Otto-von-Guericke-Universität Magdeburg

Masterarbeit

Jacobian Optimization with Multi Scaled Elastic Registration using Decomposition by the Jacobian Nullset.

Stephan Fischer
Immatrikulationsnummer: 193843
step.fischer@gmail.com

Betreuer: Dr. Dipl-Ing. Dirk Joachim Lehmann

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum Unterschrift

Contents

1 Introduction

Image registration is a very important field for applications in automotive, medical sciences, surveillance, robotics and many more. An universal approach to register data of various sources can be used in a wide range of industries, e.g. comparing data from MRI and CT images. This could make it easier to get information about a data set, which may not be found without comparing different data. To achieve this vision, this thesis discusses the idea of using the nullset of the determinant of the Jacobian matrix (short: Jacobian), which is treated as a equality measurement of data. For instance is the determinant of a 2x2 matrix equal to the area described by the row vectors of this matrix. The Jacobian matrix is created by filling such a 2x2 matrix with the first order derivatives of the underlying data. Now, to make a statement about the similarity of the data, we compute the determinant of the Jacobian matrix, which results in the Jacobian. If the Jacobian equals to zero, we treat the data as equal, although the first order derivatives at the data point is pointing in the opposite direction. Every other value than zero is counted as unequal. These are the data points we want to optimize by increasing the equality of the two data sets. For sake of complexity this thesis is only using 2D gray scale images as data sources. Using other data sources, like MRI or CT data sets, is an issue to be covered by future research. Because of the invariance of the Jacobian against noise, I will apply two kind of image filter, a Gaussian blur and an Anisotropic Diffusion filter. The Gaussian blur will ensures that the information of the Jacobian will be spread over the data set. The Anisotropic Diffusion filter, the Perona-Malik filter, will highlight features to be matched by the algorithm. Elastic registration is one of the state of the art approaches. It's based on the idea of converting the data set, here the 2D images, into a mesh with springs connecting neighboring nodes. These springs are applying a force to the mesh, which than behaves like a sheet of rubber and can be deformed in an non-linear manner. Furthermore, I will propose a regularization by distance and amount of information of the nodes movement. The experiments will show that the original algorithm and variations of it are producing satisfying, yet perfect registration results.

1 Introduction

The following chapter is introducing the related work this thesis is based on and will present some state of the art methods. Afterwards, I describe the algorithm used to get the results and insights discussed in the corresponding chapter. Finally a conclusion and an outlook for future work is summarizing this thesis.

2 Related Work

This chapter describes the image registration in general and the common histogram based methods and elastric registration which is part of the transform model estimation methods. Later, I will explain what continuous histograms are and introduce two methods, Non-Parametric Windows and Continuous Scatterplots, which are used for this thesis. The following section is based on the survey about image registration methods provided by Zitová and Flusser [11].

2.1 Image Recognition

Image recognition (IR) is a wide field in science. It is used in many applications in our all-day life. For example in the industry, image recognition is used to check the quality of products surfaces like car parts. In medicine IR is used for automated processing of medical images acquired by MRI or CT devices. In companies with secured areas IR can be used for face detection to protect them by trespassing of unauthorized personal. The next sections are describing IR in general and explains the common histogram-based methods as well as the more advanced transform model estimation methods.

2.1.1 General

Image registration has a wide variety of methods depending on the way how the data is acquired, the conditions, the used devices or methods, the prior knowledge about the content and which features should be extracted. The chosen methods depends on the later use of the results, too. Whether it is in industry, like constructing aerodynamic vehicles or in medicine for preoperative exploration of data sets from CT or MRI. Basically, image registration is structured in four steps for the majority of the methods. Feature detection,

which is the manually or automatically marking of striking objects in the picture (like edges, lines, sudden differences in the data etc.). The second step, feature matching, creates the correspondence between two images (sensed and reference image). Transform model estimation, the third step, is the step where the parameters of the mapping functions are estimated. Last but not least step four, image re-sampling and transformation, transforms the sensed image by the mapping function. The result should be a perfect matching of a sensed and reference image.

2.1.2 Histogram Based Registration

Histogram based methods in image registration are merging the first two steps, feature detection and matching, into one. This can be done, because the histogram based methods doesn't need to detect salient objects. The limitations of this method, if using the common implementation, is the use of rectangular windows to register images that only differ locally by a translation. If the images are transformed more complex, the window isn't able to cover the same scene in the reference and sensed image. Another problem is the recognition of smooth areas with no distinct details. Here, the matching will probably fail due to incorrect classification of a smooth area in the reference image to another one in the sensed image, which is not representing the same part of the image. The following methods are trying to avoid these mismatches by using different approaches.

Correlation-like Methods

The normalized cross-correlation (CC) exploits image intensity and searches for the maximum similarity of images from window pairs. The windows with the maximum similarity are treated as corresponding ones. This method isn't only capable to align mutually translated images, but can applied to slightly rotated and scaled images. General CC is capable to detect more complex geometrically deformed images by assuming every possible deformation. Some implementations are able to detect even affine transformations. For multi-modal registration the similarity measurement is done by using different sensors, which suppose to express the intensity by a function. Notwithstanding, it has drawbacks resulting from the flatness of the similarity maximum measurement and the

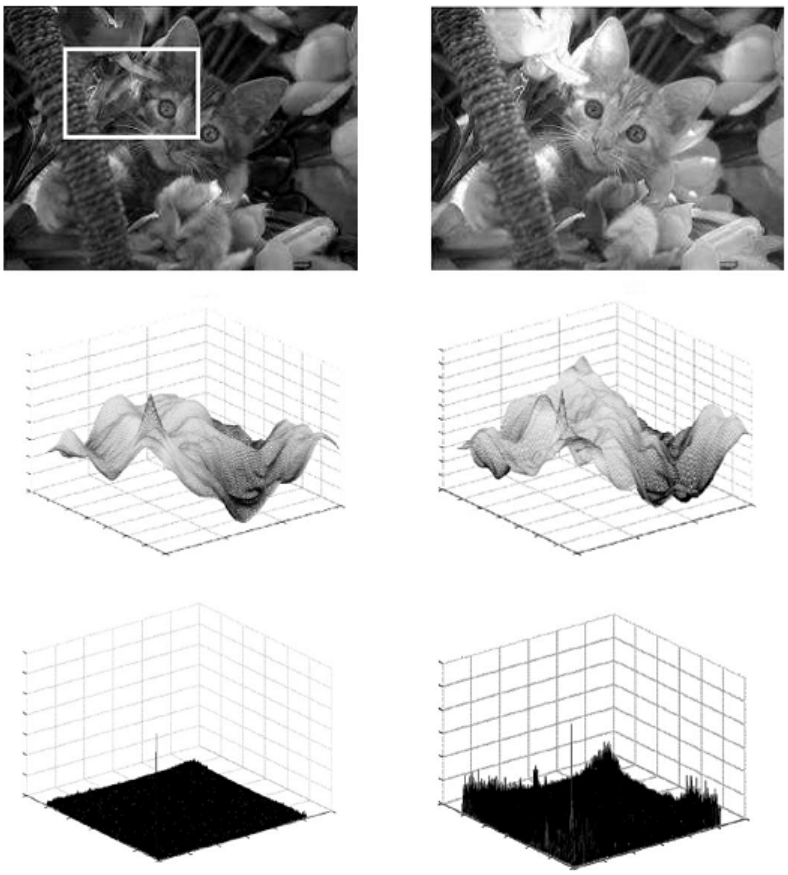


Figure 2.1: Comparison between CC and PC. Sample images of the same scene under different conditions (upper row) with template in the reference image (left). Registration with normalized cross-correlation (middle row) and phase correlation (bottom row). Red-red channel matching (left column) and red-blue channel matching (right column). [11]

high computational complexity detecting more complex deformations. The correlation-like methods are often used due to the easy hardware implementation and thus use for real-time applications.

Fourier Methods

Fourier methods, another histogram based method, are preferred instead of the correlation-like methods, if the images are acquired under varying conditions or are containing frequency-dependent noise. Working in the frequency-domain makes it possible to find transformation parameters for registering two images. This works for simple transforma-

tions like translation, rotation and scaling [1]. By using the phase correlation (PC) of two images, where one is a translated version of the other, it produce a third image which show a single peak representing the relative translation of the images [2]. The savings of computation time is increasing along with the size of the images.

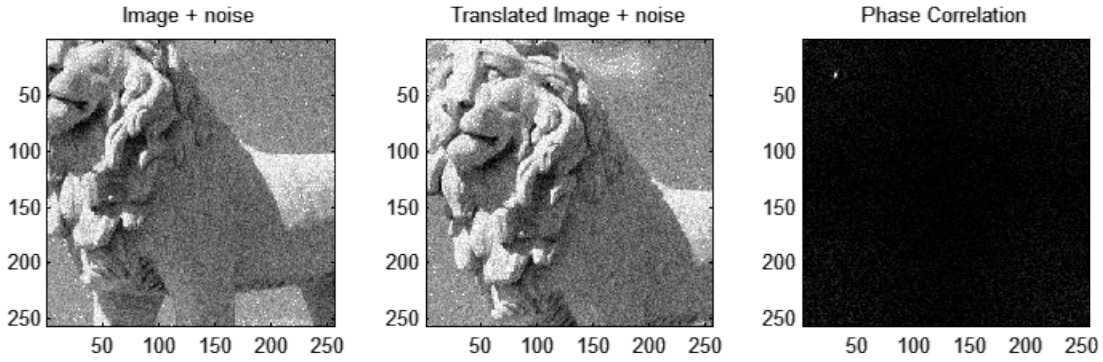


Figure 2.2: Example for phase correlation with the reference image (left) and the translated image (middle). The third image, the produced phase-correlation representation, shows a peak at (30,33), which is the translation. [2]

Mutual Information Methods

Mutual information (MI) methods is the leading technique in multi-modal registration used e.g. in medical imaging. MI is a measure of statistical dependency of two data sets and is computed from the random entropy of the probability distribution. The whole idea of the method is based on the maximization of the mutual information, which can be speed up by a coarse-to-fine strategy. Finally it can be executed on the whole image data and works with the intensities directly.

