

Charles University in Prague
Faculty of Mathematics and Physics

DIPLOMA THESIS

Jiří Čížek

Tomographic Data Processing
in Nuclear Medicine

Branch School: Computer Science - Computer Graphics

Supervisor: RNDr. Josef Pelikán
Department of Software and Computer Science Education
Faculty of Mathematics and Physics
Charles University

Consultant: Dr.Ing. Daniel Janeba
Department of Nuclear Medicine - PET centre
Na Homolce Hospital

Acknowledgements

I want to thank to RNDr. Josef Pelikán for leading this thesis. I am grateful to Dr.Ing. Daniel Janeba for lots of suggestions and comments.

My great thanks belong to my parents for their support and for making it all possible.

Special thanks go to Petra Kudějová for her love, support and many advices that helped improve this text - shortly for being my best friend.

I do declare that I have elaborated this diploma thesis on my own and that the references include all the sources of information I have exploited. I agree with lending of this diploma thesis.

Prague
April 18, 2001

Jiří Čížek

Contents

1	Introduction	6
1.1	Aim of this Study	7
2	Data Visualization	9
2.1	Principle of PET and SPECT	9
2.2	Structure of the Data	10
2.2.1	Histogram	12
2.3	Visualization of Slices	13
2.4	Color Mapping	15
2.5	Thresholding	16
2.6	Zooming of Slices	18
2.6.1	Speed-up for Transverse Slices	20
3	Data Registration	22
3.1	Data Parameters Adjustment	24
3.2	Geometric Transformation of Data	25
3.3	Similarity Measures	27
3.3.1	Count Difference	29
3.3.2	Squared Difference	30
3.3.3	Shape Difference	30
3.3.4	Mutual Information	31
3.4	Data Registration Methods	32
3.4.1	Powell's Direction Set Method	34

<i>CONTENTS</i>	4
3.4.2 Downhill Simplex Method	36
3.4.3 Conjugate Gradient Method	38
3.5 Validation of Results	39
3.5.1 Contour Matching	40
3.5.2 Chess Image Matching	41
3.5.3 Subtraction of Images	41
3.5.4 Overlay of Images	43
3.5.5 Distance Measuring	43
3.6 Evaluation of Methods	44
3.6.1 Identical Studies	45
3.6.2 Different Studies	47
3.6.3 Summary	48
4 Templates	49
4.1 Normalization of Studies	50
4.1.1 Thresholds	51
4.1.2 Total	51
4.1.3 Maximum	51
4.1.4 Quantil	51
4.2 Region of Interest	53
4.3 Template	53
5 Implementation	55
5.1 Data Structures, Techniques	55
6 User Manual	58
6.1 Program Brien	58
6.1.1 Program Installation	58
6.1.2 Main Application Window	58
6.1.3 Menu	62
6.1.4 Register Window	63
6.1.5 Options	66

<i>CONTENTS</i>	5
6.2 Program Tepito	67
6.2.1 Program Installation	67
6.2.2 Main Application Window	67
6.2.3 Normalization	69
6.2.4 ROI	70
6.2.5 Histogram Window	71
6.2.6 Statistics	72
6.2.7 Template	73
7 Conclusions	75

Chapter 1

Introduction

Medical imaging is a fast developing area of the current clinical diagnostics. The imaging modalities employed include mainly computed tomography (CT) and magnetic resonance imaging (MRI), depicting primarily morphology, and positron emission tomography (PET) together with single photon emission computed tomography (SPECT) that depict primarily functional information of tissues (metabolism, perfusion, etc.).

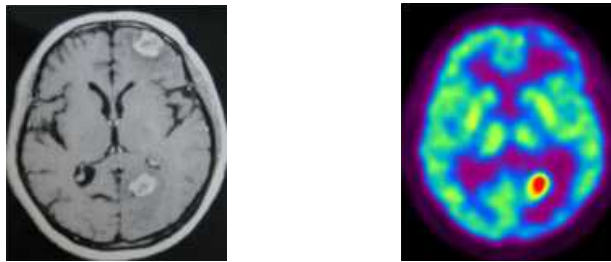


Figure 1.1: MRI, PET

Figure 1.1 shows two images taken from the same patient using MRI (left) and PET (right). The MRI image shows two suspicious lesions, while the PET image provides information about FDG consumption. By combining information from both images it is possible to identify one lesion as a tumour and the other as a radionecrotic tissue. One can combine information provided by more investigations of a patient to compare either different modalities or studies of the same modalities taken under different conditions

(before/after treatment, in stress/rest, ictal/interictal, etc.).

A proper *image registration* of useful data obtained from the separate images is therefore desired. The *registration* consists in bringing the involved images into spatial alignment. Registration of functional images following from nuclear medicine investigations (PET, SPECT) is the main goal of this thesis.

1.1 Aim of this Study

The aim of this study is to suggest and implement various methods for 3D medical data registration and to propose a convenient way for their visualization. The examined methods and elaborated procedures have been implemented in two programs that can be used for diagnostic of PET/SPECT brain studies.

This work was elaborated in cooperation with the PET centre of the Hospital Na Homolce in Prague where the source PET/SPECT data originate from. The PET centre is equipped with one PET and one SPECT scanner from Siemens company. Several high-performance graphical workstations with Solaris or SunOS operating systems from Sun Microsystems are integral components of these scanners. They are equipped with a professional software that is used for manipulation and diagnostic of output data from the scanners.

The main task of this thesis is to create a diagnostic software for common PC computers with the Microsoft Windows operating system. In contrast to the graphical workstations, these computers are much easily available. The resulting software can be used for presentation purposes and for medical research. It is not intended for treatment purposes.

The basic inspiration came from the professional diagnostic software. However, these programs have several unsatisfactory features and lack some useful functions. The documentation for the software contains very poor information about used registration and visualization methods. Nearly all methods that the resulting software uses had to be found in literature or are proposed in this work.

Chapter 2 concerns with medical data visualization. First, the general principle of PET and SPECT is introduced. Then the structure of input data is described, possible visualization approaches are summarized and a technique for data visualization is presented.

Chapter 3 begins with description of data transformations. The main part of this chapter deals with data registration, which includes finding suitable numerical methods for data alignment in conjunction with convenient measures of similarity and proposal for several methods for checking a quality of registration. The chapter ends with comparison of implemented numerical methods based on practical experiments.

In medical diagnostic, it is often necessary to determine how much a certain brain differs from a standard one. Chapter 4 deals with different approaches to creation of such a standard study.

The acquired knowledge and elaborated procedures have been implemented in two programs. Chapter 5 describes implementation details of the resulting programs not covered in the previous chapters. Chapter 6 contains a detailed user manual.

Chapter 2

Data Visualization

This chapter first introduces the general principle of PET and SPECT. The main part deals with methods of data visualization.

2.1 Principle of PET and SPECT

In clinical applications, a very small amount of radionuclide-labelled compound (called radiopharmaceutical or radiotracer) is introduced into a patient usually by intravenous injection. The radionuclide decays in patient's body and some products of the decay are detected by a scanner which allows to measure the concentration of tracer in tissue. The information about the distribution of tracer concentration in tissue allows to judge of a function or metabolism of examined organ.

In *Positron emission tomography* (PET) radiopharmaceuticals are labelled with radionuclides that emit a positron which, after travelling a short distance (3-5 mm), encounters an electron from the surrounding environment. The two particles interact and annihilate, each other resulting in the emission in opposite directions of two gamma rays of 511 keV each. The three-dimensional image acquisition is based on the external detection in coincidence of the emitted rays. Using an appropriate radiotracer one can gain specific information about tissue. For example, glucose labelled with ^{18}F (a tracer called fluoro-deoxy-glucose FDG) accumulates in tumours which

means that a high concentration of FDG may indicate a presence of a tumour.

In *Single photon emission computed tomography* (SPECT) radiopharmaceuticals are labelled with radionuclides that emit γ -ray photons (mostly ^{99m}Tc , ^{201}Tl , ^{67}Ga and ^{111}In). A scintillation camera system is able to detect emitted photons and as a result provides a two-dimensional projection image of the three-dimensional radionuclide distribution within the patient's body. By taking several two-dimensional images from different views the system is able to reconstruct a three-dimensional estimate of radionuclide distribution.

In both methods, acquired data are reconstructed to a three-dimensional matrix of intensities. This data is then saved to a file in some medical file format. The programs resulting from this thesis can read files in the *Interfile* format. It is a common file format for medical data exchange to which data in different formats can be easily converted. The specification of the Interfile format can be found in [19]. For more information about PET and SPECT see for example [4].

2.2 Structure of the Data

The input data consist of a set of square-shaped slices that together create a three-dimensional matrix of intensities (see Figure 2.1). This matrix of intensities will further be denoted as *matrix*, *volume*, *data*, *study* or *image*. (The word *image* is very ambiguous in medical imaging. It is usually used for the 3D volume of data as well as for a single 2D slice.) Intensities are usually represented by two-byte integers. A single spatial matrix element will be further denoted by a common term *voxel*. Typical dimensions of the matrix are 128x128x47 or 256x256x64 voxels. For more illustrative explanation the matrix will always be considered aligned to the three orthogonal axis of an orthogonal coordinate system as shown in Figure 2.1.

There are several parameters that describe the matrix and that allow to compute its real (i.e. metrical) dimensions. They will be used densely in several following sections and thus it is necessary to explain them briefly:

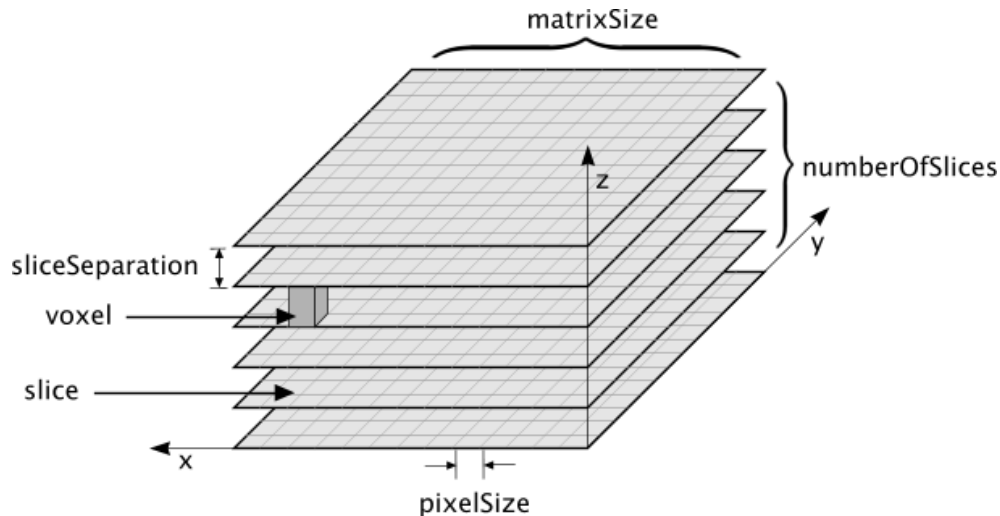


Figure 2.1: Input Tomographic Data

- **matrixSize** Denotes the dimension of a squared slice (number of elements in the direction of x and y axis). Every slice contains $\text{matrixSize} \cdot \text{matrixSize}$ elements.
- **numberOfSlices** Number of slices in the data.
- **sliceSeparation** Denotes the distance between two neighbouring slices in multiples of slice elements. For example, **sliceSeparation** equal to 1.5 means that the distance between two slices is 1.5-times the size of a slice element.
- **pixelSize** Dimension of a slice element in millimetres. It is the same for both x and y dimensions. Using **sliceSeparation** we can compute the distance between two neighbouring slices in millimetres as $\text{sliceSeparation} \cdot \text{pixelSize}$.

If the **sliceSeparation** is different from 1 (the most usual case) then the metrical size of a voxel in the xy -plane is different from the size of that voxel in the z direction. This involves using two distinct coordinate systems for determining the position of a point in the volume.

One coordinate system determines the position of a voxel by the order of that voxel in the x , y and z direction. Characters i , j , k will usually be

used for this coordinate system. The voxel $[i, j, k]$ is the i th voxel in the x direction, j th in the y direction and it lies in the k th slice. The whole volume of the spatial voxel is considered to be a single point. The components i, j, k are integers and the three-dimensional space is discrete. Further in the text it will be sometimes necessary to consider this space continuous. In such a case, the point $[i, j, k]$ denotes the centre of the corresponding voxel. For example, the point $[i + 0.5, j, k]$ lies on the border between voxels $[i, j, k]$ and $[i + 1, j, k]$. This coordinate system will be called *voxel coordinate system* throughout this thesis.

The voxel coordinate system has a property that makes it unsuitable in some cases. It does not keep proportions of the real volume if the `sliceSeparation` is different from 1. Where it is necessary to consider the real study proportions, the *metrical coordinate system* will be used. It expresses the position of a point in the three-dimensional space in millimetres. Points in this coordinate system will be usually written using characters x, y, z . An arbitrary point $[i, j, k]$ from the voxel coordinate system is expressed in the metrical system by the following equation:

$$[x, y, z] = \text{pixelSize} \cdot [i, j, k \cdot \text{sliceSeparation}]$$

In further text $M(i, j, k)$ denotes an intensity of the voxel $[i, j, k]$ of the matrix M . If any of the components i, j, k is not integral, $M(i, j, k)$ is the value interpolated from the neighbouring integral points using an interpolation technique (Section 2.6 tells details). Maximum intensity value in the study will be marked *max*.

2.2.1 Histogram

Histogram is a valuable tool used to view the intensity profile of data. It provides information about the contrast and overall intensity distribution of the data. It is obtained by dividing the interval between the minimum and maximum intensity value in the data into equally spaced bins. Each voxel is assigned to the bin that surrounds its intensity. An example of a typical SPECT study histogram, using 128 bins, is displayed in Figure 2.2 together with the corresponding image. The figure shows that the intensity values near zero totally predominate. Note that the height of the presented

histogram corresponds to the height of the third histogram bin. The first and the second bin have been cut off. Without this adjustment the only visible bar would be the first bar and the other bars would be suppressed.

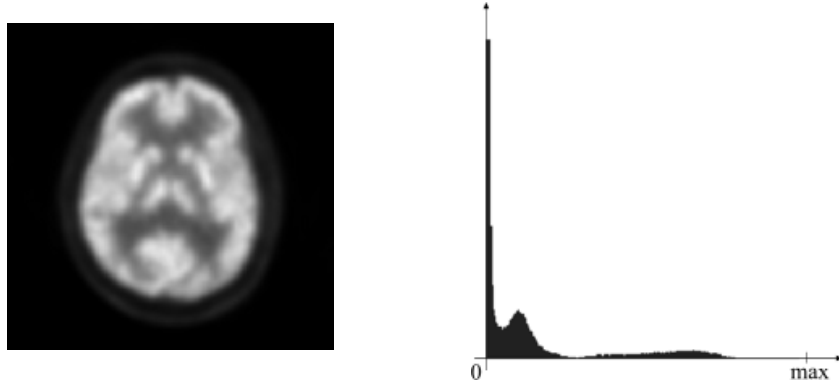


Figure 2.2: Typical SPECT study and its histogram

The high peak consisting of the first few bars and the second, lower peak, originate mainly from voxels located in the image background outside the perimeter of the object and also from voxels located at positions of material inside the object that is invisible for SPECT. The small bars around the half-maximum intensity value originate from voxels of tissues inside the object that are visible for the used modality. Only a small portion of the whole study contains useful data.

A *normalized histogram* is obtained from the image histogram by division of each bar by the number of voxels in the image. It can be considered as an estimate of the probability distribution of intensity in the volume. The normalized histogram of the volume M will be denoted as H_M and its values as $H_M(i)$, $0 < i < n$, where n is the number of bins of the histogram. Thus $\sum_i H_M(i) = 1$. The probability distribution of intensity of the volume M will be denoted as $P(M)$.

2.3 Visualization of Slices

The data can be displayed on a screen either as a set of 2D slices or as a rendered object (volume, surface rendering or maximal intensity projection).

For the purpose of study registration the most suitable way is to display the data as a set of 2D slices. Consequently, a slice will be often called *image* and elements of the slice will be denoted as *pixels*.

