A Global Linear Optimization Framework for Adaptive Filtering and Image Registration

by

Gustaf Johansson

Department of Biomedical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Linköping 2015
This is a Swedish Licentiate's Thesis

Swedish postgraduate education leads to a doctor’s degree and/or a licentiate’s degree. A doctor’s degree comprises 240 ECTS credits (4 year of full-time studies). A licentiate’s degree comprises 120 ECTS credits.

Copyright © 2015 April

ISSN 0280–7971
Printed by LiU Tryck 2015

URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-117024

2
Contents

1 Introduction 9
  1.1 Medical Images ............................................. 10
  1.2 Image denoising ............................................. 10
  1.3 Image registration ........................................... 14
  1.4 Regularization and context atlases ......................... 15

2 Prerequisites 17
  2.1 Introduction .................................................. 17
  2.2 Linear algebra and filtering ................................ 17
  2.3 Local Structure Estimation .................................. 24

3 Global Linear Optimization 27
  3.1 Consequences of minimization ............................... 27
  3.2 GLO theory ................................................... 29
  3.3 Local linear descriptors .................................... 33

4 Applications 35
  4.1 Overview ...................................................... 35
  4.2 GLO Image Denoising using Local Structure Tensor Metric 37
  4.3 Global Anisotropic Regularization using Local Tensor Metric 42
  4.4 A Tensor Decomposition for Preservation of Sliding Motion 43
  4.5 Regularization Constraints for Preserving Volumes and Areas 48

5 Discussion 51
  5.1 Overview ...................................................... 51
  5.2 Results and performance ..................................... 51
  5.3 Comparing GLO with former methods ......................... 52
  5.4 Implementation details ...................................... 53
  5.5 Future work ................................................... 54

6 Publications 55
  6.1 Overview ...................................................... 55
List of Figures

1.1 Two medical images which can be registered - calculating how to deform one of them to match the other

2.1 1D convolution matrices for mean value and differentiation.
2.2 2D convolution matrices for differentiation.

3.1 The three small basis functions in our $S_F$ space example.
3.2 From upper left to lower right: $\epsilon_h = 10^{-\{5,3.2,1.0,-3\}}$

4.1 SSIM scores for GLO-LoSTM vs BM3D for man and Lena512
4.2 Original and noisy Lena512.png
4.3 Visual comparison for Lena512.png
4.4 Original and noisy Lena512.png
4.5 Visual comparison for Lena512.png
4.6 Original and noisy man.png
4.7 Visual comparison for man.png
4.8 Original and noisy man.png
4.9 Visual comparison for man.png
4.10 Table for explaining symbols and notation.
4.11 Setup for the planar surface experiments.
4.12 Results for the planar surface experiments.
4.13 Setup for the circular surface experiments.
4.14 Results for the circular surface experiments.
4.15 Setup and result of in-compressible field.
4.16 Resulting deformation after 20 and 33 iterations.
4.17 In-compressible field without contact
4.18 In-compressible field with contact
Abstract

Digital medical atlases can contain anatomical information which is valuable for medical doctors in diagnosing and treating illnesses. The increased availability of such atlases has created an interest for computer algorithms which are capable of integrating such atlas information into patient specific data processing. The field of medical image registration aim at calculating how to match one medical image to another. Here the atlas information could give important hints of which kinds of motion are plausible in different locations of the anatomy. Being able to incorporate such atlas specific information could potentially improve the matching of images and plausibility of image registration - ultimately providing a more correct information on which to base health care diagnosis and treatment decisions.

In this licentiate thesis a generic signal processing framework is derived: Global Linear Optimization (GLO). The power of the GLO framework is first demonstrated quantitatively in a very high performing image denoiser. Important proofs of concepts are then made deriving and implementing three important capabilities regarding adaptive filtering of vector fields in medical image registration:

1. Global regularization with local anisotropic certainty metric.
2. Allowing sliding motion along organ and tissue boundaries.
3. Enforcing an incompressible motion in specific areas or volumes.

In the three publications included in this thesis, the GLO framework is shown to be able to incorporate one each of these capabilities. In the third and final paper a demonstration is made how to integrate more and more of the capabilities above into the same GLO to perform adaptive processing on relevant clinical data. It is shown how each added capability improves the result of the image registration. In the end of the thesis there is a discussion which highlights the advantage of the contributions made as compared to previous methods in the scientific literature.
Acknowledgments

First I would like to thank my primary supervisor Hans Knutsson for putting his trust in me, for all the stimulating discussions, ideas and lots of constructive criticism and helping me to proof-read articles. Thanks to co-supervisor Mats Andersson for getting me started and keeping me updated with everything from helping with computer trouble to filter optimization, proof-reading articles and helping me understand and to teach our course. Thanks to Anders Eklund for introducing me to the department and helping me out with lots of questions in my first time as a post-graduate student. Thanks to Daniel Forsberg for introducing me to the field of image registration, the morphon image registration algorithm and helping me to get started on my first article. Thanks to Johan Wiklund and Thomas Johansson for keeping my computer happy. Thanks to everyone on administration and HR who have endured my never ending and stupid questions about which forms to fill in and how.

Thanks to all teachers and professors in my excellent schooling and university studies and thanks to everyone who have helped boost my interest and curiosity in mathematics, the natural sciences and technology.

And to my friends and colleagues at work, from my university studies in Linköping and to my childhood friends from my home town Säffle.

Finally thanks to my parents Karl-Erik and Viktoria for their never ending and unconditional support regarding everything in life and to Andreas for being the best big brother in the world.

And to Dr. Stein for growing funny creatures. Let their time be right.

This work has been financed by the Swedish Research Council (Vetenskapsrådet) as part of grant nr. 2011-5176 for project: Dynamic Context Atlases for Image Denoising and Patient Safety. It is also part of CADICS, a Linneaus Research Environment.
Chapter 1

Introduction

In this licentiate thesis a theoretical framework within multidimensional signal processing is presented and derived: Global Linear Optimization (GLO). Chapter 2 contains the elementary algebra and signal processing required for the framework. Chapter 3 is focused on building the framework and exploring some of its important properties. Chapter 4 contains an overview of how the applications are constructed within the framework. Chapter 5 contains a discussion regarding the contributions made in this thesis. Finally the last chapter contains a short overview of the publications followed by the papers. The applications are derived in Chapter 4:

1. Image denoising with local anisotropic metric.
2. Regularization with local anisotropic metric.
3. Costs for object sliding along a common surface.
4. Area and volume preserving regularization.
5. Model fusion of patient specific data and atlases.

Application 1 is a standalone demonstration of the power of the framework. Application 2 is derived in paper 1 and application 3 is derived in paper 2. Applications 3 and 4 are demonstrated in journal paper 3.

Images in the thesis should be viewed in electronic version on a high-end computer monitor as the printing process removes much detail. The rest of this chapter contains introductions to sub fields of image processing which will be important for the contributions made.

Figure 1.1: Two medical images which can be registered - calculating how to deform one of them to match the other.
1.1 Medical Images

There are many kinds of medical images. A specific technology for acquiring an image is called a *modality*. Examples of modalities are:

- X-ray
- Computed Tomography (CT, 4DCT)
- Ultrasound, 2D & 3D + time
- Magnetic Resonance Imaging (MRI)
- Diffusion Tensor Imaging (DTI)
- Mammography
- Elastography (Ultrasound & MR)
- Positron Emission Tomography (PET)
- Single Positron Emission Computed Tomography (SPECT)
- Endoscopy

How the images are obtained and what they represent varies very much between different modalities. Both patient specific information and anatomical circumstances decide the choice of modality. For instance, metal implants will cause artifacts which degrade the image quality. Since MRI uses an extremely strong magnetic field, it can not be used if the patient has metal implants.

1.2 Image denoising

All kinds of images can have more or less noise. Noise is defined as parts of an image which are “unwanted” in some sense. Typically for medical images - any part of the signal which makes it more difficult to make a correct diagnosis. Many things can impact the type of noise and the amount of noise in images.

X-ray, CT and Mammography are all examples of modalities in which ionizing radiation is used for imaging. A high exposure can create images with higher Signal to Noise Ratio (SNR), but since the ionizing radiation is hazardous it is desirable to reduce the dose. Here denoising can decrease the noise level while maintaining a low exposure.
Other imaging modalities like MRI have other limits. There higher resolution and higher SNR can be acquired if scanning time is increased. So better denoising could mean screening more patients per time unit.