2.1.3 Transform Model Estimation

The transform model estimation step is constructing a mapping function, which should transform the reference image to fully cover the sensed image. This contains choosing the right mapping function and the right parameter respectively. At the same time the

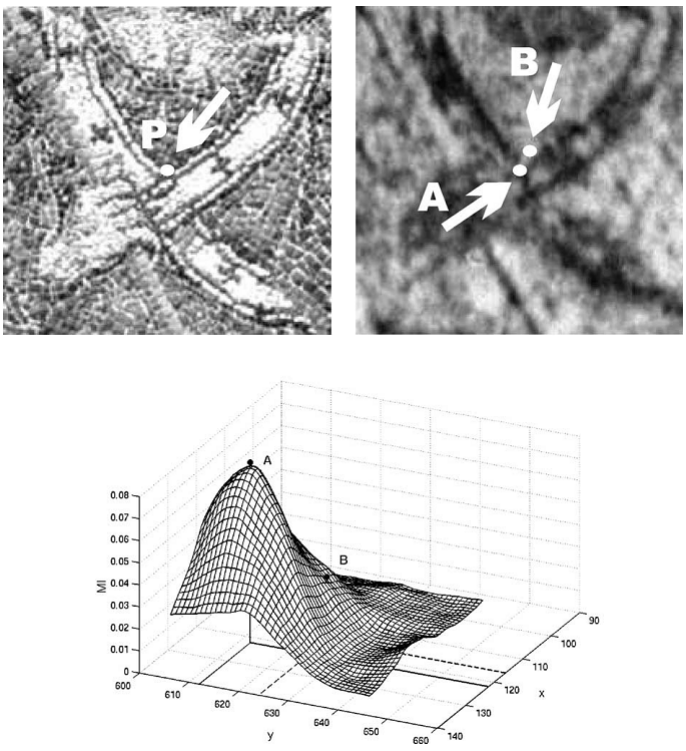


Figure 2.3: Example for Mutual Information. Upper row: Two images of the same scene in different quality. Point A: The correct matching position to point P. Point B: The chosen point by a human. Lower image: The calculated mutual information of the two images around the point P. [11]

mapping function should match the assumed geometric deformation of the matching image.

Elastic registration

The approach of elastic registration is not to use any parametric mapping function, but to find the best parameters for a geometric deformation. Therefore the images are viewed as some kind of elastic material like rubber. The registration tries to stretch the image into alignment by locating the minimum energy state. The advantage is that this method combines the feature matching and mapping steps into a single one. This can be achieved in a simultaneously and/or iterative manner as well. Like I mentioned before at the mutual information method, the elastic registration can be done in a coarse-to-fine way, too.

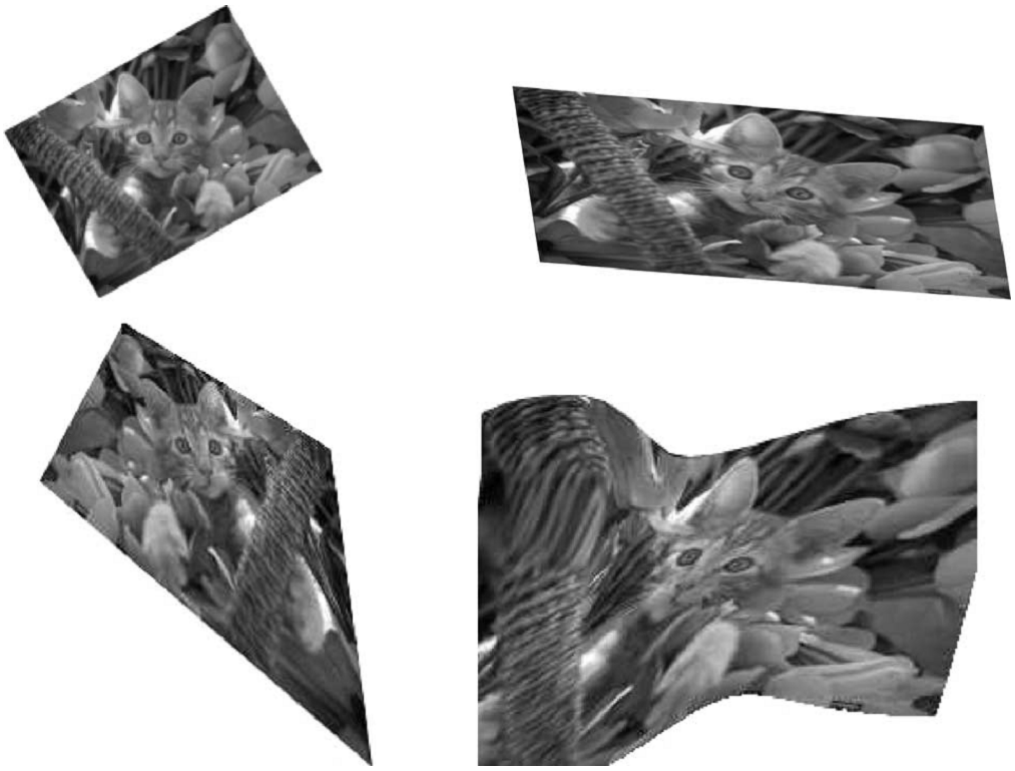


Figure 2.4: Examples for mapping functions. Top left: similarity transformation. Top right: affine transformation. Bottom left: perspective projection. Bottom right: elastic transformation. [11]

2.2 Continuous Histograms

A histogram in statistic is a method to visualize a distribution of data over a certain domain. In sense of images as underlying data a histogram is commonly used to count the number of occurrence of an intensity value within an image. Because of the discrete resolution of a histogram, continuous histograms are often use to visualize a distribution in a gap-free manner.

2.2.1 Non-Parametric Windows (NP-Windows)

Non-Parametric Windows (NP-Windows) Method, by Kadir and Brady [7], is a way calculating probability density functions (PDF) of sampled signals by representing each piecewise polynomial section of the interpolated signal by an expression corresponding to the interpolation. That means that they're not up-sampling the signal by an infinite factor,

but instead using the mathematical approximation to calculate the PDF. This method was first proposed in [7] for 1-D and 2-D signals and than used with mutual information for image template matching in [5]. This approach uses the NP-Windows to calculate the mutual information of the images by creating intensity polygons, which is just computing the PDF with respect to the neighborhood described by the images, to render them into an histogram after the overlapping polygons are combined and re-meshed.

2.2.2 Continuous Scatterplots (CSP)

Bachthaler and Weisenkopf [3] purposed a model to visualize correlations of two (discrete or continuous) data sets into one continuous scatterplot (CSP). These CSPs are a development of scatterplots, which are used in statistics for information visualization. Likewise common scatterplots, which takes two sets of discrete data points as input, CSPs are using those point collections as a data field defined in a continuous domain. The output provided by CSPs is a plot, which visualize the relationship between the data sets in a continuous and dense manner. Bachthaler at al provided a mathematical model to compute generic CSP with independence of the input dimension, which in common cases are two to three dimensional (plus one dimension if it's time-dependent), and doesn't need any input parameter. Furthermore, there is a restriction to the input, because both data sets have to have the same dimension size. In [3] they present cases where the input domains and the domain of the CSP are both equal as one is greater than the other, too. In addition they provided the mapping function from the spatial domains, where the input data points are defined, and the data domain, where the CSP is defined. Lehmann and Theisel [8] researched discontinuities in continuous scatterplots, in the case, where both spatial and data domain has a dimension of two. These discontinuities are described as a sudden event in a non-continuous way and can be classified in two different types, Boundary Curves and Critical Curves. The first type is formed by the bounds of the spatial domains. Due to the fact, that they're marking the end of the sampled signals, these curves are not as interesting as the other type. Critical curves are formed by the nature of the mathematical model of the CSP. They are consisting of all points in the null set of the CSP mapping function. Those curves are marking regions where the density is infinite.

2.3 Spring Lens Model

Spring Lenses provided by Gremer et al [4] is a method to apply non-linear transformation to images. The main idea consists out of mapping an image to a grid, where every node is connected with a spring to each of its neighbor. These springs are applying a certain force to their connected nodes like the real world model. By computing the overall force within the grid, each spring is contributing to the transformation, which makes it possible to describe complex distortions of an image. Furthermore the model is capable to assign constrains to nodes to fix there positions within the grid and thus applying a transformation in a certain region without running out of its boundaries. The main advantage of this method is the prevention of overlapping of regions by prohibiting folding of the surface by definition.

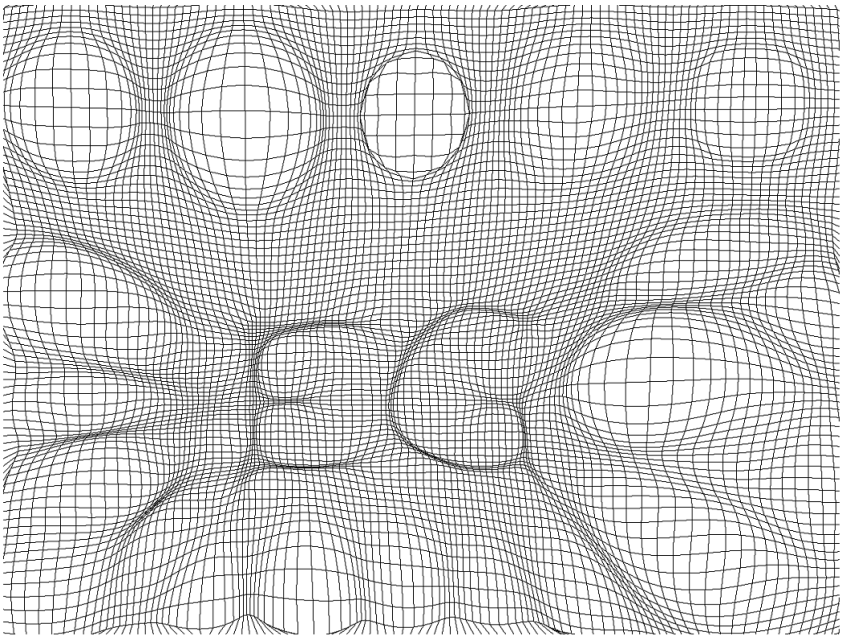


Figure 2.5: Distortion of a grid by the spring lens method.[4]

3 State of the Art Methods

Nonrigid Medical Image Registration Based on Mesh Deformation Constraints by Lin et al [9] is a method to minimize the sum of squared differences of two medical images. The method uses a triangulated mesh with a regularization constraint (a spring analogy method) to perform a nonrigid medical image registration on MRI images of a human brain. The spring analogy is treating the connections between the mesh nodes as a deformable spring. Like a spring in real world, this model is defining a term, which is the reciprocal of the nodes distance, to describe the force of a compressing spring. That means that the closer the points are, the force between them is increasing and thus it avoids overlapping of the mesh triangles and provides a better render quality of the final result.

Yin et al proposed an algorithm based on a cubic B-spline-based hybrid registration framework that is using landmark information with intensity patterns in [10]. They are combining landmark-based and intensity-based matching to register large deformation of CT lung datasets between 90% and 20% of vital capacity. With a multi resolution approach using a cubic B-spline function as basis, the algorithm starts on a coarse level and becomes more detailed over the calculation time of the registration. The intensity-based matching is achieved by applying a minimization of the sum of squared intensity differences. The landmark-based matching is based on an initial configuration of the B-spline control points which is matching corresponding ones from two different sets of landmarks.

Diffeomorphic registration of images by Janssens et al [6] is based on the idea of a nonparametric registration with a multiscale approach. The presented method utilizes a phase-based approach called Morphons to perform a registration based on three steps (Field computation, accumulation and regulation) and extend it with diffeomorphic behavior. Morphons are matching transitions between dark and bright regions and not intensities, like the common method nowadays. This makes the algorithm more robust against contrast enhancement of the original data sets.