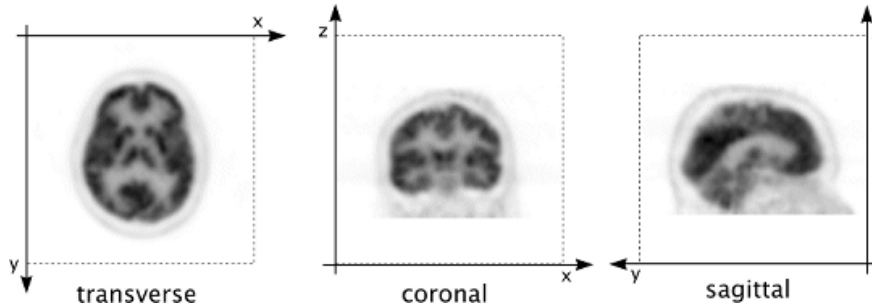


Figure 2.3: Transverse, Coronal and Sagittal orientation

Three main orthogonal slice orientations of a human body are called *transverse*, *coronal* and *sagittal* (see Figure 2.3). (*Note:* Meaning of the word *slice* is a little bit different here than in the description of the data matrix. It denotes a set of voxels lying on any plane intersecting the data matrix.) The data in the matrix are always oriented so that the xy plane corresponds with the transverse slices. By convention, the direction of the x axis corresponds to the direction from right ear towards left ear of patient's head. For the y axis it is the direction from the front towards the back part of head and the z axis goes in the direction from bottom to top. Thus a transverse slice is completely determined by its position on the z axis, a coronal slice by position on the y axis and a sagittal slice by position on the x axis.

The process of bringing a single slice onto a screen can be divided into three steps:

1. Intensities of slice elements are modified according to selected thresholds.
2. The slice is zoomed.
3. Colors are mapped to intensities according to a color table.

In practice all these steps are merged into one. Thresholding is covered in Section 2.5. Description of color mapping is the objective of the following Section 2.4 and zooming of slices is described in Section 2.6.

2.4 Color Mapping

The input matrix is a three-dimensional array of intensities (in the range of $\langle 0, max \rangle$). To display a part of the matrix (a single slice) on a screen it is necessary to assign the intensities to the colors of the output monitor. Monitors display colors using the RGB color model [22]. It means that all displayed colors are composed of three basic colors - red, green and blue. Nearly all commonly used monitors are already able to display colors using one byte for every basic color. Every color is then represented by three bytes and there are 256^3 (more than 16 millions) possible colors. This representation is commonly called *true color*. A portion of a basic color present in an arbitrary color can be expressed by an integer number in the range $\langle 0, 255 \rangle$.

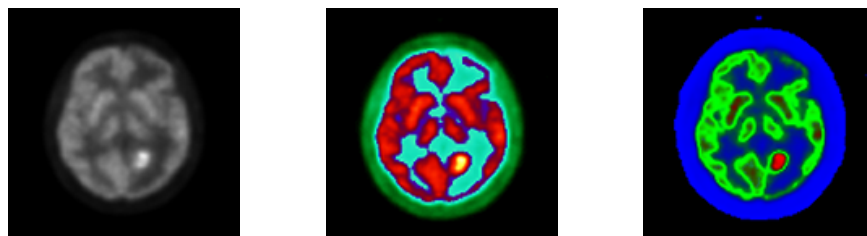


Figure 2.4: A single study displayed using different color tables

A directive that maps intensities to colors is implemented using a *color table* of size 256 so that every intensity value is assigned one of the 256 colors in the color table. The color table can contain any 256 colors of the 16 mil. possible colors in true color representation. Since the intensities are in the range $\langle 0, max \rangle$, an intensity is first linearly transformed into the range $\langle 0, 255 \rangle$ and then it is assigned a color according to the color table.

Using of 256 colors is a common practice in medical imaging. This amount is completely sufficient for displaying the tomographic data. One of the most used tables is the gray-scale palette. To get a gray color, all three components (red, green, blue) must have the same value. Since true color uses 8 bits for one color component, there can be only 256 grades of the gray color. There is no exact description how these color tables should look like. In medical imaging, they are created empirically due to specific needs. Well-designed color tables can provide very interesting effects. Different color tables may be

used for highlighting of various parts of a tomographic image. An example can be seen in Figure 2.4 on a PET study taken from patient with epilepsy.

2.5 Thresholding

Thresholding in computer graphics usually means a procedure whose object is to divide an image into two segments. The criterion for assigning a voxel to either segment is whether its value is lower or higher than a prescribed threshold value. A binary image is obtained by marking pixels with values less than the threshold with zeros and the remaining pixels with ones.

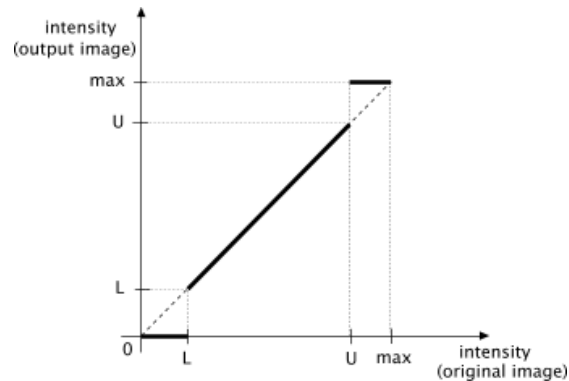


Figure 2.5: Thresholding

In nuclear medicine imaging it is usual to use the term *thresholding* for a little bit different operation as well. Given two threshold values U , L - an *upper* and a *lower* threshold - the operation assigns the zero value to the voxels with intensity lower than the lower threshold, assigns the maximum value to the voxels with intensity higher than the upper threshold and *does not alter* the voxels with intensities between the thresholds. This operation (belonging to the *look-up table operations* [22]) is illustrated graphically in Figure 2.5. The figure shows which intensity values of the original image (horizontal axis) are mapped onto which intensities in the output image (vertical axis). Throughout this thesis, the word *thresholding* always means the just described operation.

Thresholding is an important operation that allows to display only a certain

part of the study containing voxels with intensities in a selected range. For example, a tumour appears at a SPECT brain image as an area of voxels with higher intensities. Using appropriate thresholds it is possible to cut off the surrounding area and thus highlight the tumour position. The Figure 2.6 presents such case. In the right image the lower threshold is set to 64% of the maximum value in the study.

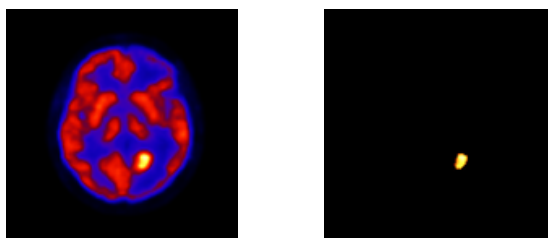


Figure 2.6: Extraction of tumour

After the process of tomographic data acquisition it can occasionally happen that the data matrix contains several voxels with extremely high intensities in comparison to the rest of the matrix. The acquisition process failed in some way in these voxels and they do not contain valid data. It makes problems with displaying of slices - the output images are dark. They could also influence the registration process. To eliminate their influence another threshold G was suggested and implemented. It is an user-supplied intensity from the range $< 0, max >$. Setting the threshold G means that the study is searched for the intensity value I_m that is the maximum from all values lower than or equal to the value G :

$$I_m = \max_{i,j,k} \{M(i, j, k); M(i, j, k) \leq G\}$$

The value I_m is the new maximum and all voxels with higher intensities will be considered to have intensity I_m from now on. The thresholds L and U are applied according to maximal intensity value determined by the threshold G .

2.6 Zooming of Slices

As stated before, a common input data matrix has dimensions in order of hundreds. An observer would not see much on such a small image, so it is necessary to enable some sort of zooming.

Enlarging an image causes that values of many pixels in the enlarged image are missing and they must be estimated using the values of the known neighbouring pixels - the image must be *resampled*. There are several well-known methods that deal with image resampling [22], [5]. The general principle of all methods is to find an approximate continuous image function corresponding to the discrete image, to transform it (i.e. to stretch it when zooming in) and then to resample the transformed image. In practice this means to pass through pixels of the output (zoomed) image and for each pixel to search its new value by interpolating in the source image (i.e. in the matrix of intensities). This technique is commonly called *backward mapping*.

The methods differ mainly in quality of continuous function reconstruction. Obviously, more accurate reconstruction needs more time to compute, so choosing an appropriate method is a compromise between quality and speed.

The most common methods are *nearest neighbour interpolation*, *bilinear interpolation* and *bicubic interpolation* [22].

Nearest neighbour interpolation is probably the simplest method for continuous image approximation. This method computes value of a missing pixel so that it simply copies the value from the nearest neighbour.

Bilinear interpolation method calculates the intensity of a point by calculating the weighted average of the four closest pixels (a 2x2 array) based on distance. The method consists of three successive one-dimensional linear interpolations. The value $f(x, y_0)$ of a point $[x, y_0]$ lying between the two closest points $[x_0, y_0]$ and $[x_1, y_0]$ is given by equation

$$f(x, y_0) = f(x_0, y_0) + \frac{x - x_0}{x_1 - x_0} (f(x_1, y_0) - f(x_0, y_0))$$

Analogously the value $f(x, y_1)$ is computed from the values $f(x_0, y_1)$, $f(x_1, y_1)$.

Finally, the value $f(x, y)$ is interpolated from the values $f(x, y_0)$ and $f(x, y_1)$

$$f(x, y) = f(x, y_0) + \frac{y - y_0}{y_1 - y_0} (f(x, y_1) - f(x, y_0))$$

More accurate methods use higher-order polynomials, goniometric or exponential functions and other functions that are computationally much more expensive than the nearest neighbour or the bilinear interpolation.

Bicubic interpolation is one of the most common nonlinear methods. It computes the value in a point as a weighted average of the nearest sixteen pixels (a 4x4 array).

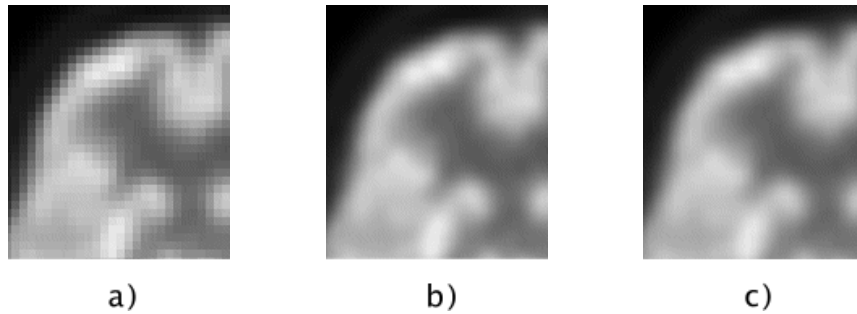


Figure 2.7: a) Nearest neighbour interpolation, b) Bilinear interpolation, c) Bicubic interpolation

Figure 2.7 shows a typical SPECT image 5x zoomed and interpolated using nearest neighbour, bilinear and bicubic interpolation. Nearest neighbour interpolation is simple and very fast and does not alter the original data. The disadvantage is that it highlights edges and sharp changes of intensity (stair-step effect) and causes the image to appear 'blocky' (Fig. 2.7a). This 'blockyness' can be completely removed by using bilinear interpolation that is still quite fast and simple. In general this method smoothes edges in image. Fortunately this is not a problem in our case since typical SPECT or PET images contain no edges. Thus this method seems to be more appropriate than the simple nearest neighbour. Bicubic interpolation is considerably more computationally intensive but Figures 2.7 b), c) show that there is nearly no difference between an image zoomed using bilinear and bicubic interpolation. For reasonable extent of zooming the bilinear interpolation seems to be the best choice and it has been chosen as the most appropriate compromise between quality and computational speed.

2.6.1 Speed-up for Transverse Slices

Transverse slices of typical PET/SPECT tomographic data have dimensions 128x128. Thus, they require approximately three times as much computational time to be resampled and displayed than the coronal or sagittal slices. Fortunately, transverse slices has a certain property that allowed me to modify slightly the common bilinear interpolation algorithm. With this modification the interpolation of transverse slices is approximately three or four times faster. This technique uses the *forward mapping* of pixels.

Transverse slices retain their proportions in both x and y axis when taken from the original matrix onto the screen. This means that if a slice is zoomed four times, then on the resulting image every fourth pixel will be the one from the original matrix. This is not generally true for the coronal and sagittal slices since the size of a voxel in the z direction usually differs from its size in the x and y direction.

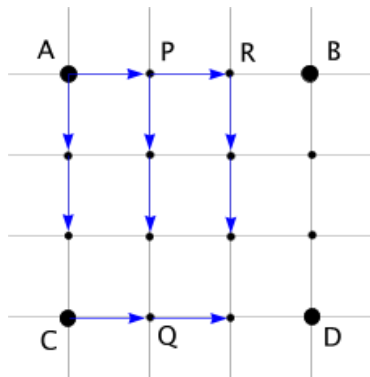


Figure 2.8: Forward mapping in a transverse slice

In Figure 2.8 there is a part of a three times zoomed transverse slice image. Using the notation established in Section 2.1, the transverse slice consists of voxels $M(i, j, z)$, $0 \leq i, j < matrixSize$, where z is the slice number. For simplicity, the parameter z will be omitted and *pixels* of the slice will be denoted as $M(i, j)$.

The four points $A = [i, j]$, $B = [i + 1, j]$, $C = [i, j + 1]$, $D = [i + 1, j + 1]$ are the original source points which all points in between have to be interpolated from. The difference of intensity between points P and A is equal to that

between points R,P and B,R (due to linearity of the interpolation). This difference, marked as dAB , is given by $dAB = \frac{1}{3}(M(i+1, j) - M(i, j))$. In the same manner $dAC = \frac{1}{3}(M(i, j+1) - M(i, j))$ and $dCD = \frac{1}{3}(M(i+1, j+1) - M(i, j+1))$.

Now, the intensities in the points between A and C are computed simply by adding dAC to $M(i, j)$ and the intensity in the point P , (resp. Q) is gained by adding dAB (resp. dCD) to the intensity in A (resp. C). Now we compute dPQ from intensities in the points P and Q and continue by computing values between P and Q . In the same manner we compute all remaining values as shown by the arrows in Figure 2.8. Then we move to the next four points $[i+1, j], [i+2, j], [i+1, j+1], [i+2, j+1]$ and repeat the procedure.

This modification considerably improves performance but the most significant speed-up comes with the following modification. As shown in Section 2.2.1 majority of voxels of the input matrix of intensities have intensity value equal to zero or near zero. Moreover, these zero voxels are mostly situated in a continuous area along borders of the matrix while higher-intensity voxels can be found mostly near the centre of the matrix. It is evidently a waste of time to make an interpolation in an area where all voxels have the same value. Thus, it is convenient to slightly modify the algorithm. A condition checking whether all the four source points have the same values is put in every loop of the just described algorithm. In the case of equality the whole square is directly filled using the appropriate color. This can be improved even more. If the algorithm finds out that $M(i, j), M(i+1, j), M(i, j+1)$ and $M(i+1, j+1)$ are equal, it takes a glance at the next two points $M(i+2, j), M(i+2, j+1)$ and repeats this for $i+3, i+4, \dots$ until a couple of points is encountered where at least one differs. Then the whole area explored this way can be filled by a single fast command for filling a rectangle.

Chapter 3

Data Registration

The aim of the study registration is to find a transformation mapping one volume (the *reslice* volume) to another volume (the *reference* volume) such that they are perfectly aligned [2]. The reference study (marked as M_r in the following text) is stationary during registration. All transformations are applied only on the reslice study (marked further as M_s). A registration of two data volumes is usually accomplished by using an iterative method that searches the space of linear transformations for the one maximizing some measure of alignment of the two volumes. Depending on the selected measure of similarity this is a *minimization* or a *maximization* process. Since minimization of a function f is equal to maximization of a function $-f$ all *minimization* methods described here can be easily used with similarity measures that need to be maximized to align volumes. There exist several methods that try to compute the transformation analytically. However, the iterative numerical methods in the frame of SPECT or PET images are much more popular. Note that the fact that a certain similarity measure has been minimized does not imply that the studies are aligned.

The registration methods used in this work operate on the full image content through the registration process and do not use any additional information (e.g. significant features or landmarks). For the purpose of clinical use it is completely sufficient to restrict the linear transformation to a combination of scaling, rotation and translation. These three transformations and their combinations belong to the *affine* transformations. (An affine transformation

transforms by definition parallel lines onto parallel lines again.)

The space of the affine transformations searched by a minimization method is nine-dimensional - three dimensions for translation, three for rotation and three for scaling. When two studies to be registered originate from the same patient it is clear that the only misalignments that can occur are shift and rotation. Therefore it is reasonable to prohibit scaling of the reslice image in such case. Employing this constrain reduces the number of degrees of freedom to six and provides a considerable speed-up. Transformations considering only translation and rotation are commonly called *rigid body* transformations.