Denoising is a mature sub field of image processing. Many concepts and methods have been introduced over the years. Some of the most popular approaches are (with examples inside parentheses):

- Classical methods (Wiener filtering, Kalman filtering [1])
- Rank / Order / Sort filters (median filtering, description: [2])
- Local adaptive filtering ([3], [4])
- Diffusion based (Perona-Malik [5])
- Total Variation (Chan [6], Rudin [7])
- Transform-domain (DCT, Wavelets: GMM [8], shrinkage [9])
- Non-Local filtering (Non-Local Means [10], BM3D [11])
- Learning of Dictionaries (Sparse Dictionaries [12], LSSC [13])

Classical methods
Wiener filtering and Kalman filtering are two very popular types of classical filtering techniques. One paper on denoising which is inspired by both of these is [1]. These classical methods and ideas are often present in modern papers in various ways. Another popular type of filtering is the rank/order/sort filters where the pixel intensities are sorted (often locally) and then the image is processed based on one or several of the sorted values. A paper briefly describing these is [2]. Maybe the very most popular of these filters is the median filter. It’s popularity probably much from combination of reasonable results and ease to understand.

Local methods
The early local adaptive filterer [3] used filters with specific orientations designed in the Fourier domain to locally decide how much of the high-pass information should be let through in different directions. The filters had a limit of spatial support up to $15 \times 15$, but it is explained how iteration can be used to get a more powerful processing. Several developments of the same basic concepts and development of new ones: for example introducing the structure tensor [14], monomial filters [15], processing in both 3 and 4 dimensions and acceleration by graphics card hardware has lead to publications like [4].
One paper which made diffusion approach for image denoising popular is the paper by Perona-Malik in 1990[5]. Many denoising algorithms can be viewed as some kind of diffusion process, either deliberately constructed as one or be analyzed as such. What they have in common is that they somehow measure locally how to perform the diffusion, in this same belonging to the same kind of algorithm as the local adaptive filterers above.

Total Variation (TV) is another concept which usually is applied to features measured locally, but using some kind of optimization of specific metrics other than the $L^2$ norm, for instance $L^1$ norm. It has been shown that for some images these approaches can give better results for image denoising.

**Transform domain methods**

The transform domain methods are not easily classifiable as strictly either local or non-local methods. For instance the non-local denoiser BM3D below uses transform domain filtering (wavelets), but many “local” denoisers also exist, for instance the many different soft and hard thresholding techniques. However most transforms are mixture between frequency and spatial features. Discrete Cosine Transform (DCT) based methods are usually blocked to achieve a mixture of spatial and frequency information as larger DCTs contain far too much focus on frequency components over large spatial supports. Wavelets have a built in trade-off between spatial and frequential information, usually making blocking of the transform not so desirable. There is much inspiration of the knowledge from transform coding community, where both DCTs and DWTs have proven very powerful and are present in many modern image and video compression standards such as the image codec Jpeg2000 (uses wavelets) and the video codec H264 (uses DCT). In image denoising, a technique called “thresholding” (pushing low-energy high-pass wavelet coefficients towards 0) was presented in [9]. Another method which gained recognition for great performance at the time was the Portilla [8] method which modeled the image using gaussian mixture models in the wavelet domain.

**Non-Local methods**

The paper [10] from 2005 introduced the Non-Local means (NL-means or NLM) which showed impressive performance triggered an interest in Non-Local methods which has not waned for at least the last 10 years.

The non-local methods all have in common that they make assumptions that the statistics of the signal is strictly non-local, i.e. they allow themselves to let the value of one pixel be influenced directly by values of pixels which are spatially very far away. It is important to remember these assumptions if comparing error and performance measures with other classes
of methods.

One of the most well known non-local methods is the BM3D [11], which performs extremely well on most test-images and also on low SNR. Various improvements have been made, SA-BM3D (Shape Adaptive) [16] and BM3D-SAPCA (with local PCA) [17], for each new addition presenting marginal gains compared to their predecessor, but of course at the cost of an increased complexity of the denoising algorithm. However the performance of the BM3D was so good at it’s time that it is still hard to outperform and often used in bench marking for new denoisers which aspire to be state of the art.

Dictionary and machine learning methods

Paper [12] introduced denoising methods for images by using different types of Machine Learning methods to learn statistics over hundreds, thousands or even more images. This is of course more aggressive than the “Non-Local” methods which are limited to measure and aggregate local patches all over one image. The most modern and powerful methods such as the LSSC[13] performs even better than BM3D - but is also more aggressive in modeling the signal - putting constraints on what is data and what is noise.

Reasonable assumptions → reasonable processing

One important thing is to be observant regarding the assumptions made. Clues in medical data as to whether a specific image indicates a disease or pathology are often contained locally in the image. This means that we can use a local method that does not assume that the local neighborhood has relations to other spatially distant image neighborhoods. If we use a Non-Local method assuming non-locality we may risk to treat a local variation which is due to a pathology as noise - since it is only present at one neighborhood of very many which in other aspects are the same.
1.3 Image registration

Image registration is a sub field of image analysis where the objective is to match one data set to another. The registration is expressed in terms of a movement or a displacement. Usually this displacement is represented as a vector field, moving data from the grid of one data set to the other. The applications are many, from motion analysis in video to medical image registration where the goal is to calculate what has changed before and after a treatment or how different parts of the anatomy moves in the case of a time sequence data.

Since the focus application of the thesis - regularization of displacement fields - is directly related to image registration, we are now going to present and briefly explain some popular image registration approaches.

To simplify, registration can be put into different categories: Transformation based v.s. elastic and local v.s. global models. We are mainly concerned with elastic registration because there are many more degrees of freedom for elastic than for transform based registrations.

Overview of registration algorithms

A very early and influential work on deformable image registration is the PhD thesis by Broit: [18]. A good book of numerical algorithms for deformable image registration is [19]. A good but rather lengthy and detailed investigation on obtaining smoothly varying vector fields for image sequences in video is [20]. A modern and versatile image registration algorithm is the Morphon [21] which performs a non rigid registration using a multi-scale approach based on local quadrature phase of the images.

Registration with locally adapted regularization

One interesting type of registration algorithms uses locally adapted regularization. This is very important to our application (medical image registration) since we want to be able to locally adapt the regularization to the local organs and tissues in the image data. Therefore, we will cover some different papers on this type of regularization.

A first example is the work [22] which uses a scalar “stiffness” image deciding how easily each pixel should move. They then show that it manages to preserve stiffness of a tumor - which is expected to be stiffer in their context.

One example is [23] which uses the well known Demons [24] registration algorithm coupled with a local regularizer. Locally adapted regularization is discussed in many more articles, for instance [25] which uses a $L_1$ norm for regularization ; something which gained popularity in the last years for
image denoising (for instance Total Variation). A grid approach is used in a paper from 2004 by Stefanescu, Pennec and Ayache[26]. This approach discusses several concepts including explicit and implicit schemes. Here the implicit schemes have a resemblance to our concept of Global Linear Optimization although the diffusion based constraints used are different. Also various possible schemes of how to implement in hardware are considered which may be of interest.

The paper [27] uses normalized convolution [28] together with regularized template matching in a a multiresolution pyramid to achieve a regularized registration. A reliability-driven approach is presented in [29], where a non-linear noise-gating for both gradient and curvature is used. These noise gatings are put together into the non-linear reliability measure. Then four different schemes are presented and investigated which each include this reliability measure in different ways to solve a mathematical minimization problem.

1.4 Regularization and context atlases

Regularization in image registration is the process of ensuring a smooth and plausible vector field. Unparameterized motion estimations need more careful regularization because the degrees of freedom are many more. We are mostly concerned with regularizing unparametrized registrations in this thesis, so the focus will be more on these types of image registration. One of the objectives of this thesis is to build a data processing framework which can incorporate context information from medical atlases into the regularization to make patient adapted data processing. Here we will present some papers which explicitly try and achieve at least one of these functionalities and then compare them to the GLO approach.

