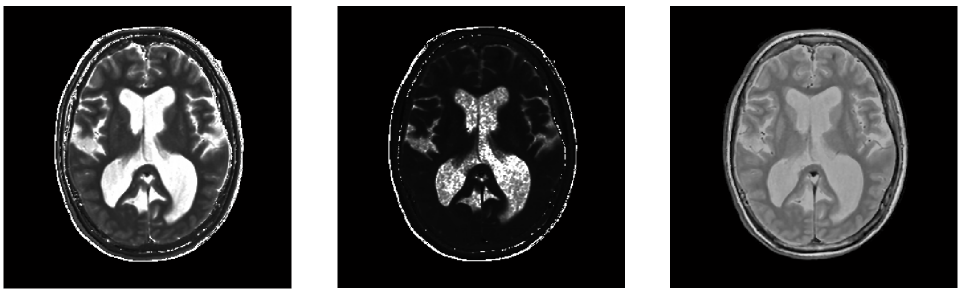
In the cases of ROI 3 and 7 the spatial gradients in different directions are of interest and special cases are used when these ROIs are selected to calculate the desired gradients. The gradients are supposed to be calculated in a radial direction out from the ventricles, but since the ROIs are thin and placed at positions directly above and to the side of the ventricles, the radial direction is approximated as the x and y-direction in ROI 3 and 7 respectively. Since the brain and thus also the ROIs might be rotated with an angle the algorithm first re-rotates the ROIs (with the pixel values) back so that they have a square shape. When this is done, mean value and σ of each row or column (depending on ROI) are calculated and saved in the structure *subjectStats* so that the data can be presented in a separate table.

4 Result

A *graphical user interface*, GUI, has been built in Matlab were the ROIs are semi-automatically inserted. The user chooses subject, ROI and the suitable slice and the program automatically places the selected ROI in the desired slice based on predetermined preferences. The position of the ROI can be seen as a suggestion and the user can move the ROI if the position is considered incorrect. When the next scan of the same patient is chosen, the program compares the current scan sequence with the first sequence to see if the objects are translated or rotated compared to the first scan. If that is the case, the ROIs are automatically rotated and translated the same way to allow to place the ROI in a position as close as possible to the one in the first scan. The user still has the possibility to translate the ROIs manually one pixel at a time if the ROIs are not placed satisfactory. When a ROI has been placed the mean value and standard deviation of each ROI is calculated and presented in the GUI. If desired, all data that has been acquired can be presented in a separate table where it is easy to transfer the data to a spreadsheet program for further analysis.

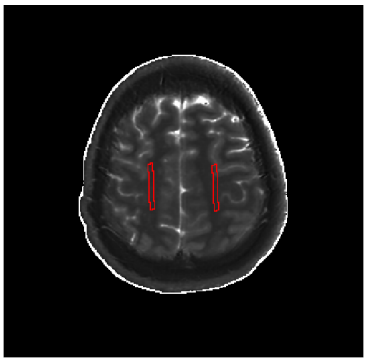
To give better means to assure that the ROIs are placed in the area of interest, the three different image maps obtained from the synthetic MRI scan can be consulted. These are the T_1, T_2 and PD maps. In Figure 8 these three maps are shown, and in the GUI it is easy to change the setting to view the ROI on the preferred map.

The results of the automatic ROI positioning is shown in Figure 9 for all ROIs in one scan sequence. In this example no manual manipulation has been done to correct the positions. For examples of ROIs before and after adjustment see Figure 10 and 11. All seven scans for one patient are shown in these two figures and the adjustment in this example mainly consists of level adjustments as in Figure 10a-10f whereas lateral adjustment can be seen between Figure 10g and 10h. In Figure 11e and 11f an artifact from the shunt implant distorts the image in scan 7 but the algorithm properly places the ROI.

