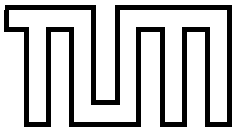
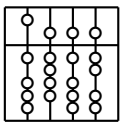


Technische Universität München
Fakultät für Informatik



IDP - Interdisciplinary Project

Image Registration and Merging for a new Optical Tomography

Moritz Blume

Advisor:	Prof. Dr. Josef Friedrich
Supervision:	Prof. Dr. Nassir Navab, Prof. Dr. Hans-Ulrich Dodt, Dipl.-Inf. Wolfgang Wein
Date:	October 15, 2005

Contents

1. Introduction

Tomographic procedures are a common way to retrieve three dimensional volumetric information of data. Well known and widely used in medicine are CT, PET and MRT. However, each of those techniques has its specific advantages and drawbacks.

A relatively novel imaging technique is Optical Tomography (OT). Instead of X-rays, positrons or magnetic fields light of the visible or close to the visible spectrum is used to retrieve information from the tissue. Especially with near-infrared (NIR) light considerable efforts have been made in the last years ([3], [1], [2], [11] and [7]).

First experiments in pilot studies for monitoring blood oxygenation, hemorrhage detection, functional imaging of brain activities, Alzheimer diagnosis, early diagnosis of rheumatic disease in joints and breast cancer detection have been made. However, the images retrieved don't have much in common with the known CT, PET or MRT images. Reconstruction algorithms still suffer of the high complexity of photon scattering in the tissue.

1.1. The transparent brain

The optical imaging technique discussed in this work doesn't have much in common with those approaches. Instead of trying to overcome the numerical problems for the simulation, Prof. Dr. Dodt reduced the scattering in the tissue. The outcome are tomographic images with an outstanding and not yet seen quality and resolution. Since the so received volumetric datasets are extremely huge, it's also quite challenging to handle them on ordinary computers.

Of course, the whole visualization is taking place in vitro. Since the resolution of the retrieved images is about five micrometers, promising prospects for this new technique is the study of neural networks.

Most preparations used during the time of this work were made of mouse embryos and brains. Also operatively extracted mouse hippocampi were imaged.

1.2. Outline

Chapter 2 will give an introduction to the new method of Prof. Dodt. It will also discuss where the drawbacks are and how they can be reduced with image registration and merging. Chapter 3 shows in detail how the varying quality inside the tissue can be modeled. Chapter 4 gives a brief introduction to image registration. Then, an adoption of the normalized cross-correlation will be introduced which includes the

previously worked out quality measure. Finally, the adopted method will be compared to the original one. Chapter 5 shows how the images can be merged after their registration. Chapter 6 gives a perspective on what still needs to be done. Finally, the appendix is meant to be a brief instruction manual on how to use the provided software.

2. Optical Tomography

The method invented by Prof. Dr. Dodt is basically a combination of two already well known techniques:

- Making anatomical preparations transparent

  The lightening of anatomical preparations is an old method, developed by the German Werner Spalteholz (*1861 - †1940). In the world of anatomy Werner Spalteholz is well known for his ”Handatlas der Anatomie des Menschen” (Hand Atlas of Human Anatomy), according to [4] a ”classic milestone in the history of anatomical illustration” and ”one of the most elegantly illustrated anatomical atlases of all time”.

  Spalteholz worked in the ”Leipziger Institut für Anatomie” (Institute for anatomy of Leipzig) under the leadership of Wilhelm His. His took care of both collecting and developing anatomical preparations. Unfortunately almost the whole collection was destroyed in 1945 by bombings. The best conserved preparation is a brightened cranium with injected inner ear. It can still be seen in the recently restored gallery in Leipzig.

  The method is described in [6]. In rough words, it consists of exchanging the water by a mixture of benzoin and benzyl alcohols. The goal is to achieve a unique refraction index in both the biological tissue and the surrounding liquid. Scattering is so reduced to a minimum and the tissue becomes transparent for light in the visible spectrum.

- Fluorescence Microscopy

  Fluorescence microscopy is based on the normal light microscopy. As the name suggests, fluorescence microscopy is an imaging technique which takes advantage of the physical phenomena of fluorescence. The preparations can be marked with fluorescing probes and thus interesting parts of it can be made visible.

  Most biological tissues already have a certain amount of autofluorescence which makes it possible that not only specially marked parts can be visualized but also the rest of the surrounding tissue.

2.1. Imaging procedure

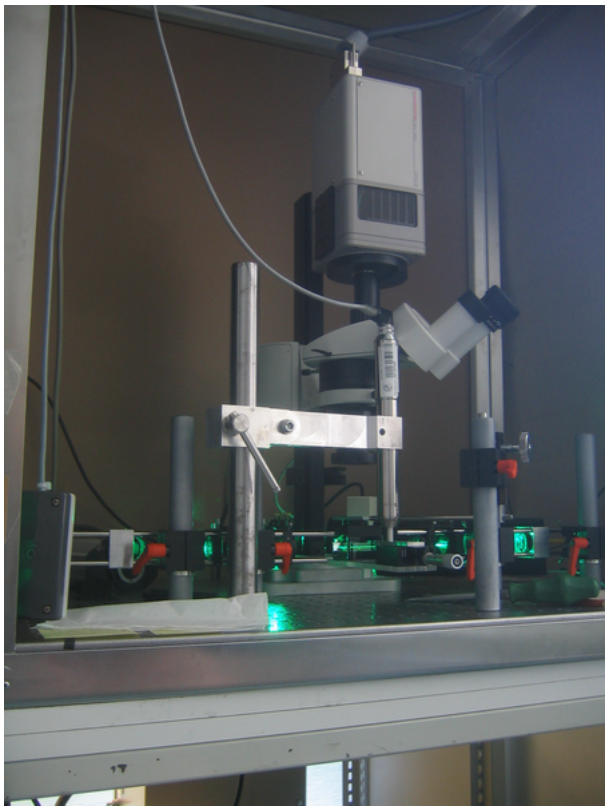


Figure 2.1.: Photo of construction

Figures 2.1 and 2.2 show how the imaging works: A laser plane shines from the side through the transparent preparation. The fluorescing parts are emitting light and can be imaged by the microscope and a camera on the top.

Moving the object with a stepping motor returns slices of equal distance. At each step two images are taken: one with the laser shining from the left side, and the other with the laser shining only from the right side.

The laser is an argon laser with 488nm wavelength. Having a power of about one Watt, after transportation in fiber optics and splitting the beam only about 10% will be used for the final illumination of the preparation.

Since the wavelength of the fluorescence light is higher than the one of the laser, a Zeiss filter with a range of $505\mathrm{nm}$ to $555\mathrm{nm}$ makes sure that only light coming from the fluorescence is getting to the imaging device.

Thus, in one imaging process the result is two different sets of 3D volumetric data. Knowing the slice distance and the pixel sizes of the image taken by the camera, one can reconstruct the 3D volumetric information. Usual values are a slice distance of

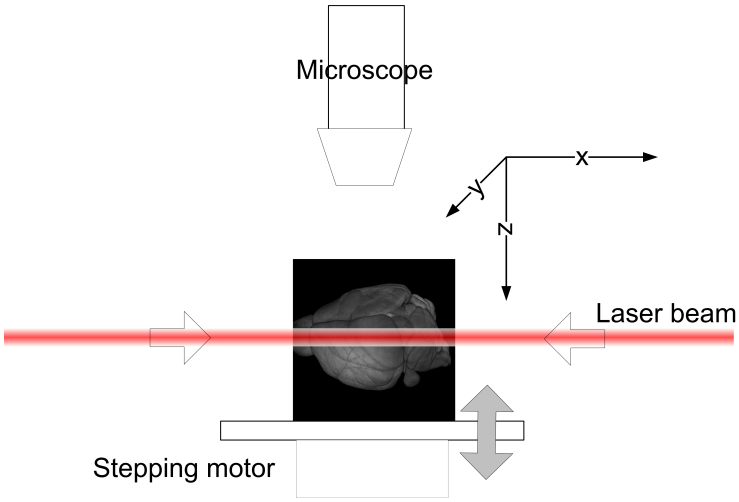


Figure 2.2.: Schematic drawing of construction

$12.5\mu m$ and pixel sizes of about $10\mu m$. The pixel size depends on the lenses used in the microscope. For an object with height of about $6.25cm$ thus 500 slices are taken. Depending on the camera the resolution is 8-bit or 12-bit. Assuming a 12-bit camera, since one image is of size $1392x1024$, the size of the whole volume (assuming about 500 slices) is more than one gigabyte! This fact leads to further problems in both registration and merging considering time and memory.

2.2. Drawbacks

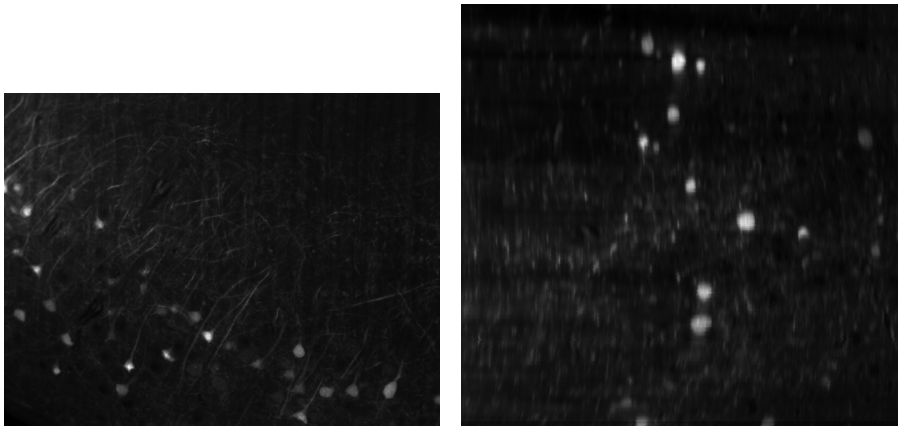
We will now discuss several issues respect to the quality of the volumetric datasets that we got. Three main effects will be discussed in more detail in this work:

- Convolution
  Figure 2.3 compares a x-y-slice with a y-z-slice. As one sees immediately, the y-z-slice is blurred quite a lot in the z-direction.

  Figure 2.4 shows how this effect can be explained: Since the laser beam has a certain width, the actual image that the camera can take is in fact the sum of several intensities in z-direction. Let $I(x, y, z)$ be the real image intensity at an infinitely small point at x, y and z. Then the intensity imaged by the camera is:

$$\int_{x-\Delta x}^{x+\Delta x} \int_{y-\Delta y}^{y+\Delta y} \int_{z-\Delta z}^{z+\Delta z} I(x, y, z) \cdot g(x, y, z) dz dy dx$$

  Since Δz is big compared to Δx and Δy, this leads to the blurring in z-direction.



(a) x-y-slice: Even axons are sharply displayed.

(b) The y-z-slice is a virtual slice since it had to be computed out of the slice stack: A quite strong blurring in z-direction is visible. One pixel in z-direction has an approximate size of three micrometers.

Figure 2.3.: Comparison of a x-y- and y-z-slice of a mouse hippocampus

The mathematical term given is nothing else than a convolution of the image in z-direction, the kernel g is the laser profile in z-direction.

- Scattering of the laser beam

  Figure 2.5 shows a x-y-slice with z-coordinate relatively close to zero.

  One easily observes that the quality decreases from the left to the right. The higher the penetration depth of the laser in the tissue is, the more the initially thin plane is spreading due to scattering of light in the tissue. This leads to a bigger convolution effect in z-direction, and obviously this effect also leads to blurring in x- and y-direction.

- Scattering of the fluorescence light.

  Figure 2.6 shows a x-y-slice with z-coordinate close to the bottom of the preparation.

  Two things can be noticed:

  - The image is generally blurred more than the one in figure 2.5. This is also due to scattering in the tissue (but shouldn't be confused with the previous convolution caused blurring): Since the fluorescence light has to pass the rest of the tissue until it can be imaged, it suffers from scattering and this leads to blurring.
  - Again, as seen in figure 2.5, the blurring increases from the left to the right. This is also due to the scattering of the laser beam.