Before making any transformations or computing similarity measures it is necessary to adjust the two volumes to have the same metrical dimensions and other parameters. This problem is discussed in Section 3.1.

Section 3.2 deals with the implementation of a fast and sufficiently precise transformation of images.

Section 3.3 presents a short overview of the important similarity measures used in medical imaging. The measures implemented in the program Brien are described together with an analysis that led me to choose them. Their advantages and disadvantages are discussed and important implementation details are also included.

An overview of the most commonly used numerical methods is presented in Section 3.4 together with a short summary of their pros and cons. According to this analysis several methods were implemented and tested.

Some kind of visual inspection on the quality of a registration is very important since the automatic registration can fail occasionally. Several simple but very efficient methods were developed and implemented. They are described in Section 3.5.

The efficiency and quality of different combinations of similarity measures and registration methods has been tested on typical applications from medical practice. They are summarized in Section 3.6.

3.1 Data Parameters Adjustment

The matrix parameters (Figure 2.2) very often differ for different studies. This case is illustrated in Figure 3.1 for two studies M_r , M_s . The matrices are displayed using their relative metrical proportions. Before starting any registration the studies must be adjusted so that voxels in the two studies have the same dimensions and that an arbitrary voxel $M_r(i, j, k)$ is placed on the same position in the xyz space as the voxel $M_s(i, j, k)$.

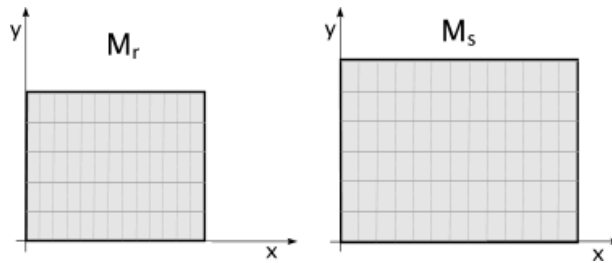


Figure 3.1: Two studies with different matrix parameters

One way to solve this situation is to resample the study with larger voxels to have the same parameters as the second study. The advantage of this approach is that the degree of image degradation is lower than in the case where the study with smaller voxel size would be resampled. The disadvantage is that the volume of the study with the smaller voxel size may be insufficient to contain all the information from the larger-voxel study. This is the case when it would be more convenient to resample the study with smaller voxels.

It can happen that the study with smaller voxel size in the xy plane may have larger voxel size in the z direction. In this case it is difficult to choose the proper study to resample and the decision depends on the current situation. Since there is not a straightforward way to choose which way is the best I decided to let user select the study that is desired to be resampled. A trilinear interpolation is always used for image resampling. (Trilinear interpolation is a 3D generalization of the 2D bilinear transformation described in Section 2.6).

3.2 Geometric Transformation of Data

This section describes the process of transformation of a tomographic study volume that was suggested and implemented in the program Brien.

Every linear transformation can be represented as a 4x4 matrix \mathbf{A} :

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{pmatrix}$$

Any point $\mathbf{P} = [x, y, z]$ of a three-dimensional orthogonal space that is to be transformed by the matrix \mathbf{A} must be first expressed in the corresponding *homogenous* coordinates as $\mathbf{P}_h = [x', y', z', w]$. It is not possible to express an arbitrary linear transformation by a single matrix without employing the homogenous coordinates. For the purpose of this thesis it is sufficient to state that the homogenous coordinates of an arbitrary point $\mathbf{P} = [x, y, z]$ are $\mathbf{P}_h = [x, y, z, 1]$. Precise definition of homogenous coordinates and their use can be found in [22]. Transformation of an arbitrary point \mathbf{P} onto \mathbf{P}' using the matrix \mathbf{A} and homogenous coordinates is performed as

$$\mathbf{P}'_h = \mathbf{P}_h \mathbf{A} \quad (3.1)$$

A matrix gained by matrix multiplication of several matrices like \mathbf{A} also represents a linear transformation. Because matrix multiplication does not commute, the precise form of the transformation matrix depends on the order in which the individual transformations are performed. Recall from the introduction to the chapter 3 that only translation, rotation and scaling are considered in this work. It seems intuitive to me to make scaling first and then rotation followed by translation. If \mathbf{A}_s , \mathbf{A}_r and \mathbf{A}_t are matrices representing scaling, rotation and translation, then the whole transformation of an arbitrary point \mathbf{P} is given by

$$\mathbf{P}'_h = \mathbf{P}_h \mathbf{A}_s \mathbf{A}_r \mathbf{A}_t \quad (3.2)$$

I will mark the transformation matrix resulting from multiplication of matrices \mathbf{A}_s , \mathbf{A}_r and \mathbf{A}_t in the equation 3.2 as \mathbf{A}_p . The 3D process of transformation of a study M using matrix \mathbf{A}_p is illustrated in two dimensions in

Figure 3.2. The Figure 3.2 a) shows the initial state of the study. The point \mathbf{C} is the centre of the study. The matrix is first translated by the vector $-\mathbf{V}_c$ so that the point \mathbf{C} lies in the origin of the coordinate system (Figure 3.2 b). Then the transformation matrix is applied (Figure 3.2 c) and the study is translated back by vector \mathbf{V}_c (Figure 3.2 d). The study M after transformation does not cover the same space as before transformation. For correct evaluation of similarity measures (see Section 3.3) it must be ensured that the study is always aligned with the coordinate axis x , y and z and that it keeps always the same (metric) dimensions. Thus it is necessary to resample the transformed study M into a new study (new matrix of intensities) M_o . Intensity values of the missing voxels are set to zero and the redundant are discarded.

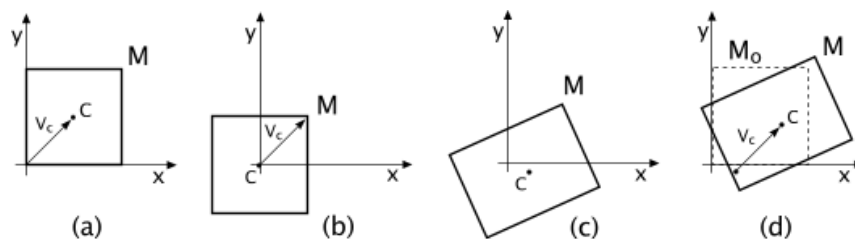


Figure 3.2: Transformation of a study

Here is an important thing to realize. Every study corresponds to a real object and all transformations must be applied using its real proportions. It means that transformations must be performed in the *metrical* coordinate system. Applying a transformation directly in the *voxel* coordinates would lead to incorrect results.

To transform an entire 3D study M into the new study M_o , each voxel location $([x, y]$ in Figure 3.2) in M could be transformed using \mathbf{A}_p , and then the voxel value from M written into the voxel in M_o that most closely corresponds to the transformed coordinate $([x_o, y_o])$. This approach is highly unsatisfactory, because it does not guarantee that each voxel in M_o will be assigned a value. A solution is to use the inverse matrix \mathbf{A}_p^{-1} to transform the coordinate $[x_o, y_o, z_o]$ of each voxel in M_o into the coordinates $[x, y, z]$ of M , take the value of the point nearest to $[x, y, z]$ and assign it to $[x_o, y_o, z_o]$. Since the centre of each voxel in M_o may not map exactly to the centre of a voxel in M , it is convenient to use the trilinear interpolation to calculate a voxel value for M_o instead of simple nearest neighbour.

Using just described schema, computation of every voxel of M_o would require one matrix multiplication (by \mathbf{A}_p^{-1}) and one 3D interpolation. A great speed-up can be provided by using the fact that the transformation is linear. This notion led to the following observation. Let $[x_o, y_o, z_o]$ be a point of the study M_o and $[x, y, z]$ be the corresponding point of the study M gained by using of \mathbf{A}_p^{-1} and interpolation. Let $[i_o, j_o, k_o]$ and $[i, j, k]$ be their respective voxel coordinates. The point $[i_o + 1, j_o, k_o]$ (in voxel coordinates of M_o) corresponds to the point $[x, y, z] + \mathbf{V}_x$ (in metrical coordinates of M) where \mathbf{V}_x is a vector that represents the change of metrical coordinates in the study M corresponding to the change of the voxel coordinate i coordinate in M_o by one. Knowing the analogous vectors \mathbf{V}_y and \mathbf{V}_z , the point of M corresponding to the point $[i_o + a, j_o + b, k_o + c]$ of the study M_o can be computed simply as $[x, y, z] + a.\mathbf{V}_x + b.\mathbf{V}_y + c.\mathbf{V}_z$.

Thus to make the whole transformation it is sufficient to apply the matrix multiplication only once to a "starting point" and three vectors \mathbf{V}_x , \mathbf{V}_y and \mathbf{V}_z . After that, values of all voxels in M can be computed by looping over the study M_o and using just addition (or subtraction) of vectors and interpolation.

3.3 Similarity Measures

The term *Similarity measure* as used in medical imaging denotes an intensity-based cost function that shows how well two data volumes are aligned. The cost function is calculated from all voxels of the reference and the reslice volume. It has a global extreme when the two volumes are perfectly aligned.

For illustrative purposes it is convenient to introduce the concept of the *joint histogram*.

Joint Histogram

Joint histogram is a 3D bar graph with the intensity values of the reference volume along the x axis and the intensity values of the reslice volume along the y axis [2]. The whole range of intensities is divided into several (usually 128 for PET/SPECT studies and 256 for MRI studies) equally spaced bins.

The height of a bar at the position $[x, y]$ represents the number of voxels which have the particular intensity surrounded by the bin x in the reference volume and at the same time have the intensity falling into the bin y in the reslice volume. Computation of the joint histogram has been implemented as follows: The two volumes - reference and reslice - are simultaneously passed through. For every voxel $[i, j, k]$ with intensity $M_r(i, j, k)$ in the reference volume and $M_s(i, j, k)$ in the reslice volume the value of the bar at the position corresponding to the intensities $[M_r(i, j, k), M_s(i, j, k)]$ is increased by one.

A straightforward way to display a joint histogram is to make an image where each pixel represents one bar and where the pixel intensity corresponds linearly to the height of that bar. Such an image would not bring much information since the bar $[0,0]$ and a few surrounding bars are always (for tomographic data) much higher than the other bars in the joint histogram that would be totally suppressed. More convenient way is to take a logarithm of a pixel intensity as the height of the corresponding bar.

In analogy with the Section 2.2.1 it is convenient to introduce the *normalized joint histogram*. It can be gained by dividing each value of the joint histogram by the number of voxels in a single study. (The two volumes must have the same dimensions.) The normalized joint histogram can be considered as an estimate of the joint probability distribution of intensity in the two volumes. The normalized histogram of the volumes M_r and M_s will further be denoted as H_{M_r, M_s} and its values as $H_{M_r, M_s}(i)$, $0 < i < n$, where n is the number of bins of the histogram. The joint probability distribution of the two volumes will be marked as $P(M_r, M_s)$.

Joint histogram is a tool that enables some sort of visual evaluation of tomographic volumes alignment. Ideally, for two identical volumes from the same modality the joint histogram is a diagonal line with the highest intensity near the point $[0,0]$ fading to the second end of the line (Fig. 3.3 a)). For real volumes the joint histogram differs more or less from this ideal situation.

The Figure 3.3 b) shows the joint histogram of two identical studies where one has been rotated around the z axis by 2 deg, while in the Figure 3.3 c) one has been moved by 80mm in the y axis. In Fig. 3.3 d) there is a joint histogram of two different intra-modality studies of one patient. The studies were previously aligned using the program Brien.

The linear trend present in the joint histogram is a specific property of studies

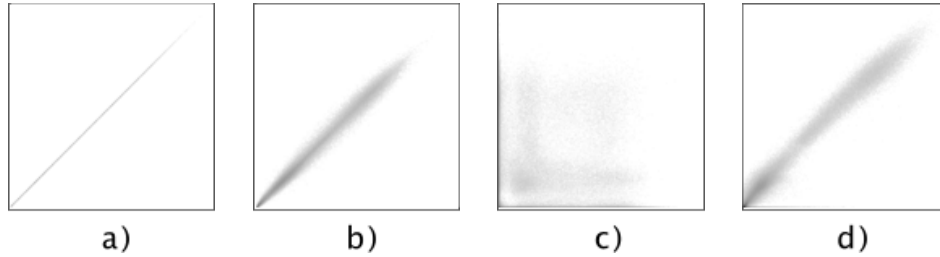


Figure 3.3: Joint histograms

from the same modality. Within different modalities it would be generally nonlinear.

Most of similarity measures essentially gauge the distance between pixels in the joint histogram and a straight line. They are mostly based on an arithmetic operation that is made on each pair of corresponding voxels in the two volumes. Outputs of these suboperations are combined together and result in a value showing a measure of alignment of the two volumes. This is the case of the first three similarity measures presented in following sections.

The fourth, latest similarity measure that I have chosen to implement is completely different. It is based rather on the probability intensity distributions of the two images and it estimates the extent of similarity of the two intensity distributions. Thus it does not operate on individual pairs of corresponding voxels but operates on the probability information gained from the whole volumes.

3.3.1 Count Difference

One of the simplest similarity measures is *count difference* [16]. The count difference $CD(M_r, M_s)$ of two volumes M_r and M_s is defined as the sum of absolute count differences between all voxels of M_r and M_s normalized to the number of voxels in the volume:

$$CD(M_r, M_s) = \frac{1}{vol} \sum_{i,j,k} |M_r(i, j, k) - M_s(i, j, k)|$$

where $vol = numberOfSlices \cdot matrixSize^2$.

By application of this measure it is assumed that the voxel values of the two studies are calibrated to the same scale so that corresponding objects exhibit the same intensity value. In this case the count difference measure tends to zero value in alignment. In case the studies differ more than by only a Gaussian noise the behaviour is less predictable. Thus this measure is very good for aligning nearly identical volumes (e.g. studies of the same patient intended to investigate progression or regression of disease, etc.).

3.3.2 Squared Difference

The sum of squared intensity differences between two volumes (shortly *squared difference measure*) should be the optimum measure in case the two studies differ only by Gaussian noise [7]. For two volumes M_r and M_s it is given by equation:

$$SD(M_r, M_s) = \frac{1}{vol} \sum_{i,j,k} ((M_r(i, j, k) - M_s(i, j, k))^2)$$

where $vol = numberOfSlices \cdot matrixSize^2$.

Usage of this measure is similar to the count difference.

3.3.3 Shape Difference

This measure determines the alignment quality of shapes of the volumes M_r , M_s [6]. Both volumes are first transformed into (3D) binary masks B_r , B_s as follows: for a single volume, all voxels with the intensity value higher than some fixed threshold are marked as 1 while the rest of the volume is marked as 0. Thus all voxels marked as 1 create together a three-dimensional body approximately depicting the shape of that volume (corresponding to the threshold value). The shape difference $SH(M_r, M_s)$ of volumes M_r and M_s is the number of voxels in the two binary masks B_r , B_s that do not overlap (do not have the same parity):

$$SH(M_r, M_s) = \frac{1}{vol} \sum_{i,j,k} (B_r(i, j, k) \text{ XOR } B_s(i, j, k))$$

where $vol = numberOfSlices \cdot matrixSize^2$.

3.3.4 Mutual Information

The similarity measure based on mutual information is quite a new concept brought to the medical imaging independently by Collignon et al. in 1995 and by Viola and Wells in 1996. It is a new approach with big expectations. The possibilities of using this measure are open so far. The measure of mutual information is used in communication theory to describe the information carried by a communication channel, by relating the information content of the transmitted and received symbols [17].

The previously described methods take pairs of corresponding voxels in the two images one by one to compute the value of a similarity measure. The general idea of the similarity measure based on mutual information is completely different. Instead of using directly voxels this method estimates the differences between the probability intensity distributions of the two volumes. Thus it makes no assumptions about the relationship between the intensities in the images to be registered. The measure is based on the Kullback-Leibler distance.

Kullback-Leibler Distance

The *Kullback-Leibler distance* (*KL distance*) is designed to gauge how different two distributions are. The role of this measure in probability matching is similar to the role of Euclidean distance in data matching.

The *KL distance* between two probability density functions $f(x)$ and $g(x)$ is defined as [9]:

$$KL(f, g) = \int f(x) \log \frac{f(x)}{g(x)} dx$$

where the logarithm is understood to the base e . By convention $0 \log \frac{0}{0} = 0$ whenever $f(x) = 0$.