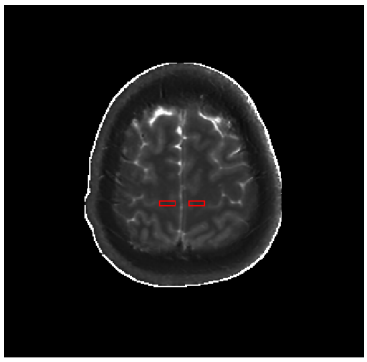


(a) T_1 (b) T_2 (c) PD

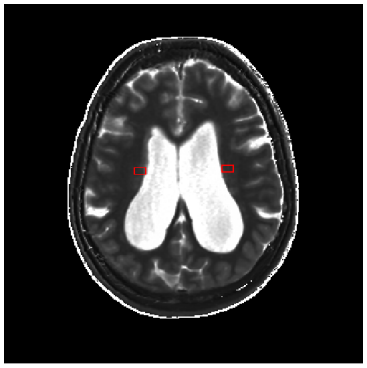
Figure 8: The three different types of maps obtained from a synthetic MRI scan are a) T1, b) T2 and c) PD.



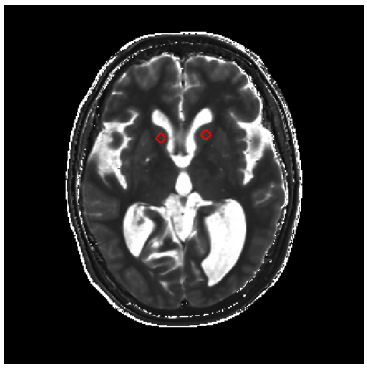
(a) ROI 1



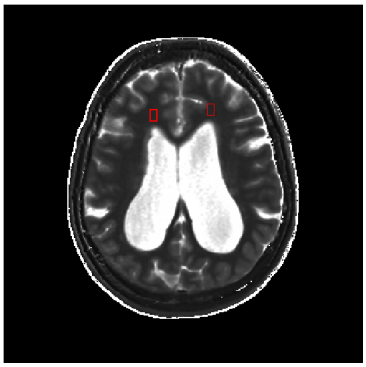
(b) ROI 2



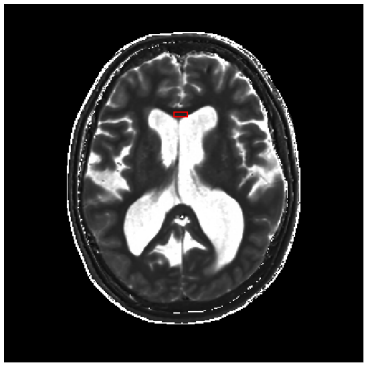
(c) ROI 3



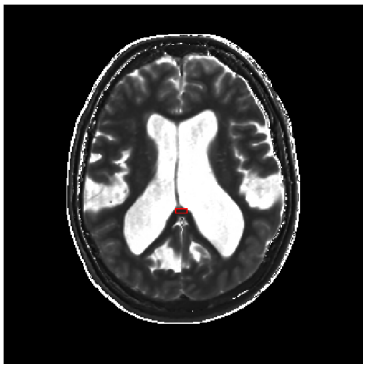
(d) ROI 6



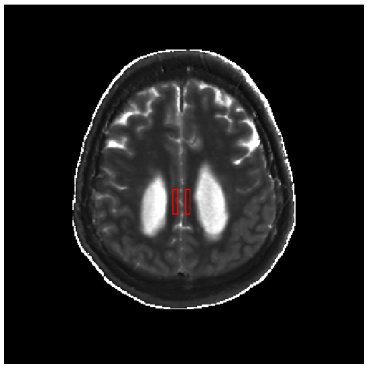
(e) ROI 7



(f) ROI 10

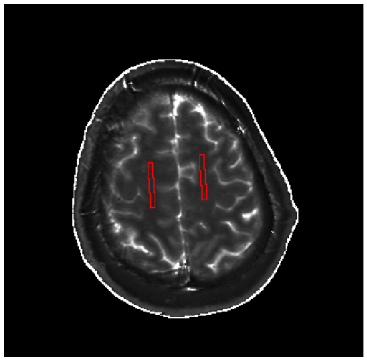


(g) ROI 11

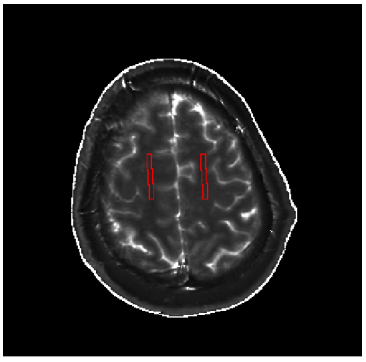


(h) ROI 12

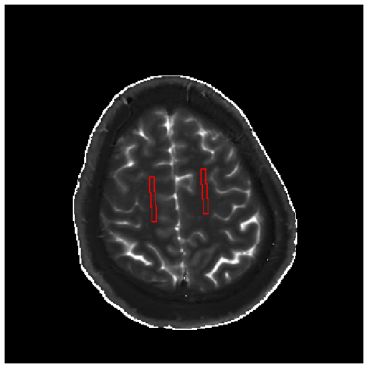
Figure 9: The ROIs as positioned by the program. Compare with Figure 1.