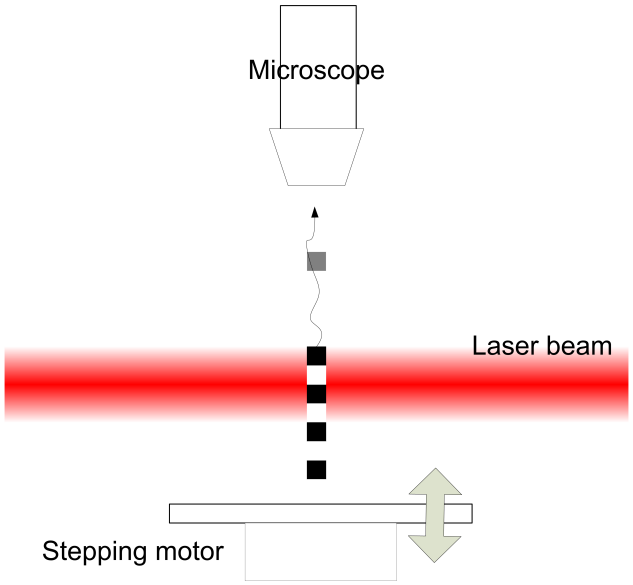


Figure 2.4.: Since the laser beam is not infinitely thin, slices are blurred in z-direction.

Further loss of quality is due to spherical abberation and last but not least the fact that the lenses used are not corrected exactly for the mixture of benzoin and benzyl alcohols used for the preparation. We suppose that those effects lead to minor quality loss and thus won't be described in detail in that work.

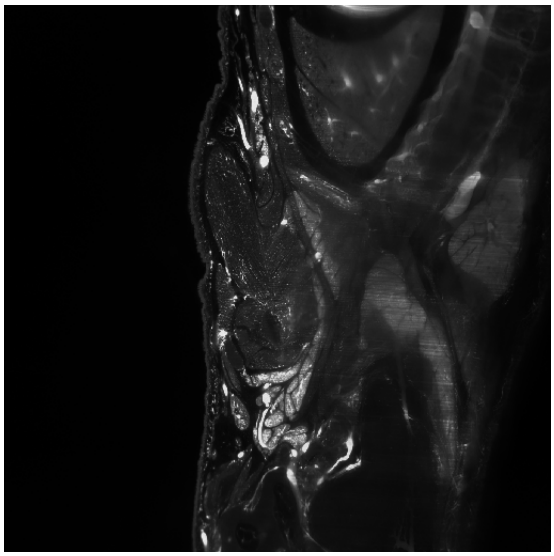


Figure 2.5.: The slice is situated relatively close to the microscope. The laser plane comes from the left. The blurring is increasing from the left to the right.

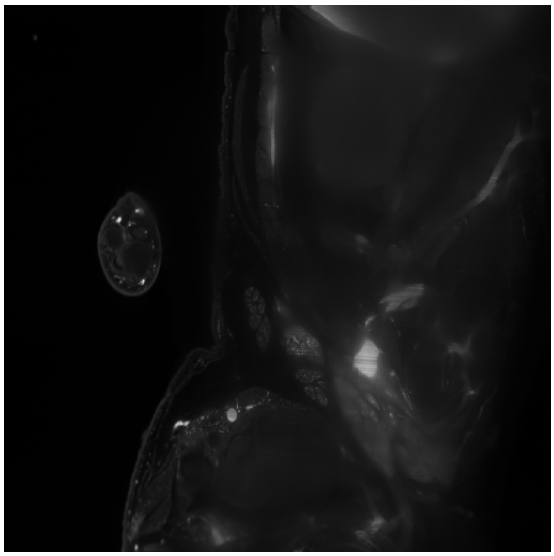


Figure 2.6.: The slice is a bottom slice and thus relatively far away from the microscope. Since the light has to pass the tissue before coming to the imaging device, the image in general is blurred quite a lot. Again, the laser plane comes from the left, which leads to increasing blurring from the left to the right.

2.3. Overview of possible solutions

This section will give a brief overview of how computer science can help to solve the mentioned problems. Two different methods will be considered in this work. Both of them are based on a very basic idea:

The quality of the received volumetric datasets varies locally and depends on the orientation of the preparation.

2.3.1. Scattering of the fluorescence light

The idea to overcome this problem is to turn the preparation about approximately $180°$ and record another dataset. The most blurred parts on the bottom of the first image would then be the sharpest and vice versa. Then, one would have to put the two together. Figure 2.7 illustrates the idea.

Figure 2.7.: Turn the preparation about $180°$ and put them together. The result should contain best parts of both

Even if this sounds very easy and straight forward, there are several problems:

- It's not possible to turn the preparation about exactly $180°$.
- How do we know which part of the data we should prefer in the merging process?

The first problem can be solved by a technique called *image registration*. It will be explained in detail in the next chapters. In order to solve the second problem, we have to find a model for the recorded quality in a volume. Once we have that model and the exact orientation the *merging* can be done.

2.3.2. Scattering of laser beam

In a very similar manner we can again take two volumes, one taken with the illumination from the negative x-side and the other one from the positive x-side. In theory

this is easier since the preparation itself needn't be turned around because the laser can be controlled to only shine from one side.

In practice, the two laser planes from the different sides are not exactly the same and thus also in this case image registration is necessary. Then again, we have to find a model for a quality measure before merging can take place.

2.3.3. Convolution

Even if we look at the good parts of a volume, we always have the problem of blurring in z-direction due to convolution. To overcome this problem very soon the idea raised that if we turned the preparation about $90°$ about the y-axis, we would have data which wouldn't be blurred in the formerly blurred direction.

Again, image registration would be needed in order to orientate the volumes but then a simple quality model is not enough. How exactly one can deal with the problem will be described in chapter 5.

3. A Quality function

As seen in the previous chapter, there are several sources of quality loss. It was shown briefly how image registration and merging can reduce the discussed drawbacks. However, when we want to merge two datasets, we have to generate one volume out of two input volumes. For each voxel in the output volume we have two corresponding voxels of the input volumes. How do we know which one we should take? Similarly, when we do registration, we might want to take into account local quality values.

So, it seems to be useful to have a function

$$Q : \mathbb{R}^3 \rightarrow [0; 1]$$

which is taking a image position in a 3D volume and returning its quality value. A value of 0 stands for worst quality and 1 for best quality.

This function Q can describe how the quality changes locally, e. g. the scattering of the fluorescence light and the scattering of the laser beam that leads to increasing convolution in the z-direction. However, Q can only model *local* quality measures. Since also without scattering of the laser beam we have a convolution due to its width, this would be a *global* effect that does not depend on the position in the volume and can not be described by Q.

So, Q can take into account the scattering of the laser beam and the scattering of the fluorescence light. In the case of the fluorescence light a Monte Carlo Simulation will be used to find out about the quality values. This will be shown in the next sections. For the scattering of the laser beam and the resulting higher convolution, only a simple model will be used.

3.1. Scattering of fluorescence light

If we assume that scattering is only taking place in the tissue and not in the liquid surrounding the tissue, then the most interesting parameter is the distance that light is propagating inside the tissue. This distance will be referred as the *depth*. Even in one slice, pixels will have different depths, as figure 3.1 illustrates.

How is scattering affecting the image? Figure 3.2 illustrates an image point P and three light rays representing the infinite number of rays that leave P in all directions. The optics of the microscope will project those rays to one point in order to get a sharp image.

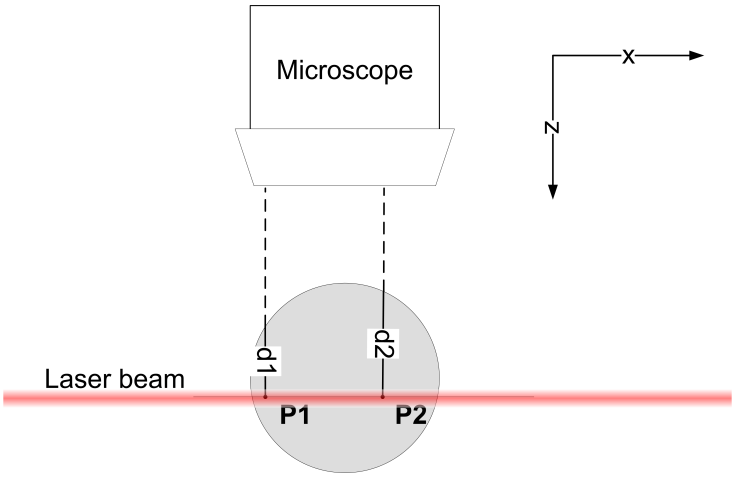


Figure 3.1.: P1 has a lower depth than P2

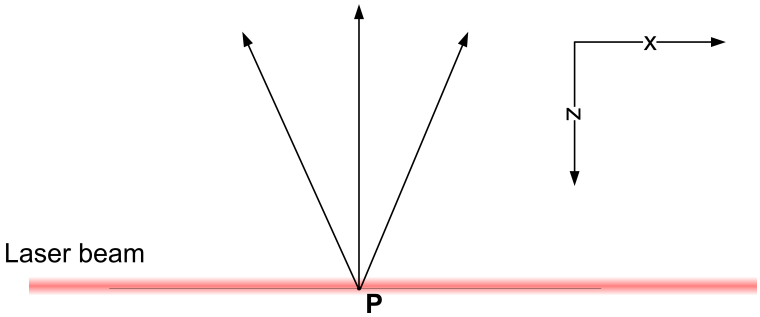


Figure 3.2.: A point and its rays

Due to scattering, the rays will change their directions in some manner. If a ray changes its direction, it seems that it comes from a different point source and thus the image of one point results in the image of several virtual points. This leads to blurring and is illustrated in figure 3.3.

If the virtual points lie outside of the image plane then they won't be considered by the optics of the microscope. Thus, the effect of blurring is limited: if the angle as result of scattering is too high, the optics of the microscope will ignore it.

The higher the depth of a voxel is, the more it will be scattered. Since the depth varies also in between one slice, there are parts which are blurred more and others that are blurred less. Figure 3.4 shows a slice from a mouse brain. The pixels at the borders had a much lower depth (like in the circle of the previous figures), and thus there is less scattering and finally less blurring.

Basically there are two methods to describe the scattering in tissue: Analytic ones adopting the diffusion theory and statistical Monte Carlo methods. In this thesis a Monte Carlo method was used for simulation.

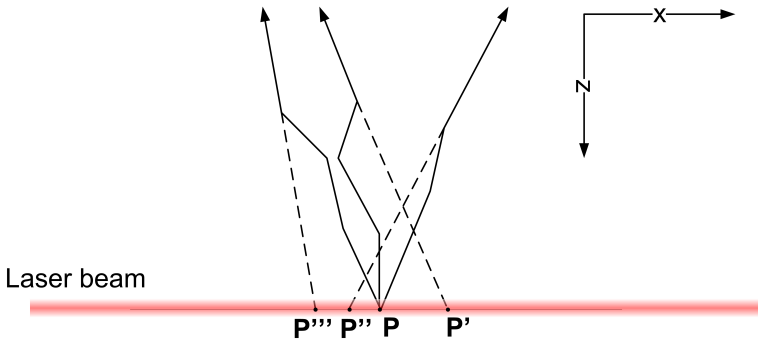


Figure 3.3.: How scattering leads to blurring

3.1.1. A Monte Carlo method of light propagation in tissue

The method described here is mostly inspired by [8]. Since this thesis should be self-contained, for terminology reasons and last but not least because of some differences the method will be described in detail here.

For most accuracy, the simulation has to be performed for every voxel. Considering that the simulation time for just one voxel would need several minutes, this would lead to an tremendous amount of calculation time.

We can reduce the computational time if we treat voxels with same depth equally. The simulation itself doesn't have to know about the geometry of the object. Only the depth value decides on how the probability distribution for one voxel will look like. The voxel is considered to be in an object with infinite width, and the height is its depth value.