One of the most important properties of the *KL distance* is **non-negativity**: $KL(f, g) \geq 0$ with equality in the case of identical distributions; another

important properties are **asymmetry** and **convexity**. Precise formulations and proofs of these theorems can be found for example in [9] or [20].

This measure has been successfully adapted to image registration by defining the probability distributions of each of the image volumes as $P(M_r)$ and $P(M_s)$ and by defining the joint probability distribution of the two volumes $P(M_r, M_s)$ [2]. Then the measure

$$MI(M_r, M_s) = \int P(M_r, M_s) \log \left(\frac{P(M_r, M_s)}{P(M_r)P(M_s)} \right) dM_r dM_s$$

may be used to gauge the amount of mutual information in the two images by estimating the difference between the joint distribution and that calculated from the assumption of complete independence.

Recall from the Sections 2.2.1 and 3.3 that the (marginal) probability distributions $P(M_r)$ and $P(M_s)$ can be approximated by the corresponding normalized histograms H_{M_r} and H_{M_s} and the joint probability distribution $P(M_r, M_s)$ can be estimated by H_{M_r, M_s} . Then $MI(M_r, M_s)$ can be estimated as

$$MI(M_r, M_s) = \sum_{i,j=0}^n H_{M_r, M_s}(i, j) \log \left(\frac{H_{M_r, M_s}(i, j)}{H_{M_r}(i)H_{M_s}(j)} \right) \quad (3.3)$$

Maximization of the mutual information implies minimization of the information provided by pairs of values occurring together in the combined image in relation to that provided by the two images separately. Registration of images using mutual information means searching for a transformation that transforms the reslice volume to a position where the amount of information that one volume contains about the other is maximal.

To speed-up the computation, the mutual information of two images is evaluated simultaneously with the computation of the marginal histograms and the joint histogram.

3.4 Data Registration Methods

The general requirements on a registration method that come up from the clinical use can be summarized into the following points:

- **accuracy:** The accuracy of the image registration is very difficult to define and measure. It can be partially supplied by a sort of visual inspection. Several methods for visual accuracy checking implemented in program Brien are described further in this chapter.
- **robustness:** The algorithm should be insensitive to local minima as much as possible. Small variations in an input should result in small variations in an output.
- **resource requirements:** These resources include computational time, computer memory requirements and also the degree of needed user interaction.
 - **computational time:** Most of the computational time is spent on evaluating of the cost function. The algorithm should try to reduce the number of evaluation of the function as much as possible. The computational time should be reasonably relative to the clinical merit obtained from the registration and must be adapted to the time and resource constraints of the clinical environment.
 - **memory:** All common minimization algorithms require storage space of order not higher than $O(n^2)$, where n is a number of dimensions of the space that is searched by the minimization algorithm. Considering only linear transformations we have nine independent variables maximally, so memory requirements are out of question here. However, considerably larger space must be provided for computing values of the cost-function that is to be minimized, but this is not a matter of the minimization algorithm.

Taking into account these requirements the following three minimization methods has been chosen and implemented in the program Brien: *Powell's direction set* method, *Downhill simplex* method and *Conjugate gradient* method. Roughly speaking, the Powell's method was chosen because it is supposed to be fast and accurate, the Conjugate gradient method is very robust and the Downhill simplex method is very often able to give quite good results quickly.

All minimization methods that I have employed search a minimum of function of n independent variables $f(\mathbf{P})$ where $\mathbf{P} = [p_1, p_2, \dots, p_n]$ by evaluating the

function f (or even its gradient ∇f) at different points and by using this information to adjust the search in the next steps. This general approach has been implemented in the application Brien in the following way:

The space searched for minimum of f is nine-dimensional and a point $\mathbf{P} = [p_1, \dots, p_9]$ from that space represents a linear transformation applied on the reslice volume: p_1, p_2, p_3 stand for translation in x, y and z directions, p_4, p_5, p_6 mean rotation around axis x, y, z and p_7, p_8, p_9 represent scaling in x, y, z directions. For example, $p_2 = 1$ means that all voxels are shifted by one voxel in the direction of the y axis, $p_4 = 2$ determines that the volume has been rotated by two radians around the x axis with the centre of rotation in the middle point of the yz slices. When $p_9 = 1.1$, the study is scaled to 110% in the z direction.

Next, let M_r be the reference (stationary) volume and M_s the reslice volume and $h_{M_r, M_s}(\mathbf{P})$ function that evaluates a measure of similarity (e.g. CD, SH, SD or MI) of the reference volume M_r and the reslice volume M_s , transformed according to the point \mathbf{P} . The minimization methods have been implemented so that they evaluate the function $h(M_r, M_s)$ in different points \mathbf{P} and they adjust the search according to this information to find the point \mathbf{P}_m where h has its minimum (with some tolerance).

3.4.1 Powell's Direction Set Method

This method has been first published by M.D.J Powell in 1964 [12]. The implementation of the Powell's method is based on [13]. The *direction set* method finds the minimum of a function with n independent variables by a succession of one-dimensional minimizations. It profits from the fact that one-dimensional minimization is an easier problem to solve numerically than a n -dimensional minimization. Each iteration of Powell's method consists of n one-dimensional minimizations. Considering only translations, rotations and scaling the space to be searched is nine-dimensional. For rigid body transformation it has six dimensions.

For the first iteration it is usual to do one-dimension minimizations along the axes of the nine (six) dimensional space (i.e. to take the unit vectors as the initial set of directions). After each iteration the method adjust the set of searched directions by discarding one of them and replacing it by a new

direction.

Let \mathbf{P}_0 be the initial position in the searched space (i.e. the initial transformation, see Section 3.4) and $\mathbf{u}_i, i = 1, ..n$ the initial set of directions. The vectors \mathbf{u}_i are initially set to the basis vectors \mathbf{e}_i . A single iteration consists of the following steps:

1. For $i = 1, \dots, n$, move \mathbf{P}_{i-1} to the minimum along the direction \mathbf{u}_i and call this point \mathbf{P}_i .

Let i_{max} be the index of the direction $\mathbf{u}_{i_{max}}$ where the function h_{M_r, M_s} made its largest decrease.

2. Set $\mathbf{u}_{i_{max}} = \mathbf{u}_n$. Set $\mathbf{u}_n = (\mathbf{P}_n - \mathbf{P}_0)$.
3. Move \mathbf{P}_n to the minimum along the new direction $\mathbf{u}_{i_{max}}$ and call this point \mathbf{P}_0

The direction that is discarded is the direction where the largest decrease of the function h_{M_r, M_s} occurred during one iteration. The reason is that this direction is likely the major component of the newly added direction $\mathbf{P}_n - \mathbf{P}_0$ and it is better to discard it to avoid building of linear dependence.

As an addition to the just described general schema there are few cases when the direction set is kept unchanged. The conditions that must be met to skip adjusting of the direction set are not too much intuitive. They can be found in [13] or [12] together with more detailed description of all nuts and shells of the algorithm.

The algorithm finishes when an iteration fails to improve the similarity measure evaluation by more than some defined tolerance tol . If f_{j-1} is the value of the function h_{M_r, M_s} after $(j - 1)$ th iteration and f_j is the value after j th iteration, then the minimization stops if the following expression is true:

$$\frac{f_{j-1} - f_j}{\frac{1}{2}(|f_{j-1}| + |f_j|)} \leq tol$$

According to Maes [10] the choice of ordering of the searched directions can influence the success of the minimization. Since typical medical images have

higher resolution in the xy plane than in the z direction, Maes suggests to search the in-plane degrees of freedom first (i.e. components p_1, p_2, p_4, p_7 and p_8 of a point \mathbf{P}).

The implementation of the one dimensional line minimization subprocedure uses bracketing of minimal value in the searched direction followed by the Brent's line minimization method as described in [13].

3.4.2 Downhill Simplex Method

The *downhill simplex* method was published by Nelder and Mead in 1965 [11]. This method requires only evaluations of the function h and not its derivatives. The idea is to initially evaluate the function $h_{M_r, M_s}(\mathbf{P})$ in $n + 1$ points of the n -dimensional space and then to iteratively replace individual points with new points lying closer to a minimum. In the case of registration of two studies n is nine, thus ten initial points must be evaluated. (For the rigid body registration $n = 6$.) These points, together with all their inter-connecting line segments, polygonal faces, etc. create a geometrical object called *simplex*. The points are vertices of the simplex. For illustration, in two dimensions simplex is a triangle and in three dimensions it is a tetrahedron. Given an initial estimate vertex \mathbf{P}_0 (which is usually a zero vector), it is convenient to set the remaining nine initial vertices to $\mathbf{P}_i = \mathbf{P}_0 + \lambda_i \mathbf{e}_i$ where \mathbf{e}_i is the unit vector in the i -th dimension and λ_i 's could be considered as initial estimates of the area where the minimum might lie. (This does not mean that the method will search for the minimum only in the area given by the set of λ_i .) Further I will denote \mathbf{P}_h the vertex where the value of function h is maximal (the worst vertex): $\mathbf{P}_h = \mathbf{P}_i$, where $h_{M_r, M_s}(\mathbf{P}_i) \geq h_{M_r, M_s}(\mathbf{P}_j), \forall i, 1 \leq i \leq n + 1, i \neq j$. Analogously \mathbf{P}_l denotes the vertex with minimal value of h_{M_r, M_s} (the best point) and \mathbf{P}_{nh} the vertex with the second-highest value. For simplicity I will denote h_i the function value in the point P_i , i.e. $h_i = h_{M_r, M_s}(P_i)$.

Further I will denote as \mathbf{P}_s the centroid ("gravity centre") of the face of the simplex opposite to \mathbf{P}_h : $\mathbf{P}_s = \frac{1}{n} \sum_{i, i \neq h} \mathbf{P}_i$. In each iteration the point \mathbf{P}_h is replaced by a new point. Three operations can be used: *reflection*, *contraction* and *expansion*. They are graphically illustrated in Figure 3.4 for a two dimensional case. Figure 3.4 a) shows the initial simplex. The first

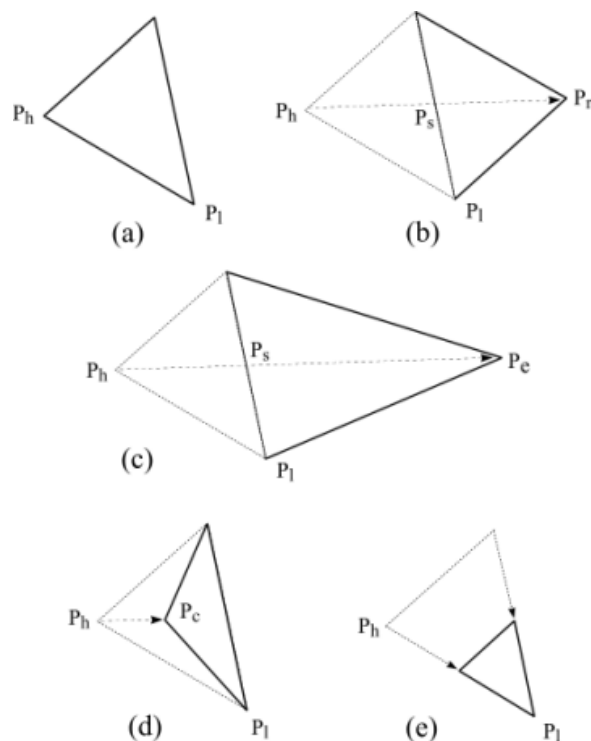


Figure 3.4: Possible steps of the downhill simplex method in one iteration: a) initial simplex, b) reflection, c) expansion, d) one-dimensional contraction, e) contraction around P_l

step is to reflect the highest point \mathbf{P}_h through the opposite face (line in a 2D case) of the simplex to a new point \mathbf{P}_r , i.e. \mathbf{P}_r lies on the line joining \mathbf{P}_h and \mathbf{P}_s , on the far side of \mathbf{P}_s from \mathbf{P}_r (Fig. 3.4b):

$$\mathbf{P}_r = \mathbf{P}_s + (\mathbf{P}_s - \mathbf{P}_h)$$

If $h_l \leq h_r \leq h_h$ then \mathbf{P}_h is replaced by \mathbf{P}_r and the method goes on with the new simplex.

If the reflection has produced a new minimum ($h_r < h_l$) then \mathbf{P}_r is expanded to a new point \mathbf{P}_e (Fig. 3.4c):

$$\mathbf{P}_e = \mathbf{P}_s + 2 \cdot (\mathbf{P}_r - \mathbf{P}_s)$$

If $h_e < h_l$ then \mathbf{P}_h is replaced by \mathbf{P}_e . On the other hand, if $h_e \geq h_l$ (expansion

failed) then \mathbf{P}_h is substituted with \mathbf{P}_r . The new simplex goes on for the next iteration.

If the point \mathbf{P}_r reflected from \mathbf{P}_h still has the maximum value in the simplex, i.e. $h_r > h_i, \forall i \neq h$, then a new \mathbf{P}_h is set to be either the old \mathbf{P}_h or \mathbf{P}_r , whichever gives lower value. The new \mathbf{P}_h is then contracted along the line joining \mathbf{P}_h and \mathbf{P}_s to a new point \mathbf{P}_c (Fig. 3.4d):

$$\mathbf{P}_c = \mathbf{P}_s + \frac{1}{2}(\mathbf{P}_h - \mathbf{P}_s)$$

If the contraction found a better point, i.e. $h_c \leq \min(h_h, h_r)$, \mathbf{P}_c is accepted to be a new \mathbf{P}_h and the algorithm continues with the next iteration. In case of failed contraction the simplex contracts itself around the best point \mathbf{P}_1 . This contraction means that every vertex of the simplex except \mathbf{P}_1 is replaced by the point in the middle of the line connecting \mathbf{P}_1 and the vertex under consideration (Fig. 3.4e). The resulting simplex continues for the next iteration.

The termination criterion for downhill simplex method is different from that of Powell's method. The decrease of function value in one iteration cannot be used because it is usual that one iteration passes without any decrease of $h_{M_r, M_s}(\mathbf{P}_1)$. (It happens for example when the simplex only contracts itself.) Thus the criterion is that the fractional range from the highest point \mathbf{P}_h to the lowest \mathbf{P}_1 must be smaller than a given fractional convergence tolerance tol to stop the minimization process [13]:

$$\frac{h_{M_r, M_s}(\mathbf{P}_h) - h_{M_r, M_s}(\mathbf{P}_1)}{\frac{1}{2}|h_{M_r, M_s}(\mathbf{P}_h) + h_{M_r, M_s}(\mathbf{P}_1)|} < tol$$

The value of tol is usually between 0.05 – 0.1.

3.4.3 Conjugate Gradient Method

As the name suggest, the *conjugate gradient* method uses gradient $\nabla f(\mathbf{P})$ in addition to function values $f(\mathbf{P})$ for minimization of a similarity measure function f . Like the Powell's method, the conjugate gradient method consists of a sequence of line minimizations along various directions. The difference is

that the conjugate gradient method uses information about a local function gradient to select the most convenient directions for one-dimensional search.

The method is based on the *steepest descent method* [13]. This method minimizes a function by making a series of line minimizations along the direction in which the function f decreases most quickly, which is the direction opposite to ∇f . This idea may be expressed as:

1. Start at a point \mathbf{P}_0 .
2. As many times as needed, move from the point \mathbf{P}_i to the point \mathbf{P}_{i+1} by minimizing along the line from \mathbf{P}_i in the direction $-\nabla f(\mathbf{P}_i)$.

The problem of the steepest method is that in some cases the direction of the largest function decrease is not the most convenient one. For example, if the function has a shape of a long narrow valley and the method begins at an upper rim of one side of the valley, it has to perform many small steps from one side to the other side (because that is the direction of the largest function decrease) instead of making one long step in the direction of the valley.

Thus the method uses much more complicated schema for selection of a new search direction. The schema is completely described in [15]. The implementation of the conjugate gradient method is based on [13].

3.5 Validation of Results

Registration of tomographic studies using an automatic (performed by a computer) registration method can never get along without some sort of visual inspection that validates the results. Numerical algorithms involved in the registration are still just numerical algorithms that are likely to fail in finding a correct solution from time to time. Similarity measures, as well, are only approximations of the exactly indefinable state when two different studies are aligned. It is important to check the result that the used automated method claims to be the right one. This validation is usually made by a medical specialist.

Similarity measures naturally provide a *numeric* information about alignment. This section deals with *visual* information that the two registered studies can give.