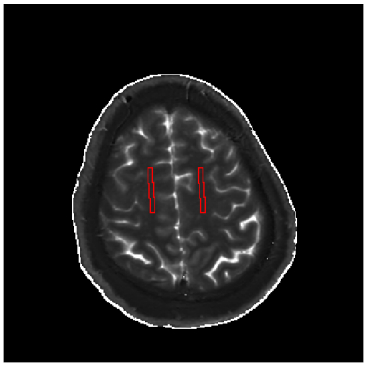
(a) Automatic, scan 1



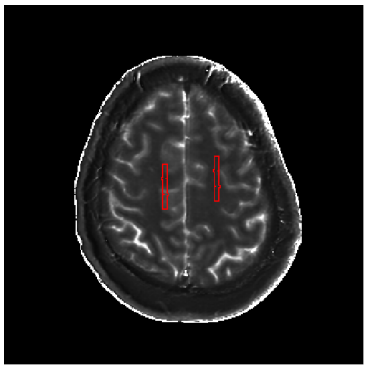
(b) Adjusted, scan 1



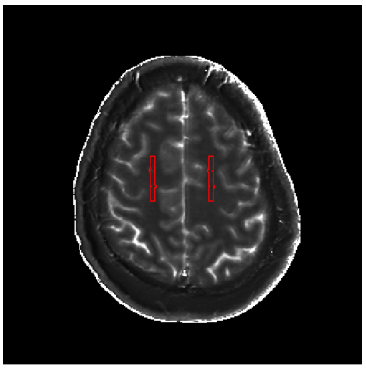
(c) Automatic, scan 2



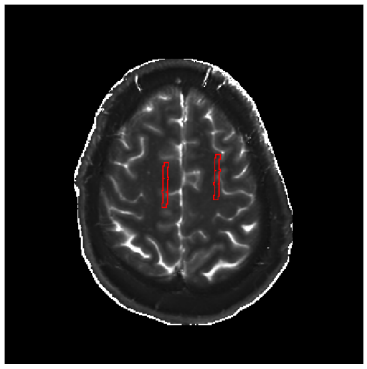
(d) Adjusted, scan 2



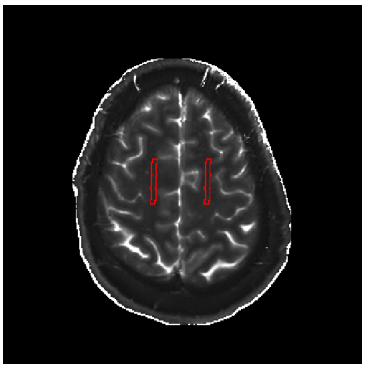
(e) Automatic, scan 3



(f) Adjusted, scan 3

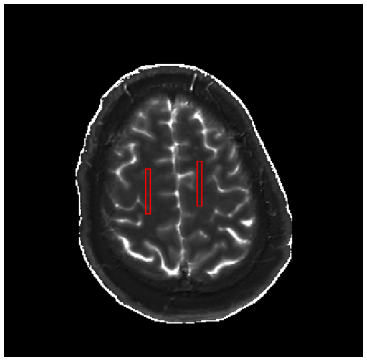


(g) Automatic, scan 4

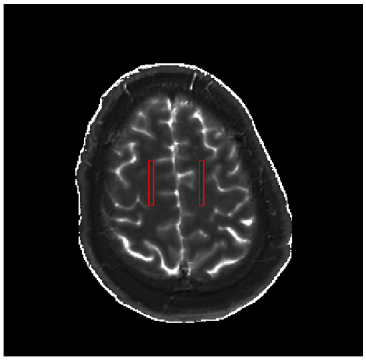


(h) Adjusted, scan 4

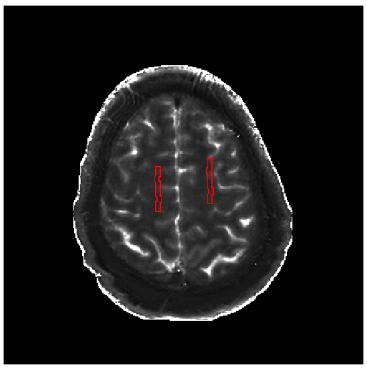
Figure 10: The left column shows ROI 1 as positioned by the program for scans 1-4 of one subject. In the images in the right column the ROIs have been moved manually to better fit the anatomy. See below for scan 5-7.