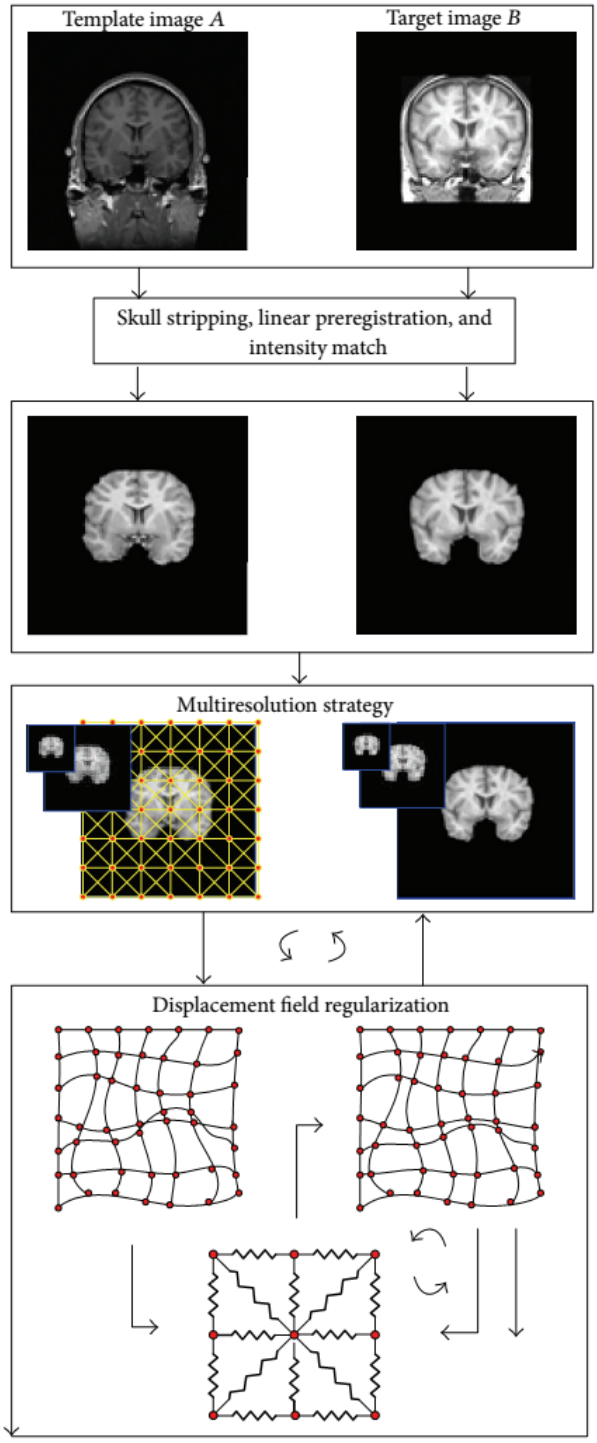


Figure 3.1: Overview of the Nonrigid Medical Image Registration Based on Mesh Deformation Constraints algorithm. [9]

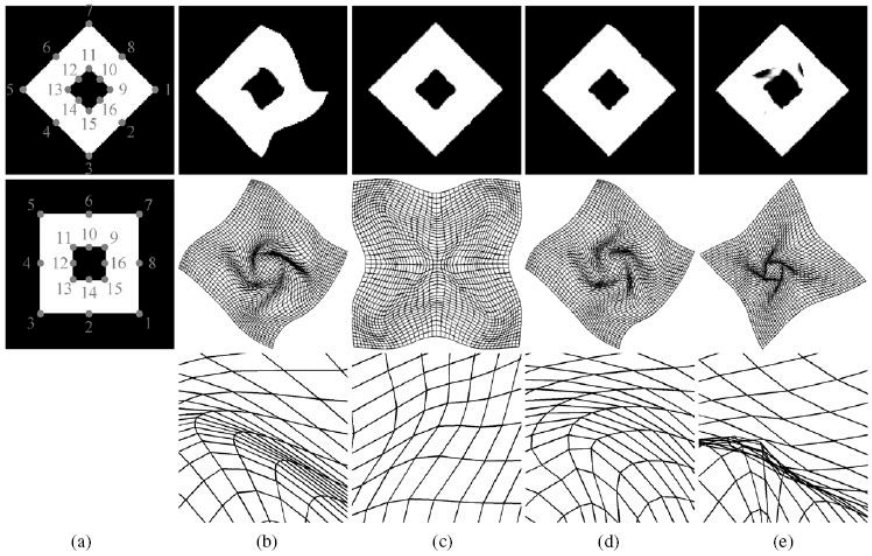


Figure 3.2: The cubic b-spline-based registration results. Column (a) is showing the correspondence of the landmarks. The columns (b)-(e) are showing the reference image used for matching (top), the deformed grid (middle) and a close up view of the deformed grid (bottom). Column (b) is just the intensity-based matching applied, (c) the landmark-based matching applied,(d) both matching methods and (e) shows the results of a thin-plate spline based algorithm. [10]

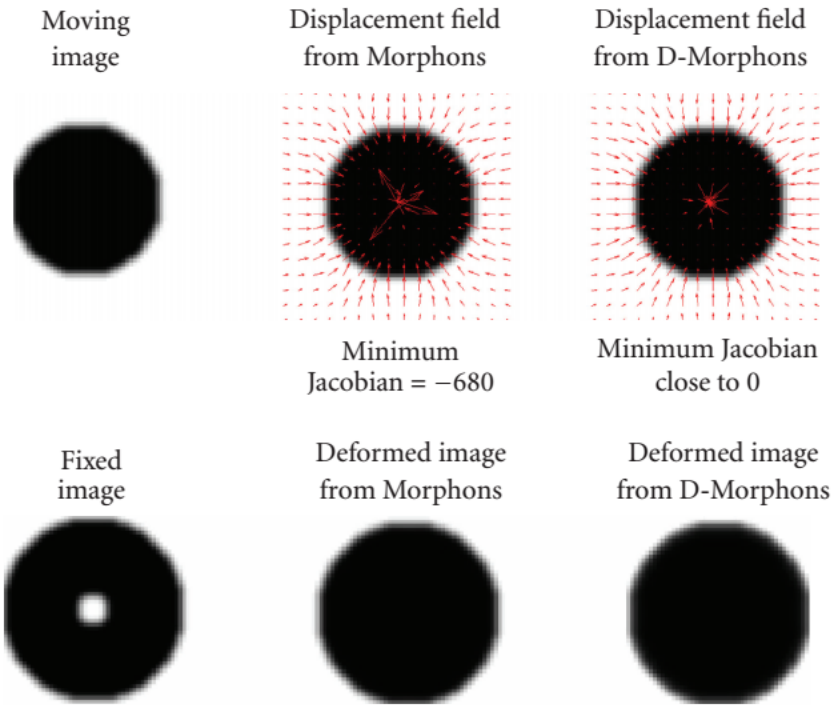


Figure 3.3: Registration example of the morphons and diffeomorphic morphons proposed by [6].

4 Jacobian Based Optimization

In this chapter I like to introduce an elastic registration of two data sets (here the data sets are two two-dimensional gray scaled images) by optimizing the determinant of the Jacobian matrix (called Jacobian). Based on discontinuities in continuous scatterplots, the back projected feature lines are used as an initial model for an elastic registration with a simple regularization method for the node movement inspired by [9] and [4] with respect to the information amount (the length of the gradient at an element in the data set). Due to the fact that the Jacobian is very noise depending, I use two filtering methods to improve the image quality prior to the optimization process. After this noise handling, the Jacobian is calculated in each element pair and the resulting distribution can be used to construct feature lines, which forms a decomposition of the two data sets used for a local optimization. The aim of the algorithm is to match features on the data sets solely by optimizing the Jacobian.

4.1 Noise Handling

Due to the fact, that the Jacobian is very noise depending, the noise handling is an important task prior optimizing the Jacobian. The noise in an image can result from various sources. Does the image contain artifacts produced by the compression (like for .jpg or .png images) and noise, which is the result of the measurement conditions (e.g. quality of the CCD-Chip in a camera, light conditions), it distorts the information in the image and thus provides a wrong database for the algorithm (figure 5.2). In this thesis I am experimenting with a Gaussian Blur, an Anisotropic Diffusion Filter (also called Perona-Malik Filter) and a combination of both.

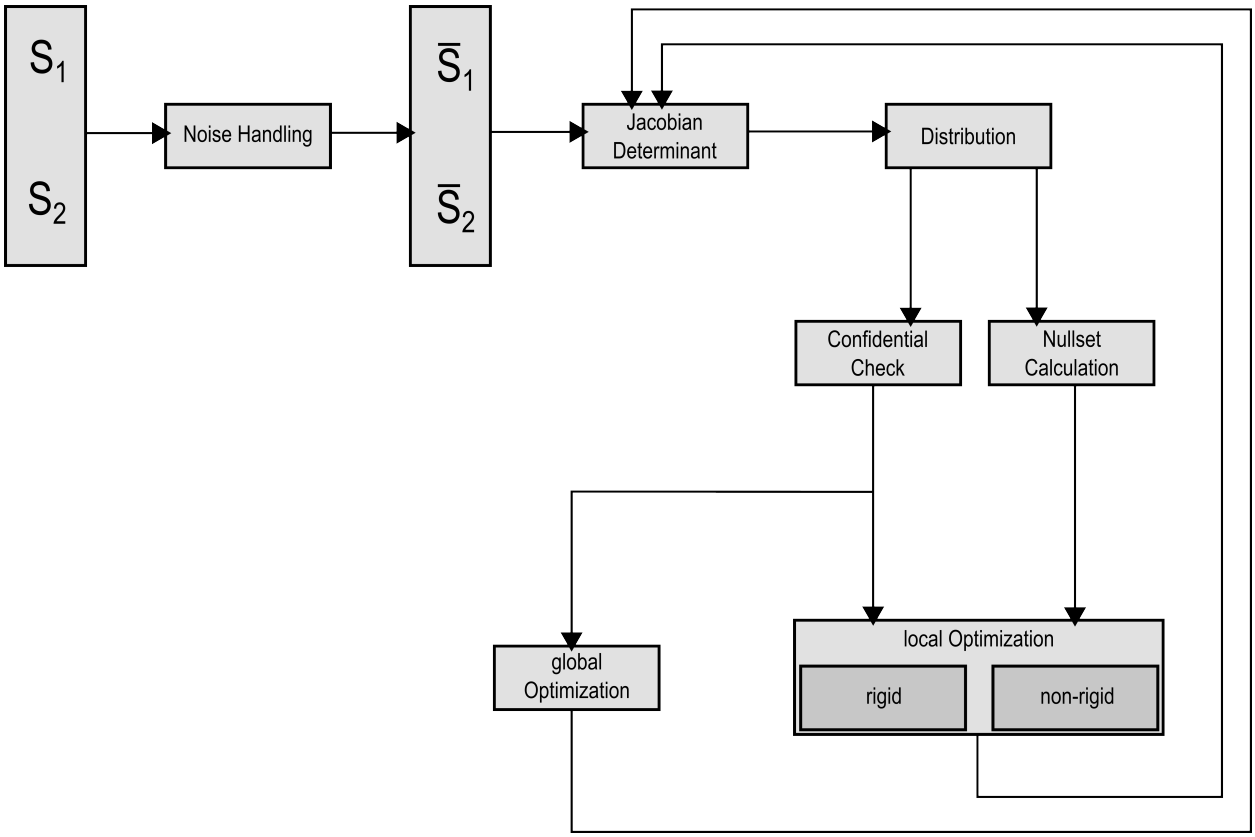


Figure 4.1: The whole algorithm at a glance. The two images S_1 and S_2 are preprocessed by the "Noise Handling". The next step is to calculate the Jacobian distribution and the Nullset of it (only for local optimization). If the system isn't failing the confidential test, one optimization step is calculated. The whole process starts again by updating the Jacobians. For the non-rigid local optimization, the Nullset is updated before every step. One optimization cycle ends if the confidential check fails.