As figure 3.5 illustrates this can lead to problems, since it is possible that the ray is already outside of the tissue (and thus won't be scattered anymore) but in the approximation it is still inside. On the other hand, as shown with the right ray, it can happen that the ray is still inside the tissue but in the approximation it already left. Thus, somehow these drawbacks are compensating each other.

So, the effect that blurring at borders is lower than blurring in the middle of the object will be simulated correctly because depth values of voxels close to borders are lower (or almost equal to zero!) than the ones of voxels in the middle.

Figure 3.6 shows the algorithm.

Instead of a single photon, photon packets will be considered. This will drastically reduce the computational amount of time for the simulation. A photon packet has a certain weight at the beginning and will loose parts of its weight with each absorption event.

For simplicity, the simulation assumes that there are so called "centers", where both scattering and absorption occurs. These scattering and absorption centers are uniformly distributed in the whole tissue.

Light is propagating the tissue without being scattered for a certain distance, the

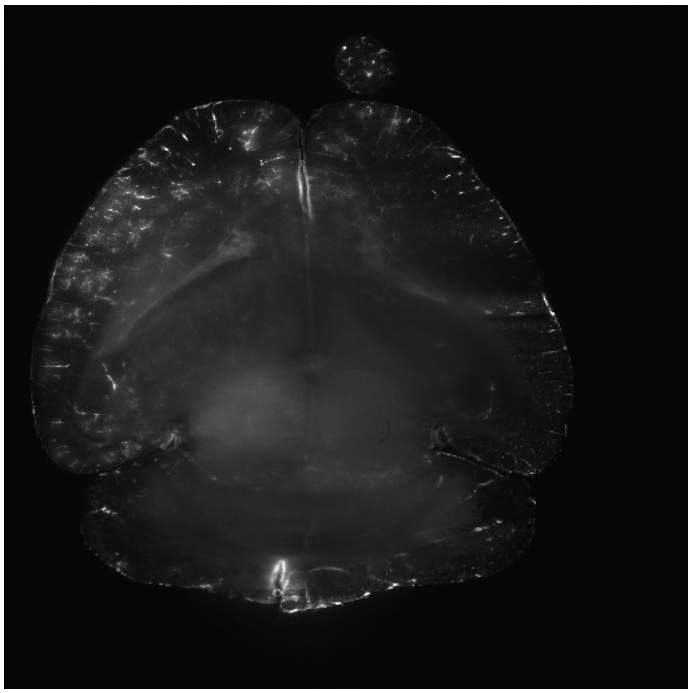


Figure 3.4.: Since the depth of the border pixels is lower they are much sharper than those in the middle of the image

free pathlength. Then, it will be scattered and absorbed. The weight decreases and the direction of propagation changes.

The simulation ends if either the packet has left the tissue or if the weight is under a certain threshold.

3.1.1.1. Initialization

The actual position of the photon packet is stored in $\mathbf{X}$. $\mathbf{X}$ will be initialized with the entry point $\mathbf{P} = (0, 0, -d)^T$. d is the depth of the voxel.

The weight of the photon packet is initialized by $w = 1$.

The direction vector $\mathbf{D} = (0, 0, 1)^T$ is a norm vector and points towards the propagation of the photon packet.

3.1.1.2. Free path length

The goal of this section is to find a probability distribution for the free path length.

First, we will look for a function $N(s)$ which returns the number of photons that entered the tissue by length s and didn't hit a center. We are interested in how many photons will pass the tissue without hitting a center at distance $s + ds$. This will be $N + dN$ photons, with dN being a negative number or zero since some photons or

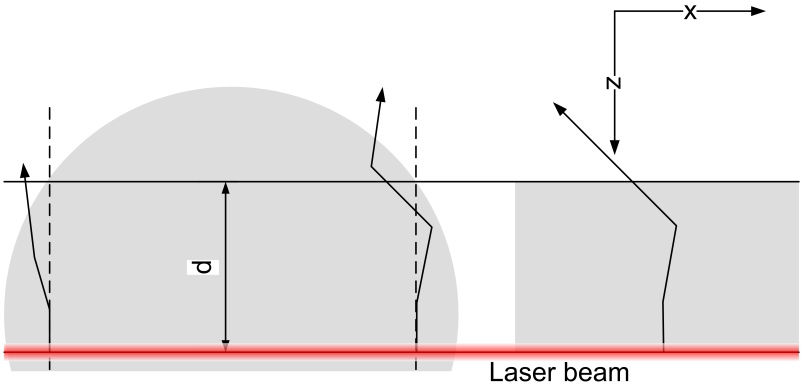


Figure 3.5.: Rays with same depth d will be treated by only one simulation

none will hit a center. If we assume that ds is very small, then dN depends linearly on N by a coefficient μ_t and ds:

$$dN = -\mu_t N \, ds$$

Thus, the higher μ_t is the more photons are absorbed or scattered in the interval $s + ds$. μ_t is called the transmission coefficient.

The differential equation can be easily solved by the method of separation of variables:

$$\begin{aligned}
\frac{dN}{N} &= -\mu_t \, ds \\
\int_{N_0}^{N} \frac{1}{N} \, dN &= -\mu_t \int_0^s ds \\
[\ln N]_{N_0}^{N} &= -\mu_t s \\
\ln \frac{N}{N_0} &= -\mu_t s \\
N &= N_0 e^{-\mu_t s}
\end{aligned}$$

Now, we know that if we have N_0 photons at the beginning, $N(s)$ photons will pass the tissue at distance s without hitting a center.

In order to get estimates for the free path length in the simulation, we need to have a probability distribution:

$$\begin{aligned}
\int_0^\infty N(s)\,ds &= 1 \\
\int_0^\infty N_0 e^{-\mu_t s}\,ds &= 1 \\
N_0\left[-\frac{1}{\mu_t}e^{-mu_t s}\right]_0^\infty &= 1 \\
-\frac{1}{\mu_t}N_0(e^{-\infty}-e^0) &= 1 \\
N_0 &= \mu_t
\end{aligned}$$

and thus the probability density function f_S for the random variable S which denotes the free path length is

$$f_S = P(S = s) = \mu_t e^{-\mu_t s}.$$

What is μ_t?

Let's have a look at the mean value of this distribution:

$$\begin{aligned}
&\int_0^\infty s\mu_t e^{-\mu_t s} \\
&= \frac{\mu_t}{\mu_t^2}\left[e^{-\mu_t s(-\mu_t s-1)}\right]_0^\infty \\
&= \frac{1}{\mu_t}
\end{aligned}$$

Thus, the mean free path length is equal to $\frac{1}{\mu_t}$. Since we wanted to have centers where both scattering and absorption occurs, we can also relate μ_t to the scattering and absorption coefficient:

$$\mu_t = \mu_s + \mu_a$$

and finally the mean free pathlength is

$$< s >= \frac{1}{\mu_t} = \frac{1}{\mu_s + \mu_a}$$

Transformation of a random variable

Since it is not common to have a random variable generator which is returning such a distribution, in the following it will be described how it can be simulated by a uniformly distributed random variable X. Basically, we look for a function $s = g(x)$ which generates a value for a free path length distributed by f_S.

The transformation is done with the distribution functions of S and X:

$$\begin{aligned} P(S < s) &= P(X < x) \\ \int_0^s f_S \, ds &= \int_0^x f_X \_ dx \\ \int_0^s \mu_t e^{-\mu_t s} \, ds &= \int_0^x dx \\ -\left[e^{-\mu_t s}\right]_0^s &= x \\ e^{-\mu_t s} &= 1 - x \\ s &= -\frac{1}{\mu_t} \log(1 - x) \\ g(x) &= -\frac{1}{\mu_t} \log(1 - x) \end{aligned}$$

3.1.1.3. Updating the position

$\mathbf{X}$ is updated by

$$\mathbf{X} = \mathbf{X} + s\mathbf{D}$$

3.1.1.4. Photons still inside of tissue?

The simulation ends if either the weight goes under a certain threshold or the photons left the tissue.

As mentioned, the tissue is infinitely wide and has height d (the depth). The photons are outside of the tissue if

$$\mathbf{X}_3 > 0.$$

The result of the simulation is both x and the final weight w. Figure 3.7 shows that x is the point of intersection of the ray with the image plane. This is the virtual point $\mathbf{P}_v$ where the light seems to come from and thus leads to blurring.

For $\mathbf{P}_v$ we get

$$\mathbf{P}_v = \mathbf{X} + k\mathbf{D}.$$

Since we know that $\mathbf{P}_{v3} = -d$, k is calculated by

$$k = -\frac{d + \mathbf{X}_3}{\mathbf{D}_3}$$

and finally

$$x = \|\mathbf{P}_v - \mathbf{P}\|.$$

3.1.1.5. Updating the weight

After the position was updated scattering and absorption has to be treated. Scattering only will affect the angle of the photon packet, and absorption affects the weight. The weight decreases by Δw:

$$w = w - \Delta w$$

and Δw depends on the absorption:

$$\Delta w = \frac{\mu_a}{\mu_t}.$$

3.1.1.6. Termination

For computational cost reasons we cannot track a photon packet forever, and thus we have to decide when we are not interested anymore in propagating it. Since photon packets of very low weight won't affect the distribution very much (only few photons will hit the image plane), we define a threshold weight w_T and if the actual weight w lies under this threshold value we start thinking about termination.

Note that we can't simply terminate the simulation if $w < w_T$, because then we would "loose" energy and therefore violate energy conservation.

A roulette method can be used to solve that problem: If $w < w_T$ we decide with probability P_T whether the packet should be lost or still tracked. If the result is that it's still tracked then its new weight is

$$w = \frac{1}{P_T} w$$

In average the energy will be conserved.

3.1.1.7. Changing the photon's direction

Once we computed the new weight and decided that the photon propagation will be continued we have to scatter the photon's direction.

As shown in [8, p. 107] scattering in tissue can be characterized by the Henyey-Greenstein phase function (which originally was proposed in astrophysics). It is a probability density function for the scattering angle.

$$f_{\mathrm{HG}} = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\phi)^{\frac{3}{2}}}$$

g is the anisotropy factor, and the higher g is the less probable it is to get high values for ϕ and the more forward directed will be the scattering.

Figure 3.8 shows the function for some values of g.

As [8, p. 107] shows, the Henyey-Greenstein function can be transformed in order to get a generating function. X is a uniformly distributed random variable.

$$\cos\phi = \frac{1}{2g}(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2gX}\right)^2)$$

The new direction updates by the following equations:

$$\begin{aligned}
\mathbf{D}_1' &= \frac{\sin\theta}{\sqrt{1 - \mathbf{D}_3^2}} (\mathbf{D}_1\mathbf{D}_3\cos\phi - \mathbf{D}_2\sin\phi) + \mathbf{D}_1\cos\theta \\
\mathbf{D}_2' &= \frac{\sin\theta}{\sqrt{1 - \mathbf{D}_3^2}} (\mathbf{D}_2\mathbf{D}_3\cos\phi + \mathbf{D}_1\sin\phi) + \mathbf{D}_2\cos\theta \\
\mathbf{D}_3' &= -\sin\theta\cos\phi\sqrt{1 - \mathbf{D}_3^2} + \mathbf{D}_3\cos\theta
\end{aligned}$$

3.1.2. Results

Figure 3.9 shows the distribution for the distances with respect to the point source. All photons that have some distance will lead to a blurred image.

Of most interest is the relationship between depth and the variance of that probability distribution. This is shown in figure 3.10.

This justifies that one can simply take the depth of a voxel as a measure of quality (of course some scaling will be necessary).

3.2. Scattering of laser beam

As already mentioned in chapter 2, also the laser beam is being scattered and thus the convolution is changing locally. This is illustrated in figure 3.11.

Since it is complex to model both scattering and convolution, a simplification was made: As with the scattering of the fluorescence light, the quality will decrease linearly with the penetration depth of the laser beam.