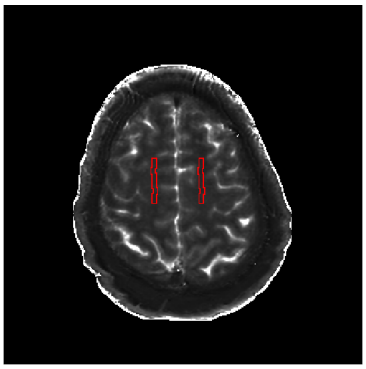
(a) Automatic, scan 5



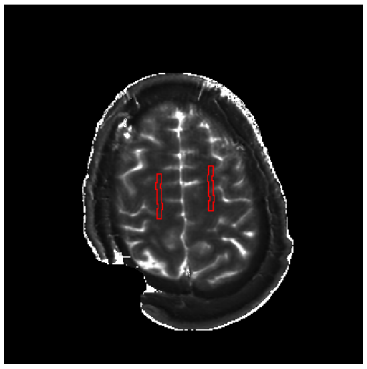
(b) Adjusted, scan 5



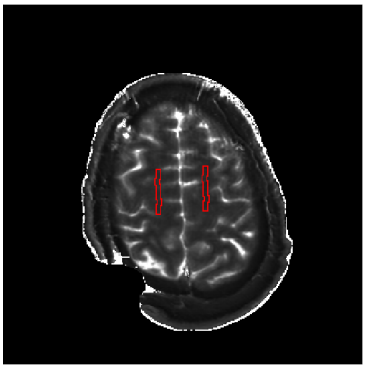
(c) Automatic, scan 6



(d) Adjusted, scan 6



(e) Automatic, scan 7



(f) Adjusted, scan 7

Figure 11: The left column shows ROI 1 as positioned by the program for scans 5-7 of one subject. In the images in the right column the ROIs have been moved manually to better fit the anatomy. See above for scan 1-4.

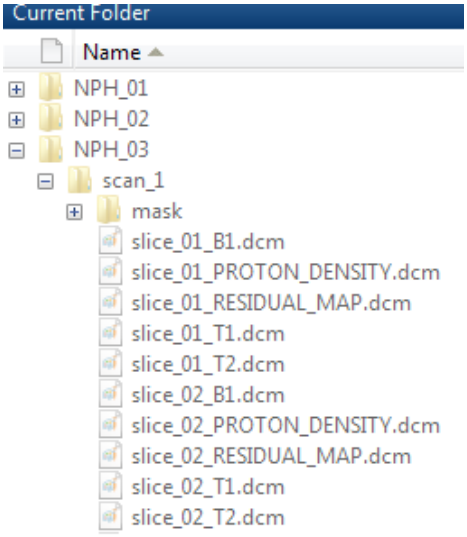


Figure 12: Part of the folder structure in the current Matlab folder which must be the same as where the program files are located. Folder and file names must be as shown here. For the control group, the folder name should be $NPHFK\_01$ and so on.