Gaussian Blur

The Gaussian blur, or Gaussian smoothing, is a low-pass filter which uses a Gaussian function and reduces the high frequencies in the data. Points with a very high frequency, which is treated as noise, can't find a place on the reference data set to optimize their Jacobian and will "wander" through the whole domain. Applying a Gaussian Blur will carry along the pixel information over the domain and not only reduces the noise, it gives the data points of the matching image the possibility to increase their movement radius, too.

Anisotropic Diffusion

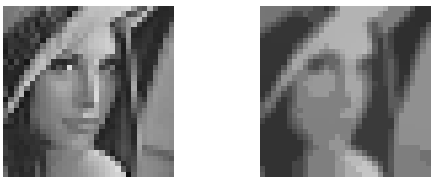


Figure 4.2: Right: The original image. Left: After applying the Perona-Malik Filter. You can see that it produces homogeneous regions and hard edges.

The anistropic diffusion (also called Perona-Malik filter) is a method for smoothing an image without removing significant parts, like edges, from the data. In regions, with moderate noise, the Perona-Malik filter behaves like a Gaussian blur (figure 4.2). At edges, which has a higher frequency than the surrounding, the method has an anisotropic behavior, which keeps this "gap" in place and doesn't smooth it like in homogeneous regions. This preserves significant features and produce plateaus in the data set. Another side effect of these plateaus is that Jacobians of an element on a plateau are zero and thus marking regions of elements without information.

The Combination of Gaussian Blur and Perona-Malik

The combination of both, the Gaussian blur and the Perona-Malik filter, is one of the main issues I want to address, before I start talking about acquiring the prerequisites

for the optimization database. Like I mentioned before, the Gaussian blur distributes the information over the data domain. The Perona-Malik filter is creating plateaus with a nearly constant intensity and sharp edges between them. By first applying a strong Perona-Malik filter on the data set and smoothing the result with a Gaussian blur, the optimization will use a database with removed information like small differences in an actually smooth region and stretched edges (figure 4.3).

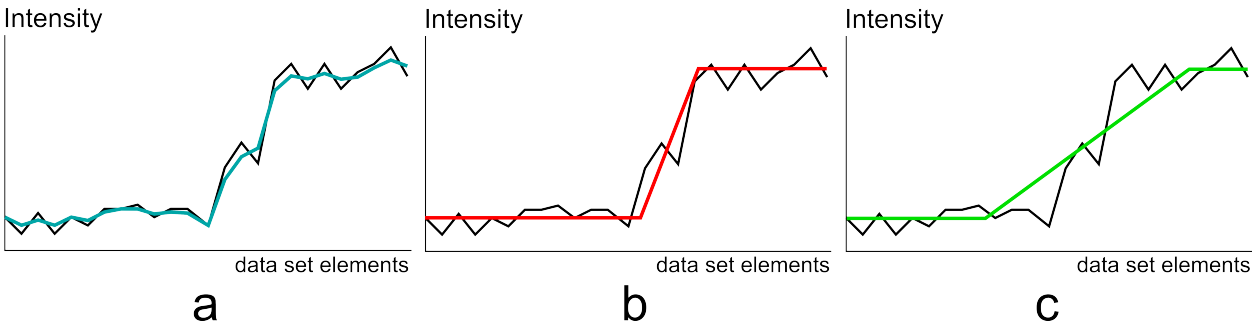


Figure 4.3: Illustration of the used smoothing functions. The black curve is the signal of the data set. a) The blue line shows how the signal looks after smoothing with a Gaussian blur. b) The signal (red line) after applying the Perona-Malik filter. c) The green line is showing the result after applying both filter after another. First the Perona-Malik filter is creating a hard edge and plateaus. Second the Gaussian Blur is stretching the edge over the domain.

4.2 Jacobian Calculation, Distribution and Decomposition

At first, I will describe the common way the Continuous Scatterplot created and how the feature lines are constructed. But due to the reason of the time consumption, the last section describes the way the algorithm is constructing the feature lines for the decomposition.

Constructing the Continuous Scatterplot

Continuous scatterplots (CSP) are first proposed by Bachthaler and Weiskopf in [3]. Based on the idea of discrete scatterplots (DSP), CSPs have the advantage that the visualization of data correlation is plotted in a continuous manner, rather than just populating the plot

with a finite number of data points. In [3] a CSP is defined as a mapping of a n-dimensional input field, called spatial domain, onto a m-dimensional domain of the scatterplot, called data domain. I will use $n = 2$ and $m = 2$, because the data base consists out of two 2-D images that will be mapped on a 2-D CSP.
Assume we have two data sets $\mathbf{S}_1, \mathbf{S}_2$ with $\mathbf{S} \subset \mathbb{R}^2$ and the function value $\xi = \tau(\mathbf{x}) \in \mathbb{R}$ at an element $\mathbf{x} \in \mathbf{S}$ of the spatial domain. The first step is to compute the gradients $\mathbf{G} = \nabla(\mathbf{S})$ with $\mathbf{g} = (\xi_x \quad \xi_y) \in \mathbf{G}$ of the data sets $\mathbf{S}_1$ and $\mathbf{S}_2$, were x and y are the first derivatives of the proper axis. The resulting scalar fields $\mathbf{G}_1 = \nabla(\mathbf{S}_1)$ and $\mathbf{G}_2 = \nabla(\mathbf{S}_2)$ are than used to construct the Jacobian matrix

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{g}_1, \mathbf{g}_2) = \begin{pmatrix} \xi_{1x} & \xi_{2x} \\ \xi_{1y} & \xi_{2y} \end{pmatrix}$$

and to compute the Jacobian (determinant of the Jacobian matrix):

$$\mathbf{d_x} = \mathbf{det}(\mathbf{J}(\mathbf{x}))$$

For constructing the surface forming the CSP we use the polygonal approach which is proposed by [7]. Therefore we will triangulate the surface and define vertices of the triangles as:

$$\varphi(\mathbf{x}) = \begin{pmatrix} \tau_1(\mathbf{x}) \\ \tau_2(\mathbf{x}) \\ \mathbf{det}(\mathbf{J}(\mathbf{x})) \end{pmatrix} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \mathbf{d_x} \end{pmatrix} \in \Phi$$

for each element of $\mathbf{S}_1$ and $\mathbf{S}_2$. Thus we have a mapping $\varphi : \mathbf{S} \to \Phi$ from the spatial domain $\mathbf{S}_1$ and $\mathbf{S}_2$ onto the data domain Φ.
Caused by the nature of triangulation, the CSP surface isn't really continuous through the discretization of the data sets. But for sake of complexity, we could scale up the resolution of the spatial domain to produce a finer mesh for the CSP surface.

Discontinuities in Continuous Scatterplots

In [8], Lehmann and Theisel investigated types of discontinuities produced by the mapping function and classified them in two types of curves. Critical Curves, formed by discontinuities of the mapping function. Boundary Curves, resulting in the end of the signals (here the borders of the images). Thus that Boundary Curves are marking the borders

of the data sets, we will concentrate on constructing the Critical Curves (CC). Using the definitions we made in the last section, CCs are defined as

$$\bigcup \varphi_{d=0} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ d = 0 \end{pmatrix} \subseteq \Phi$$

From the definitions in [3] and [8] the CSP is an integral over the reciprocal of the Jacobians. But through the superposition of the surface, a Jacobian of zero will produce an infinite value at the positions the CCs are going through. So we are just looking at single values, which are all points of the CSP with a Jacobian $d = 0$ and thus a density of infinite.
To easily calculate the CCs of our triangulated CSP surface, we are just intersecting each triangle with the zero plane of the CSP. The lines that will be produced are forming the CCs in the data domain.

Feature Line Back Projection

While constructing the CSP surface with the polygonal approach, we are storing the origin pixel positions of the spatial domain along side with each triangle vertex. Using the origin pixel of the computed CCs lines, we are projecting them back from the data domain to the spatial domain. It shows that the CCs of the data domain are forming feature lines in the spatial domain. These lines completely enclosing coherent regions having a same-signed Jacobian. Neighbored regions are having different signed Jacobians or a Jacobian of zero, which defines a plateau in the data domain. We can compute the positions of the feature lines subpixel precise by increasing the resolution of the CSP surface. If we're now merging the boundaries of the data sets with the feature lines, we get a entire segmentation of both data sets.

Implementation

To save resources and to enhance the computation time, the algorithm doesn't use the method to get the feature lines as described above. Instead, I am calculating the determinant distribution at the beginning of a new time step by checking every node-to-node-connection for a change of the sign of the determinant and the corresponding intersection

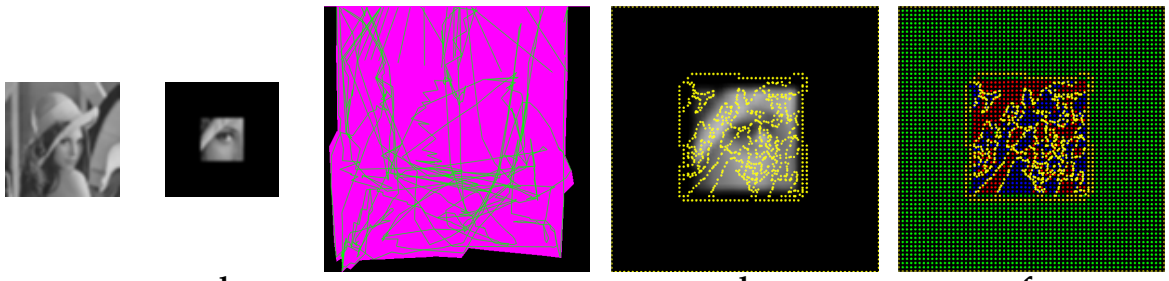


Table 4.1: An example of a CSP (**c**) constructed from a reference image (**a**) and a matching image (**b**), the back projected feature lines (**d**) and the corresponding Jacobian distribution (**f**). The green lines are the CCs and the magenta colored surface is the CSP. The yellow dots in **d** are the intersection points of the feature lines on the node connections. As you can see, the feature lines are forming enclosed regions in **f**. Note that some points have been already optimized and have changed the sign of their Jacobian.

point. The next chapter will show, how the segmentation is used for image registration.