Methods for sliding or slipping objects

The paper [30] introduced a mechanism for preserving or allowing a slipping motion, but the approach used the calculus of variations and the Euler-Lagrange equations. Other papers with the same functionality include [31] [32] [33] of which the first two explain very in depth a specific mechanism involving a weight \( w \) and a normal vector \( n \) to encode the direction to the closest slipping surface. The \( w \) is a fast monotonically decreasing function of \(|n|\) such that it is more important to allow slipping in close proximity to the tissue boundary than far away.
Methods for preservation of area and volume

Allowing to put constraints on the change of area and volume is important in medical image registration - because many organs in the human body are known to have such properties that even though they are non-rigidly deformable they are \textit{in-compressible} - i.e. not possible to change volume in 3D or area in 2D. There exist many methods which aim to achieve this regularization mechanism. Important examples include \[34, 35, 36, 37, 38\]. First of these \[34\] use the determinant of the Jacobian to put a constraint on change of Area or Volume as a determinant not equal to 1 means a local change in area or volume. The paper by Denney \[35\] formulates a stochastic motion model where the incompressibility condition is modeled as correlations between various of the displacement or velocity field gradients. The method used in \[36\] uses a divergence-free approach in a Finite Elements Model (FEM) and show that it gives very reasonable results for myocardial and blood velocity. Much like \[34\] the paper \[37\] uses the determinant of the Jacobian to preserve volume. However instead of minimizing some energy they use an explicit calculation of correction to the deformation. The last of these papers : \[38\] discusses this incompressibility constraint with respect to the symmetry of forward and backward mapping, something which is not very much discussed in this thesis, but nevertheless interesting.
Chapter 2

Prerequisites

2.1 Introduction

In this first chapter we will go through various prerequisites which will be necessary to formulate and understand the GLO framework and the mechanisms involved. This is intended to be a very thorough treatment. Do not hesitate to skip the parts which seem painfully obvious!

2.2 Linear algebra and filtering

Signals, images and filters

A digital signal is a function that somehow samples an underlying continuous (or discrete) function at discrete points. An image is in this thesis considered to be a signal with 2 or more outer dimensions. Example of images with 2 outer dimensions are gray scale photographs and plane X-ray. Images with 3 outer dimensions include MRI and CT scans at a single point in time. It is common to call 3D images “volumes”, especially if they store data about 3 spatial dimensions. However there also exist 2D+time which are 3 outer dimensions, for instance classical ultra-sound can give such images. Another example is photographic video. Even higher dimensionality is possible. For instance there exist techniques in MRI, CT and Ultrasound to acquire a sequence of 3D images at discrete points in time, which therefore may be considered 4D signals (3D + time). Multidimensional signals regard signals which are of two outer dimensions or more. A filter can be regarded as a signal in these senses, although they are usually designed with a specific purpose - to measure on signals. Usually filters have a more compact support than signals. This is because usually filters are designed to measure some local property of signals.

\(^1\)The support is the set of nonzero values.
Outer and inner dimensionality

The dimensions we have talked about so far are “outer dimensions” or physical dimensions - concerning time and place. Another important concept is “inner dimensionality”. A signal can have 3 inner dimensions, if we for example measure a 3 dimensional vector at each point in space. Another example of inner dimensionality is for color photographs, usually having three inner dimensions (for instance red, green, blue). Later we will be talking about structure tensors which have even more inner dimensions than vectors. To simplify, one can say that the number of inner dimensions correspond to the number of values stored for each position of the outer dimensions.

Convolution

One of the most basic and important operations in signal processing is convolution. Convolution of a digital $N$ dimensional image $I$ and a filter $f$ can be expressed as:

$$(f * I)[x] = \sum_{x' \in \mathbb{Z}^n} f[x'] I[x - x'], \ x \in \mathbb{Z}^n$$

We can see, for each $x$, this is a scalar product between subsets of $f$ and $I$. This notation may seem a bit mathematical and we will use matrix notation once we’ve gone through basic matrix algebra.

Border handling

One issue is how to handle when the filter is outside the support of the signal. The simplest and most common is probably the zero padding, simply assuming the signal equals 0 everywhere outside of it’s support. Two other common border handling are the border extension and the border mirror. The border extension just assumes that the signal continues as a constant in every dimension. A border mirror just mirrors the value in the closest border.

Circular convolution

One important type of border handling for convolution is the circular convolution: similar to an ordinary convolution, but in the case that the signal index $t - \tau > N$ or $t - \tau < 0$, it replaces it with $t - \tau - N$ or $t - \tau + N$ respectively. A more concise way to write this is by modulo arithmetic: $\text{mod}(t - \tau, N)$. In other words, circular convolution makes the assumption that the signal is periodic in each of it’s dimensions.
Matrix arithmetic, scalar products and norms

The matrix $A$ with size $N \times M$ is an organized collection of scalars:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NM} \end{pmatrix} \quad (2.2)$$

For two matrices $A$ and $B$, if their $N$ and $M$ are the same, we can define addition as the addition of element-wise scalars: $(A + B)_{ij} = a_{ij} + b_{ij}$

Multiplication is a bit more complicated. There it is required that the *inner sizes* must match. If we write the matrix sizes from left to right: $N(A), M(A), N(B), M(B)$, we see that the sizes in the middle (the “inner” sizes) are $M(A), N(B)$. These are required to be the same for the matrix multiplication to be well defined. We then define

$$(AB)_{ij} = \sum_{k=1}^{M(A)} a_{ik}b_{kj} \quad (2.3)$$

The product matrix will have the *outer size*, i.e. $N(A), M(B)$. Furthermore the transpose operator $(\cdot)^T$ on a matrix $A$ performs: $(A^T)_{ij} = a_{ji}$, so sizes $N$ and $M$ are switched, and all elements are mirrored in the *diagonal*: where $i = j$. A vector is a $N \times 1$ matrix, and a dot product between the vectors $v$ and $w$ is: $w^Tv$, which is also called a scalar product because the result is $1 \times 1$ matrix, a *scalar*. The $L^2$ norm : $\|\cdot\|_2$ is defined by scalar product:

$$\|a\|_2^2 = a^T a = \sum_{\forall i} |a_{i1}|^2$$

The Frobenius norm of matrices is defined as:

$$\|A\|_F^2 = \sum_{\forall i} \sum_{\forall j} |a_{ij}|^2$$

This is the same as the squared $L^2$-norm of any *vectorization* of $A$.

A vectorization is any unique and invertible mapping $\mathbb{Z}^n \to \mathbb{Z}$. I.e. any unique encoding of the $n$ indices into 1 index, *representing* a matrix or a multidimensional signal as a vector or *storing* it’s values into a vector.

**Block property of matrix multiplication**

An important property of matrix multiplication is that if a matrix $A$ can be written in terms of sub-matrices or block-matrices $A_{kl}$:

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NM} \end{pmatrix} \quad (2.4)$$
Then the matrix product of two such matrices $A$ and $B$ (with the same size of the sub-matrices) is:

$$(AB)_{ij} = \sum_{k=1}^{M(A)} A_{ik}B_{kj} \quad (2.5)$$

So blocks are multiplied exactly as scalars are. An important special case of this blocking is a convolution matrix, $C$, where each block $c_k$ has one row:

$$C = \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} \quad (2.6)$$

If performing $Cv$, where $v$ has the same dimensions as $c_k^T$, the result is:

$$Cv = \begin{pmatrix} c_1v \\ \vdots \\ c_Nv \end{pmatrix} \quad (2.7)$$

Which we can see is just a set of scalar products between rows and the vectorized signal.

**Linear Filtering as Matrix Multiplication**

As we see above, we can construct matrices which perform an arbitrary set of scalar products, and in the definition of convolution in 1.2.2., we saw that the result at every point is a special scalar product. We can now start constructing matrices which represent convolutions between 1D filters and 1D signals. However we need to dig a bit deeper to systematically construct multidimensional convolutions.

**Kronecker Products**

A Kronecker product is a type of matrix product which is defined as:

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1M}B \\ \vdots & \ddots & \vdots \\ a_{N1}B & \cdots & a_{NM}B \end{pmatrix} \quad (2.8)$$

We can now systematically create index vectors $i_1, i_2, \cdots, i_N$ where $N$ is the outer dimensionality. Let $N_i$ denote the size of outer dimension $i$. Let $1_N$ be the vector with $N$ ones and $o_n = (1 2 \cdots n)^T$ we can then calculate index vectors for two outer dimensions:

$$i_1 = 1_{N_2} \otimes o_{N_1} \quad i_2 = o_{N_2} \otimes 1_{N_1} \quad (2.9)$$
This will generalize well in the sense that $i_k$ will have $o_{N_k}$ at the k:th position from right in the Kronecker product, and all other factors will be $1_{N_m}$, $k \neq m$. A set of $N$ index vectors together map the position of every value in a $N$ dimensional filter or signal. If we want to construct a vectorization of a signal we can now just go through all positions of $i_1, i_2$ and copy the value from the signal at that position into the same row of a new vector as our current position in $i_1$ and $i_2$. This construction is practical, because it seems to mimic the lineup and indexing operators in the Matlab programming language.

**Multidimensional Convolution Operators**

We now have all the tools to turn both our filters and signals into vectors. What remains is to build the actual convolution matrices. But this becomes a simple task now that we have our index vectors. The properties of the Kronecker product makes it easy to find position in the vectorization given the indices.