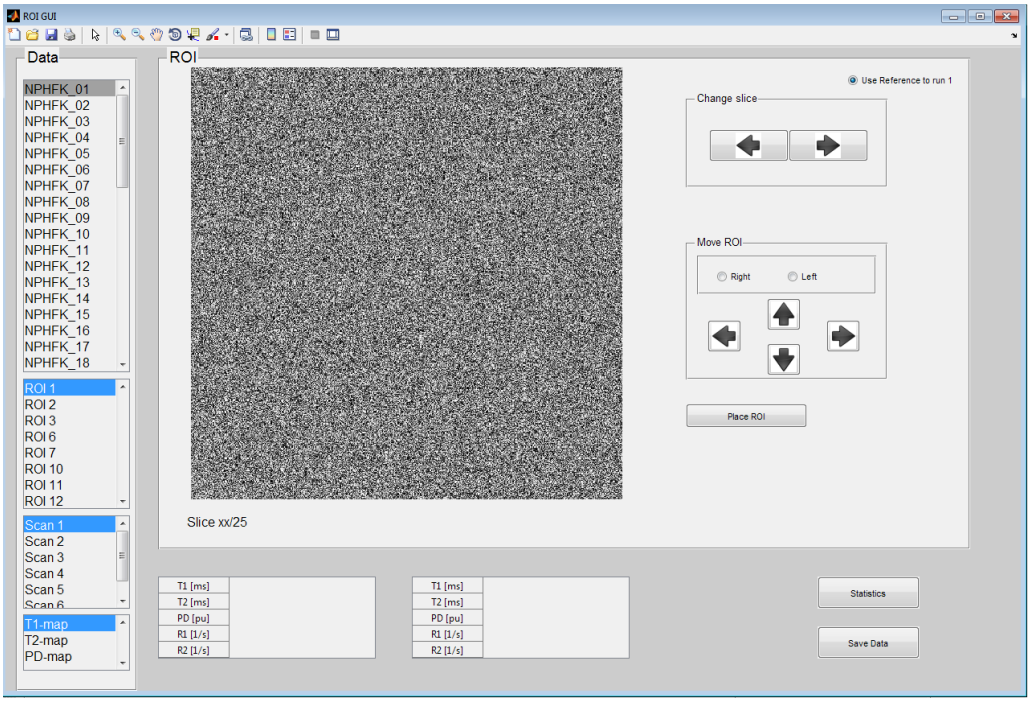
4.1 User's Guide

Here follows a short description of how to use the program with the help of the GUI.

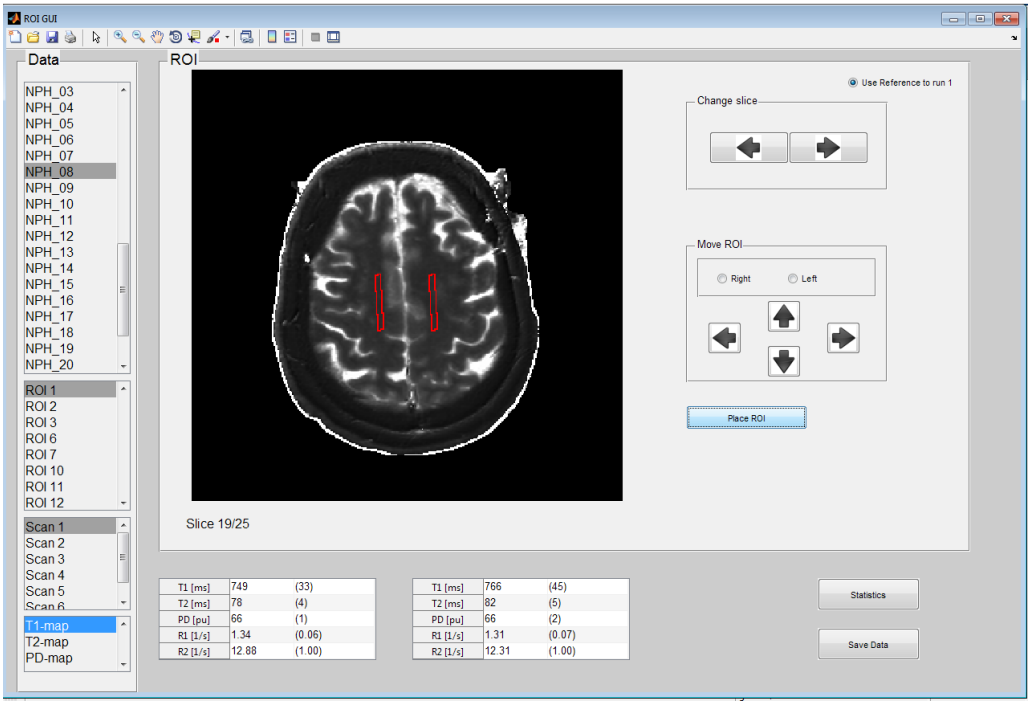
To set up the program, all data that are to be analysed need to be located in the same folder as the program files. This means that data from all patients with all scans have to be placed in the same folder as the Matlab files before the GUI can be started. Each scan needs Q-map data in DICOM format acquired from the program SyMRI along with the intracranial mask created by the same program. The mask images need their own folder within each "scan" folder, see Figure 12 for the structure and names of folders and files. It is important that the names are exactly corresponding to those in Figure 12 for the program to work.