3.5.1 Contour Matching

The idea of contour matching is to find contours of both images first and then for each image to display its contour over the counterpart image.

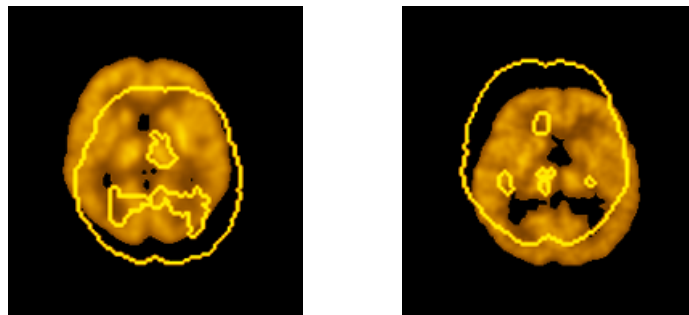


Figure 3.5: Two different misaligned SPECT studies, each with contour of the companion study

A simple method for determining a contour of an image was suggested and implemented. A contour is determined by the value of the lower threshold (see Section 2.5 for details about thresholds). The contour is a set of voxels of the currently displayed slice with intensities approximately equal to the intensity corresponding to the lower threshold. More precisely, a contour is composed of the voxels of the currently displayed slice that have an intensity higher than the lower threshold and that neighbour on at least one voxel (of the current slice) with intensity lower or equal to the lower threshold.

First, a binary mask of the current slice is created where pixels with value 0 correspond to the voxels of the slice with intensity below the lower threshold and pixels with value 1 correspond to the voxels with intensity above the lower threshold. This binary mask is then passed through and every pixel with value 1 neighbouring on a pixel with value 0 is marked as a 'contour' pixel.

The Figure 3.5 shows two misaligned SPECT studies - a reference image with

the contour of the reslice image and vice versa.

3.5.2 Chess Image Matching

Chess Image Matching is an impressive method that allows to visually check alignment of images. As the name suggest, the idea is to cut the reference and the reslice image into $n \times n$ equally sized squares and to mix the squares between the two images. Using a chess board for illustration, one of the resulting images contains squared parts of the reference image on white fields and parts of the reslice image on the black fields. Analogously, the second resulting image contains the rest of the parts of the reference and reslice image.

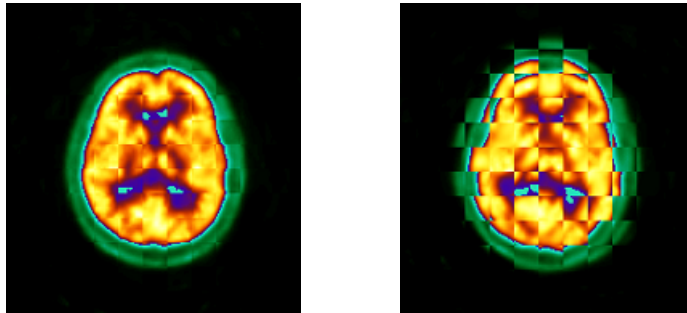


Figure 3.6: Chess image of aligned (a) and misaligned SPECT studies (b)

The Figure 3.6 a) presents a chess image of two SPECT images registered using Brien. Figure 3.6 b) presents the same images before registration in their original positions. Chess image shows mainly the continuity or discontinuity of contours of the combined image.

3.5.3 Subtraction of Images

Subtraction of images is a tool that was designed to provide a way to extract significant differences between studies and to compare an arbitrary brain study to a standard one. The general idea is to create a new volume M_s by subtracting two studies (voxelwise) and by setting intensity differences lower than some value v to zero. Two distinct ways have been employed.

The first approach is used to estimate how much a certain brain differs from a standard template (for description of templates see Chapter 4. Thus one of the studies loaded in the program Brien (either the reference or the reslice one) represents a template and the other represents a study to compare. I will mark the template volume as \overline{M} and the compared volume as M_c . The idea is to use the volume of standard deviations \overline{M}_v corresponding to the template study \overline{M} for decision whether to set a particular difference value to zero. It is convenient to express this procedure in a pseudo-code:

```

for i=1 to matrixSize {
  for j=1 to matrixSize {
    for k=1 to numberOfSlices {
       $M_s(i, j, k) = | \overline{M}(i, j, k) - M_c(i, j, k) |$ 
      if  $M_s(i, j, k) < q \cdot \overline{M}_v(i, j, k)$  then  $M_s(i, j, k) = 0$ 
    }
  }
}

```

The parameter $q, q > 0$ is a user-specified value that determines the tolerance.

The second method is suitable for extraction of significant differences between two studies. For example, one can have two studies of the same patient, before and after treatment. By subtraction of the two studies and by setting insignificant differences to zero, one can get a volume where the only nonzero intensities correspond to the area where the treatment was performed. By the same way it is possible to find out differences between studies taken in stress and rest, etc.

Having two studies M_a and M_b the procedure can be expressed as:

```

for i=1 to matrixSize {
  for j=1 to matrixSize {
    for k=1 to numberOfSlices {
       $M_s(i, j, k) = M_a(i, j, k) - M_b(i, j, k)$ 
      if  $M_s(i, j, k) < q \cdot \max_a$  then  $M_s(i, j, k) = 0$ 
    }
  }
}

```

Here max_a is a maximum value in the study M_a . The parameter $q, 0 < q < 1$ is a user-specified value that determines a portion of the maximum max_a representing the maximum tolerable difference. Note that values of M_s are not *absolute* values of differences. Thus all negative differences will also be set to zero. This implies that it is important whether M_b is subtracted from M_a or vice versa. The program Brien allows to choose the order.

For both methods, it must be ensured that intensity ranges of employed studies will not differ significantly. For this reason it is possible to normalize the studies according to the *total* or *maximum* criterion prior to subtraction. Normalization is described in Section 4.1.

3.5.4 Overlay of Images

The idea of this technique has been borrowed from the professional software in the Hospital Na Homolce. It is similar to the idea of chess images. As user moves with a mouse pointer over one image, the area under the mouse pointer is overlaid by a small square-shaped corresponding part of the other image. Thus, moving with the mouse pointer user moves the square and the two images can be compared locally.

3.5.5 Distance Measuring

Distance measuring is a supplemental tool implemented in the program Brien. It is intended to be used in conjunction with the previously presented visual inspection methods. It allows to draw lines over images displayed on the screen. Simultaneously, a length of a drawn line in millimetres is displayed. The tool is useful for measuring extent of misalignment (using displaying of contours or the chess images) or for measuring sizes of various objects like tumours or lesions.

3.6 Evaluation of Methods

Several practical experiments were performed to compare speed and accuracy of implemented numerical algorithms in conjunction with different similarity measures. Nevertheless, these comparisons have certain limitations. The main problem is comparison of quality and convenience of similarity measures since there is mostly no relation between different similarity measures.

Suppose, for example, that we have a pair of different studies and we let the Powell's method align them, once using the square difference measure and next using the mutual information. After an hour Powell decreases value of the square difference residual from 3.5 to 0.03 and increases value of the mutual information measure from 0.4 to 1.5. Since there is no relation between the two measures, it is impossible to say which combination is faster only from the resulting residual values. An acceptable (although inaccurate) way is to estimate the misalignment (in millimetres) of studies after registration. A little better is a situation in the case of two identical studies (initially misaligned). Then we know that an ideal registration would set all transformation parameters to zero. Thus different similarity measures can be compared according as they bring values of transformation parameters close to zero.

Comparison of different minimization methods using the same similarity measure is simple. The most successful method decreases (resp. increases) the corresponding residual value the most.

Behaviour of all implemented numerical methods depends on many input parameters. These include thresholds, an initial search range and a voxel interpolation method. Two similarity measures - mutual information and shape difference - also depend on initial parameters. Mutual information is sensitive to a number of histogram bins and shape difference is sensitive to a selected threshold value. Tests were performed for all minimization methods in conjunction with two similarity measures: mutual information and squared difference. These two represent two completely different approaches to measurement of study alignment. At the same time they are among the most often used measures.

Default values were used for the initial search range, no thresholds were applied to the studies and the linear interpolation was chosen for approximation

of values inside voxels. The number of histogram bins used for computation of mutual information was 128. The minimization methods are referred as CG (conjugate gradient), DS (downhill simplex) and PL (Powell). The similarity measures are denoted SD (squared difference) and MI (mutual information). Parameters that determine a transformation of a reslice study are marked as TX, TY, TZ (translation in the three axis in milimetres), RXY, RYZ, RXZ (rotation in xy , yz and xz planes in degrees) and SX, SY, SZ (scaling in percent). The word *accuracy* means how much a certain method is able to lower (raise) a residual value.

Two types of measurements were performed: registration of identical, initially misaligned studies and registration of a pair of different studies. As mentioned in Section 6.1.4, the program Brien can save information about residual, number of similarity function evaluations, time, etc. to a text file. These text files were used for evaluation of methods. The studies and text files used for comparison are included on the enclosed CD-ROM. All studies originate from the Hospital Na Homolce. The studies were chosen to be as most representative as possible. The presented results conform with the practical experience gained during the development and testing of the program Brien. All experiments were performed on a computer with Celeron 333 MHz processor, 192 MB RAM, running Windows NT4.0 operating system.

3.6.1 Identical Studies

The source study for the measurement of identical studies is contained in the file 081.a00 on the CD-ROM. It serves as a reference study and is denoted as M_1 . The source for the reslice study M_2 is the same except that it is initially misaligned.

The transformation parameters of the misaligned reslice study are the following: TX=20mm, TY=-10mm, TZ=8mm, RXY=-3 deg, RYZ=12 deg, RXZ=4 deg, SX=97%, SY=105%, SZ=104%.

Figure 3.7 shows the process of residual improvement for the three minimization methods using square difference measure (left) and mutual information (right).

The left graph shows that in the case of SD measure there are no significant

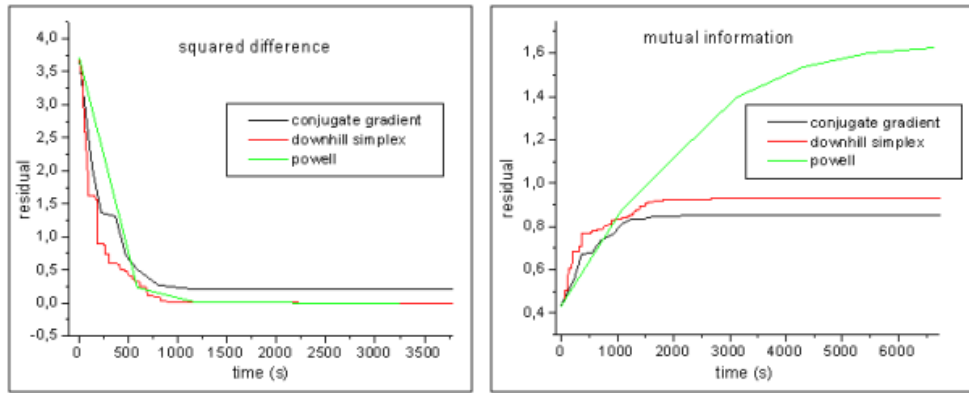


Figure 3.7: Registration of identical studies

differences between the three minimization methods. However, DS and PL methods are able to lower the residual more than CG. There is nearly no difference in speed and accuracy between DS and PL but DS gives partial results more often and thus it is able to give better results than PL at the beginning of registration.

For mutual information (right graph) the situation is quite different. Although a bit slower at the beginning, PL gives much more accurate results than both DS and CG.

To compare the two used similarity measures, transformation parameters are compared after 3000s and 6000s of registration. They are summarized for all examined combinations of minimization methods and similarity functions in the Table 3.1. For combinations DS+MI and PL+MI, the upper row relates to 3000s and the bottom row corresponds to 6000s. For the remaining combinations the transformation parameters already do not change after 3000s (at the three significant digits after the decimal point).

As we can see, the least accurate method is the conjugate gradient. The combinations DS+SD and PL+SD are approximately equally accurate. For the MI measure, PL is much more accurate than DS. The table also shows that the results that PL+SD and DS+SD give after 3000s are very similar to the results that PL+MI and DS+MI give after 6000s. Thus squared difference seems to be more convenient for registration of very similar studies than mutual information.

	TX	TY	TZ	RXY	RYZ	RXZ	SX	SY	SZ
DS + MI	0,074	0,043	-0,026	0,093	-0,010	0,013	105,942	100,119	100,034
	0,076	0,042	-0,021	0,087	-0,040	0,009	105,941	100,121	100,025
PL + MI	-0,042	-0,021	-0,292	-0,003	-0,006	-0,03	99,997	100,038	100,247
	-0,001	-0,001	-0,010	-0,001	0,00	-0,01	100,001	100,002	100,010
CG + MI	1,833	1,184	-1,184	0,803	0,871	1,646	100,358	100,261	102,658
DS + SD	0,036	0,03	0,002	-0,012	0,001	0,0	100,433	99,813	100,009
PL + SD	0,0	0,0	-0,053	-0,002	0,001	0,0	100,001	100,001	100,046
CG + SD	1,478	1,831	1,583	0,828	1,373	1,127	100,232	99,874	99,406

Table 3.1: Results of registration for different similarity measures and minimization methods for studies M_1, M_2

3.6.2 Different Studies

The studies 101.a00 and 032.a00 were used as a sample for registration of two completely different studies. The improvement of residual values for MI and SD measures is shown in Figure 3.8.

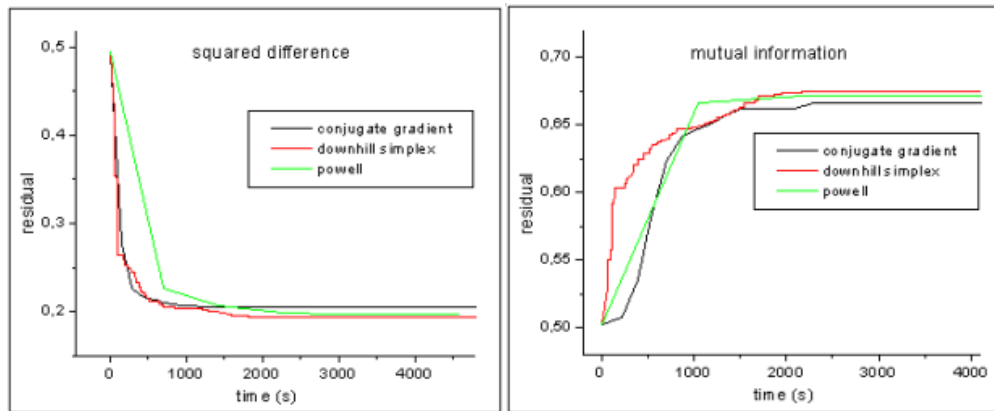


Figure 3.8: Registration of different studies

As a criterion for comparison of results the contour tool of the program Brien was used to display contours corresponding to 20% of the maximum intensity in the registered pair of studies. After that the maximal variation between contours (in millimetres) of the two studies was estimated. These estimates are summarized in the Table 3.2:

The estimates show that the two similarity measures SD and MI give very

method	estimated max. variation (mm)
DS + MI	6
PL + MI	8
CG + MI	7
DS + SD	9
PL + SD	6
CG + SD	7

Table 3.2: Estimated maximal variations

similar results for completely different studies.

3.6.3 Summary

The results gained in the Section 3.6.1 for two identical studies can be applied to any pair of studies that are supposed to be very similar, e.g. two studies from the same patient taken under similar conditions. In such a case the most convenient combination seems to be the Powell's method or the downhill simplex method in conjunction with the squared difference similarity measure. The conjugate gradient also gives quite good results, but it is not so accurate.

The mutual information similarity measure is more suitable for studies that are considerably different.

All results described in the previous two sections provide only a directory information. They represent the most typical cases and should not be taken as a rule. Different studies may give quite different results, although in general the results should be similar to those presented here.

Chapter 4

Templates

In medical diagnostic, it is often necessary to compare, how much a certain study differs from a standard one. This helps in localization of various anomalies in the study (e.g. tumour, centre of epilepsy). One can have standards (templates) for normal studies as well as for various anomalies and use them for detection of anomalies present in a study or to estimate the type of anomaly.

However, it is difficult to decide which study should be taken as a standard for normality, resp. as a standard for a certain type of anomaly. A convenient way to create a standard for some type of study is to take several studies that are thought to satisfy conditions to be qualified as a study of that type, and compute a mean study (a volume of mean values of voxels). Without adjusting the involved studies prior to making a template, this approach would lead to incorrect results. For example, creation of a template from two normal but very misaligned studies surely will not result in a good prototype.