1. For each row in the matrix, we can check the indices of the corresponding image.
2. For each position in the filter, we can find the suitable offsets for the convolution as differences between filter index and image index.
3. Finally we can transform back to vectorized indices using the Kronecker construction above. Now we know which columns our filter coefficients should be inserted for the current row. And we are good to carry on with the next row.

Item 3 can conveniently be found as a linear regression:

$$[1_k, i_1, \cdots i_N]p = i(:,) \quad (2.10)$$

This can be solved in terms of a parameter vector $p$:

$$p = [1_k, i_1, \cdots i_N]^\dagger i(:,) \quad (2.11)$$

Here $i(:,)$ is the vector of positions in the vectorized images. Furthermore, $\dagger$ denotes the Moore-Penrose pseudo-inverse:

$$A^\dagger x = (A^*A)^{-1}A^*x \quad (2.12)$$

If $A$ has full column rank, and

$$A^\dagger x = A^*(AA^*)^{-1}x \quad (2.13)$$

If $A$ has full row rank. In our case we can substitute:

$$A = [1_k i_1, \cdots i_N] \ ; \ x = i(:,) \quad (2.14)$$

21
Linear Equation Systems

A linear equation system \( Mx = b \) can be solved by many methods. What is important to note is that all of our \( M \) matrices will be derived from combinations of convolution matrices. One important property that those will have is that they will be \textit{sparse}. This means that almost all entries will be 0. For such matrices there exist especially fast algorithms to try and solve. One popular such family of algorithm is the \textit{Krylov subspace} methods in general and the \textit{Conjugate Gradient} in particular. They are iterative and only use relatively cheap and simple matrix operations such as matrix-vector multiplication and addition. The Conjugate Gradient requires the matrix to be symmetric positive semi definite, but luckily as we will see later on, we can systematically make sure that our equation systems will have this property.

Filters as matrices

We start with two simple 1D approximations to low-pass and derivative: the \( \frac{1}{2} [1, 2, 1] \) low-pass filter and the \([1, 0, -1] \) derivative filter. We continue with demonstrating matrices for the 2D \textit{Sobel} derivative approximating filters for (very small) 2D images:

\[
\begin{align*}
    d_x &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} &
    d_y &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}
\end{align*}
\]

(2.15)

Figure 2.1: Left: \( \frac{1}{4} [1, 2, 1] \) filter matrix. Right: \([1, 0, -1] \) filter matrix.
Adaptive filtering - Spatial filter adaptation

The matrices we have shown so far have rows consisting of the very same values (the filter coefficients) at every row - the same values - but permuted to different positions. Since we can express linear combinations of the signal in every neighborhood, we realize that a spatially invariant filter is very restricted, compared to what we are able to express with these matrices. In practice we could adapt the filter in any position by just finding it with the index vectors and then changing it based on some arbitrary constraint. We will be doing this a lot later on, although sometimes we won’t mention it. For instance multiplication with local structure tensor $T$ is exactly such an operation, since $T$ has been measured somehow and can be different on every location in the image. Most of the adaptations we make in this thesis are limited to having localized filters. However the framework is able to express any linear combination over the whole image for every position in the image.

Vectorization of inner dimensions

In this thesis we have chosen to stack the inner dimensions onto each other in the vectorization. A 3D vector field $v = (v_1 \, v_2 \, v_3)^T$ will be vectorized as:

$$\text{vec}(v) = (\text{vec}(v_1) \, \text{vec}(v_2) \, \text{vec}(v_3))^T$$

(2.16)

Thanks to the block property of matrix multiplication, using this vectorization will give the systematic and practical expressions with Kronecker products in the applications chapter. If this is still unclear, pay special attention to the details of paper 1 which describe how to build GLO matrices without the help of Kronecker products.
2.3 Local Structure Estimation

The Fourier Transform

The discrete Fourier transform of an image \( I \) of \( d \) dimensions is defined as:

\[
\hat{I}(u) = \sum_{\forall x} I(x) \cdot e^{-\frac{2\pi i}{N} u^T x} : \quad u \in [0, N - 1]^d
\]  

(2.17)

The circular convolution is extremely important as it has very close ties to circulant matrices and the Fourier Transform, which are used extensively in signal processing. It can be shown that a circular convolution becomes a multiplication in the Fourier domain. If both signal and filter are of size \( N \), the complexity of a convolution is \( O(N^2) \) and the complexity of multiplication of \( N \) values is \( O(N) \). There exist many Fast Fourier Transform (FFT) algorithms which operate at complexity \( O(N \log(N)) \). These facts can be used to speed up circular convolution by performing multiplication in the Fourier domain instead of convolution in the spatial domain.

The Quadrature Concept

The quadrature concept is based on the fact that a filter which is zero in one half of the Fourier domain and real in the other half will in the spatial domain have a even real part and an odd imaginary part. It can be shown that such a filter will measure all of the phases of the frequencies in it’s pass band.

Radial and Directional Components

When designing quadrature filters, it is practical to do so in the Fourier domain, because of the half-plane support. Furthermore we can build the filters as the product of radial \( R \) and directional \( D \) components. Many functions can be used for \( R \) and \( D \). By tradition the directional factor can be some power of a cosine. This is nice because it is very conveniently expressed as powers of scalar products of normalized vectors. But one important difference: if said scalar product is negative it should be set to 0 ( to achieve 0 in the “backwards” half-plane ). A popular choice of radial function is the log-norm function, which has two parameters: a center frequency and a band-width.

Filter Optimization

Since the quadrature filters are designed in the Fourier domain, we need to make sure they are somewhat suitable in the spatial domain. This is usually done with a process called filter optimization. In this process one can design constraints in spatial, Fourier domain and also other domains (if wanted) to produce practically useful filters. More of this can be read in chapter 5 in the book [39].
Tensor Definition

In the context of this thesis, tensors are special cases of matrices. First and foremost: tensors are always unitarily diagonalizable.

\[ T = S^T D S \text{ s.t. } S^T S = SS^T = I \]  \hspace{1cm} (2.18)

This means the eigensystem is always orthogonal. Furthermore it can be shown that \( D_{ii} \geq 0 \), i.e. the eigenvalues are real and strictly non-negative. We can write this as:

\[ T = \sum_{i=1}^{N} \lambda_i \hat{e}_i \hat{e}_i^T \text{ with } \lambda_1 \geq \cdots \geq \lambda_N \geq 0 \]  \hspace{1cm} (2.19)

And also the eigensystem is orthogonal:

\[ \hat{e}_i^T \hat{e}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \]  \hspace{1cm} (2.20)

Construction of Tensors

One construction which always produces a tensor is a sum of outer products:

\[ T = \sum w_i \hat{v}_i \hat{v}_i^T, \text{ if } w_i > 0 \]  \hspace{1cm} (2.21)

This is a very convenient construction and will be used later. Also all positive powers of tensors will always be uniquely well defined, even negative powers will be defined if the tensor is strictly positive definite. Because of the guaranteed existence of an eigenvalue decomposition this amounts to performing the power operator separately on each eigenvalue.

Representing Local Structure and Orientation

We now have all the tools required to start measuring and representing local structure and orientation in images. We can design quadrature filters in different orientations. Chapter 10 in the book [39] contains many details of how to build and filter images with tensors.

Non-linear Tensor remapping

For practical applications it is sometimes desirable to perform non-linear scaling of structure tensors. Two such important mapping functions are the \( m \) and \( \mu \) mappings described in chapter 10 of the book [39]:

\[ m(x, \sigma; \alpha, \beta, j) = \left[ \frac{x^\beta}{x^{\beta+\alpha} + \sigma^\beta} \right]^{1/j} \]  \hspace{1cm} (2.22)
\[
\mu(x; \alpha, \beta, j) = \left[ \frac{(x(1-\alpha))^\beta}{(x(1-\alpha))\beta + ((1-x)\alpha)^\beta} \right]^{1/j} \tag{2.23}
\]

Then the \( m \) function is applied to the tensor magnitude:

\[
\gamma_1 = m(|T_{lp}|, \sigma; \alpha_1, \beta_1, j) \tag{2.24}
\]

The \( \mu \) function is applied to the tensor shape in the following sense:

\[
\mu_1 = \mu \left( \frac{\lambda_n}{\lambda_{n-1}} ; \alpha_1, \beta_1, j \right) \tag{2.25}
\]

The eigenvalues are assumed to be sorted so that \( 0 \leq \lambda_n \leq \lambda_{n-1} \), we can assure that this is always a number between 0 and 1. Because the telescoping properties of multiplication of the \( \mu_k \) above, the control tensor can be constructed iteratively like this:

\[
C = \gamma_1 (\hat{e}_1\hat{e}_1^T + \mu_2(\hat{e}_2\hat{e}_2^T + \cdots + \mu_N(\hat{e}_N\hat{e}_N^T))) \tag{2.26}
\]
Chapter 3

Global Linear Optimization

3.1 Consequences of minimization

Philosophical prologue

Usually, when working with image processing - or engineering at large - the philosophical stance and question when addressing a problem would be: *What do we want to achieve?* A positive description of what we want. When stating and solving a minimization problem however, we would rather ask another question: *What do we want to avoid?* That - in contrast - is a negative question - describing what we do not want. It is therefore necessary to build a theoretical framework capable of expressing this in an orderly fashion.