3.3. Result

By combining the results of the two previous sections, the quality function used in this work is the sum of the penetration depth of the laser beam and the depth of the fluorescence light. With a scaling factor we get

$$Q(\mathbf{x}) = \frac{d_{\text{laser}}(\mathbf{x}) + d_{\text{fluorescence}}(\mathbf{x})}{d_{\text{max}}}.$$

Since we already have the volume and together with a threshold value that is distinguishing between background and tissue voxels, this function can be discretly precalculated. A matlab script doing that is part of the project.

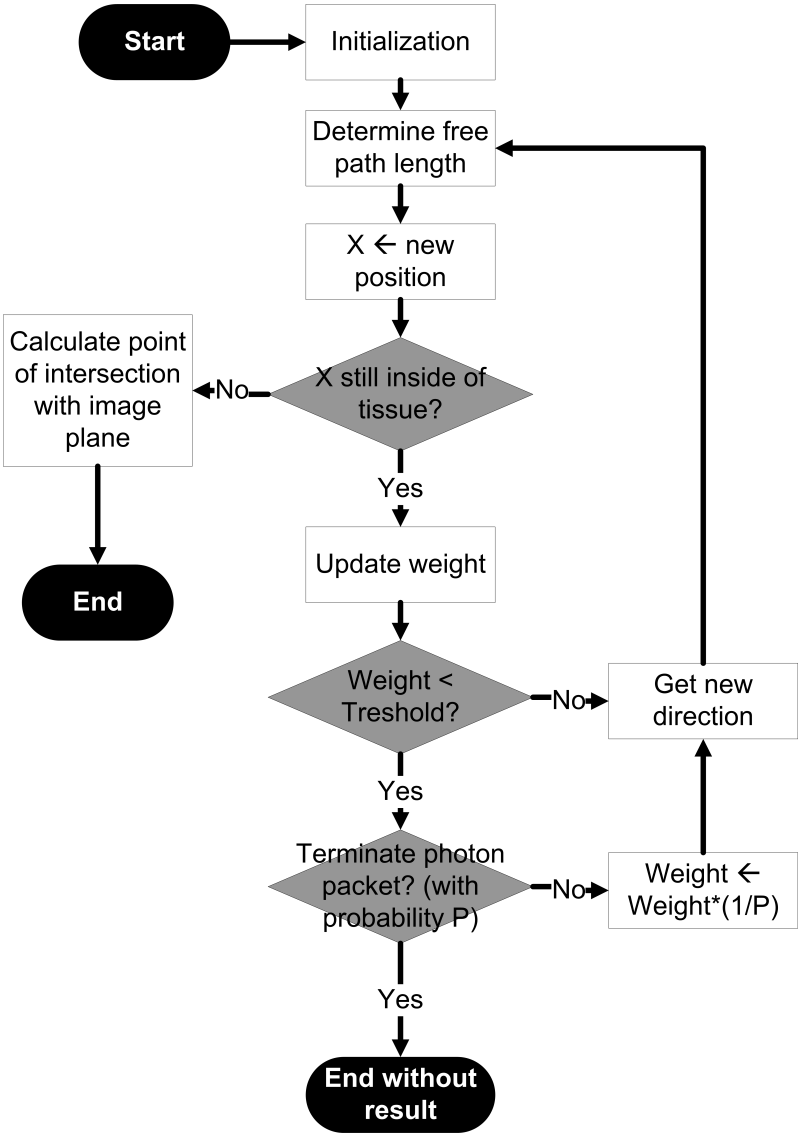


Figure 3.6.: Light propagation in tissue

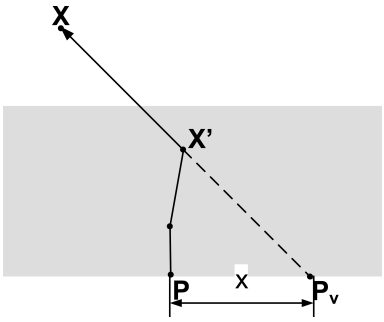


Figure 3.7.: Light propagation in tissue

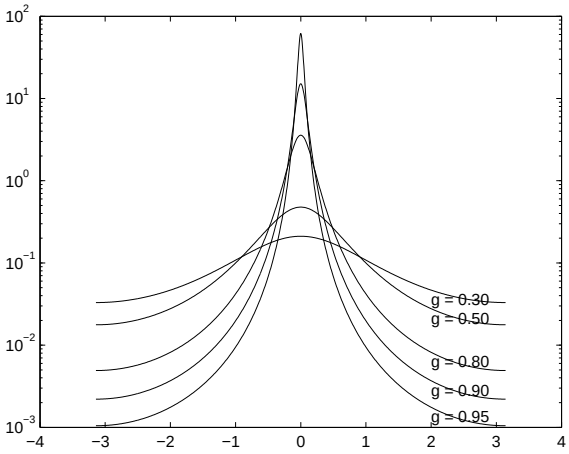


Figure 3.8.: Henyey-Greenstein density function for different values of g

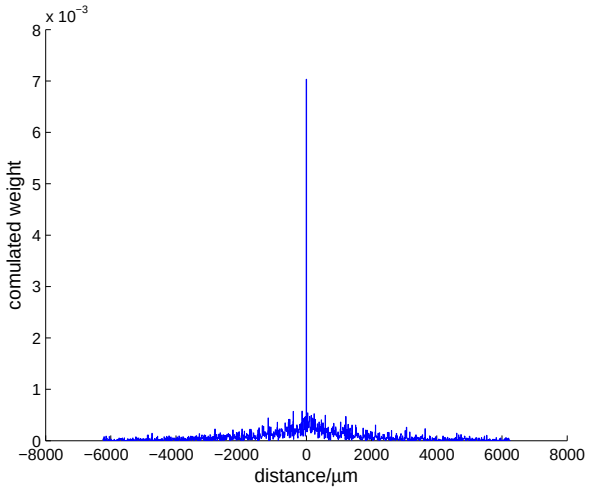


Figure 3.9.: Number of photons for different distances to the original point source: most of the photons are not scattered

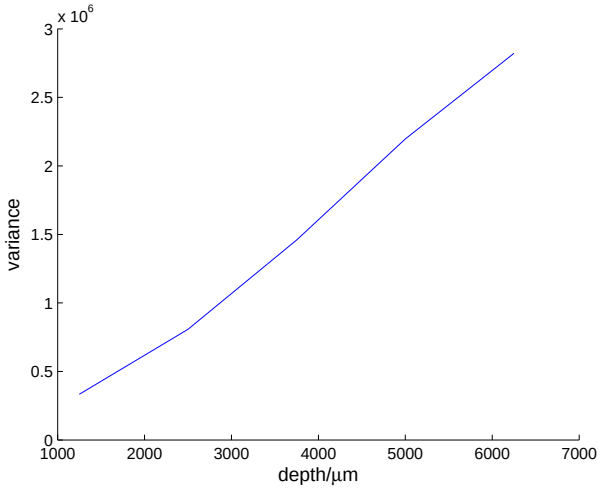


Figure 3.10.: The depth related to the variance: approximately a linear relationship

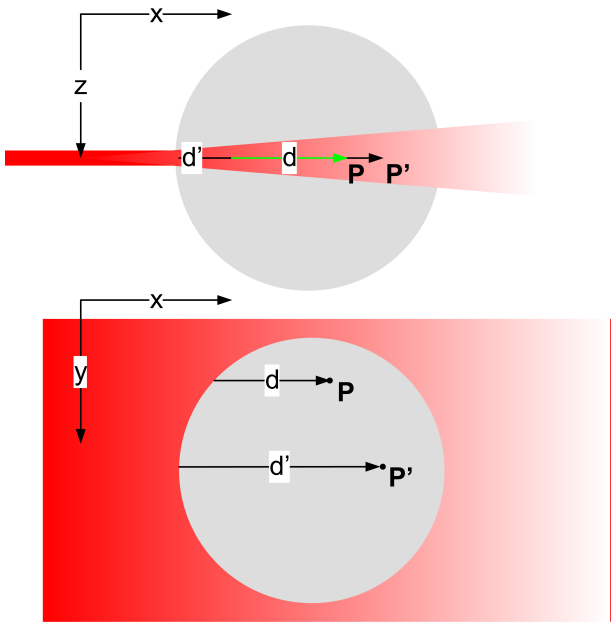


Figure 3.11.: How scattering of the laser beam affects the convolution

4. Image registration

Image registration in 3D consists of finding a transformation function φ such that a reference image $R : \mathbb{R}^3 \to \mathbb{R}$ and a template image $T : \mathbb{R}^3 \to \mathbb{R}$ become most similar. The transformation will be applied to the template image. Basically two methods should be distinguished:

- Rigid registration
  φ is a rigid body transformation and thus only consists of translations and rotations. It can be written as

  $$\varphi(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$$

  where R is a rotation matrix and $\mathbf{t}$ is a translation vector.

- Non-rigid or deformable registration
  Additionally to the rigid body transformation other non-rigid transformations are considered. This is especially useful in medicine, where one e. g. wants to register two CT images taken at different points of time. Bowels can have changed their size or exact position and thus a rigid transformation wouldn't lead to a satisfying result.

In our case, since the imaged data is already dead and doesn't deforme in a significant way, only rigid registration was used and will be explained in more detail here.

Similarity measures

A similarity measure is a function

$$S : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$$

that takes two images and returns a similarity value. The more similar the images are, the higher should be the similarity measure. There are also distance measures: here, a lower value leads to a higher similarity.

The goal is to find a transformation φ that maximizes the similarity measure, thus

$$S(R(\mathbf{x}), T(\varphi(\mathbf{x}))) \to \text{maximize!}$$

Common similarity/distance measures are:

- Sum of Absolute Differences
- Sum of Squared Differences
- Cross-Correlation
- Normalized Cross-Correlation (NCC)
- Mutual Information

Since it was used in this work, the Normalized Cross-Correlation will be described in more detail later in this chapter.

4.1. Rigid registration

In rigid registration there are several different methods and this work does not aim to give a complete overview of image registration. Methods chosen for this thesis were mostly inspired by [5]. The main ingredients for rigid image registration are:

- An initial estimate
- A similarity measure
- A non-linear optimization algorithm

There are several methods to get an initial estimate, but in this project not too much effort was put into that field since it was easy to let the user define this initial estimate: It is a rotation about the x-axis of $90°$ or $180°$. The more precise the initial estimate is, the less time will be required for the registration since less evaluations of the similarity measure are necessary. The provided software Q-RAM allows to manually define a very detailed initial pose.

Finally, a non-linear global optimization algorithm like Best Neighbor, Powell-Brent or Gradient Descent can be used. I can share the good experiences with the Best Neighbor optimizer that W. Wein made in [10].

The Best Neighbor algorithm (also referred as Hill-Climbing) is given a parameter vector P which is the parameter for the cost function that has to be optimized. This cost function is the similarity measure we use, thus the NCC or WNCC. In each iteration step, the cost function will be evaluated several times for different variations of P, and the one that returns the highest value for the cost function will be taken for the next iteration step. In our case, P is a 6-vector containing the translation and rotation applied to the moving volume. For each entry P_i, both a higher value and a lower value are tried, depending on the given step size. If none of the steps achieves a higher similarity measure, then the step size is being decreased and the algorithm tries again. Best Neighbor terminates after a specified number of iterations, a specified number of cost function evaluations or after the reached parameter changing is lower than a specified tolerance.

4.2. Quality-based registration

Quality based registration means that the quality of voxels will be taken into account in order to perform the registration. Comparing voxels of very low quality with other voxels doesn't make sense, because we cannot be sure about the intensity information of the voxel with low quality. Somehow the quality of a voxel can be seen as a measure of information that we have about the voxel. If some parts of a volume have low quality and thus contain few information, it is not useful to respect them in the same way for the registration as we do for high quality parts of the volume.

As described in part 3 we have a model for the quality in each volume. What I now will propose is a way to include this quality measure in the registration.

4.2.1. Including the quality measure into the registration

The rigid registration as an iterative process doesn't have to be changed at all. We can still use the optimization algorithms that we used to use, and the goal still is to maximize the similarity between the two similarity measures.