To open the program, first open MATLAB and make sure that the current path is set to the correct folder and then simply type ROI_GUI into the command window. When the program starts it will open the GUI without any selected patient, ROI or scan, see Figure 13a. From here it is relatively easy to use the program, just choose the patient in the top left column, the ROI in the field below and the scan in the third field. Note that the ROIs in the consecutive scans (scan 2-7) are based on references to the current ROI in scan 1. Thus it is necessary to start out the procedure from the first scan. Once a ROI in scan 1 is positioned as desired, the processing order of scan 2 - 7 does not matter. It should be emphasized that a ROI must be placed in all scans before the next ROI can be handled (see Section 5.1.1 for a discussion about this). When the patient, ROI and scan has been selected and before the ROI has been placed, the user can change the slice to what seems to be the best one for the ROI selected. This is done with the arrows in the upper right corner. The program places the ROIs when the user press the button "Place

ROI". The ROIs (marked with red, see Figure 13b) are placed after predefined criteria and if the user is not satisfied with the position it is possible to move the ROIs with the four arrows in the button group "Move ROI". If the ROI in the left side of the brain is to be moved, the button marked left has to be pressed and the same goes for the ROI in the right side of the brain. (Since the patient is seen from below, the right side of the brain is to the left in the image and vice versa). It is also possible to switch between T_1, T_2 and PD maps when looking at the ROIs, this is done in the field in the bottom left corner. When the ROIs are placed it is possible to see mean T_1, T_2, PD, R_1, R_2 values in the two tables in the center of the GUI. The standard deviations are shown in parenthesis behind the mean values. When the button "Statistics" is pressed two new windows are opened containing all data acquired at the moment. From here it is easy to copy the data for further evaluation. Upon pressing the button "Save Data", all data including the ROIs in image format will be saved in the current MATLAB folder and remain even after the program has been closed. If the user desires to delete all current data and re-start from scratch the two MATLAB files named *subjectImages.mat* and *subjectStats.mat* containing the previous data have to be removed from the folder where the program is situated. To close the program simply press the red cross in the upper right corner and remember that all progress not saved will be lost.



(a)



(b)

Figure 13: a) The GUI upon starting b) The GUI with data selected and the ROIs positioned in the image. Two fields with mean values corresponding to the right and left ROIs are shown below the image.

5 Discussion

Overall, the algorithms perform well in placing the ROIs in the correct areas. In the majority of cases, minor adjustments had to be done with the arrow buttons to move the ROIs in the order of some pixels and in rare cases adjustments to the code were needed to account for individual problems in the images due to the anatomical structure. The data acquired can easily be copied to a spreadsheet program for further statistical analysis which makes it simple for the user to work with the data.

The program does not control the accuracy of the data thus leaving to the user to ensure that the measurements are reliable. Extra care need to be taken in scans with artifacts from the shunt implant, where the level of distortion in the images differs between patients and scans.