Linear vector spaces have been used to describe and manipulate signals for a very long time. The focus has been on trying to describe parts of the signal that are *significant*, *important* and so on - regarding a specific application. It is always possible to build a complete *basis* of signal neighborhoods - but focus has mostly been to look on specific subspaces containing the information of interest.

We are now going to flip that coin and look on the other side - start looking on the subspaces spanning either uninteresting or unwanted aspects of our data.

Complementary spaces

A *complementary space* is a vector space that is required to be added to a given *feature space* in order to get a complete description of a signal. Together a *feature space* $\mathcal{S}_F$ and a *complementary space* $\mathcal{S}_C$ completely describe a signal. The complementary space is in most of our applications orthogonal
to the feature space. Unwanted parts of a signal may lie in the complementary space, whereas parts that are important to the task at hand lie in the feature space. If we can find a way to span our complementary space - then we are able to define cost functions for those undesired signal components.

**Decomposing a sample 3 × 3 basis**

Here we show an example of a very simple 3 × 3 basis that is decomposed into signal and complementary parts. First we use a low-pass filter: a weighted mean value filter. For the feature space $S_F$ we use the two very basic Sobel gradient filters $d_x$ and $d_y$:

\[
\begin{align*}
    l &= \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \\
    d_x &= \frac{1}{8} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \\
    d_y &= \frac{1}{8} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}
\end{align*}
\]

Figure 3.1: The three small basis functions in our $S_F$ space example.

Then, to be able to fully describe our 3 × 3 neighborhood, we need at least another $9 - 3 = 6$ basis functions. We might use even more basis functions if we want to span a frame for our complementary space, but as for now - let’s focus on a basis for the space.

**The simplest possible complementary basis**

The simplest complementary basis would probably be to use a random number generator together with an orthogonalization scheme. For instance the Gram-Schmidt orthogonalization.

**How to generate basis vectors?**

So how should we choose our basis vectors? Well the answer to that may really depend on the application at hand. It may even be determined by the data as such. For instance we may decide which basis vectors to use after first investigating the feature space of an image. Sometimes we may be able to - or it may be necessary to learn what basis functions should help span our feature space. Here is also room for multi-pass algorithms with an *a-priori* $S_F$ from which we together with some data can learn new basis functions to include in an *a posteriori* $S_F$ - successively refining the model of our signal.
3.2 GLO theory

To construct a GLO means to pose a Global Linear Optimizations problem with respect to the data in a multidimensional signal. All data points in the signal are part of a big linear equations system. That is, an equation system of the form: \( \mathbf{Mv} = \mathbf{b} \), where \( \mathbf{M} \) has size \( n \times m \), \( n \) being the number of equations and \( m \) the number of data points of our signal. The fact that most images are in the range of \( 10^6 \) to \( 10^7 \) pixels nowadays makes this a very large matrix, especially if the number of equations are close to the number of data points (which it is in most of our applications). For practical reasons it is important to keep the equation system sparse. Memory requirements and computational costs quickly become an issue as the number of non-zero elements in the matrices grow.

**How \( S_F \) and \( S_C \) are related to GLO**

Definitions:
\( d \) - original signal vector.
\( v \) - additive signal update vector.
The original assumption:
1. Feature space contains data we want to preserve / enhance.
2. Complementary space contains data we want to remove.

This means, that in practice for our optimization, we want to:
1. Impose costs on changes \( v \) in the feature space.
2. Impose costs on result \( v + d \) in the complementary space.

This is possible, since in GLO we can express both constraints on linear operators of changes as well as on the result. As we will see in the applications below, this will be quite useful.

**Posing a double sided cost function**

\( F_k \) - matrix representing definition of cost metric \( k \) in feature space.
\( C_k \) - matrix representing definition of cost metric \( k \) in complementary space.

Our objective function can then be stated as follows:

\[
O(v) = \sum_{k=1}^{\dim(S_F)} \alpha_k \| F_k v \|^2_F + \sum_{k=1}^{\dim(S_C)} \beta_k \| C_k (v + d) \|^2_F \tag{3.1}
\]

The terms in the first sum are the costs related to destroying features and the terms in the second sum are the costs to preserve complementaries.
Differentiating and minimizing the cost function

If we expand the Frobenius norms of our cost function, we get:

\[ O(v) = \sum_{k=1}^{\dim(S_F)} \alpha_k v^T F_k^T F_k v + \sum_{k=1}^{\dim(S_C)} \beta_k (v + d)^T C_k^T C_k (v + d) \] (3.2)

If we then differentiate with respect to \( v \) and equate to 0 we get:

\[
\begin{pmatrix}
\sum_{k=1}^{\dim(S_F)} \alpha_k F_k^T F_k \\
\sum_{k=1}^{\dim(S_C)} \beta_k C_k^T C_k
\end{pmatrix} v = - \sum_{k=1}^{\dim(S_C)} \beta_k C_k^T C_k d
\] (3.3)

This is positive semi-definite because of SVD together with Brauers theorem \([40]\) and non-negative real numbers are closed under addition. If we are not careful in choosing our division into complementary and feature space, we may need further regularization to ensure we are not on a saddle point. A simple and straight-forward way would be Tikhonov Regularization, by adding \( \lambda \| I v \|^2_F \) with \( \lambda > 0 \). However, the more careful and powerful approach is to construct a complete local basis division, as explained later.

Since the right hand side is a vector, this is a matrix-vector equation system: \( Mv = b \). If \( M \) is sparse there exist many numerical methods to solve this type of system. The Krylov subspace methods are one group of such methods which makes a series of matrix-vector multiplications to approximate a solution. This can also be used for \( F_k \) or \( C_k \) having certain properties, some of which are not necessarily sparse. However this is outside the scope of the thesis.

Complete local basis divisions

Here we demonstrate the safety of dividing a local patch into basis functions belonging to \( S_F \) and \( S_C \) respectively. Consider the 2 \( \times \) 2 filters

\[ S_F = \left\{ \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \right\} \] (3.4)

If we just use these two filters in our GLO, so that \( S_C = \{\} \), then we have no control over

\[ ? = \left\{ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right\} \] (3.5)

Which can be devastating as the subspace is then locally unconstrained. A trivial solution for minimizing derivatives or gradients if the means are allowed to be pushed to be the same everywhere - the constant function. Needless to say, the constant function is rarely a solution we are looking for. If not punishing the “chessboard” filter in any way, then we may get
arbitrary chessboard effects in our result. Ensuring that $\mathcal{S}_x \oplus \mathcal{S}_c$ spans a basis for (some) local neighbourhood guarantees us to avoid those issues. Of course this also requires $\alpha_k > 0$ and $\beta_l > 0$ for all $k$ and $l$.

**Adaptive processing and Regions of Interest**

The $\alpha_k$ and $\beta_k$ scalar weights can be generalized to weighting matrices. An important special case for this is of course the ability to choose different features in different areas of the image: which would imply multiplication to the left with a diagonal binary matrix where the ones select the “regions of interest” and the zeros select where the constraint does not apply. The matrix does not need to be binary, it could instead measure an “importance” or a “certainty” of the different constraints in various parts of the image.