The adoption takes place in the similarity measure. As shown in earlier sections, the similarity measure of choice is the normalized cross-correlation. In order to adopt this similarity measure we will have to take a closer look on it.

4.2.2. Normalized Cross-Correlation

The normalized cross-correlation is a measure of difference of two images/volumes. One can substract the average voxel intensity in order to get rid of brightness differences. The resulting image will be

$$R'(\mathbf{x}) = R(\mathbf{x}) - \overline{R} = R(\mathbf{x}) - \frac{\sum_{\mathbf{x}} R(\mathbf{x})}{N}.$$

The same applies to T.

Since we cannot deal directly with continuous images, we have to know how to represent them discretely. Discrete images can be represented by a vector:

$$R'(\mathbf{x}) \approx \mathbf{r}'_{\alpha(\mathbf{x})} \text{ with } \alpha(\mathbf{x}) = x_1 + n_1 x_2 + n_1 n_2 x_3.$$

The more similar the images are, the smaller is the angle between the two image vectors. Consequently, the normalized cross-correlation is defined by

$$\mathrm{NCC}(\mathbf{r}, \mathbf{t}) := \cos(\angle(\mathbf{r}', \mathbf{t}'))$$

Note that since we have $\mathbf{r}'$ and $\mathbf{t}'$ (and not $\mathbf{r}$ and $\mathbf{t}$) for the definition it is invariant to brightness differences.

In addition it is invariant to contrast, because different contrast would just be a different scale of one or both of the vectors, and thus doesn't lead to a different angle.

Since for $\angle(\mathbf{r}', \mathbf{t}') = 0$ the similarity is best, a value of 1 for the normalized cross-correlation stands for the highest possible similarity. For $\angle(\mathbf{r}', \mathbf{t}') = 180$ the similarity will be worst and the cosine is -1.

With the scalar product we get:

$$\begin{aligned}
\cos(\angle(\mathbf{r}', \mathbf{t}')) &= \frac{\mathbf{r}'^T \mathbf{t}'}{\|\mathbf{r}'\| \|\mathbf{t}'\|} \\
&= \frac{\sum_i \mathbf{r}'_\mathbf{i} \mathbf{t}'_\mathbf{i}}{\sqrt{\sum_i \mathbf{r}'^2_\mathbf{i}} \sqrt{\sum_i \mathbf{t}'^2_\mathbf{i}}} \\
&= \frac{\sum_\mathbf{x} R'(\mathbf{x}) T'(\mathbf{x})}{\sqrt{\sum_\mathbf{x} R'(\mathbf{x})^2} \sqrt{\sum_\mathbf{x} T'(\mathbf{x})^2}} \\
&= \frac{\sum_\mathbf{x} (R(\mathbf{x}) - \overline{R})(T(\mathbf{x}) - \overline{T})}{\sqrt{\sum_x (R(\mathbf{x}) - \overline{R})^2} \sqrt{\sum_\mathbf{x} (T(\mathbf{x}) - \overline{T})^2}}
\end{aligned}$$

4.2.3. Adopting the Normalized Cross-Correlation

How can the quality function Q be included in the normalized cross-correlation? Remember, that every image intensity value is corresponding to a quality value:

$$\begin{aligned}
R(\mathbf{x}) &\leftrightarrow Q_R(\mathbf{x}) \\
T(\mathbf{x}) &\leftrightarrow Q_T(\mathbf{x}).
\end{aligned}$$

The idea is to define a weighing between two corresponding image points $\mathbf{x}$:

$$w(\mathbf{x}) = Q_R(\mathbf{x}) Q_T(\mathbf{x})$$

Now, if we define new images

$$R^*(\mathbf{x}) = w(\mathbf{x}) R'(\mathbf{x})$$

and

$$T^*(\mathbf{x}) = w(x) T'(\mathbf{x})$$

the weighted normalized cross-correlation (WNCC) is defined by

$$\mathrm{WNCC}((\mathbf{r}, \mathbf{t})) := \cos(\angle(\mathbf{r}^*, \mathbf{t}^*)).$$

Remember that the normalized cross-correlation is $\cos(\angle(\mathbf{r}', \mathbf{t}'))$. What we want is that bad quality voxels don't affect $\angle(\mathbf{r}', \mathbf{t}')$ as much as good quality voxels do. Every voxel corresponds to an entry of $\mathbf{r}'$. If this voxel for example has quality zero, then the angle $\angle(\mathbf{r}', \mathbf{t}')$ should not be affected by this entry: we simply don't want to let it take part in the normalized cross-correlation. Since its weight would be zero if one of the quality values is zero, this is the case. On the other hand, if for example both voxels have a quality value of one, then the resulting weight is also one. Then they contribute to $\angle(\mathbf{r}', \mathbf{t}')$ just like in the NCC.

4.3. Evaluation

A quick look on the results of the registration confirms: the registration is quite good. But for applications like studying neural networks in the mouse brains it's not enough to know that the result is "quite good", we really want to know in more detail how big the error is. Another interesting aspect is to know whether the quality-based technique is better or not, since at first sight a difference with respect to the normalized cross correlation can't be observed.

In order to evaluate one has to have two datasets with a known transformation. A common way to get such datasets is to synthetically generate ground truth data. It has the advantage that its accuracy is really perfect since we know the exact pose from one to the other. The drawbacks of ground truth data for our application is that it is difficult to produce really good synthetic data. Even though we already have a quality measure and thus just had to apply blurring, we didn't consider all the physical effects that could occur and thus the two generated datasets don't seem to be able to imitate real circumstances.

Fortunately, we have real datasets and their (almost) exact transformation: Since the preparation in one step is illuminated once from the left and once from the right, we would have really perfect data if the laser's illumination planes would overlap exactly. So, in order to use that data, one first has to roughly estimate how big the error of the given transformation might be.

4.3.1. How to compare transformations

The task is to compare a given transformation (the actual registration transformation) to a known transformation which should be achieved. Both in the case of synthetic data as in the real data this will be the identity transformation

$$\varphi(\mathbf{x}) = \mathbf{x}.$$

This might sound simple at first view but in fact we have to deal with the problem to compare both translations and rotations, thus, how to compare apples and oranges.

4.3.1.1. Separate comparison

One idea is to compare translations and rotations separately. Comparing translations is easy since we just could look at the euclidean distance, but how do we compare the rotations? We have three angles α_x, α_y and α_z with α_i standing for a rotation about the i-axis with angle α_i. They are given in XYZ-order meaning that we first apply the x-rotation, then the y-rotation and finally the z-rotation. In order to compare them one idea is to transform this XYZ-rotation into a rotation about an axis and only one angle α. We would be interested only in α as a comparable quantity. If R is a rotation matrix representing the given XYZ-rotation then according to [9] we get:

$$\cos\alpha = (\text{Trace}R - 1)/2.$$

The approach has the following drawbacks:

- It's difficult to judge how good the registration really was in the end. The method is only valid for relative comparison of the performance of two registration methods, but the value itself is difficult to judge.
- If we compare two registration methods to each other, what happens if method A is performing better in the translation than method B and for the rotations it's the other way round? Then it's up to the viewer of the diagram to decide which one is really performing better.

4.3.1.2. Combined comparison - compare apples and oranges

What we really want is one single quantity that's telling us how good the registration was. The key idea is described in [5] (although here I don't make the difference between Target Registration Error and Fiducial Registration Error): we are interested in what effect does the transformation have on points in the image. In the case of only translation and no rotation, the effect is equal in the whole volume. A voxel on the border of an image is transformed in the same manner as one in the middle. For rotations it's different: since rotations are always meant to be about the center of the volume, the effect is higher at the borders. In order to get an upper bound for the error, it seems to be reasonable to apply the transformation on the corner points of the bounding box. If T is the transformation stored in a 4×4 homogeneous matrix, and the corner points of the bounding box are P_1, ..., P_8, then the average error e is

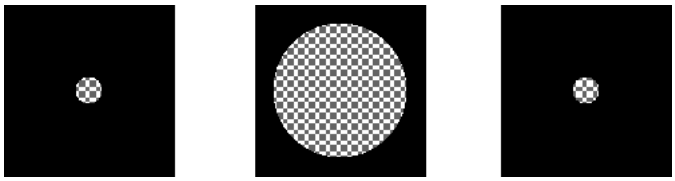
$$e = \frac{1}{8}\sum_{i=1}^{8} \|TP_i - P_i\|.$$

Attention: we have to take care for the units, because T is normally given in μm and we finally want to have the error in pixels of the used image.

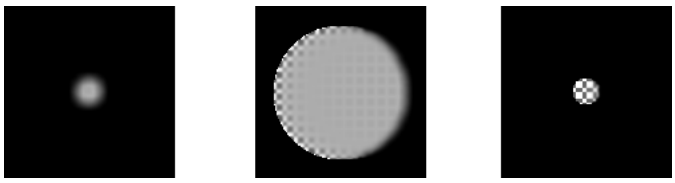
4.3.2. Ground truth data

The synthetic data used was a sphere with a 3D checker pattern inside. The quality volumes were used in order to blur the the synthetic data and so two datasets with different blurred parts were produced. Figure 4.1 shows how the original and the blurred volumes look like.

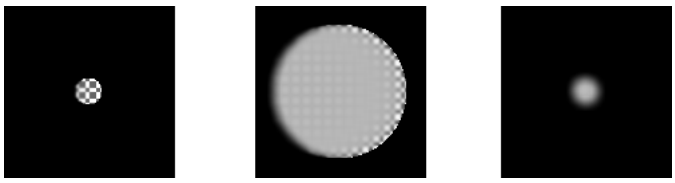
Figure 4.2 shows how the error for the translation and rotation (both separated) performs. No significant difference can be observed between the quality based (the weighted NCC) and the normal approach.



(a) The original sphere. The leftmost slice is from the top of the sphere, the middle slice from the middle and the rightmost slice is from the bottom.



(b) The sphere with simulated blurring: The laser plane is coming from the left and the microscope is situated on the top



(c) The sphere with simulated blurring: The laser plane is coming from the right and the microscope is situated on the bottom

Figure 4.1.: A sphere with 3D checker pattern used as synthetic data

Figure 4.3 shows how close an average corner voxel comes to the perfect position. The error can be reduced from more than 5 pixels to about 1 pixel. This is quite astonishing taking into account how much the data was blurred.

4.3.3. Real data

The evaluation was made for a mouse embryo lying on the stomach. As explained above, in one single step we get two images: first the laser shines from the left and then from the right. The effect of blurring because of the convolution due to the scattering of laser light was explained in detail above. This effect gives us two different datasets which should be not moved nor rotated at all.

4.3.3.1. How exact is the transformation?

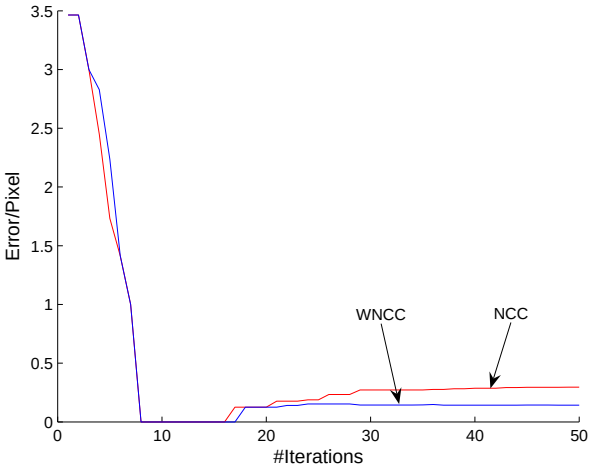
If the laser planes would overlap perfectly, we wouldn't have any translation/rotation. Unfortunately, this is not perfectly the case. One can observe that by looking at corresponding slices: the end of the mouse's head in the first sequence occurs about four

images later than in the second sequence. Since for the mouse embryo the pixels are of the same size, this means that the transformation we are facing is a translation of about four pixels. Obviously this is only a rough estimation. A simple trick helps us out: If we reduce the size of the image about the factor 8, then we know that the two images are equal up to an error of 0.5 Pixels.