4.3 Confidential Check

The idea of the confidential terms is to set a lower bound while the optimization is valid in meaning of optimizing the signal and not the noise. During the experiments, I used the sum of absolute determinants (SAD), the median of the determinants (MD) and the median of absolute determinants (MAD). It showed that every sample converges against different bounds of the three condifential terms. Therefore the resulting confidential check is relying solely on the oscillation of the SAD. While the SAD is varying around a certain value, the nodes are still moving in a way that the SAD is still decreasing itself, caused by nodes which getting closer to their optimum position. Once the system is oscillating around a certain SAD value, it's counting the number of times it is increasing compared to the last iteration step and if its exceeds a certain limit the optimization will stop.

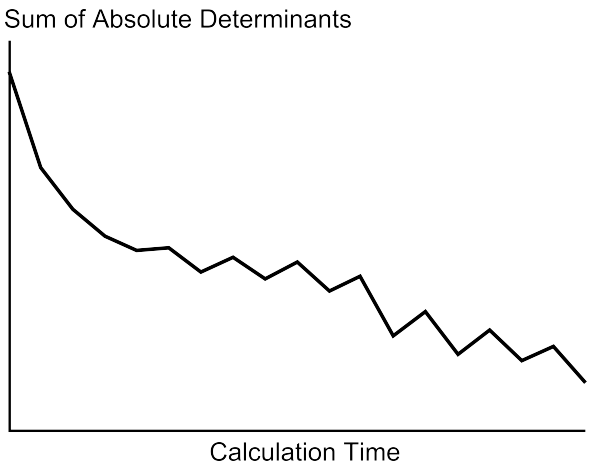


Figure 4.4: A schematic example for the oscillation. The graph shows an continuous decreasing of the sum of absolute differences and thus of the determinants.

4.4 Optimization

4.4.1 Optimization Algorithm

By definition, the determinant describes the volume of the parallelepiped spanned by the column vectors of a matrix. If this volume is equal to zero, all column vectors are linear dependent. Concluding from this definition, all gradients of both data sets at the feature lines are linear dependent, because of a Jacobian equal to zero. The idea of the registration process is based on the assumption that those positions are supposed to be matching.
To realize an elastic registration based on the optimization of the Jacobian, I construct a mesh $\mathbf{M}$ out of the data set $\mathbf{S}_2$. Every node $k \in \mathbf{M}$ corresponds to an element $\mathbf{x} \in \mathbf{S}_2$. Every k gets the initial position $\mathbf{p}_k(0) = \mathbf{x}$, the Jacobian $\mathbf{d}_k(p_k(t))$ at the position $\mathbf{p}_k(0)$, the gradient $\nabla_k(\mathbf{x})$ and a velocity $\mathbf{v}_k(t)$, which is initially the zero vector $\mathbf{0} = (0,0)^\top$. Starting from step $t = 0$ we're computing $t + 1$ as

$$\mathbf{p}_k(t+1) = \mathbf{p}_k(t) + \mathbf{v}_k(t)$$

with

$$\mathbf{v}_k(t) = \mathbf{g}(\mathbf{p}_k(t)) + \mathbf{r}(\mathbf{g}(\mathbf{p}_k(t)))$$

where $\mathbf{g}(\mathbf{p}_k(t))$ is the gradient of the determinant calculated with

$$\mathbf{g}(\mathbf{p}_k(t)) = w_d \cdot h \cdot \mathbf{RK4}(\mathbf{p_k}(\mathbf{t}))$$

$\mathbf{RK4}(k)$ is the estimated gradient measured with the Runge-Kutta method of grade 4, w_d is a determinant based damping term defined as

$$w_d = \begin{cases} |\mathbf{d}_k(\mathbf{p}_k(t))| & if\ |\mathbf{d}_k(\mathbf{p}_k(t))| < 1 \\ 1 & otherwise \end{cases}$$

and a step size h.
$\mathbf{r}(\mathbf{p}_k(t))$, described in the next section, is the regularization of the node movement to prevent overlapping of the nodes, which improves the rendering of the registration result.

4.4.2 Regularization of the Node Movement

To preserve the topology of the mesh $\mathbf{M}$, we need to regularize the movement of each node during the optimization procedure. Using a weighted velocity approach, similar to [9], over the eight neighbor around each node, we can compensate the overlapping of nodes. If every node n, that is not on the border of $\mathbf{M}$, is connected to their eight neighboring nodes n_n, n_{ne}, n_e, n_{se}, n_s, n_{sw}, n_w, $n_{nw} \in \mathbf{N}$, we can regularize the movement of every node k by:

$$\mathbf{r}(\mathbf{p}_k(t)) = \frac{1}{\sum\limits_{n \in N} \mathbf{w}_n} \cdot \sum_{n \in N} \mathbf{w}_n \cdot \mathbf{v}_n(t)$$

where $\mathbf{w}_n$ is calculated as

$$\mathbf{w}_n = \mathbf{w}_{dist} + \mathbf{w}_{info}$$

Like Lin et. al in [9] I am using the proposed distance weight defined as

$$\mathbf{w}_{dist} = \frac{1}{|p_n(t) + p_k(t)|}$$

The experiments show, that a regularization solely by distances results in a restless movement of every node. By adding a weight $\mathbf{w}_{info}$, which is describing the amount of information of a node (the length $||$ of the gradient ∇ at $\mathbf{x}$) with

$$\mathbf{w}_{info} = \frac{1}{d_n} \cdot |\nabla(\mathbf{x})|$$

, the optimization prioritize important nodes with a significant gradient during the regularization and gives them the capability to hold their optimal position on the reference image, because the neighbored nodes can't push them back in a less optimal position.

4.4.3 Local Optimization

Like mentioned before, the back projection of the feature lines from the CSP onto the data sets is decomposing the spatial domains into regions, which consists of determinants of the same sign. By adding new boundary nodes to the neighborhood of every node, the algorithm can be parallelized to increase the calculation speed. That means that every region can be processed by one thread, but after one step, the proposed algorithm has to synchronize the node positions, if a non-rigid local optimization is used. This synchronization ensures the consistency of each region, because nodes can change the region affiliation or form a new region within an existing one. Every boundary node b

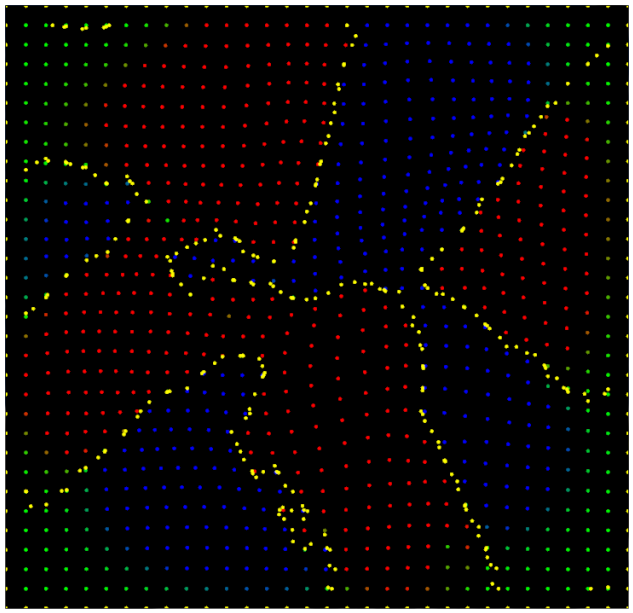


Figure 4.5: This example shows the boundary lines (yellow dots) as intersection points between the node point. The red color indicates a negative, the blue a positive and the green a near-zero Jacobian.

is treated as a normal node k. But unlike the node from $\mathbf{S}_2$, the boundary nodes have a velocity $\mathbf{v}_b = \mathbf{0}$ and a gradient $\nabla_b = \mathbf{0}$. The position $\mathbf{p}_b(t)$ is the respective feature line

intersection between the nodes of the spatial domain. The local rigid optimization uses the initial constructed feature lines as decomposition but isn't synchronizing the nodes after every time step. The local non-rigid optimization is updating the feature lines and is synchronizing the nodes after every time step. The effect of the update is, that the border lines are wandering with the moving nodes. The main advantage of the local optimization is a possible parallelization of the calculation process, because the borders are decoupling the connections between the node regions in $\mathbf{M}$.

4.4.4 Global Optimization

The global optimization works like the local one, but without the border points from the feature lines. The regularization of the node movements will take the whole data set into count and thus a parallelization can't be realized anymore. Notice that regions in the global optimization, unlike in the local optimization, can influence each other, caused by the direct connection of the adjacent nodes. This creates situations where one region can displace each other.

4.5 Multi Scale Approach

In the previous sections I described the algorithm from beginning to the final result by stopping the optimization due to a failing confidential check. The first step, before the Jacobian distribution is calculated, is the noise handling. While I was experimenting with the noise handling, Lin et al [9], who is scaling up the resolution of the image from coarse to fine during optimizing the SSD (Sum of Squared Differences), inspired me to extend the algorithm with a multi scale solution (figure 4.6). The Gaussian Blur is smoothing the image over the whole domain. All the information will be "mixed up", so every pixel gradient will be "carried" over a certain distance. This results in changing the process by adding a down scaling of the Gaussian Blur, to start a new optimization cycle with the calculated node positions $\mathbf{p}_k$ of the previous one, after the last confidential check has failed.

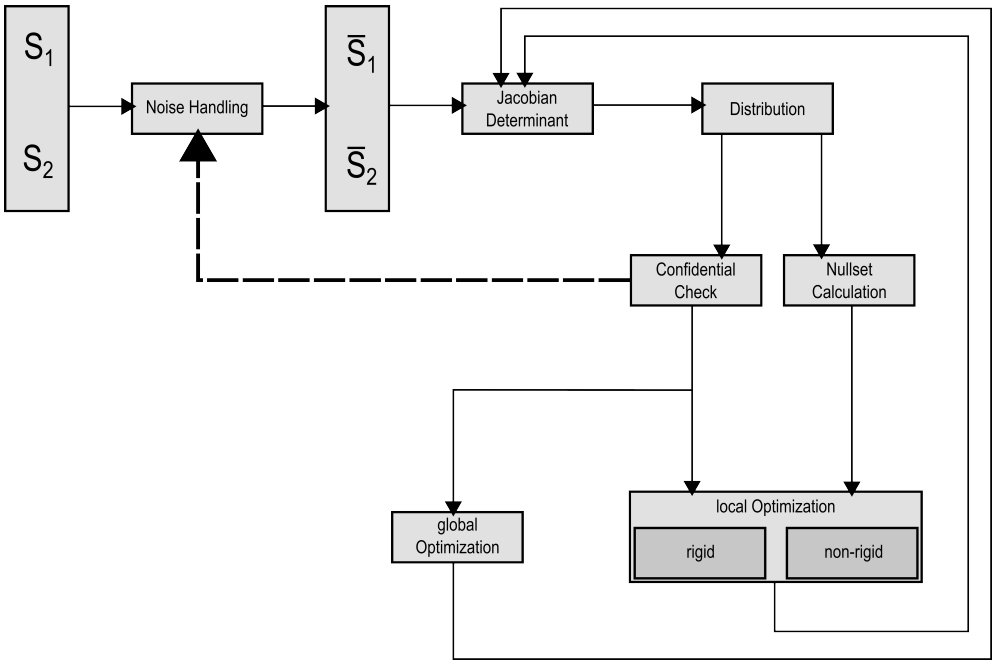


Figure 4.6: An overview of the extended algorithm. After the confidential test fails, a new optimization cycle starts with a decreased Gaussian Blur on S_1 and S_2. If the blur count reaches zero the optimization finishes. One variation of the algorithm is to just blur the reference image. This enables a matching of the original data (yet filtered with the Perona-Malik filter) and no distortion of the original feature distribution.