**An example of an incomplete local basis division**

Here we present a test GLO to demonstrate the possible effects of neglecting some local subspace in the $\mathcal{S}_x \oplus \mathcal{S}_c$ division. We create a test image where the upper and lower rows of pixels consist of a high frequency wave multiplied by random numbers in the sets $[0,1]$ and $[-1,0]$ to ensure that they have opposite sign. On average if we wanted the pixels in between to have as small a gradient as possible, we should be getting a vertical gradient in the result. First we want to punish changes to the initial conditions, this can be done by replacing $\alpha_1$ with $A_1$, a binary diagonal matrix being one for the first and last rows. Then the corresponding feature matrix simply becomes $F_1 = I$. Next we want to punish large gradients and chessboard effects on the result. We do this by placing the gradient filters together with the chessboard filter into the complementary space:

$$\mathcal{S}_c = \left\{ d_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} , \ d_x = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} , \ h = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \right\}$$

(3.6)

Now, let the GLO term for the chessboard filter be:

$$\epsilon_h \| H(v + d) \|_F^2$$

(3.7)

We see the effects for varying values of $\epsilon_h$ in figure 3.2 on page 32. By continuity, the smaller the $\epsilon_h$ is, the closer it is to being left out of the subspace division. Also beware that too large regularization parameters can give strange and undesirable results. How to tune parameters for a particular application is outside the scope of this thesis.
Figure 3.2: From upper left to lower right: $\epsilon_h = 10^{-\{5,3;2,1;0,-3\}}$
Local neighbourhoods and filter validity

One practical consideration that is important to mention is to be careful with validity of basis functions. This can be understood if we consider conventional convolution on a signal. At the borders, it is difficult to say. If one constraint should hold in a specific RoI, the equations should only be active at the pixels where the filters fit the RoI. In the example below, this is a non-issue because there the filter is the identity filter so the set of valid pixels equals the RoI.

From validity of RoI to certainty adaptation

We can extend RoI from being a discrete 0/1, inactive/active constraint, to being a continuous constraint, where for each neighbourhood and each pixel we have a certainty both for the data, as well as for the equations. In that setting, our validity mask above turns into some kind of local importance measure. We could for instance use normalized differential convolution on the filters on $S_F$ and $S_C$. This allows for even more adaptability of the data as well as increased freedom in formulating GLO.

Scales and Resolution

When there exists some low-pass constraint $F_j$ with a large corresponding $\alpha_j$, multi-scale decompositions become possible. We can then continue our processing on a sub-sampled $F_jd$ to allow for a pyramidal [41], sub-band, or wavelet [42] processing.

Also it does not even necessary have to be a low-pass filter. For instance, in vector analysis, we may want to further investigate high-pass responses, such as divergence, rotation, potential, gradients, laplacians et.c. Then we can use other sub-space-divisions. This leads to wavelet packet analysis or even adaptive wavelets [43], but in-depth investigation is outside the scope of this thesis.

3.3 Local linear descriptors

The GLO is a very powerful framework - able to incorporate a multitude of filters and techniques into the optimization. This section give short descriptions to some of the previously introduced methods, techniques & filters which can be expressed in a GLO.

Generic differential operators

Differential operators of any order are possible to express, as long as a suitable discretization can be found. Examples are the $\nabla$ operators for scalars as well as for vectors. These operators are used in all of the regularization
and denoising applications. This also opens up a wealth of expressibility because any (discretized) differential equation become possible to approach - even higher order differential equations. For instance by mimicking models in the physical sciences.

**Structure tensors**

Structure tensors \[44\] are constructed to represent local orientation in images but are also used in various physical sciences. For instance thermodynamics, fluid-dynamics, aero-dynamics and electromagnetism. They have some properties which have proven very useful in image processing for adaptive filtering and it is always an asset to be able to use them.

**Polynomial expansion**

Polynomial Expansion \[45\] is a technique which uses filters designed in the spatial domain. Since all linear filters can be expressed within the GLO framework, also Polynomial Expansions can be incorporated. They are possible to factor / decompose for speed, a feature which carries over very nicely to certain algorithms for solving GLOs.

**Normalized convolution**

Since the GLO is able to express adapted filters at each point in the data set, both ordinary normalized convolution as well as normalized differential convolutions \[28\] can be incorporated into the cost functions.

**Separability and decomposition**

In the cases, when a technique is factorizable / decomposable, this can be utilized to speed up computations into several of sparse linear solvers, especially among the Krylov Subspace methods.

**Multiple resolution algorithms (MRAs) and wavelets**

If a low pass component punishes \(v\) enough, then effectively stops frequency components below that frequency from being processed. We can then make a MRA to make multi-scale processing if we want to do processing on the lower frequency components.
Chapter 4

Applications

4.1 Overview

Here we explain the various contributions made using the GLO framework. The explanations aim to start with simple concepts in low-dimensional data to gradually expand to covering more complicated content.

Global image denoising: the GLO-LoSTM denoiser

We start out with an application to gray-scale image denoising. This is because it is conceptually easier to explain due to the data only having one inner dimension i.e. it is a scalar field. The result is the GLO-LoSTM denoiser. Here we use a local structure tensor metric which gives a well structured sparse system for fast and efficient solutions. We show that there exist parameters for our method which achieve objective performance according to the popular Structural Similarity Index (SSIM) which for several of the classical test images are comparable to the state of the art Non-Local denoising algorithm BM3D.

Global anisotropic regularization

An important sub field of Image Registration is Regularization - the process of noise filtering the displacement fields to achieve a plausible deformation. Assume that we have a prior estimate of displacement vector field \( \mathbf{d} \) which we want to update with an update vector field \( \mathbf{v} \) to obtain posterior field \( \mathbf{d} + \mathbf{v} \) which should be smoother, without destroying the match between the two data sets (too much). This is the first application to result in a publication: paper 1.

\footnote{Local Structure Tensor Metric}
Constraints for preservation of sliding motion

The global regularization of displacement fields displayed the power of the GLO method which lead to the hope that even more expressive and adaptive regularization could be made. In this application, the objective is to be able to express knowledge of borders of objects which are allowed to slide against each other. This is the second application to appear in a publication: paper 2.

Constraints for preserving areas and volumes

In this application, we show that we can use the divergence in a constraint to produce vector field that is “incompressible”. Such a constraint is useful if we want to allow deformable motions, but still not allow expansion or compression of area / volume. As is well known, there are many organs and tissues in the human body where such a deformation is expected. Being able to express such a constraint is therefore desirable. This mechanism is derived and demonstrated in paper 3.

Model integration demonstrated on 4D CT data

Here we show that the previously introduced mechanisms for regularization can be integrated into one single GLO. The RoI mechanism in 3.2 on page 31 can be used to encode which subsets of the data should allow sliding and which shouldn’t. This allows for spatially adaptive Atlas based regularization. This feature is then demonstrated on the DIR-lab set of deformable image registration data - 3D+time computed tomography scans of the lungs. Three GLOs of increasing complexity are used and it is shown that it is possible to reduce the registration error for each new mechanism introduced in the data processing. The regularization models which are tested on the medical data are:

1. Knowledge of displacement for a set of previously known landmarks, some manually added landmarks and minimizing gradients everywhere.
2. Add knowledge of lung border to allow sliding motion (paper 2).
3. Add knowledge of lung border with anisotropic certainties (paper 1).

This capability for model integration is included in paper 3 together with the introduction of the model for in-compressible areas and volumes.

However, this functionality has no section in the Applications chapter.
4.2 Image denoising using local tensor metric

Definitions

\( L \) - 2x2 Low pass box-filter.
\( H \) - 2x2 High pass checkerboard.
\( T \) - Local structure tensor.
\( \nabla \) - gradient operator (2x2 filters)

Anisotropic tensor based GLO for denoising

\[
\begin{array}{c|c}
\text{Space} & \text{Weight} \\
\hline
C_1 = (I - T)\nabla & \beta_1 = 1 - \alpha \\
F_1 = T \nabla & \alpha_1 = \alpha \\
F_2 = |T|^4 & \alpha_2 = \beta \\
F_3 = L & \alpha_3 = \gamma_L \\
F_4 = H & \alpha_4 = \gamma_H \\
\end{array}
\] (4.1)

As we can see, \( \nabla \) together with \( L \) and \( H \) build a complete local basis division. In fact this is precisely the Haar basis for a single resolution. We see that if we substitute the values above, we get the following GLO:

\[
v_0 = \arg\min_v \{(1 - \alpha) \| (I - T)\nabla (v + d) \|^2_F + \alpha \| T\nabla v \|^2_F + \beta \| |T|^4 v \|^2_F + \gamma_L \| L v \|^2_F + \gamma_H \| H v \|^2_F \} \] (4.2)

Experiments and results

Figure 4.1: SSIM scores: Noisy, GLO-LoSTM and BM3D. \( x = \sigma \) of gaussian noise. Best scores for our method which is restricted to a very localized metric and very simple filters. This shows how promising GLO is for signal processing - even a very localized metric is able to to compete with several of the most high performing non-local methods.
Figure 4.2: Left: Original Lena512 Right: Noisy Lena512, $\sigma = 20$

Figure 4.3: Results for Lena512. Left: GLO-LoSTM. Right: BM3D.
Figure 4.4: Left: Original Lena512 Right: Noisy Lena512, $\sigma = 20$

Figure 4.5: Results for Lena512. Left: GLO-LoSTM. Right: BM3D.
Figure 4.6: Left: Original man Right: Noisy man, $\sigma = 20$

Figure 4.7: Results for man. Left: GLO-LoSTM. Right: BM3D.
Figure 4.8: Left: Original man Right: Noisy man, $\sigma = 20$

Figure 4.9: Results for man. Left: GLO-LoSTM. Right: BM3D.
4.3 Global anisotropic regularization using local structure tensor metric

Introduction

Image registration is a field where two data-sets (images or volumes) are mapped onto each other in a way that is “sensible” with respect to a certain application. For instance if registering two images in a video-sequence, the mapping of one image to the next would be related to achieving a “sensible” motion estimation in video coding or in the application of medical informatics - to compare the anatomy of a patient prior to treatment and after treatment. Which organs have moved, changed in size and so on. Usually vector fields are used together with interpolation techniques to represent this registration mapping. The process of regularization is supposed to impose “sensibility” restrictions on the mapping such that the vector field becomes smoother without increasing some “similarity error” too much. Usually such restrictions involve having various differentials of the vector fields to not be too large.