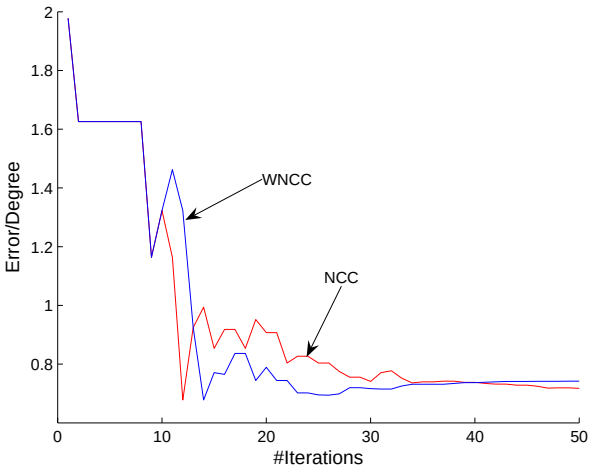
With the fact that we have two images which up to 0.5 pixels have exactly the same pose one to the other, we can start a registration with a higher initial error and see how the performance is. Once we know that the registration is exact up to e pixels for an image of this size, we can assume that it will be exact up to e pixels for larger images, too.

Let's have a look at the results: figure 4.4 shows that in both translation and rotation the two approaches end up without any significant difference.

Combining the apples and oranges, figure 4.5 shows that after a worse performance of the quality-based approach it ends up with almost the same error. The registration is perfect up to two pixels. This is good news for the final merging of the data which will be explained in detail in chapter 5.



(a) The translations compared to each other



(b) The rotations compared to each other

Figure 4.2.: Ground truth data: translations and rotations separeted

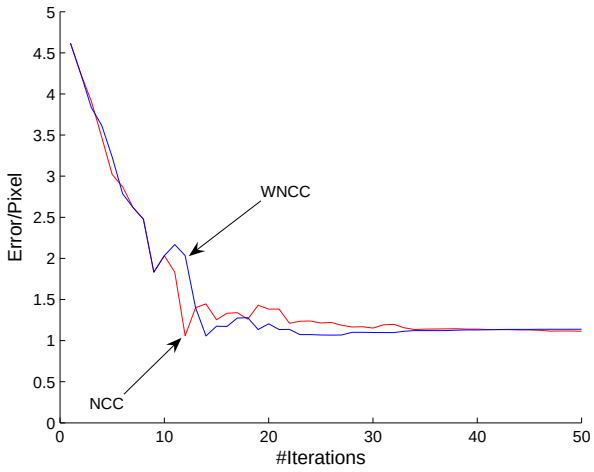
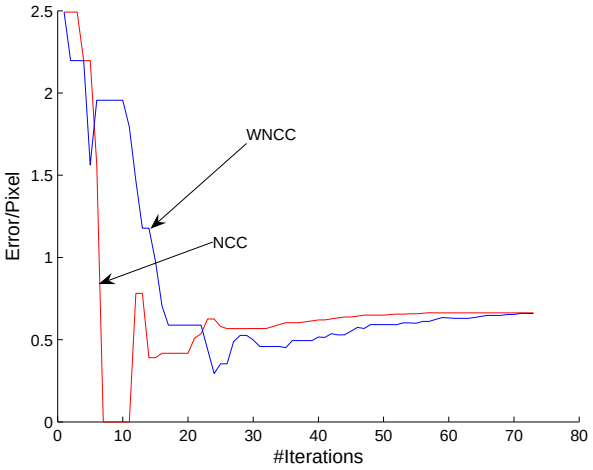
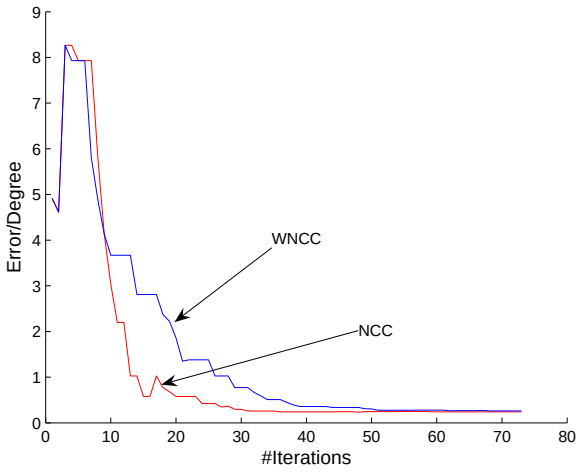


Figure 4.3.: Ground truth data: the two approaches finally result in the same error.



(a) The translations compared to each other



(b) The rotations compared to each other

Figure 4.4.: Real embryo: translations and rotations separated

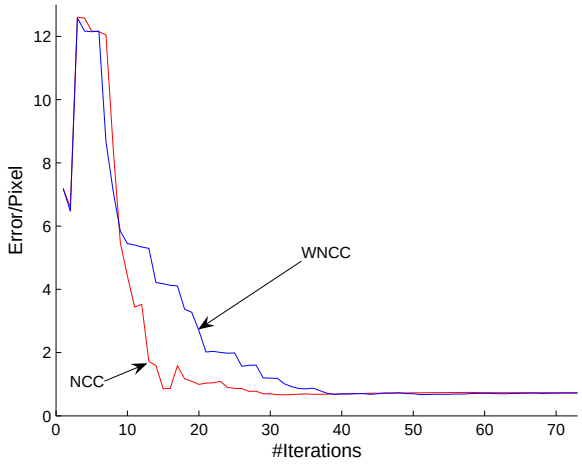


Figure 4.5.: Real embryo: the two approaches finally result in the same error.

5. Merging the data

Once the final rigid transformations between the different volumetric datasets are known one can put them together into one dataset. This process is referred as the merging of the data.

The main problem we face when we are merging the data is that we have way more information than we would need: For one resulting volume, we can choose out of several volumes. Thus, a decision criteria is necessary: If V is the resulting volume, for the voxel at position $\mathbf{x}$ we have n corresponding voxels of the volumes $V_1, V_2, ..., V_n$.

As described in chapter 3, at the point of merging we already dispose of a measure for quality for each voxel of each volume. As consequence, one simple possibility is to choose the corresponding voxel with the highest quality value, or to define a blending function, which takes into account all the corresponding voxels of the volumes $V_1, V_2, ..., V_n$ and prefers the one with the highest quality value.

Unfortunately the quality measure described in chapter 3 does only consider two of the three described causes of blurring. Remember the distinction of *local* and *global* quality measures made in part 3. So, the merging process has to respect both

- the local quality measure and
- the convolution.

5.1. Classifying the volumes

In order to do that, it seems to be useful to classify the given volumes which should be merged. Volumes that have an angle of $\approx 0^\circ$ or $\approx 180^\circ$ with the x- or y-axis (relative to each other) belong to the same class, since the z-blurring arising from the convolution is in the same direction. In this work I assume that there are only two classes of volumes according to that classification: The blurring direction of the first class is approximately perpendicular to the one of the second class.

Merging algorithm

So, the merging algorithm consists of two steps:

- Merging the volumes of the same classes
- Merging the two classes

The merging of volumes associated to the same classes can be done as described above: Since we already have a quality measure, we can simply choose the best voxel out of the volumes $V_1, V_2, ...V_n$ of the same class, or we mix all the corresponding voxels and favor those with higher quality.

Merging the two classes can be done in two ways: We can apply a deconvolution as the inverse transformation to the convolution, and then merge them in the same manner as we did before. For the deconvolution we have to know exactly the PSF (Point spread function) of the laser beam. In this work, a second way is proposed: the idea is that the gradients in x- and y-direction are not that much affected by the blurring in z-direction as the z-gradients. A system of linear equations can be built which just has to be optimized. The next section will discuss these two ideas.

5.2. Reducing the convolution effect - merging the classes

5.2.1. Deconvolution

One idea that first comes to ones mind would be a deconvolution. The basic idea is the well known convolution theorem:

$$\mathcal{F}(f * g) = \sqrt{2\pi}\,(\mathcal{F}f) \cdot (\mathcal{F}g)$$

$\mathcal{F}f$ stands for the fourier transform of the function f. Since we know $f * g$ (this is the image we measure!) and if we know g, by juggling with the convolution theorem we get

$$\begin{aligned}\mathcal{F}(f * g) &= \sqrt{2\pi}\,(\mathcal{F}f) \cdot (\mathcal{F}g)\\ \mathcal{F}f &= \frac{\mathcal{F}(f * g)}{\sqrt{2\pi}\mathcal{F}g}\\ f &= \mathcal{F}^{-1}\frac{\mathcal{F}(f * g)}{\sqrt{2\pi}\mathcal{F}g}.\end{aligned}$$

5.2.2. Gradient-based optimization

As described in the first chapter, the convolution leads to blurring in the direction of the optical axis. This blurring can be seen as a smoothing of the gradient in this direction. A short look on the slices confirms: the blurring has only little effect on the gradients in other directions perpendicular to the optical axis.

This leads to a basic idea: One could turn the preparation about 90 degrees and build another set of 3D volumetric data.

For each of the three directions we now find a non-blurred direction in one of the two provided datasets. Figure 5.1 illustrates these facts.

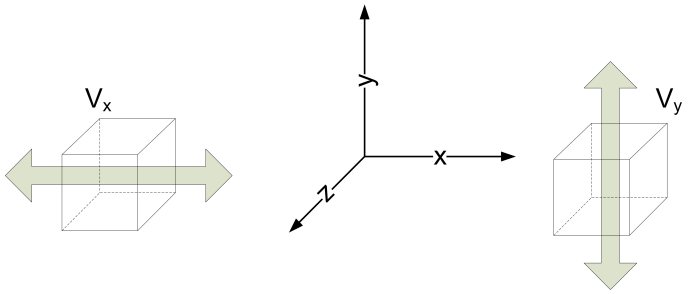


Figure 5.1.: V_x is blurred in x-direction and V_y in y-direction

5.2.2.1. Getting partial differential equations

How can we take advantage of that? Formally written, we have a volume V_x which is blurred in x-direction and a second volume V_y that is blurred in y-direction. We want to have a volume V that has as less blurring as possible.

We assume that for the volume V_x the gradient in y-direction (written as $\frac{\partial}{\partial_y} V_x$) and the gradient in z-direction $\frac{\partial}{\partial_z} V_x$ are not affected by the blurring. Same assumptions are made for V_y: $\frac{\partial}{\partial_x} V_y$ and $\frac{\partial}{\partial_z} V_y$ are supposed to be not affected.

Thus, the goal is to find a V that satisfies

$$\frac{\partial}{\partial y}V - \frac{\partial}{\partial y}V_x + \frac{\partial}{\partial z}V - \frac{\partial}{\partial z}V_x + \frac{\partial}{\partial x}V - \frac{\partial}{\partial x}V_y + \frac{\partial}{\partial z}V - \frac{\partial}{\partial z}V_y = 0$$

This can be rewritten to

$$\left(\frac{\partial}{\partial y} + \frac{\partial}{\partial z} + \frac{\partial}{\partial x} + \frac{\partial}{\partial z}\right) V = \frac{\partial}{\partial y}V_x + \frac{\partial}{\partial z}V_x + \frac{\partial}{\partial x}V_y + \frac{\partial}{\partial z}V_y$$

5.2.2.2. Solving the partial differential equations with finite differences

Since V_x and V_y are already known, the right part of equation 5.1 is a constant. As already seen in chapter 4, discrete images can be represented by a vector. Similarly, the partial differential operators can be represented by matrices (supposing that the size of the volumes is $s_x \times s_y \times s_z$):

$$\frac{\partial}{\partial x} \approx G_x \text{ with } G_{x_{ij}} = \begin{cases} -1 & \forall i,j : i = j \\ 1 & \forall i,j : j = i+1 \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial}{\partial y} \approx G_y \text{ with } G_{y_{ij}} = \begin{cases} -1 & \forall i,j : i = j \\ 1 & \forall i,j : [i = x + ys_x + zs_xs_y] \ \wedge \ [j = x + (y+1)s_x + zs_xs_y] \wedge \\ & (x < s_x) \ \wedge \ (y < s_y) \ \wedge \ (z < s_z) \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial}{\partial z} \approx G_z \text{ with } G_{z_{ij}} = \begin{cases} -1 & \forall i,j : i = j \\ 1 & \forall i,j : [i = x + ys_x + zs_xs_y] \ \wedge \ [j = x + ys_x + (z+1)s_xs_y] \wedge \\ & (x < s_x) \ \wedge \ (y < s_y) \ \wedge \ (z < s_z) \\ 0 & \text{else} \end{cases}$$

.