Consider a set of N studies M_p , $p = 1, \dots, N$ that are intended for creation of a template study \overline{M} . These studies must be resampled first to have mutually equal parameters `matrixSize`, `numberOfSlices`, `pixelSize` and `sliceSeparation`. This can be accomplished using Brien. The intensity values of template voxels are given by equation:

$$\overline{M}(i, j, k) = \frac{1}{N} \sum_{p=1}^N M_p(i, j, k)$$

A volume of standard deviations \overline{M}_v corresponding to the study \overline{M} is given by [1]:

$$\overline{M}_v(i, j, k) = \sqrt{\frac{1}{N} \sum_{p=1}^N M_p(i, j, k)^2 - \overline{M}(i, j, k)^2}$$

In the frame of this set of studies, one should exert an effort to make the values of standard deviations in the volume \overline{M}_v as low as possible. For quality comparison of templates created from a fixed set of studies it is better to use the average v of all standard deviations instead of the whole volume:

$$v = \frac{1}{vol} \sum_{i,j,k} \overline{M}_v$$

where $vol = numberOfSlices \cdot matrixSize^2$.

Given a fixed set of studies, the quality of a template \overline{M} created from this set is then evaluated by v . The first step to make it lower is to align the studies spatially. One of the studies is depicted as the reference study and all other studies should be registered to the reference study (using for example Brien). However, not even this set of registered studies is suitable for template creation. The studies fit spatially but their intensity ranges may (and usually do) vary. The intensity ranges of all studies must be *normalized* to the range of the reference volume.

4.1 Normalization of Studies

Normalization of a study M_p , as understood in this work, is a process where the intensity value of every voxel in the study M_p is multiplied by a fixed *normalization factor* f_p : $M'_p(i, j, k) = f_p \cdot M_p(i, j, k)$.

Thus a normalization is fully determined by the choice of the normalization factor. It is usually different for every study and is chosen so that after normalization there is a property that is equal for all studies in the set. There are several ways how to choose a convenient normalization factor according to various properties that are to be equalized. The methods tested and used in this work are described in the following sections. They are implemented in the program Tepito.

4.1.1 Thresholds

4.1.2 Total

A simple and straightforward way is to make all studies have the same sum of intensity values. First, the sum of all intensity values (*total value*) sum_p for each study M_p is found. The question is then, what shall be the value that all sum_p should be equal to after normalization. I used the average sum_{avg} of values sum_p : $sum_{avg} = \frac{1}{N} \sum_p sum_p$. The normalization factor is then given by:

$$f_p = \frac{sum_{avg}}{sum_p}, p = 1, \dots, N$$

.

4.1.3 Maximum

The studies are normalized to have the same maximum value. Maximum value of each study M_p is max_p and max_{max} denotes the maximum of values $max_p, p = 1, \dots, N$. The value max_{max} is the reference maximum that all studies should have after normalization. This can be expressed as:

$$f_p = \frac{max_{max}}{max_p}, p = 1, \dots, N$$

.

Since this type of normalization depends on a single value of every study, it is likely not as robust as normalization according to the averages.

4.1.4 Quantil

The use of quantil values is a method that has been developed in this work. The idea comes from the specific shape of a SPECT/PET study histogram. In Figure 4.1 there are histograms of four SPECT studies taken from different patients. The lower threshold was set to the 2% of maximum intensity value in each study and the histograms do not display the first two bars. The maximum intensity is the same for each histogram and is set to the maximum

intensity max_{max} value in the whole set. The respective maximum intensity values of the studies are marked as max_1, \dots, max_4 . Thus it is possible to compare how the intensity ranges in studies differ. The shapes of histograms are very similar and we can distinguish quite clearly two peaks separated by a valley. Since a histogram is an approximation of the probability distribution function, the bars of a histogram approximately correspond to quantils [14]. (Since the lower thresholds are set to the 2% of the maximum values in each study, the histogram bins corresponding to voxel intensities below 2% of maximum value in each study are ignored during computation of the quantils.)

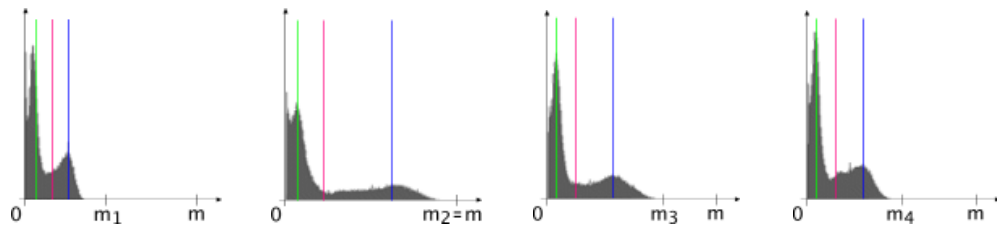


Figure 4.1: Histograms of four different studies

Experiments showed that the two peaks and the valley correspond to nearly the same quantils in all studies. In Figure 4.2, the green lines correspond to the 20% quantil, red lines to the 59% and blue lines to the 81% quantil. Thus the idea is to normalize the studies so that the intensity values corresponding to a certain quantil are the same in all studies. Figure 4.2 shows histograms of the studies after normalization according to the 81% quantil corresponding to the second peak. The new maximum values that arose from normalization are marked m'_1, \dots, m'_4 . The new overall maximum is marked m' . All histograms were created using the program Tepito.

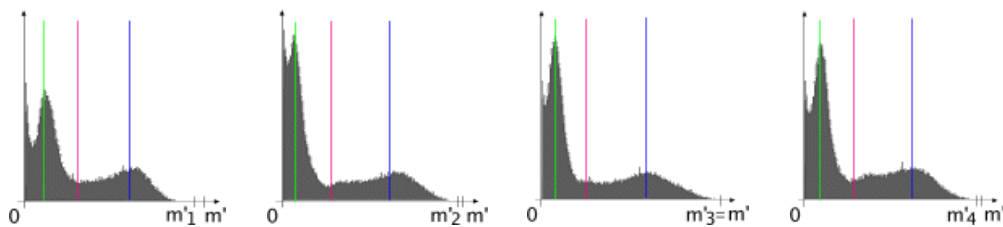


Figure 4.2: Histograms of four different studies normalized according to the 81% quantil.

More formally, having a set of N studies $M_p, p = 1, \dots, N$, the first step is to select a quantil q . A convenient way is to display histograms and choose a quantil corresponding to a significant feature. This selection is made by user. For every study M_p , the value $I(M_p, q)$ corresponding to the selected quantil q (and selected thresholds as well) is found and then the average $I(q)$ of the values $I(M_p, q)$ is computed. The studies are then normalized using the normalization factor

$$f_p = \frac{I(q)}{I(M_p, q)}$$

4.2 Region of Interest

In medical practice it is often desired to compute the normalization factor by using only a small part of the whole volume. For example it is known that the area of cerebellum in brain usually has very similar intensity distribution in most studies. Considering only this part the resulting template is often better (measured by average deviation v) than when using the whole volume.

An user-selected area that will be used instead of the whole study is called *region of interest* (ROI). In the program Tepito, ROI is implemented as a set of voxels of *one* slice. One study of the set of studies is used for selection of voxels that will belong to a ROI. This set of voxels is the same for all studies.

User first selects a study and a slice of that study. The selected slice is displayed in a new window where it is possible to draw an outline of the ROI. Voxels surrounded by the outline are the ROI voxels. The ROI can consist of more disjunct parts, thus it is possible to draw more outlines. Outlines that overlap are merged.

4.3 Template

It is not possible to decide definitely which normalization method is the best. It is even difficult to tell what does it mean to be the best. In the introduction to the chapter 4 the volume of standard deviations \overline{M}_v and the corresponding average v has been proposed as a measure of quality of the normalization

methods. The most convenient method differs for different sets of studies. Thus users should try to make several templates using different methods (and different quantils for the normalization by quantil) with/without ROI, make corresponding templates and select the one with the lowest value of v .

The program Tepito is able to display statistic information for the whole studies as well as for a selected ROI. This information can be consulted when choosing the most appropriate template.

Chapter 5

Implementation

This chapter acquaints with more technical details of the programs Brien and Tepito.

Both programs were developed using Microsoft Visual C++ 6.0 and MFC 4.2 libraries. They were tested primarily on Microsoft Windows 95/NT/2000 but they should run also on Windows 98/ME. The program Brien needs a graphic card with a minimum resolution of 800x600 pixels. The program Tepito makes do with any reasonable resolution. For smooth run at least 48MB RAM is recommended.

5.1 Data Structures, Techniques

First, important structures that are same for both programs will be described and then a few remarks will be made for individual programs.

Each study is stored in a class that contains all functions for manipulation with the data. This includes making of interpolated bitmaps from slices, three-dimensional transformation, thresholding, mapping of color tables, computing histogram and contours.

Intensities of the data matrix can be in various number formats - integers, floats, two-byte, four-byte, etc. In the case of the data from the Hospital Na Homolce, they are always two-byte unsigned integers. However, the same

set of operations must be implemented for matrices with different number formats. Only their implementation will differ. For this reason I created an abstract class that contains all these operations as pure virtual functions. A small subset of functions does not depend on number format. These functions are implemented directly in the abstract class.

Classes representing various number formats can be derived from this class. Currently, only the class for two-byte unsigned integers is implemented. Support for other number formats was not added because there was no way to test the results. However, this design allows further extensions in the future.

Brien

Transformation of a study is always computed from the original matrix. For this reason the class representing a study must keep two versions of the matrix - the original M_o and s transformed M . The original M_o is used only as a source for transformation while M is used as a target. The interpolated slices displayed on a screen are always computed using M .

Since registration of images is a lengthy operation, it was implemented as a second thread. This allows to do nearly all normal operations with the studies (browsing, changing color tables, thresholding, etc.) while the second thread silently computes on background. After each iteration, the registration thread sends new transformation parameters to the main application thread that displays them in a registration dialog window.

The appearance of the main application window is handled by a class that contains two classes with studies - one for the reference and one for the reslice study. It handles all messages and events and reacts appropriately.

In the program Brien, it is possible to save a currently viewed slice to an image file. I decided to support the *Portable Network Graphics* file format since it uses a lossless compression with a very good compression ratio. The widely used library *libpng*, in conjunction with the *zlib* compression library, was employed. The support for saving image files is not included in program Tepito because it would be useless there.

Tepito

Classes representing studies are contained in a single class that provides nearly all operations with the studies. These include mainly adding/removing studies from the set, normalization, thresholding, computing of histograms and creation of templates.

Chapter 6

User Manual

This chapter describes controlling of the two programs Brien and Tepito.

6.1 Program Brien

6.1.1 Program Installation

The program *Brien* runs on Microsoft Windows 95/98/NT4/2000. It is available as a zip file *Brien.zip*. There is no real installation of the program. The only thing to make it functional is to unzip the content of *Brien.zip* into any directory.

The program starts by running the *Brien.exe* file. First, a dialog window appears asking for two files containing a reference and a reslice study. It is possible to choose the same file for both studies by checking a checkbox. The supported file format is Interfile v3.3.

6.1.2 Main Application Window

After selecting the source files the program checks whether both studies have the same parameters `matrixSize`, `numberOfSlices`, `pixelSize` and

`sliceSeparation` (see Section 2.1 for explanation). If they differ, a dialog window appears asking whether to resample one of the studies. It is possible to skip resampling at this moment, however, only viewing of slices will be enabled in such case. Studies can be resampled later using a command from the main menu.

When the studies are loaded, the main application window opens. Appearance of the window can be seen in Figure 6.1.

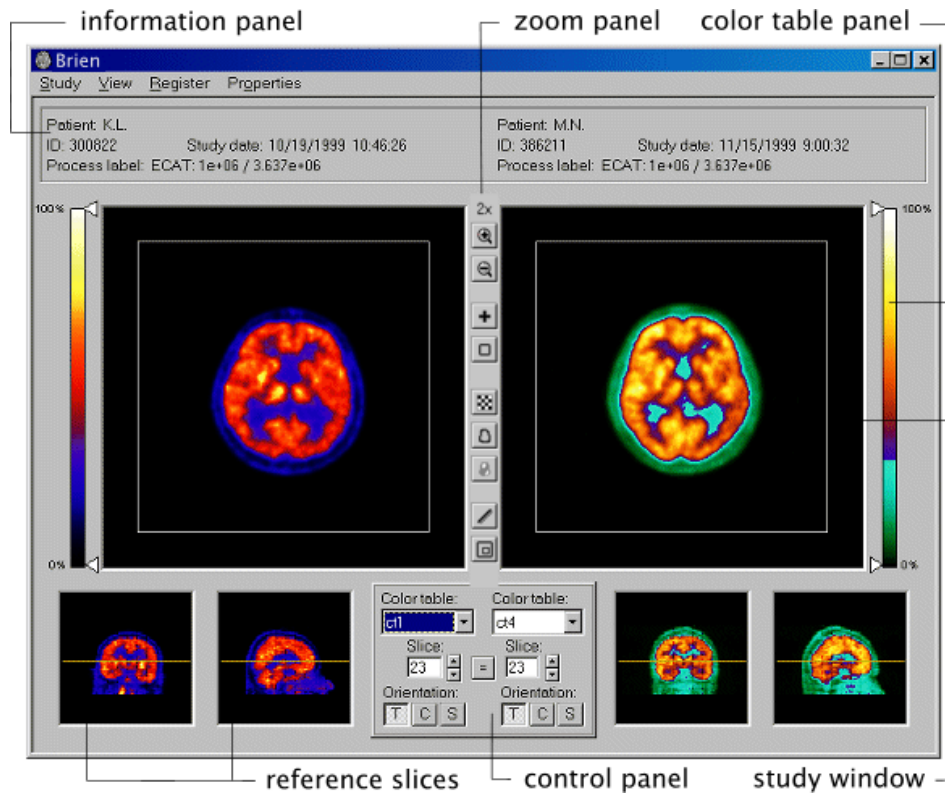


Figure 6.1: Main window

The window is divided into two parts. The left part belongs to a reference study and the right part to a reslice study. The *study window* always displays one slice of a study. Below the study window there are two *reference windows* showing complementary slices corresponding to the slice displayed in the study window (e.g. if the study window displays a transverse slice, the reference windows display coronal and sagittal slice).

There is an *information panel* on the top, showing the most important information about the loaded studies: patient name, patient ID, date of study acquisition and a short description of the study.

Zoom panel

Zoom panel (Figure 6.2) enables zooming of slices in both study windows using the *zoom in* and *zoom out* buttons. Minimum size of a displayed slice corresponds to its original size in pixels and the slice can be zoomed up to 8 times. The initial zoom is always such a (maximal) zoom when the slice still fits into the study window.

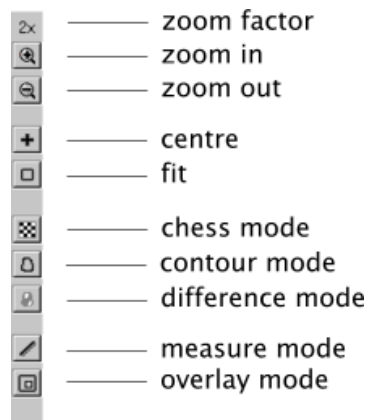



Figure 6.2: Zoom panel

When the size of a zoomed slice is larger than the study window, only a part of it is displayed and it is possible to drag the slice using left mouse button to see all parts. The *centre* button aligns the centre of the displayed slice with the centre of the study window. The *fit* button zooms the slice to fit into the study window.

The lower part of the zoom panel contains a group of buttons for switching to different modes. Pressing the *measure* button switches both study windows to the measure mode. Click with the left mouse button anywhere in a study window and drag. A line will be drawn and the length of that line in millimetres will be displayed above the study window on the right.

The remaining four buttons allow to switch to *chess mode*, *contour mode*, *difference mode* and *overlay mode*. For explanation of these modes see Section 3.5.

Control panel

Control panel (Figure 6.3 left) enables to change the orientation of displayed slices (**T**ransverse, **C**oronal, **S**agittal) and to browse through a set of slices. Pressing the  button causes binding of slice number and orientation for both studies. It means that both study windows display a slice with the same number and orientation. Changing the number of displayed slice or orientation for one study immediately changes that number and orientation for the second study. The control panel also enables to change color tables used for displaying of slices.