Local structure tensors

Local structure tensors have been used for a long time to represent structure in images. In this application, we use these local structure tensors $T$ to build a local metric to control the regularization.

An anisotropic tensor $GLO$ for regularization

First let us define the feature space $S_F$ and complementary space $S_C$:

1. $F_1 = T$ - punish changes in the direction of the local structure.
2. $C_1 = \nabla$ - punish irregularities in the result.
3. $\alpha_1 = \alpha \in [0,1]$
4. $\beta_1 = 1 - \alpha$

We see that by making these substitutions we get the following:

$$v_0 = \arg\min_v \left\{ \alpha \|Tv\|^2 + (1 - \alpha) \|\nabla(v + d)\|^2 \right\} \quad (4.3)$$

Details of how to build the equation system as well as application to medical image registration and results can be found in the paper \[^2\].

[^2]: “Globally Optimal Displacement Fields in Image Registration” in IEEE-Xplore
4.4 A tensor decomposition for preservation of sliding motion

Expressing structural and tangent spaces

We call $T$ the tensor which represents the local structure of a friction-less surface. We do not regard how this information is gathered, we just assume at this point that such information is possible to acquire. Furthermore, we define

$$P = I - \hat{T} \quad : \quad \hat{T} = \frac{T}{\lambda_1(T)}$$

Where $P$ is the complementary tensor. I.e. the tensor which is large in any orientation which $T$ isn’t. So in the neighborhood of a surface, $T$ has at least one large eigenvalue, and $P$ has at least one small eigenvalue, with $\lambda_P = 1 - \lambda_{\hat{T}} \in [0,1]$ for each such pair of eigenvalues. Far from any friction-less surface, $P$ will be the identity tensor, but close to it will span the local tangent space of the surface.

Oriental decompilation

We can now perform the decomposition of a vector field $d$, we can decompose into normal and tangential components with the construction above:

$$d_n = \hat{T}d, \quad d_t = Pd : d_n + d_t = Pd + \hat{T}d = (P + \hat{T})d = Id = d$$

So this decomposition is sure to capture all information in our vector.

$d_n = \hat{T}d$ is the normal component and $d_t = Pd$ is the tangential component. We now have the basic building blocks required for our application.

The tangent costs

For simplicity, let’s assume vectorization of $d : [d_1, \ldots, d_N]^T$, where $d_k$ is the vectorization of the inner dimension $k$. This will make possible the elegant and efficient formulation using Kronecker products involving the gradient operator later. We will now derive the smoothness in the $P$ orientations.

We want the vector field to be as smooth as possible along the tangent space. The tangential change of the gradient in the tangential orientation is given by:

$$(I_N \otimes P) \nabla P$$

$P$ first picks out tangential components of vector field. Then the gradient calculates all differentials, the “smoothness” of the tangential components.

---

3. Again, by Brauer’s theorem together with $\lambda(T) \in \mathbb{R}^+$ eigenvalues, eigensystems of $T$ and $P$ will coincide.
Finally we only want to punish the parts of those gradients which are in the tangential directions, so we have a final multiplication on each scalar gradient by $P$ from the left. Since we want to disallow irregularities in the tangent orientation on the result, this constraint belongs to $S_C$.

**The normal costs**

Now we investigate reasonable costs in the orientation normal to the surface: $\nabla T$, we want to remove any irregularities of the normal component. But this time, we want to do that on the changes of the field. Therefore this constraint will be in the $S_F$.

**Filling out the spaces**

For the gradients above, we choose the $2 \times 2$ partial differential filters above. We then add the checkerboard filter $H$ to our $S_C$ space, not wanting any such high frequency components in our results. Also the mean value box filter $L$ is used, punishing the results of low pass components $S_C$.

**Accumulating the GLO**

\[
\begin{align*}
\textbf{v}_0 = \arg\min_{\textbf{v}} & \left\{ \| (I_N \otimes P) \nabla P(\textbf{v} + \textbf{d}) \|_F^2 + \| \nabla T(\textbf{v}) \|_F^2 + \\
& \| (I_N \otimes L)(\textbf{v} + \textbf{d}) \|_F^2 + \| (I_N \otimes H)(\textbf{v} + \textbf{d}) \|_F^2 \right\} \\
& \quad (4.7)
\end{align*}
\]

**Test images and initial conditions**

We have two test images with two test cases each:

1. Test image with flat border and known movements along border on each side, but no known motion far from the border.
2. Test image with flat border and unknown movements along border on each side, but known translative motion of two objects, far from the borders.

3. Test image with circular border and known motion of two objects, one inside and one outside.

4. Test image with circular border and known motion of two objects, both inside.

**Velocity vs. displacement**

Since the constraint on our vector field is tangential, our regularized field is actually valid for velocity fields, and not displacement fields. However as we know from physics, we can get displacement from velocity by integration. It is a bit outside the scope of this application to consider this, but we will return to this as it is important in practical use cases.
Figure 4.11: Setup for the planar surface experiments.

Figure 4.12: Results for the planar surface experiments.
Figure 4.13: Setup for the circular surface experiments.

Figure 4.14: Results for the circular surface experiments.
4.5 Regularization for preservation of volume and area

We have previously constructed several GLOs with the $\nabla$ operator. In this application we will find use for yet another of the classical tools from vector analysis: The divergence $\nabla \cdot \mathbf{v}$ of vector field $\mathbf{v}$. In physics divergence is used to describe natural phenomena in many fields such as mechanics, electricity, magnetism and fluid dynamics. We will here show that we can create a GLO which finds a regularized deformation that preserves the area of objects.

Definition

The divergence is the trace of the gradient. For 2D, we have:

$$\text{div}(\mathbf{v}) = \nabla \cdot \mathbf{v} = \text{Tr}[\nabla \mathbf{v}] = \text{Tr} \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} \end{bmatrix} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \quad (4.8)$$

The GLO

As usual, we first explain the symbols and costs:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1 = I_2 \otimes I$</td>
<td>Punish changes to the vector field.</td>
</tr>
<tr>
<td>$B_1$</td>
<td>Diagonal RoI matrix. Replaces $\beta_1$</td>
</tr>
<tr>
<td>$C_1 = \nabla$</td>
<td>Punish gradient of results equally.</td>
</tr>
<tr>
<td>$C_2 = I_2 \otimes H$</td>
<td>Punish checkerboard effects on result.</td>
</tr>
<tr>
<td>$C_3 = I_2 \otimes L$</td>
<td>Punish box-filter on result.</td>
</tr>
<tr>
<td>$C_4 = \nabla \cdot (\cdot)$</td>
<td>Punish divergence on result.</td>
</tr>
<tr>
<td>$C_5 = I_2 \otimes D$</td>
<td>Punish vector norm on result. $D$ is inertia.</td>
</tr>
<tr>
<td>$\alpha_1 = \epsilon_n$</td>
<td>Parameters for stability.</td>
</tr>
<tr>
<td>$\alpha_2 = \epsilon_H$</td>
<td></td>
</tr>
<tr>
<td>$\alpha_3 = \epsilon_L$</td>
<td></td>
</tr>
<tr>
<td>$\alpha_4 = \alpha_I$ &amp; $\alpha_5 = \alpha_D$</td>
<td>Parameters for incompressibility &amp; inertia.</td>
</tr>
</tbody>
</table>

$$\mathbf{v}_0 = \arg\min_{\mathbf{v}} \left\{ \epsilon_L \| (I_2 \otimes L)(\mathbf{v} + \mathbf{d}) \|^2_F + \epsilon_n \| \nabla (\mathbf{v} + \mathbf{d}) \|^2_F + \epsilon_H \| (I_2 \otimes H)(\mathbf{v} + \mathbf{d}) \|^2_F + \\
+ B_1 \| (I_2 \otimes I)\mathbf{v} \|^2_F + \alpha_I \| \nabla \cdot (\mathbf{v} + \mathbf{d}) \|^2_F + \alpha_D \| (I_2 \otimes D)(\mathbf{v} + \mathbf{d}) \|^2_F \right\} \quad (4.9)$$
**The experiment**

We have a 2D test setup with 2 equally large parallel plates (lines) at a distance from each other. The lower plate should be fixed in position and the upper plate is forced downwards with a constant velocity. In the middle is a circular region which is supposed to be squeezed. We then approximate an integration along our vector field by repeated interpolation to get a displacement field from our velocity field.