5 Results

This chapter is presenting the results I have got during the experiments with the proposed algorithm from the last chapter. All chosen example images aren't preprocessed in any kind. At first I talk about the overall performance of the algorithm. In the last two chapters, I discuss the main problems the method has to deal with. For sake of complexity I focus on the multi scale approach, because it creates more satisfying results. For all 32x32 pixel image pairs I used a conductance term of 0.2 for the Perona-Malik filter and a blur iteration of 32.

5.1 Overall Performance

The overall performance of the method is not satisfying, however we can construct examples that are producing the expected results. By turning of the regularization, all the nodes are moving ahead the optimum of their Jacobian and thus the sum of absolute determinants (SAD) is minimized to the global optimum. With regularization turned on, the points doesn't overlap in non-extreme situations. But it can be forced by increasing the sampling width of the velocity calculation. From table A.3 we can see that all approaches, which are not using multi scaling, produces nearly the same results, apart from the smoothing of the Perona-Malik filter. More interesting are the results with multi scaling. At first we can see that the nodes can overcome a greater distance away from their start position and thus has a better probability to get closer to an optimum. Second, the whole algorithm can optimize every level of detail for itself. The global approach is the simplest one and optimizes the Jacobian in a global manner. Thus every point influences all the other points due to the regularization, the whole grid is in a never ending movement. Just a handful of nodes can bring the system in a restless state, which results in a "kick-back" of nodes not found an optimum position. Both local approaches are decomposing the matching image in enclosed regions. The difference between them is the updating of the boundary. The

local rigid and the local non-rigid approach has nearly the same results. The difference between them is the update of the boundary lines, where nodes can change there region affiliation. The local rigid approach has the drawback that the regularization forces within a region can immobilize all nodes in it, because the boundaries are non-moving nodes and "pushing back" those nodes which want to cross the border. Depending on the size of such regions this results in a stop of the node movement and of the optimization process in this area. This is the advantage of the non-rigid local approach, if nodes near the nullset are moving towards the boundary, the border may displace and the region is moving as a whole.

5.1.1 The Jacobian

Another issue the algorithm is facing with, is the calculation of the Jacobian itself. If the determinant of the Jacobian matrix is calculated like mentioned in the section 4.2.1 the following situations producing valid results during the optimization:

1. If both gradients are pointing in the same or opposite direction.
2. If the length of the gradient in the reference image is smaller while the direction stays the same.
3. If the gradient in the reference image equals the zero vector $(0\,0)^\top$ (describing a plateau).

Number one is the condition defining the equality of the two nodes. The second point in the list results from the calculation of the gradient in the images. Because the gradients are not normalized to calculate the distribution of the Jacobian and the nullset itself, the length of the gradients are varying from zero to around 360 (due to the 256 gray scaled values). If one node is surrounded by nodes with the same gradient length, the algorithm will compute the direction of the node with best oriented gradient. If at least one node of the neighborhood has a smaller gradient than the other, the optimization will probability chose this one (see 5.1). The third point in the list, the plateaus, is discussed in the next section.

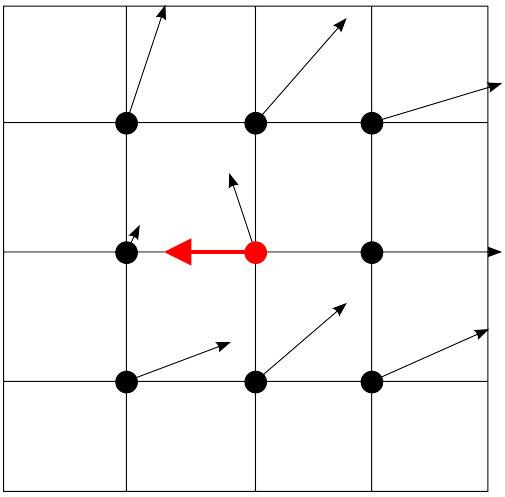
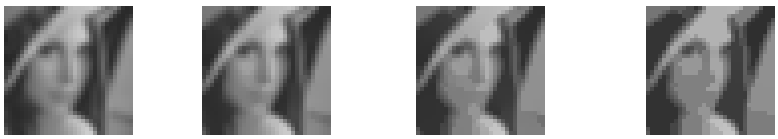


Figure 5.1: The node currently optimizing will move in the wrong direction (red arrow), if one node of his neighborhood has a significant smaller gradient length.

5.1.2 Plateaus

A plateau is defined as an area, where all gradients in each node are equal to the zero vector $(0\,0)^\top$. Likewise I am forcing the generation of plateaus with the Perona-Malik filter as a step to reduce the noise in the images, those plateaus are disadvantageous for the optimization process. Again, the calculation of Jacobian is the reason of this issue, because a zero gradient always resolves in a Jacobian $\mathbf{det}(\mathbf{J}(\mathbf{x})) = 0$. As you can see in the last column of Table A.1 all points has moved towards the forehead of Lenna, because the intense Perona-Malik filter generated a plateau (see the right column in Table 5.1), which attracted nearly all nodes of the matching image.



0 Iterations 10 Iterations 100 Iterations 1000 Iterations

Table 5.1: Example of the Perona-Malik filter smoothing an image and creates plateaus within it. You can see that the regions are getting more and more homogeneous while the iterations are increasing. The conductance term was set to 0.2.

5.1.3 The Regularization

The regularization of the node movement guarantees the preservation of the topology of the original sources and preserves the nodes neighborhood association to each other (nodes should never change their neighborhood relationship to each other). That means that the force, illustrated in [4] and [9], will increase as the distance of the nodes are decreasing, until they stop moving to prevent overlapping. The result is a better quality of the rendered results. The main drawback of this approach are points who increase their distances to each other will pull back their neighbor. In a situation a node has found an optimal position, where the Jacobian is at its optimum, the node may be pulled back towards a neighbor position. To compensate this behavior, I introduced a weight dependent damping term to prevent a light-weight node (one who hasn't found a optimum position yet) be pulled back by a heavier one. Another disadvantage is a push back of a node in optimal position performed by a heavy one, which has a small gradient but a low determinant. That leads to the introduction of the information damping term by taking the length of nodes gradient into count. But using all three damping terms for the regulation, the system shows unwanted behavior:

1. The distance that two nodes can have is limited by the movement force they evolve during the optimization.
2. Nodes near the image borders have a more restricted movement radius.
3. The whole movement is slowed down at a certain time in optimization and...
4. ... is restless if the majority of the nodes has found an optimum position.

5.1.4 Compression of the Source Images

The last issue I like to mention is the quality of the source images. Furthermore the image compression if one is used. While I was implementing the algorithm, I had to validate the correctness of some calculations like the Jacobian distribution of the spatial domains. I already mentioned the noise dependency of the Jacobian in the last chapter, where the image compression is a reasonable source of noise. Figure 5.2 shows an illustrative example, how image compression can break the calculation of the Jacobian distribution.

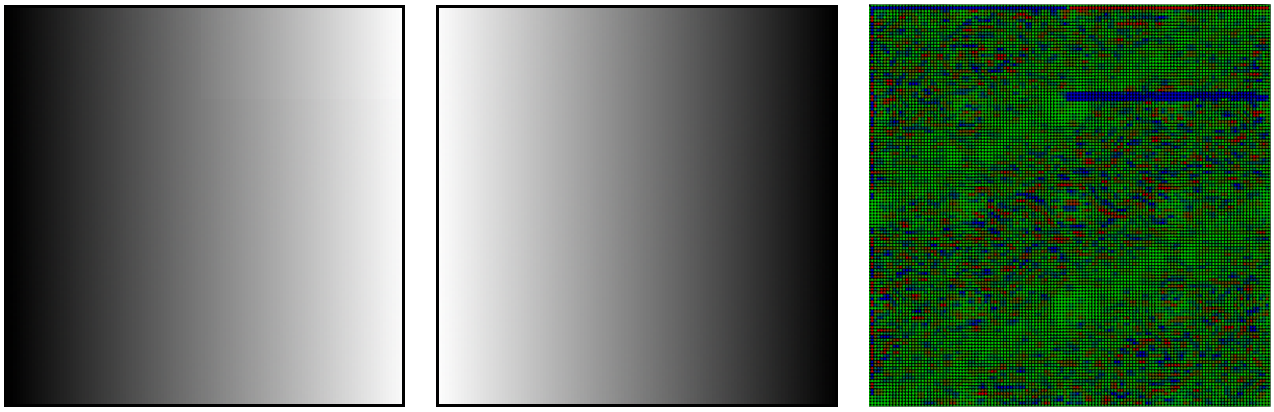


Table 5.2: Incorrect calculation of the Jacobian distribution (right image) due to image compression of the left and middle image. The black border around the left and the image in the middle is just for differentiating them from the white background.

5.2 Method Variations

The last section was about problems that the optimization is facing. Now i want to discuss two variations of the original procedure to show how the optimization behaves, if we change the processing chain with the knowledge from the experiments.

Blur Applied to One Source Image Only

The blurring of the image with an Gaussian smoothing is "mixing" up the information of every pixel. From the nature of the Gaussian blur, the length of the resulting gradients will decrease as the amount of blurring steps is increased. That means, that in the worst case scenario the gradients will vanish into zero vectors, because the whole image or an (probably small) area is one homogeneous surface. Referring to the Jacobian optimization, the algorithm can't solve the system anymore and the nodes will wander in one direction as one cluster, which later results in a wrong solution. Not only that both images has to be blurred, but the amount of blurring is crucial and correlates with the size of the data sets. See table A.4 and A.5 for an example.

Perona-Malik Applied to One Source Image Only

Applying the Perona-Malik filter to the reference image only is not feasible, because it is creating plateaus with all its disadvantages. But if we use a Perona-Malik filtered matching image and leaving the reference image untouched, we optimize just the remaining features. Depending on the image size, those features can push away the insignificant nodes forming the plateau, but as more and more nodes with a zero gradient are created, it is slowing down the movement of the whole system. Notice that the distance nodes can move is decreased, because of the removed blur on the matching image.