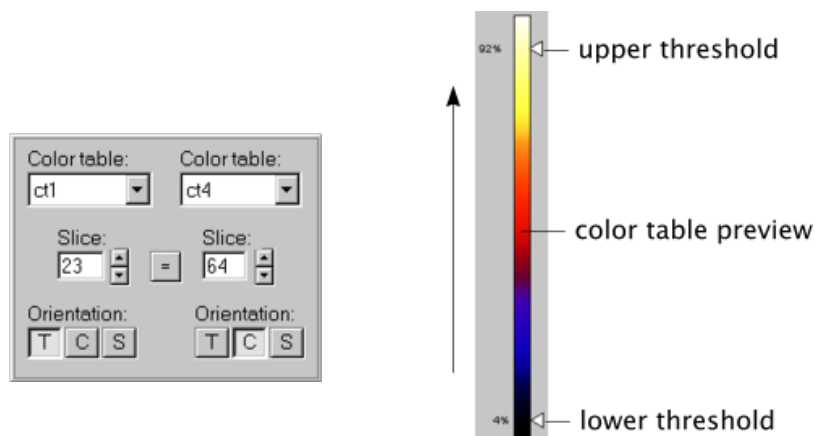


Figure 6.3: Control panel and Color table panel

Color Tables, Thresholds

The outlook of the currently used color table is drawn on the *color table panel* (Figure 6.3 right). The arrow shows a direction of increasing voxel intensity values corresponding the the colors in the color table. Dragging the *upper threshold* or *lower threshold* slider sets the respective threshold in percents of maximal intensity value in study.

6.1.3 Menu

The main application menu consists of four sub-menus: *Study*, *View*, *Register* and *Properties*. The following paragraphs describe the commands contained in these submenus.

Study submenu commands

Load new studies Brings up a dialog window asking for new studies to load. The current studies will be released.

Load new reference/reslice study Loads the respective study and releases the current.

Reference <-> Reslice Swaps the reference and the reslice study.

Save reference/reslice study as... Saves the studies. Currently only the Interfile v3.3 format is supported.

View submenu commands

Study information Brings up a dialog window displaying additional information to the information provided by the *information panel* in the main window - name of the input file, values of parameters `matrixSize`, `numberOfSlices`, `pixelSize` and `sliceSeparation`.

Histograms Shows a window with histograms of both studies corresponding to the current lower and upper thresholds.

Note: The height of the window corresponds to the height of the *third* histogram bin. The first two bins are always cut off.

Joint histogram Shows a window with a joint histogram. The darker a pixel of the joint histogram is, the higher is the corresponding histogram bin.

Note: The relation between heights of histogram bins and darkness of pixels is *logarithmic*.

Register submenu commands

Register Brings up a window that controls registering of studies. Using of this window is described in Section 6.1.4.

Resample If the studies differ in parameters `matrixSize`, `numberOfSlices`, `pixelSize` and `sliceSeparation`, this command enables to adjust them.

Properties submenu commands

Global Thresholds Brings up a dialog window displaying where it is possible to change the global thresholds. See Section 2.5 for details.

Options Changing of settings. Explanation of all options can be found in Section 6.1.5.

6.1.4 Register Window

The *Register window* is the most important part of the application. It controls the reslice study reorientation and the overall process of study registration.

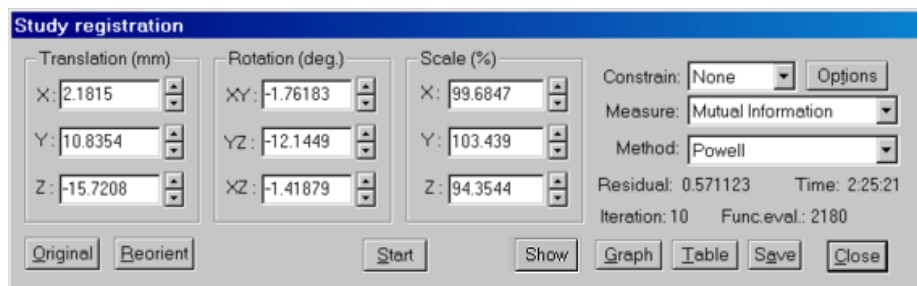


Figure 6.4: Register window

There are three groups of edit boxes in the Register window: *Translation*, *Rotation* and *Scaling*. It is possible to enter any transformation parameters and the *Reorient* button then transforms the reslice study according to the entered parameters and tells the study window and the reference windows

to update their contents to reflect the new transformation. In this manner the reslice study can be manually roughly registered prior to an automatic registration. This initial estimate can considerably help to speed-up the registration and decrease the possibility of failing. Pressing the *Original* button returns the reslice study to the initial state.

Three combo boxes on the right side of the window determine the parameters of a registration. The *Constrain* can be *None* or *Scaling*. The *Measure* determines the similarity measure that will be used for study registration. The available measures are: *Squared difference*, *Mutual information*, *Count difference* and *Shape difference*. The *Method* combo box offers three minimization methods: *Powell's Direction Set* method, *Downhill Simplex* method and *Conjugate Gradient* method.

Residual displays the value of the selected similarity measure function evaluated using currently the set transformation.

A registration is started by pressing the *Start* button. The button changes its caption to "*Stop*" and the registration begins. The registration runs as a background thread and thus it is possible to work with the main window (i.e. change color tables, browse through slices, etc.) even when the registration is in progress.

After each iteration the background thread sends information about its progress to the register window. The window then updates the displayed transformation parameters and residual value. If the *Show* button is pressed down, the study window of the reslice study repaints the currently displayed slice to reflect the new transformation.

Iteration displays number of passed iterations. *Time* is the time spent so far with the registration currently in progress. *Func. eval.* is the total number of similarity function evaluations in the current registration process.

Information about Iterations

The *Table* button shows a window with a table containing information about all passed iterations (Figure 6.5). It is possible to save this information to a text file using the *Save* command in the menu of this window or using the *Save* button in the registration window.



Number	Residual	Time	Func.eval.	Transl.: X	Transl.: Y	Transl.: Z	Rot.: XY	Rot.: YZ	Rot.: XZ	S
12	0.525349	13	2	-0.074	5.928	-8.049	2.295	2.295	2.295	11
13	0.525349	13	2	-0.074	5.928	-8.049	2.295	2.295	2.295	11
14	0.520720	13	2	-3.213	7.788	-9.234	1.260	2.165	1.785	11
15	0.507575	13	2	-0.146	7.437	-8.195	0.804	0.855	5.355	11
16	0.419744	9	1	-0.227	11.524	-12.687	-1.584	1.312	2.609	11

Figure 6.5: Information about iterations

Graph

The *Graph* button brings up a window with a graphical illustration of the current registration progress (Figure 6.6). The vertical axis stands for residual values and the horizontal axis represents elapsed time. The horizontal size of the window corresponds to a certain fixed number of seconds. If the registration process takes more time, a horizontal scroll bar appears allowing to see the parts of the graph exceeding the window size. Number of seconds determining the portion of a graph that fits into the graph window can be set in *Options* (see Section 6.1.5). One reference value is marked on the vertical axis and the part between the zero value and the reference value is divided into ten equally spaced parts marked with gray dashed lines. (Thus they correspond to percents of the reference value.) Note that graph is only intended as a supplemental tool to illustrate the registration process.

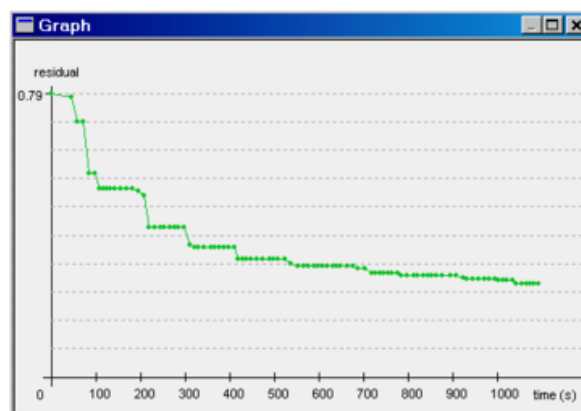


Figure 6.6: Graph

6.1.5 Options

The program Brien provides a set of options that can be customized in the *Options window* (Figure 6.7). This window is accessible from the *Properties* menu or using the *Options* button in the Register window. The Options window consists of two sheets, one dealing with registration and the other with all other options. The options are loaded at the start of the program from the file *main.conf* placed in the same directory as the program executable. On quit of the program all options are saved to this file.

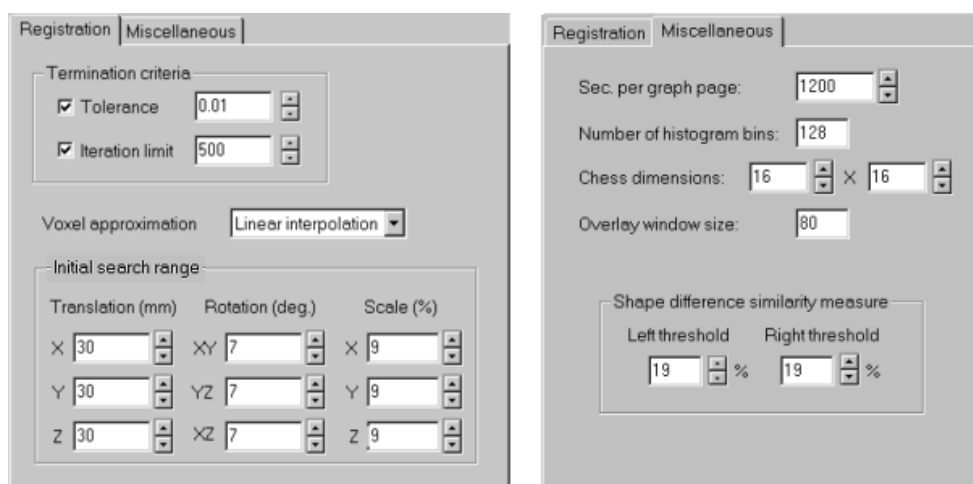


Figure 6.7: Options window

The meaning of individual options is explained in the following paragraph:

Options

Termination criteria Termination criteria for minimization algorithms. The process of registration stops on meeting either of these conditions. The meaning of the *Tolerance* criterion differs for different minimization methods. See Sections 3.4.1, 3.4.2 and 3.4.3 for details.

Voxel approximation Method for approximation of values inside voxels. The *Nearest neighbour* approximation is faster but not very accurate. The *Linear interpolation* is recommended.

Initial search range An advanced option that provides initial parameters for minimization methods and thus allows to 'tune' the algorithms. For the Powell's method and for conjugate gradient it is the initial length of the bracketing steps (see Sections 3.4.1 and 3.4.3 for details). For the downhill simplex method, the search range corresponds to the initial values λ_i (Section 3.4.2). This option is included only for testing purposes. Normally there is no need to change the default values.

Sec. per graph page Number of seconds determining the portion of a graph that fits into the graph window (Section 6.1.4).

Overlay window size Size of a square-shaped overlay window (Section 3.5.4).

Number of histogram bins Number of bins for marginal and joint histograms.

Chess dimensions number of parts in which to split a slice image in chess mode

Shape difference similarity measure Threshold values used for the shape difference measure. See Section 3.3 for details.

6.2 Program Tepito

6.2.1 Program Installation

The program *Tepito* runs on Microsoft Windows 95/98/NT4/2000. It is available as a zip file *Tepito.zip*. There is no real installation of the program. The only thing to make it functional is to unzip the content of *Tepito.zip* into any directory. The program starts by running the *Tepito.exe* file.

6.2.2 Main Application Window

The appearance of the main application window can be seen in Figure 6.8. It consists of a *control panel* and a *display window*. Nearly all controls and menu items are grayed until there is a study loaded.

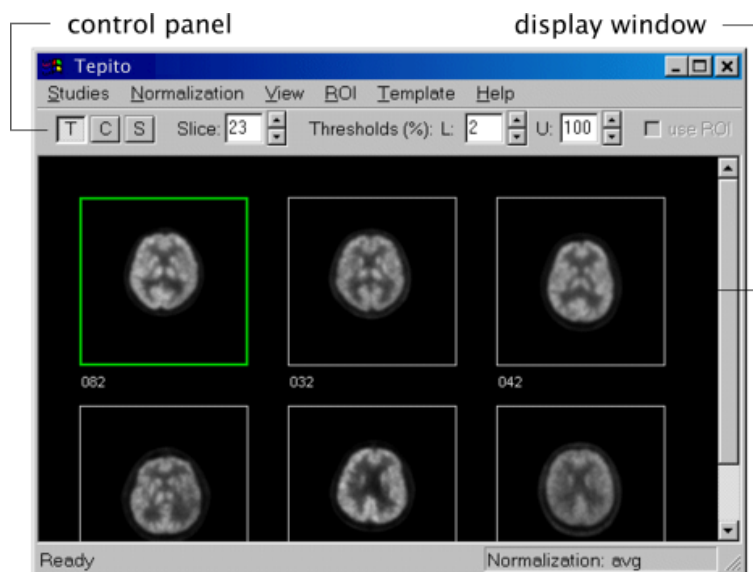


Figure 6.8: Main window

Adding and Removing Studies

New studies can be added by using the menu item *Studies*→*New studies* or eventually by clicking with the right mouse button inside the *display window* and then selecting *Add new studies* from a context menu. A *File Open* dialog appears where multiple files containing study data can be selected. The application currently supports the Interfile v3.3 file format (usually files with extension **.a** and **.i** or **.v* and **.i*).

Note: All studies must have the same `numberOfSlices`, `sliceSeparation`, `pixelSize` and `matrixSize` to be loaded by the application. See Section 2.1 for explanation of these parameters. A set of studies can be prepared in the program *Brien* for the use in *Tepito*.

The studies that have been successfully loaded are shown in the *display window*. A single slice is displayed for each study. The number of that slice is the same for all studies. Under each slice there is a name of a file that contains that study (displayed without extension). One study is highlighted by a green rectangle. It will further be referred as the *active* or *selected* study. A study can be made active by left- or right-clicking on the study.

There are two ways how to remove studies from the application - to remove a single study or the whole set. A single study can be removed by clicking on it's slice with the right mouse button and selecting *Remove single study* in the context menu. Another way is to make the study active (by left-clicking on it) and then select *Studies*→*Remove single study* in the main menu. Analogously the whole set can be removed either using the context menu or the main menu and selecting *Remove all studies*.

Studies can be added and removed at any time during the work with the program.

Working with the Control Panel

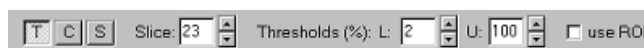


Figure 6.9: Control Panel

The studies are initially displayed in the *Transverse* direction. For the purpose of changing the view direction there are three buttons on the *control panel* (see Figure 6.9): *T* for Transverse, *C* for Coronal and *S* for Sagittal view direction.

The *Slice* edit box enables to browse through the set of slices.

Lower and upper thresholds for the set of studies can be changed in *thresholds* edit boxes. Both values are in per cents. The meaning of these values is explained in Section 2.5.

6.2.3 Normalization

The process of normalization is described in Section 4.1.

Studies are normalized using one of the following criteria: *total*, *maximum* and *quantil*. To normalize by *total* or *maximum*, simply select *Normalize*→*total*, resp. *Normalize*→*maximum*. The slices displayed in the main window will be redrawn according to the normalization.

Prior to normalization by *quantil* it is necessary to choose a quantil that will be used for normalization. This selection is employed in the *Histogram window* (see 6.2.5 for details). Once a selection is made, to normalize studies by quantil select *Normalize*→*quantil*.

Note that normalization is affected by the *lower* and *upper threshold* (see Section 4.1.1). These can be adjusted in the *Control panel* or in the *Histogram window*.

Selecting *Normalization*→*None* returns all studies to the original state (no normalization).

The current normalization type can be seen in the status bar of the main window. Normalizations are always performed on the original studies. Thus a sequence of different normalizations returns the same output as if only the last normalization was performed.

Saving studies

It is possible to save studies at any time during work with the program. Studies will always be saved in their current state (e.g. normalized by total counts, in their original state etc.). A single study can be saved by selecting *Studies*→*Save selected study*. Using *Studies*→*Save all studies* the whole set of studies can be saved.

6.2.4 ROI

By default all normalizations are computed using the whole volume of each study. Another possibility supported by the application is using of ROI (**R**egions **O**f **I**nterest) (see Section 4.2). The index of the slice that is used in conjunction with a ROI is always the current slice selected in the *Slice* edit box in the *Control panel*.

A ROI can be selected in the *ROI edit window*. This window is accessible from the main menu (*ROI*→*Edit*) or by right-clicking on a study in the *display window* and selecting *Edit ROI* from the context menu. The *ROI edit window* displays the current slice of the active study. It can be zoomed by selecting

View→*Zoom in N times*.