That the ROIs are placed in the *actual* regions of interest is naturally of great importance. It is essential to the evaluation of the measurements to know what tissue the ROI contains. The standard deviation can to some extent assist in verifying that the ROI contains *homogeneous* tissue, but it is in the current version of the program left to the user to know the properties of the tissue of interest. As a help to distinguish between the tissues, all three of the T_1, T_2 and PD maps can be consulted. Two further means of evaluation of tissue type have been identified but are not implemented in the current version of the program. The first possibility is to calculate the *weighted* T_1, T_2 and PD images, which the user might be more experienced with since these are the images obtained in traditional MRI. Secondly, using a plot of the inverted relaxation times, R_1 and R_2, for each pixel in the ROI an integrated evaluation can be obtained. For a further description of these methods, see Section 5.1.2 below.

In ROI 3 and ROI 7 a gradient is measured in the normal direction of the boundary of the ventricles. One can argue about the shape and extension of these ROIs and the impact it has on the resulting data. By comparing Figure 1c and Figure 9c one can see that the shape of the ROIs have been changed and become shorter in y-direction and longer in x-direction. This change was made to minimize the influence of the convex shape of the ventricles on the gradient and the opposite change has been made to ROI 7. A possible different approach is to measure the gradient from the ventricle all the way to the skull and give the user the possibility to choose a relative length of the gradient and thus adjust for size differences between brains. Measuring the gradient as a mean of evaluating the iNPH state is a somewhat new approach and although the method in our program can be optimized it could still give information of the effectiveness of such an approach.

In the case of iNPH, a co-registration procedure of the brains can give unreliable results because of the abnormal anatomy. The co-registration involves translation and rotation of the object and because of the discrete pixel values an interpolation has to be made to create the new image. By adjusting the ROIs instead of the brains an attempt to avoid these sources of error has been made. The script uses the positions of the eyes to find out how the brains are located in the image and then compares these values to know how to adjust the ROIs between scans. Since this procedure determines the positions and shapes

of the ROIs, it is crucial that this the code is precise. Some improvements can be done to the script to increase its reliability and a more in depth discussion about this can be found in 5.1.1. Since the current algorithm is not fail safe, the user has an option to change the positions of the ROI after the program has positioned them. Although this increases the potential error of the "human factor" on the results, it is considered necessary in the current version. The real size of a pixel is about 1 mm and the slice thickness is 5 mm, so the effort to measure in the same place in all scans to get values that are comparable not only comes down to the accurate positioning of the ROIs in the image, but also to assuring that the slices are placed in the same z-positions at all times. This is done by ascertaining that the scanning sequence starts at the exact same place in all scans. If the slices are shifted in the z-direction, a source of error is introduced to the measurements. The scanning data available in this study has the defect of not being translation free and therefor this uncertainty must be taken into account when evaluating the numerical results.

5.1 Improvements

There are improvements that can be made to the program to make it more reliable and to work faster. There are also several modifications that can be made to give the user a greater freedom and adjust the program for more diverse studies. Below, some aspects are presented to attain these goals. The first subsection contains improvements that can be done with already existing algorithms, while the second subsection presents additional functions that can be added to the program for further use.

5.1.1 Improvements to the algorithms

Some of the algorithms can be improved for stability and/or speed, two of which are the functions *eyeFind* and *ventricleSegmentation*. The eyes of the patients are found in *eyeFind* and are used to measure the translation and the angle of rotation between scans. The present algorithm finds circles, that are supposed to be eyes, with the help of the Hough transformation and can sometimes find shapes that resembles eyes but that are not. When this happens it results in erroneous values of the translation and the angle and the ROIs get an incorrect shape when placed in the image. Since the risk for this error was known, it has been possible to create a temporary and individual solution for the cases where this problem did arise. An integrated and permanent solution to this problem should be found if the program is to be used any further. One way to do this, is to set additional conditions on the circles found with the Hough transformation, such as the distance between them, their relative sizes et cetera.

The function *ventricleSegmentation* is used for almost all ROIs and is therefore an important part of the program. This function performs well on images of patients with iNPH, but can encounter problems when patients with normal sized ventricles are examined. This is because the ventricles are too small in some of these cases resulting in that the conditions set in the algorithms are not met. At the moment, the slice on which this function operates is hard-

coded. It would be desirable to set conditions and let the program find the most suitable slice for the segmentation, for example the slice with the largest ventricles. This problem is possible to work around in the current version by manually changing the slice number in the code when the problem arises.