6 Conclusion and Future Work

This thesis showed how the determinant of the Jacobian matrix can be used for feature based image registration by optimizing the Jacobian based on the idea of discontinuities in continuous scatterplots. Using the Nullset of the Jacobian and thus a complete decomposition of the images, I described a method to match the contained features solely by moving them along the gradient of their Jacobian. To improve the renderings of the results, I applied a regularization of the node movement during the optimization process. This regularization is used for the boundary lines, formed by the nullset of the Jacobian, to realize a multi-threaded computation, too. But the used regularization, as a mixture of best fit and importance by the information held by the node, makes the whole system more restless the more nodes are moving. For more convenient results I introduced a multi scale approach to enable a longer movement distance for the nodes by optimizing from a coarse (very high Gaussian blur) to a more and more detailed data source. At least I showed a variety of results and described the most common problems the optimization with the Jacobian is faced with. The most crucial part of the algorithm is the non-uniqueness of the Jacobian and the mathematical nature, that a zero gradient always results in a zero determinant and thus attracting the nodes at most, which is indeed the wrong solution in most of the time. Another problem is the Perona-Malik filter which has to be adjusted for every image source again. The Gaussian blur has to be adjusted as well, to get the optimal balance between the maximal amount of "mixing up" the information and keeping them as significant as possible. I tried to deal with these problem by changing some of the steps in the algorithm. Leaving the matching image untouched by the Perona-Malik filter, I wanted to keep as most of the information to avoid the problems plateaus bringing along, for instance. Maybe an optimization of the Jacobian along side with a maximization of the information may produce better results and simplifies the regularization by using the spring analogy only. Notice that extending the method to 3D doesn't work, because the nullset in a three dimensional data set is a line too and thus no surfaces can be generated to decompose the domains into separated regions.

Bibliography

[1] Image registration. http://en.wikipedia.org/wiki/Image_registration.

[2] Phase correlation. http://en.wikipedia.org/wiki/Phase_correlation.

[3] BACHTHALER, S., AND WEISKOPF, D. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphic 14*, 6 (November / December 2008), 1428 – 1435.

[4] GERMER, T., GÖTZELMANN, T., SPINDLER, M., AND STROTHOTTE, T. Springlens – distributed nonlinear magnifications. *EUROGRAPHICS 2006* (2006).

[5] INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION. *Image template matching using Mutual Information and NP-Windows* (2006).

[6] JANSSENS, G., JACQUES, L., DE XIVRY, J. O., GEETS, X., AND MACQ, B. Diffeomorphic registration of images with variable contrast enhancement. *International Journal of Biomedical Imaging 2011* (2011), 3.

[7] KADIR, T., AND BRADY, M. Non-parametric estimation of probability distributions from sampled signals.

[8] LEHMANN, D. J., AND THEISEL, H. Discontinuities in continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1291 – 1300.

[9] LIN, X., RUAN, S., QIU, T., AND GUO, D. Nonrigid medical image registration based on mesh deformation constraints. *Computational and Mathematical Methods in Medicinee* (2013), 8.

[10] YIN, Y., HOFFMAN, E. A., DING, K., REINHARDT, J., AND LIN, C.-L. A cubic b-spline-based hybrid registration of lung ct images for a dynamic airway geometric model with large deformation. *Physics in Medicine and Biology 56*, 1 (2011).

[11] ZITOVA, B., AND FLUSSER, J. Image registration methods: a survey. *Image and Vision Computing 21*, 11 (2003), 997 – 1000.

A Full Test Results

Source images	 The reference image	 The matching image

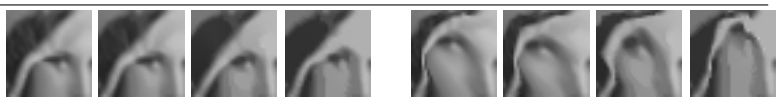
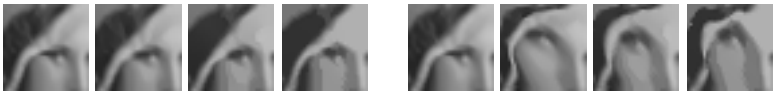
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.1: Results showing the problem with rigid movement of the nodes near the image borders. Please notice the attraction of plateaus in the last column. The conductance term for the Perona-Malik filter was set to 0.2. The 32x32 pixel-wide source images are blurred 32 times with an 3x3 kernel.

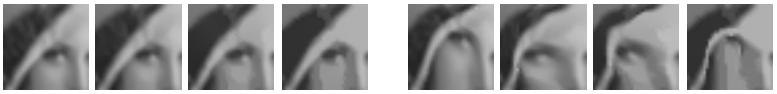
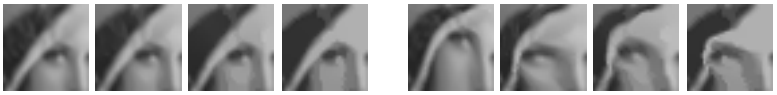
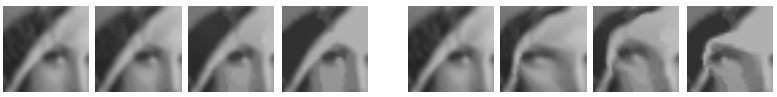
Source images	 The reference image	 The matching image						
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.2: In this experiment only the matching image was blurred 32 times. The conductance term was set to 0.2. Both images have a resolution of 32x32 pixel. All results are not satisfying. At first, we can see, that the problem with the non moving border pixels still remains. Second every approach delivers the same result, except the multi scaled global approach with 0 iterations for the Perona-Malik filter.

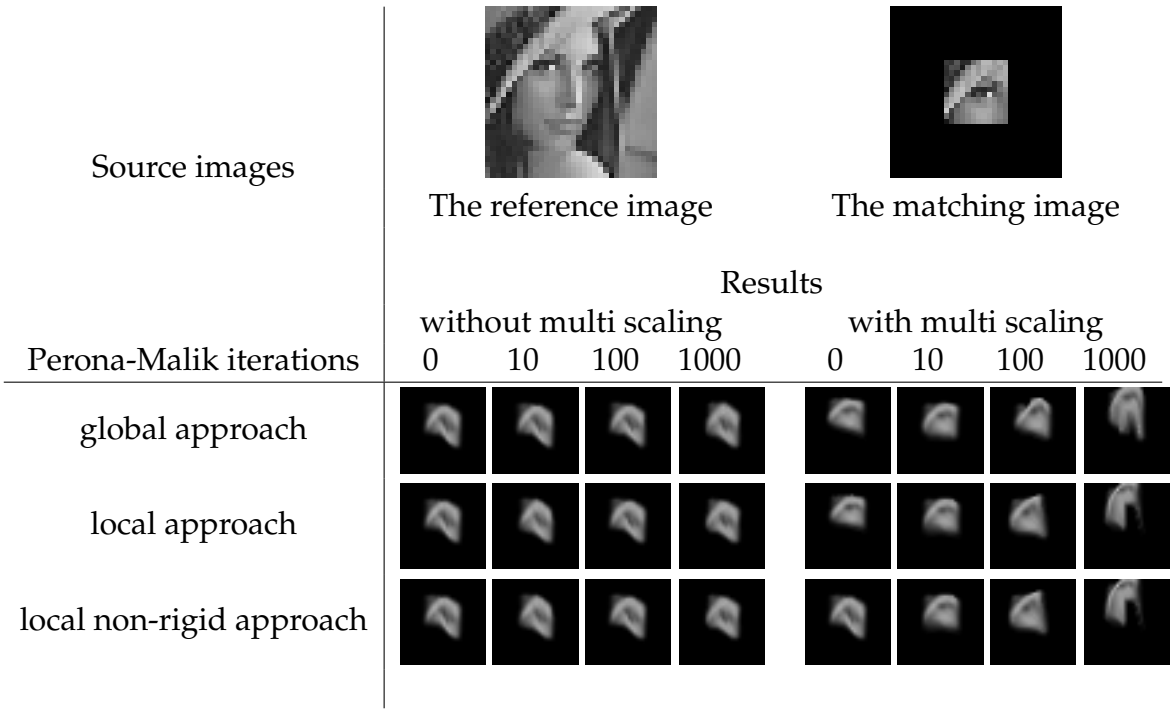
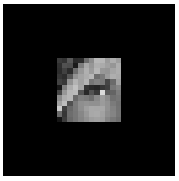


Table A.3: Two 32x32 pixel grayscale images calculated using the three modes (global, local rigid, local non-rigid), multi scaling and different Perona-Malik iterations. The conductance term for the Perona-Malik filter was set to 0.2. Both images are blurred 32 times with an 3x3 kernel. You can see that the calculation without multi scaling delivers nearly the same results. The best results are achieved from the local rigid and non-ridig approach with multi scaling and 100 iterations of the Perona-Malik filter. The problem with the non-moving border nodes is avoided by using a small template embedded in a plateau (the black border in the matching image).

Source images	 The reference image	 The matching image

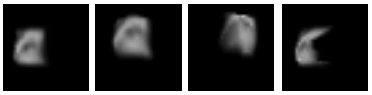
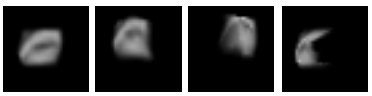
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.4: This experiment is the same as from Table A.3, but with the difference that the reference image was untouched from the Gaussian blur. As you can see, this experiment has completely different results. The best onces are achieved by the multi scaled local non- and rigid approach with 10 iterations of the Perona-Malik iterations. One thing to notice is the fact that the plateaus created in the reference image by the Perona-Malik filter are falsifying the information as described in chapter 5.1.2.

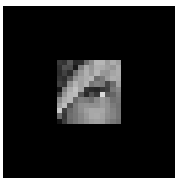
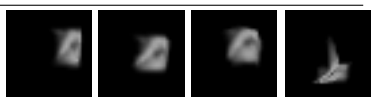
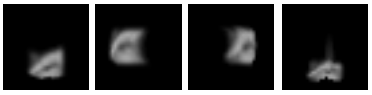
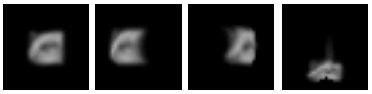
Source images	 The reference image	 The matching image						
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.5: In this experiment the matching image was untouched by the Gaussian blur. Again, the approaches without multi scaling doesn't show any difference to the matching image. The multi scaled local non- and rigid approach are producing the most convenient result with 10 iterations for the Perona-Malik filter. Both images are 32x32 pixel wide, blurred 32 times and the Perona-Malik conductance term was set to 0.2. One thing to notice is that the plateaus created in the reference image are falsifying the information again (see chapter 5.1.2 for more details).