There are two ways how to select a ROI : to draw it or to load it from a file. Drawing is performed by pressing the left mouse button on the slice image and dragging. Releasing the left mouse button finishes drawing of the area and closes the curve by connecting the last point with the starting point. The new area is then displayed on all slices in the *main window*. The ROI can consist of many disjoint areas and the areas need not be convex. If a newly drawn area overlaps with any of existing areas, these areas are merged.

A new ROI can be saved using *ROI*→*Save* and a saved one can be loaded by *ROI*→*Load*.

After an area is drawn it is filled with a single color to provide a way to check the correct behaviour of the algorithm that computes the selected area. *View*→*Don't show inner areas* switches filling of areas off.

A ROI can be deleted by *ROI*→*Clear*. Closing of the *ROI edit window* does not delete the ROI and it can be adjusted at any time by opening the *ROI edit window* again. The checkbox *use ROI* on the *Control panel* in the *Main window* determines whether the created ROI will be used for normalization and template creation.

By default, newly created ROI is drawn onto all studies in the *Main window*. This behaviour can be suppressed by selecting *ROI*→*Don't show* in the main menu.

6.2.5 Histogram Window

The *Histogram window* (Figure 6.10) can be opened by selecting *View*→*Histograms*. The window displays a histogram for every study and allows to select a quantil that will be used for normalization of studies. A quantil can be selected in the *Quantil* edit box. The positions of a currently chosen quantil in histograms are highlighted by red lines.

Thresholds can also be adjusted in this window. It is usually necessary to set the lower threshold to a value above zero to eliminate the influence of a huge number of voxels with an intensity value near zero. On start of the program, the initial value of the lower threshold is set to 2%.

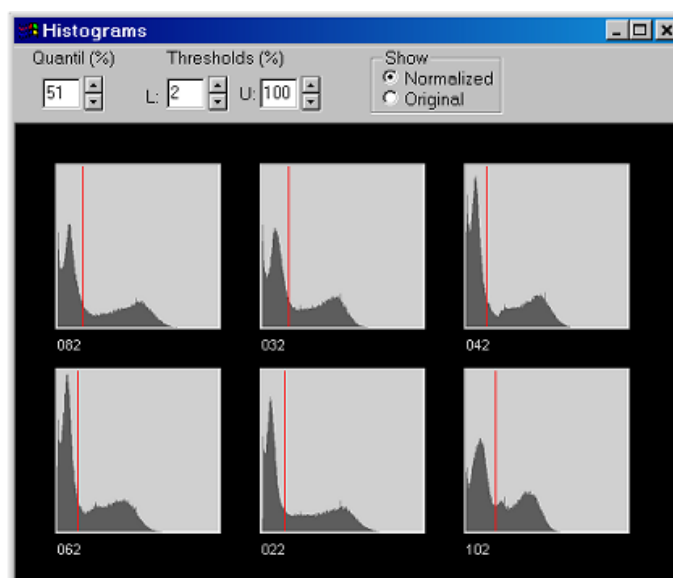


Figure 6.10: Histogram window

If the *use ROI* on the *Control panel* of the *Main window* is checked the histograms are computed only by using the current slice and the area determined by ROI.

Switching among the options *Show normalized* and *Show original* allows to compare differences between histograms of original and normalized studies.

6.2.6 Statistics

Study name	Avg value	Max value	Norm factor	Norm Avg	Norm Max	Avg (ROI)	Max (ROI)	Norm Avg (ROI)	Norm Max (ROI)
062	1037.01	21296	0.65204	676.17	13885	10537.64	17491	6670.95	11404
032	403.53	8166	1.67563	676.17	13683	3946.73	6544	6613.26	10965
042	978.26	16698	0.69120	676.17	11541	9506.87	14705	6571.12	10164
062	579.53	10154	1.16677	676.17	11847	4779.13	8216	5576.14	9586
022	508.73	12175	1.32913	676.17	16182	4465.42	8359	5935.15	11110
102	549.97	8966	1.22948	676.17	11023	4496.63	6874	5530.96	8451

Figure 6.11: Statistics window

The statistics window (Figure 6.11) brings a summary of all important characteristics for each study. There are nine values for every study there:

- *Avg. value*: Average intensity value of a study.
- *Max. value*: Maximum intensity value of a study.
- *Norm. factor*: Normalization factor. This value is equal to 1.0 for all studies if no normalization has been applied on studies yet.
- *Norm. avg*: Average intensity value of a study after multiplication by the normalization factor.
- *Norm. max*: Maximum intensity value of a study after multiplication by the normalization factor.
- *Avg. (ROI), Max. (ROI), Norm. Avg. (ROI), Norm. Max. (ROI)*: Values analogical to the previous except that they are computed using the ROI area only.

6.2.7 Template

Selecting *Template*→*Make* from the main menu creates a new template using current normalization. A new *Template window* is opened with two studies: the template and the standard deviations (Figure 6.12). It is possible to browse through slices and to change thresholds and view directions (Transverse, Coronal, Sagittal) using the *control panel* in exactly the same manner as in the *main window*.

A template can be created at any time during the work with the application. The principles of template creation are explained in Section 4.3. It is convenient to create more templates using different normalization methods and thresholds and compare them to select the most appropriate one.

Save Template

The template can be saved in the Interfile format by selecting *File*→*Save*. A *Save* dialog appears asking for the name of the file to save the new template in. The application always stores two files: one containing the template and having the chosen name and another file containing the volume of standard deviations. The latter file will be stored using the name *chosen_nameDev*.

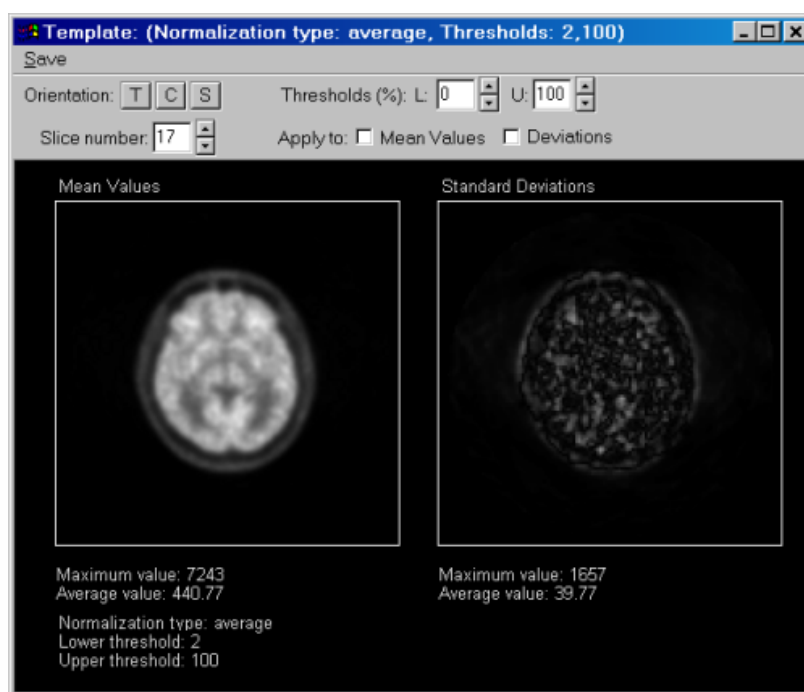


Figure 6.12: Template window

Chapter 7

Conclusions

In this work several methods for 3D medical data visualization and registration were examined and implemented.

A registration of two data volumes is accomplished by using an iterative method that searches the space of linear transformations for the one maximizing some measure of similarity between the two volumes. Three iterative numerical methods were chosen for this purpose: Powell's method, downhill simplex method and conjugate gradient method. Three commonly used similarity measures were employed: count difference, squared difference and shape difference. In addition, quite new and still insufficiently explored similarity measure based on mutual information was introduced as an approach to express an alignment of brain images.

Quality and suitability of implemented registration methods were examined by two distinct types of measurements. The first one consisted in registration of two identical, initially misaligned studies while the second measurement dealt with registration of two completely diverse studies from different patients. Tests were carried out for all implemented minimization methods in conjunction with two similarity measures: mutual information and squared difference.

Results of the first measurement evince that in the case of very similar studies all three iterative methods are able to bring the studies into a spatial alignment much faster when the squared difference is used. Mutual informa-

tion turned out to be unsuitable in this case. The measurement also revealed that conjugate gradient method is not very accurate, whether used in conjunction with squared difference or with mutual information. The Powell's method with the squared difference measure proved to be the most effective combination for this type of registration.

The conjugate gradient method remained the least suitable one even in the second type of measurement. In contrast to the first type, each method gave similar results for squared difference measure as well as for mutual information. No significant distinction in performance between the downhill simplex and the Powell's method was found as well. Although the conjugate gradient method is not very accurate, experience gained during the development of the software shows that this method is more robust than the downhill simplex or the Powell's method.

Special attention was paid to the visual verification of registration results. Several tools were implemented that enable a sort of evaluation of alignment quality. Chess tool splits both studies into squared fields and shows the combined images. If two brains are registered properly one shall get a continuous contour for both. Contour tool shows contours on a selected intensity level for both images. The better the alignment of contours is, the better is the registration. Joint histogram (used for calculation of mutual information) can also indicate the quality of registration.

Besides the registration algorithms, a convenient way of visualization of studies was suggested. The employed techniques include fast zooming of slices, mapping of color tables and thresholding.

In nuclear medicine it is often necessary to compare, how much a certain study differs from a standard one. A prototype of a standard brain study is usually created as a mean of several "normal" studies. Before a set of studies is averaged, all studies must be normalized, i.e. their intensity ranges must be equalized in some way. Besides several common normalization techniques, a normalization method based on quantils was suggested.

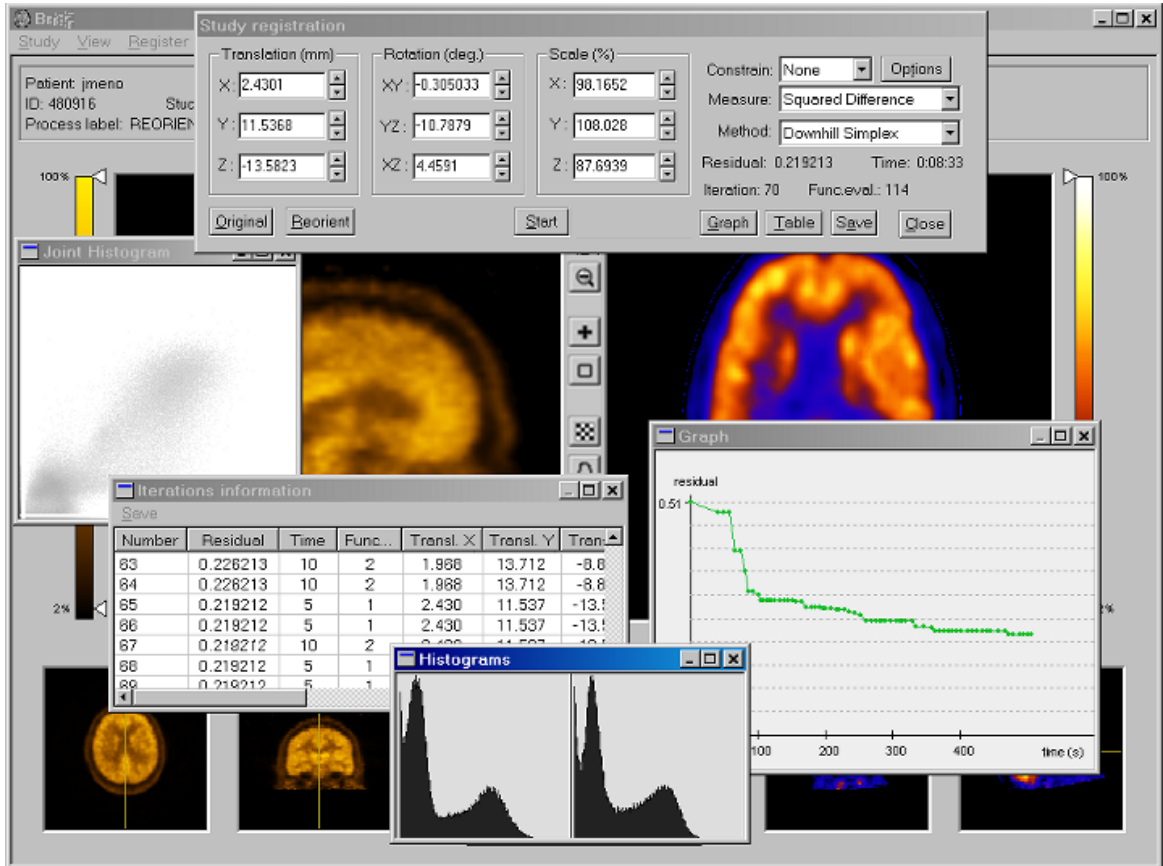
All methods presented in this thesis were implemented in two programs running on the Microsoft Windows operation system. The programs were tested and positively accepted by the medical staff in Hospital Na Homolce.

Bibliography

- [1] J. Anděl. *Statistické metody*. MatfyzPress, Praha, 1993
- [2] J.L.R. Andersson, L. Thurfjell. *A multivariate approach to registration of dissimilar tomographic images*. Eur J Nucl Med, Volume 26, pages 718-733, 1999
- [3] L.Barnden, R.Kwiatek, Y. Lau, B. Hutton, L. Thurfjell, K. Pile, Ch. Rowe. *Validation of fully automatic brain SPET to MR co-registration*. Eur J Nucl Med, Volume 27, pages 147-154, 2000
- [4] Committee on the Mathematics and Physics of Emerging Dynamic Biomedical Imaging. *Mathematics and Physics of Emerging Biomedical Imaging*. National Academy Press, Washington, 1996
- [5] A.S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995
- [6] HERMES Tomographic Applications - Multimodality (original user manual)
- [7] D. Hill, D.J. Hawkes. *Intensity-based cost functions for across modality applications*. Chapter for Handbook of Medical Imaging ed. Bankman (in press), 1999
- [8] D.L.G. Hill, C. Studholme, D.J. Hawkes. *Voxel Similarity Measures for Automated Image Registration*. Proc. Visualisation in Biomedical Computing, 1994. Proc. SPIE 2359:205-216, 1994
- [9] R. Kulhavý. *Lecture Notes in Control and Information Sciences; 216*. Springer-Verlag London Ltd., 1996

- [10] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, P. Suetens. *Multimodality image registration by maximization of mutual information*. IEEE Trans Med Imaging, Volume 16, pages 187-198, 1997
- [11] J.A. Nelder, R. Mead. *A simplex method for function minimization*. Computer Journal, Volume 7, pages 308-313, 1965
- [12] M.D.J. Powell. *An efficient method for finding the minimum of a function of several variables without calculating derivatives*. Computer Journal, Volume 7, pages 155 -162, 1964
- [13] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling. *Numerical recipes in C*. 2nd edition, Cambridge University Press, 1992
- [14] K. Rektorys et al. *Přehled užití matematiky II*. Prometheus, Praha, 6th edition, 1995
- [15] J.H. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. unpublished draft, 1994. URL: <http://www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.ps>
- [16] C. Studholme. *Measures of 3D medical image alignment*. PhD thesis, United Medical and Dental Schools of Guy's and St Thomas's Hospitals, University of London, 1997
- [17] C. Studholme, D.L.G. Hill, D.J. Hawkes. *Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures*. Med Phys, Volume 24, pages 25-35, 1997
- [18] L. Thurfjell, Y.H. Lau, J.L.R. Andersson, B.F. Hutton. *Improved efficiency for MRI-SPECT registration based on mutual information*. Eur J Nucl Med, Volume 27, pages 847-856, 2000
- [19] A. Todd-Pokropek, T.D. Craddock, F. Deconinck. *A File Format For The Exchange Of Nuclear Medicine Image Data: A Specification Of Interfile Version 3.3*. Nuclear Medicine Communications. Volume 13, pages 673-699, 1992.
- [20] I. Vajda. *Teória informácie a štatistického rozhodovania*. Alfa Bratislava, 1982

- [21] W.M. Wells III, P. Viola, H. Atsumi, S. Nakajima, R. Kikinis. *Multi-modal volume registration by maximization of mutual information*. Med Image Anal, Volume 1, pages 35-51, 1996
- [22] J. Žára, B. Beneš, P. Felkel. *Moderní počítačová grafika*. 1st edition, Computer Press, Brno, 1998



General view of the program Brien