For the current version of the program all subject data needs to be arranged as described in Section 4.1 and the total number of subjects, ROIs and scans needs to be fixed. A more flexible implementation would be desirable where, for instance, the user is prompted to select subject data upon starting the program and to choose a folder and file name for saving the data.

The tables created when the "Statistics" button is pressed shows all subjects with the value zero at all positions where no data has been set. This makes the table somewhat difficult to overview and it would be desirable that only the subjects that have any data where shown. This improvement goes hand in hand with the possibility to add and remove patients during the run of the program.

If the program is to be used for other studies it is necessary that it can handle images of different sizes and scale the ROIs after the size of the image.

5.1.2 Adding functions

Neurologists might be used to work with weighted images instead of the maps presented in the current version of the program. A future improvement would be to give the user the option to see these weighted images as well. No additional data is needed for this, but one can implement an algorithm that uses the T_1, T_2 and PD-maps to create the synthetically weighted images. The equation for the intensity, I, of each pixel in the synthetic image is

$$I = PDe^{-\frac{TE}{T2}}\left(1 - e^{-\frac{TR}{T1}}\right)$$

where T_1, T_2 and PD are the pixel values values of the respective charts. For a T_1 weighted image TE = 10 and TR = 500 while for a T_2 weighted image TE = 100 and TR = 4500.

To further help to distinguish between tissues, a plot with R_1 on one axis and R_2 on the other axis could be created. By plotting each pixel value contained in the ROI and comparing with the values of white and gray tissue one would get a fairly good approximation of what kind of tissue is contained in the ROI.

For extended use of the program it would be beneficial to let the user draw her own ROIs that can be translated and rotated by the same algorithms as the predefined ROIs are. One could also implement a function that lets the user rotate the ROIs manually if the algorithm for rotating the ROIs fails.

6 Conclusions

The program works well for the purpose it is designed for. It manages to reduce the task of manually placing over 2600 ROIs in the images and extract the desired data, which can be exported to an external software for analyzation. The ROIs are fairly accurately positioned by the program, though one would likely want to adjust most of the ROIs by some pixels before extracting the data. The data is presented in a table and is easy to survey.

Several modifications can still be made to the program. The most important of these would be to improve the reliability in the key tasks, i.e to find the correct positions of the ROIs as well as assuring the coherence between different scans. For this purpose, the functions *eyeFind* and *ventricleSegmentation* should be the focus. If the precision is increased in these two algorithms and the type of tissue is taken into account, the human factor can be minimized, a goal that is not entirely met in the current version of the program.

The outcome of this project, both when it comes to the accuracy in the ROI placement and the data presentation, has been approved by an experienced neuroradiologist. The performance of the program is therefore considered to be good.

7 Bibliography

[1] Kumar A, Welti D, and Ernst R R. Nmr fourier zeugmatography. *Journal of Magnetic Resonance*, 18(1), April 1975.

[2] McRobbie D.W., Moore E. A., Graves M.J., and Prince M.R. *MRI From Picture to Proton*. Cambridge University Press, Cambridge, 2nd edition, 2007.

[3] Bloch F. Nuclear induction. *Physical Review*, 70(7 and 8), October 1946.

[4] Lilley J. *Nuclear Physics*. John Wiley and Sons Ltd, Chichester, 2001.

[5] Sakurai J and Napolitano J. *Modern Quantum Mechanics*. Addison-Wesley, 2 edition, 2011.

[6] Serra J. *Image Analysis and Mathematical Morphology*. Academic Press, 1 edition, 1982.

[7] Virhammar J. *Idiopathic Normal Pressure Hydrocephalus. Cerebrospinal Fluid Tap Test and Magnetic Resonance Imaging as Preoperative Prognostic Investigations*. PhD thesis, Uppsala University, 2014.

[8] Allisey-Roberts P and Williams J. *Farr's Physics for Medical Imaging*. Saunders, 2 edition, 2007.

[9] Hofmann P. *Solid State Physics*. Wiley-VCH, 1 edition, 2012.

[10] Gonzalez R.C and Woods R.E. *Digital Image Processing*. Prentice Hall, 3 edition, 2008.