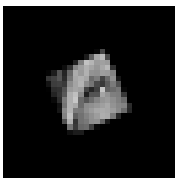
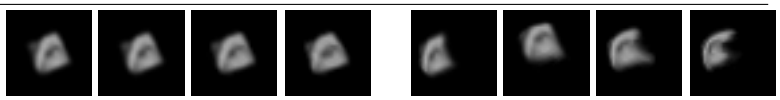
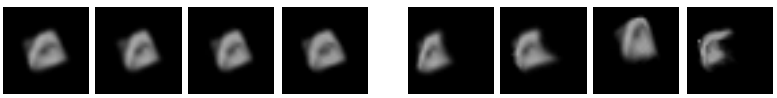
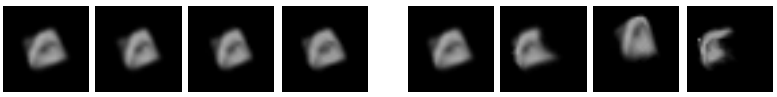
Source images	 The reference image	 The matching image						
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.6: This experiments shows the results with a matching image containing a rotation. As you can see none of the results are satisfying the expected solution. Solely the multi scaled global approach with 10 Perona-Malik iterations gets close to the expected result. Both 32x32 pixel images were blurred 32 times. The Perona-Malik conductance was set to 0.2.

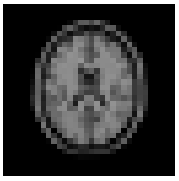
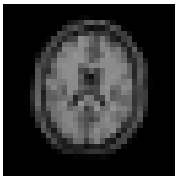
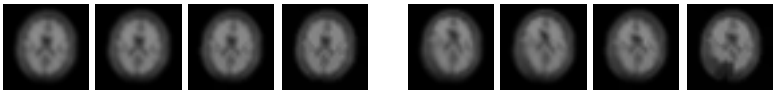
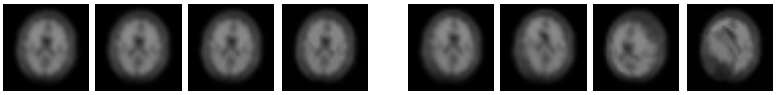
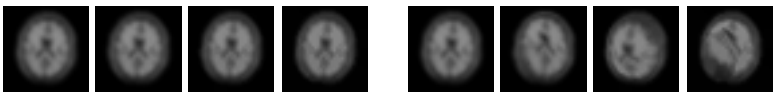
Source images	 The reference image	 The matching image						
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.7: In this experiment the matching image is rotated by 5 degree clockwise. The best result are achieved with the global approach using multi scaling and a Perona-Malik iteration of 100. As you can see this experiment is matching the border of the cerebral hemispheres of the reference and matching image. Unlike from the definition of the Jacobian, this registration may be just a random result caused by the global approach, where all points are optimized as a whole.

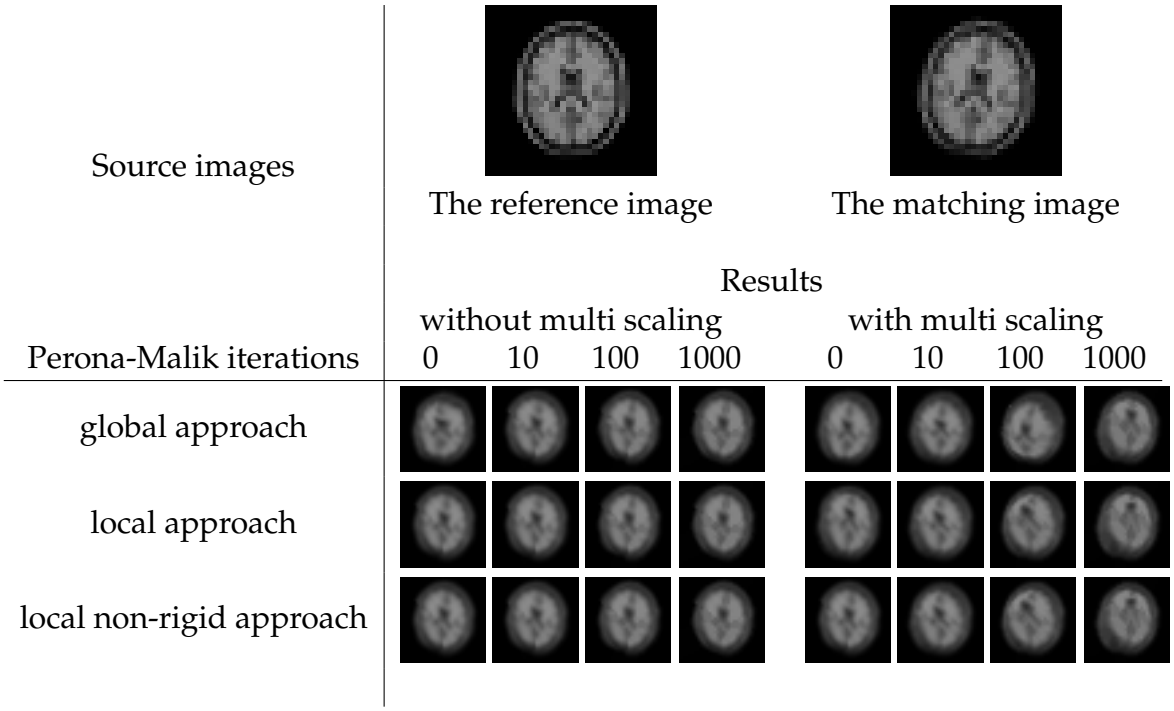


Table A.8: In this experiment the matching image is rotated by 10 degree. None is delivering satisfying results. As you can see the strong Perona-Malik filter is attracting all the points, due to the formed plateaus.

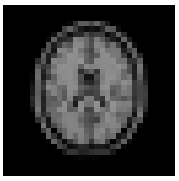
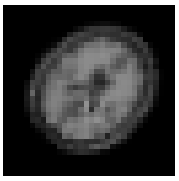
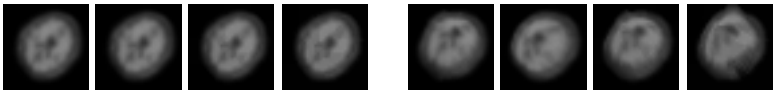
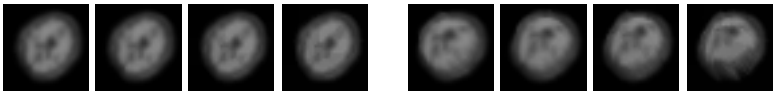
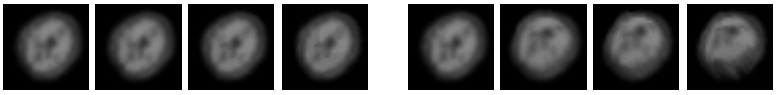
Source images	 The reference image	 The matching image						
	Results							
	without multi scaling				with multi scaling			
Perona-Malik iterations	0	10	100	1000	0	10	100	1000
global approach								
local approach								
local non-rigid approach								

Table A.9: This experiment shows the limit of the registration. The reference image is rotated by 15 degree clockwise. This rotation can't be registered by the algorithm anymore. Again, the strong Perona-Malik filter is distorting the results.

The reference image

The matching image

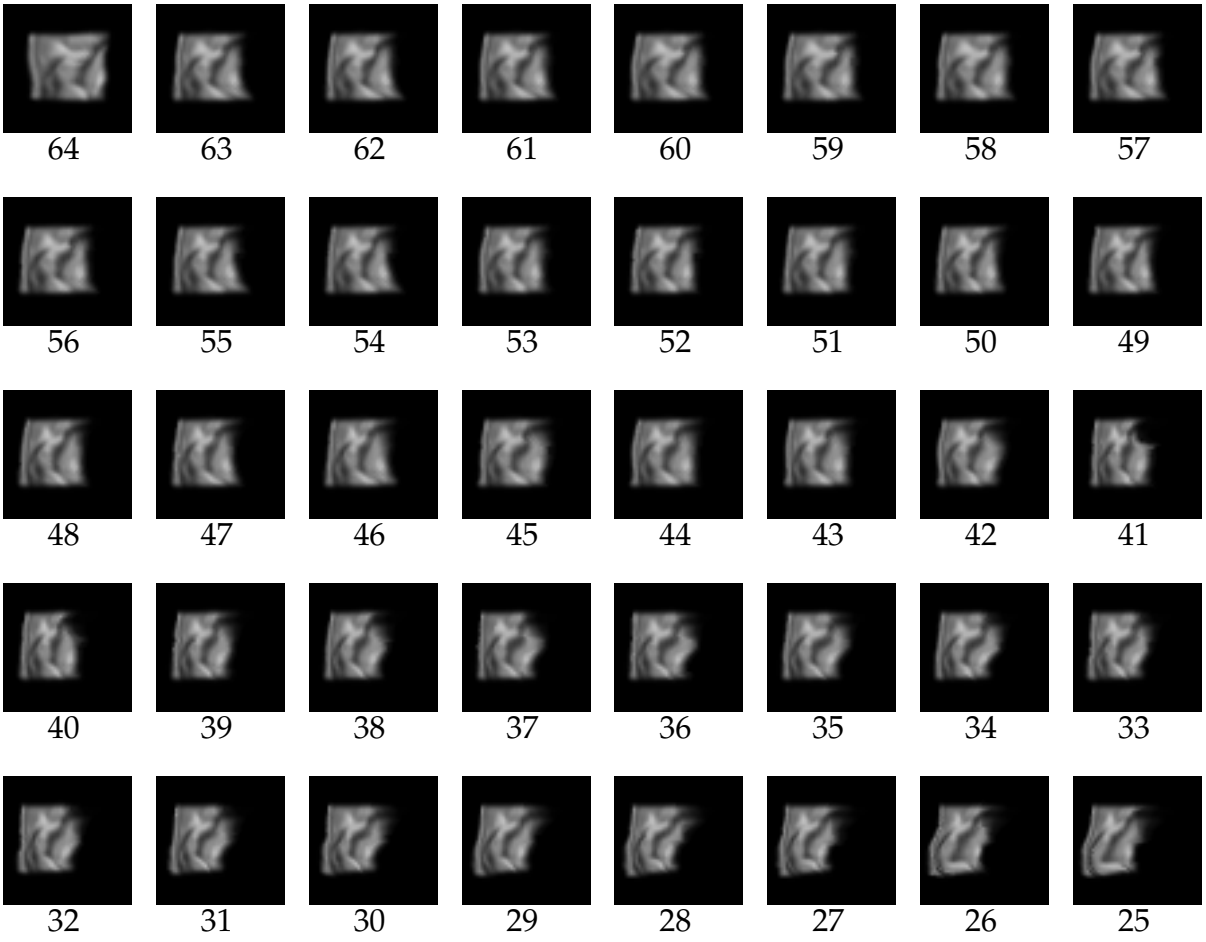


Table A.10: The first 40 results of the registration process. As one can see the Perona-Malik filter with an iteration count of 100 is forming plateaus, which are attracting the lower left part of the matching image. When the experiment terminates, all nodes of the matching image have wandered to the left side of the image.