That means that the right part of equation 5.1 can be represented by a vector $\mathbf{b}$. For the left part, we can factor out V and discretize the partial differential operators which finally sum up to a matrix A.

Thus, the task is to minimize

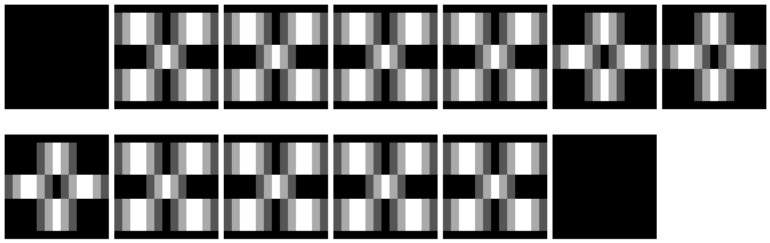
$$\|A\mathbf{v} - \mathbf{b}\|. \tag{5.1}$$

This leads to a linearized volume $\mathbf{v}$ and the delinearization of $\mathbf{v}$ is the volume V we were looking for.

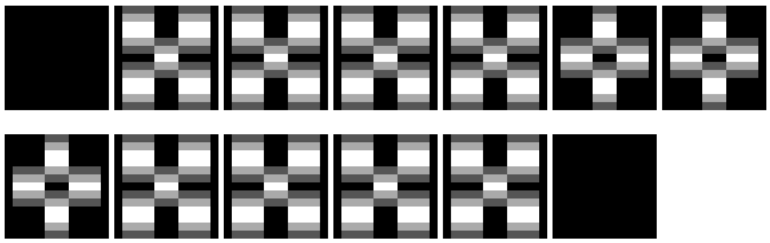
5.2.2.3. Results

In order to be able to compare the results to the original volume one has to create ground truth data and then blur it in both x- and y-directions and finally feed the linear equations with the blurred volumes. Figures 5.2 and 5.3 visualize the results we obtained for a 3D checker pattern and a 3D sphere.

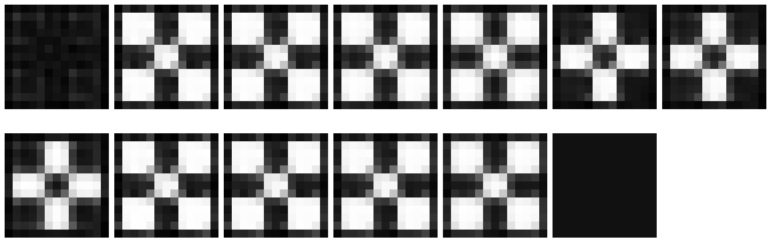
As one can see, the chess board is almost perfectly reconstructed. This is due to the fact that all lines are perpendicular to the gradient information that we have. In the case of the sphere things are similar. Again, lines perpendicular to the blurring directions can be reconstructed almost perfectly, but problems occur with diagonals.



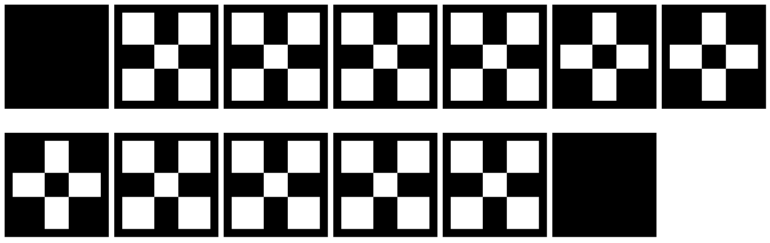
(a) Blurred in x-direction



(b) Blurred in y-direction

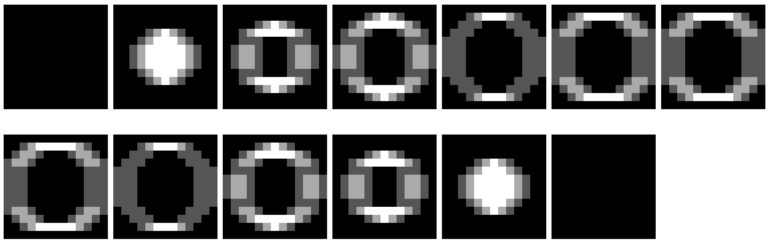


(c) Reconstruction of the two blurred cubes

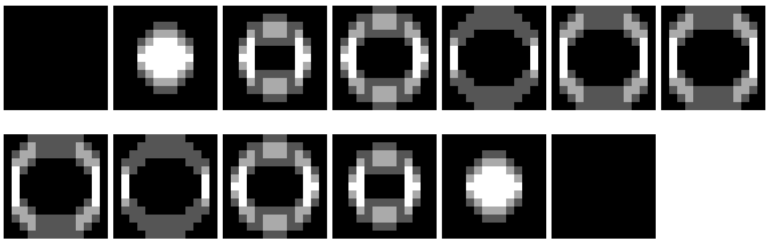


(d) The original cube

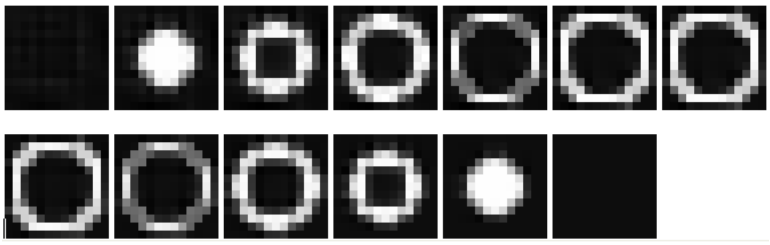
Figure 5.2.: Gradient-based optimization for a 13x13x13 3D checker pattern. The volumes are visualized by the z-slices.



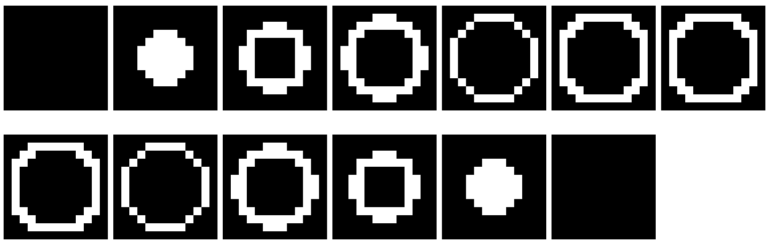
(a) Blurred in x-direction



(b) Blurred in y-direction



(c) Reconstruction of the two blurred spheres



(d) The original sphere

Figure 5.3.: Gradient-based optimization for a 13x13x13 3D sphere. The volumes are visualized by the z-slices.

6. Future work

Since this imaging technique is very new, it's clear that with one short term project not all issues with respect to the image processing can be solved and discussed. Most of the following proposals have to do with limited memory and computation speed. From my point of view the following issues deserve a deeper analysis:

6.1. Trying the gradient-based method on real data

As illustrated for the hippocampus in figure 2.3 a big problem still remains the blurring in z-direction. As shown for synthetic data, the gradient-based merging of datasets which are perpendicular to each other is very promising. At least for the synthetic data it worked quite well. Since the registration is very precise (up to two pixels) we are able to get a transformation which is exact enough in order to apply the method on real data.

However, some issues still prevent us from trying the approach:

- Because of its banana-like shape it's difficult to turn the hippocampus about 90 degrees, and until now no turned dataset is available
- The matrices in the gradient-based approach are getting extremely huge. For a volume of size $100 \times 100 \times 100$ the gradient matrix is of size $(100^3)^2 \times (100^3)^2$. Since it is a sparse matrix there might be approaches to deal with such data. But since our volumes are even larger we could come to a limit. An idea is to only solve the problem for smaller cubes and then put them together again.

6.2. Increasing registration speed

Since the volumes we are dealing with are very huge (up to several gigabytes) the registration is very slow. Until now we overcame this problem by registering with lower resolution and then applying the merging on a higher resolution. An interesting approach in order to increase the speed of registration would be to not take the whole volume but only the parts which are of more or less equal quality. Then, at least for the last stage of registration where the transformation is already quite good, a significant speed-up could be achieved. Instead of the whole volume which consists of $1000^3 = 10^9$ voxels perhaps only five smaller cubes and thus $5 \cdot 10^3$ have to be processed.

6.3. Registration and merging of multiple volumes

Another not resolved issue would be to register several volumes. According to the quality information that we already retrieved one could choose which of the volumes is most adequate in order to be the fixed volume. All the others would then be registered in order to get the relative transformation of this one. As explained in section 5.1 the order of merging is also important since it implies different methods.

6.4. Mosaicing

Last but not least, putting several volumes with only small overlapping regions together is another issue. Again, very huge datasets which are difficult to handle would be the consequence.

Another idea is that one could take an image of the whole embryo and then again and with higher resolution image the legs and the head. And finally put that together. A problem would be the mixture of low-resolution with high-resolution datasets. Since the high-resolution parts should be conserved

A. Q-Ram - Quality-Based Registration and Merging

In the scope of this project a GUI for the registration and merging was developed. It is possible to load volumetric datasets, register them and finally merge them into one single dataset. As figure A.1 shows, two volumetric datasets can be loaded at once. The left one is supposed to be static while the right one is the moving volume.

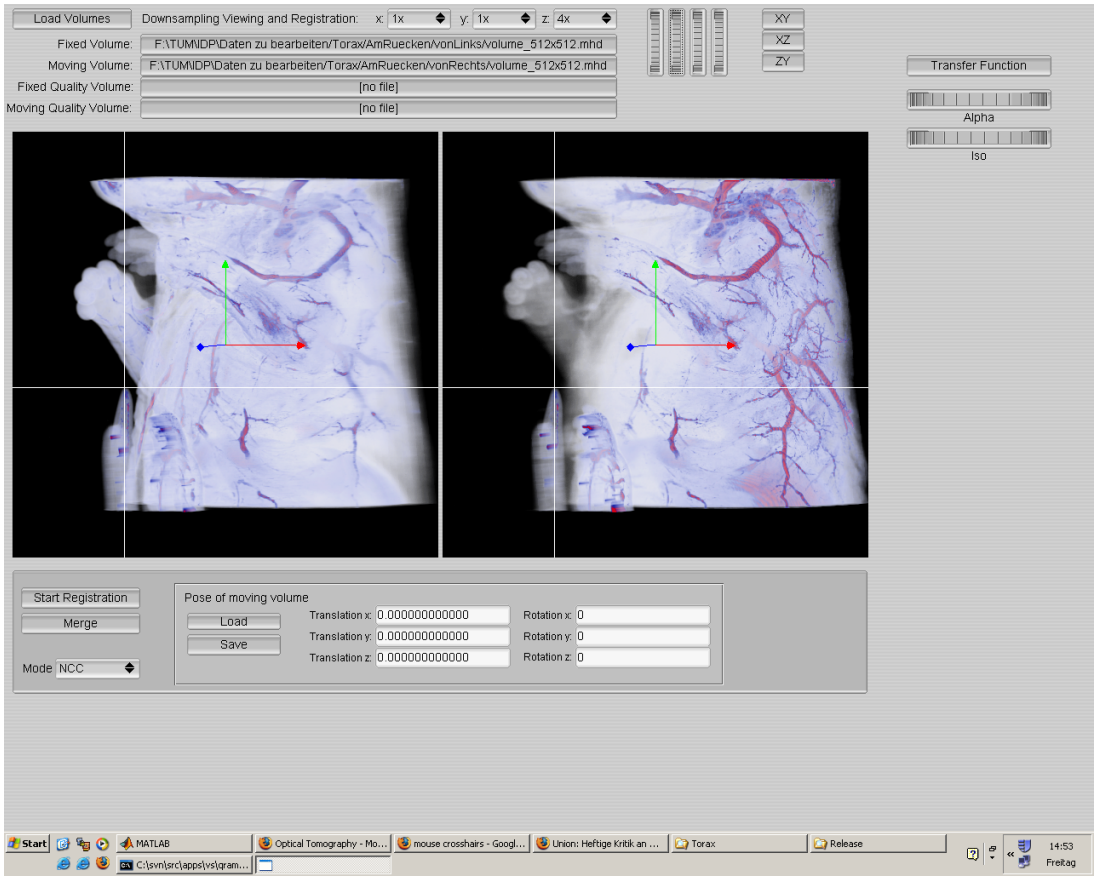


Figure A.1.: The main screen of Q-Ram

A.1. The initial pose

The user can interactively change the position and rotation of the moving volume in order to orient it to the fixed volume. For translations, the user has to press the keyboard key for the desired axis and then the left mouse button. For rotations it's the right mouse button. While moving the mouse up and down, the desired value increases or decreases. For example, a translation in x-direction is established by holding the "x"-key and the left mouse button, and then moving the mouse up and down. A rotation about the z-axis is achieved by holding the "z"-key and the right mouse button whilst moving the mouse up and down. The initial pose doesn't have to be very exact, since the registration algorithm is quite robust. The algorithm won't terminate for rotations of 180 degrees, but 30 degrees can still be handled. The initial translation should at least provide an overlapping of the two volumes.