**Results**

![Experiment setup: two thin plates of known motion.](image1)

![Experiment result: deformed circle after 20 iterations.](image2)

Figure 4.15: Left: Setup for experiment. Right: Resulting velocity field.

![Experiment result: deformed circle after 33 iterations.](image3)

Figure 4.16: Resulting deformation after 20 and 33 iterations.
Figure 4.17: Experiment with object not touching the upper plate and different inertia on the surrounding media and the object.

Figure 4.18: Experiment with object touching the upper plate and different inertia on the surrounding media and the object.
Chapter 5

Discussion

5.1 Overview

This chapter will contain various discussions regarding the thesis. These are broken down into four categories:

1. Results and Performance
2. Comparisons with former methods
3. Implementation details
4. Future work

First we will review the results and discuss the performance of the methods.

5.2 Results and performance

The denoiser with local structure tensor metric achieved SSIM values which are in range to compete with several high performing non-local methods. This indicates the power of the GLO framework. When comes to the regularization applications it is more difficult to compare performance. For paper 1, we showed how the parameter $\alpha$ affected both error measures, and smoothness measures (gradient and laplacian) when registering one MRI brain slice against another. As the parameter $\alpha$ changed there was a convincing increase in smoothness without increasing the error much at all.

In demonstrating model integration the concepts were proven using clinical data of the Dir-Lab data set of lungs. These experiments showed that incorporation of more and more models into the GLO could give a better resulting field with smaller matching errors. It was also possible to see by making interpolated animations of the clinical data that the functionality of allowing a sliding motion gave a convincingly realistic motion.
5.3 Comparing GLO with former methods

Here we will discuss some of the conceptual likenesses and differences between the GLO approach and the approaches presented in the literature.

Starting out with the concept of *certainty*. In the paper [27], Normalized Convolution has the objective to bring certainty into the data processing. In the paper [29] non-linear gating functions were used to ultimately calculate regularization weights which were to fulfill the same purpose. The total variation approach [25] uses a weighting matrix $W$ to encode certainty.

As it comes to the GLO framework, there exist several capabilities to deal with certainty in this context. It can be used to encode localization and model selection, the Region of Interest (RoI) mechanism described in the GLO framework section 3.2 on page 31. Various tensor constructions inside the norms have also been shown to be able to have this functionality.

Two concepts required to perform the proposed GLO applications are local structure and local orientation. These are implemented with the use of structure tensors [14] to achieve the anisotropic functionalities of (paper 1) and (paper 2). However, among the literature we have found, such functionality is in general not present. Therefore this is an advantage of our method: the ability to specifically encode the local orientations of the data processing.

The biggest advantage of using the GLO framework is likely the functionality to simultaneously incorporate all the types of regularization at once into one big optimization problem.

Furthermore, the representation for the surface between slipping organs in [31, 32] has the drawback that in 3 or more dimensions, it is not able to encode intrinsically 2D neighborhoods - there is only one normal direction $n$. But in a tubular structure such as a cylinder or blood vessel or along the edge of a box, such a representation would not be able to encode the surface which allows sliding - which has two orientations or two dimensions which are close. Here our contribution is mainly in using the tensor representation of the local structure where the tensor is able to hold a local ON-system of eigenvectors where the eigenvalues are large in the orientations of the local surface. In this sense (paper 2) can be viewed as an expansion or improvement on these works. But it is important to note is that these before-mentioned papers also use the calculus of variation and corresponding Euler-Lagrange equations to find a solution, which is conceptually different from our approach of building a global linear optimization problem based on tensor weighted gradients instead of divergences and diffusion equations.

When comes to the incompressibility constraint there exist several differ-
ent approaches proposed in the literature but it is difficult to make direct comparisons. What is clear however, is that there are no claims that the other methods for preservation of area or volume presented here are contained in or expressible in any generic framework which is one of the main features of the GLO.

Last but not least, the fact that the GLO approach is a framework allowing several simultaneous models described above makes it a unique approach. None of the previously described methods in the papers refer to any framework where all these functionalities can be combined into one big context adapted data-process.

5.4 Implementation details

The implementation done is in Matlab and uses built in sparse solvers. Both the “backslash” (matrix-vector division) operator and “pcg” (preconditioned conjugate gradient) have been used with success. The running time for solving can be quite dependent on parameters, but typical time it can take to perform the whole of the LoSTM denoising algorithm on a $512 \times 512$ image is 3-4 seconds. Although much of that time is spent on running rather unoptimized Matlab code and only a fraction on actually solving the GLO. This can be compared to the BM3D Matlab code which took about the same amount of time for the “non-fast” mode.

The $64 \times 64$ pixel large test-example GLO for chessboard effects took 0.01 s to solve with Matlab. The circular sliding object test image ($256 \times 256$ pixels) GLO took 3-7 seconds for the various cases using a home-built conjugate gradient solver and about 0.9 second for the built-in Matlab “backslash”. However the home-made could solve non-sparse systems (as when translation of the big disc was prohibited) in the same time frame, but the Matlab “backslash” for which was needed to explicitly calculate $S^TS$ where $S$ is a non-sparse row, required first lots of memory and took 163.7 seconds to just solve it. It is reasonable to assume a speed which is linear with respect to the number of data points. As 1 second was enough for a $256 \times 256$ 2D-slice, it is reasonable to believe that 256 seconds (4 minutes and 16 seconds) would be an approximate time requirement for processing a $256 \times 256 \times 256$ cube of clinical data in Matlab. However, there exist possibilities for “multiscale” or “multi-grid” processing, fast specialized CPU implementations, as well as GPU acceleration with CUDA or OpenCL. Benchmarks have been done where speedup factors of 5-10 times seem common when going from a high performing CPU to a high-end GPU. All in all these considerations put together with the fact that the processing times measured above were obtained on a 5 years old CPU make it reasonable that the GLO regularizers developed in this thesis can be successfully used as a component in large-scale image registration problems on high-end desktop computers.
Another important aspect is how the memory requirements scale with larger data sets of higher dimensions. As an experiment, if we were to solve an equation system with the $256 \times 256 \times 256$ data set above, how much memory would be required. At first, we would need to store a tensor in 3 dimensions, such a tensor has 6 degrees of freedom each which takes the same amount of space as an image. We would also need a vector field $\mathbf{d}$ with 3 more values at each data point. This becomes a total of $9 \times 256^3 \times 4 > 600 \cdot 10^6$ bytes - assuming single precision floats of 4 bytes each are used. Modern high-end desktops can have between 32 and 96 GB of RAM, so this should not be a problem in practice, but since the number of data points increase as a cube of the side, it will fast becomes unfeasible as the side of the cube grows. A 512 cube would require $4.8 \cdot 10^9$ bytes and a 1024 cube $39 \cdot 10^9$ just for storage. So at some resolution, by the curse of dimensionality we would have to start splitting our data-set into smaller blocks and treat them as separate sub-problems which we can then solve one at a time.

5.5 Future work

There are many ways build new features into the GLO framework to allow for even more powerful models, but perhaps the most interesting addition would be to try how the computational load and memory requirements scale on 3D data.
Chapter 6

Publications

6.1 Overview

In this chapter all the publications will be found. The papers will be presented in the following order:

1. “Globally Optimal Displacement Fields using Local Tensor Metric” (Presented at ICIP IEEE International Conference on Image Processing, October 2012)

2. “Motion Field Regularization for Sliding Objects using Global Linear Optimization” (Presented at ICPRAM International Conference on Pattern Recognition Applications and Methods, January 2015)

3. “Regularization in Medical Image Registration using Global Linear Optimization” (Submitted to the Hindawi journal International Journal of Biomedical Imaging)
Bibliography


Papers

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-117024