Since it takes quite a long time for high resolution registration, it's useful to provide an as exact as possible initial pose. A mouse crosshairs pointer can be used for a rough impression whether the position is already fine or not.

A.2. Automatic registration

Once the moving volume is oriented, the user just clicks on the "Start Registration" button and the automatic registration starts. The moving volume is oriented automatically to the fixed volume.

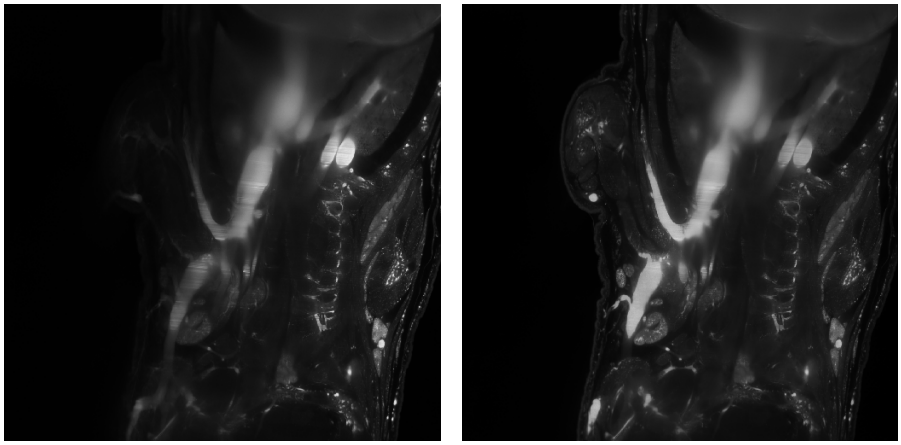
Due to computational complexity, the time for the registration is quite long. It doesn't make sense to initially start the registration with the highest possible resolution, since a better position can already be retrieved with lower resolution and thus way faster. Therefore, Q-Ram provides the possibility to downsample the data before the registration is started. Once the registration finishes for one resolution, the user can go on with a higher resolution.

Once the final transformation is established it can be saved to a file and the data can be merged.

A.3. Merging the data

Of course, even if the registration was made for a lower resolution, it's possible to merge the data in any desired resolution (as long as enough memory available). A simple click on the "Merge" button takes the selected image files, loads them and applies the merging. Note that it's necessary to provide the quality volumes in order to do the merging.

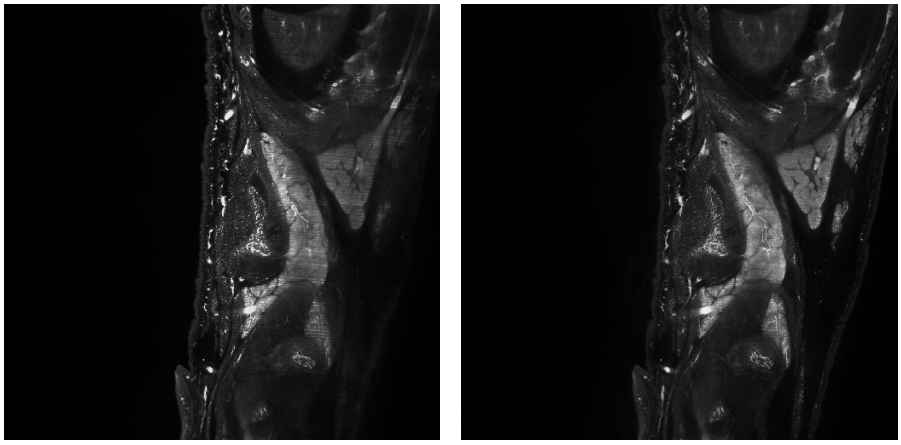
B. Results of registration and merging



(a) Slice taken with laser coming from the right

(b) Corresponding slice of the merged dataset

Figure B.1.: Embryo lying on the stomach



(a) Slice taken with laser coming from the left

(b) Corresponding slice of the merged dataset

Figure B.2.: Embryo lying on the stomach

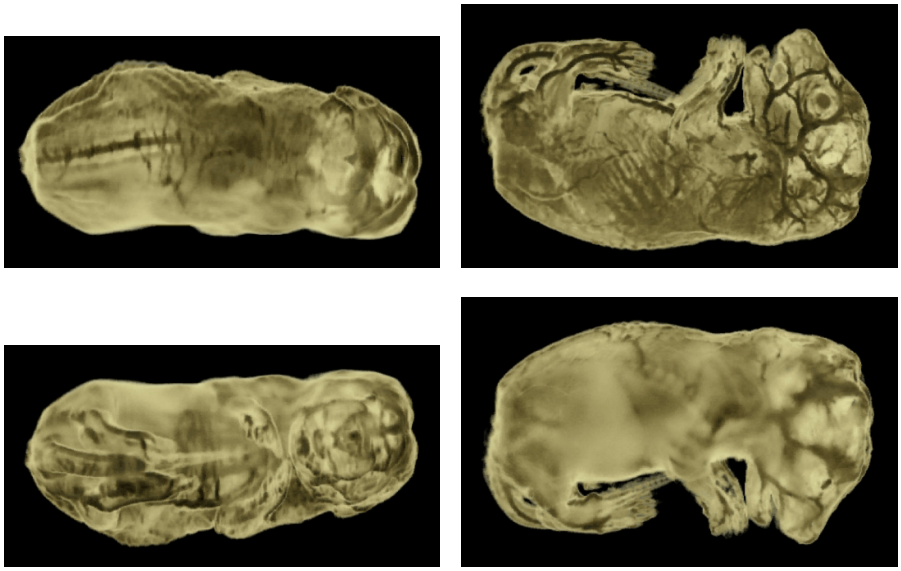


Figure B.3.: One of the two original embryos

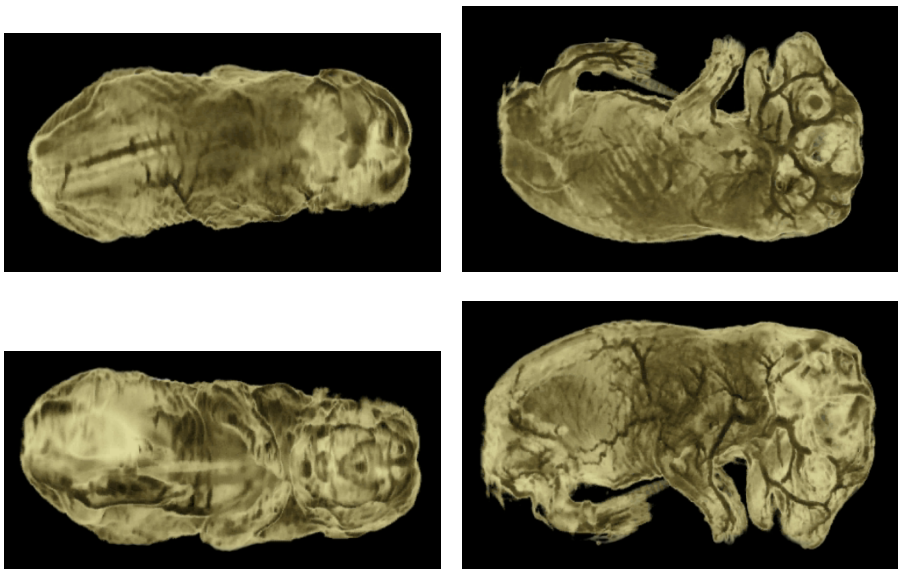


Figure B.4.: The merged embryo

Glossary

Notation	Description
Delinearization	A formally linearized image or volume is reconstructed.
Downsample	Reducing the size of a volumetric dataset.
GUI	Graphical User Interface
Linearization	Both 2D- and 3D-data can be linearized to a vector. The values will be taken out of the data in row-major order.
Penetration depth	the distance that light enters the tissue
Pixel	Picture element
Rigid transformation	Rotations and translations.
Slice	An image as result of a cut of a plane with a 3D volume. Most common are slices perpendicular to one of the axes.
Transfer function	Mapping between voxel intensities and colors. E. g. voxels with higher intensity value can be displayed red and other voxels blue.
Volume Rendering	Rendering method able to visualize volumetric datasets. A transfer function can be used to improve the visualization.
Voxel	Volume element

Bibliography

[1] R. R. ALFANO and J. G. FUJIMOTO, *Advances in Optical Imaging and Photon Migrations*, OSA Trends in Optics and Photonics Series, 2, 1996.

[2] B. CHANCE and R. R. ALFANO, *Optical Tomography, Photon Migration and Spectroscopy of Tissue and Model Media: Theory, Human Studies and Instrumentation*, SPIE Proc., 2389, 1995.

[3] B. CHANCE and R. R. ALFANO, *Optical Tomography and Spectroscopy of Tissue: Theory, Instrumentation, Model and Human Studies*, SPIE Proc., 2979, 1997.

[4] W. D. J., *The history of Werner Spalteholz's Handatlas der Anatomie des Menschen*, Journal of Audiovisual Media in Medicine, 22(4), 1999, pp. 164–170.

[5] J. HAJNAL, D. HILL, and D. HAWKES, *Medical Image Registration*, CRC Press, 2001.

[6] D. MED. ET LL. D. WERNER SPALTEHOLZ, *Über das Durchsichtigmachen von menschlichen und tierischen Präparaten und seine theoretischen Bedingungen*, S. Hirzel Leipzig Verlag, 2nd extended ed., 1914.

[7] G. MLLER, B. CHANCE, R. ALFANO, S. ARRIDGE, J. BEAUTHAN, E. GRATTON, M. KASCHKE, B. MASTERS, S. SVANBERG, and P. VAN DER ZEE, *Medical Optical Tomography: Functional Imaging and Monitoring*, SPIE Institutes for Advanced Optical Technologies Series, IS11, 1993.

[8] S. A. PRAHL, M. KEIJZER, S. L. JACQUES, and A. J. WELCH, *A Monte Carlo Model of Light Propagation in Tissue*, SPIE Institute Series, 5, 1989.

[9] L. RADE and B. WESTERGREN, *Mathematics Handbook for Science and Engineering*, Springer, 5th ed., 2004.

[10] WOLFGANG WEIN, *Intensity Based Rigid 2D-3D Registration Algorithms for Radiation Therapy*, Master's thesis, Technische Universitt Mnchen, 2003.

[11] A. YODH and B. CHANCE, *Spectroscopy and Imaging with diffusing light*, Physics Today, 48, 1995.