Robust Alignment of Wind Prognosis

Rune Klarskov Jensen

LYNGBY 2004
EKSAMENSPROJEKT
NR. 10

IMM

Trykt af IMM, DTU

Preface

This thesis is the result of work done over five months at the Mathematical Statistics Section, Department of Informatics and Mathematical Modelling (IMM), The Technical University of Denmark (DTU). The work was done in partial fulfillment of the requirements for the Master degree of Science in Engineering, M.Sc.Eng.

It is assumed that the reader has a basic knowledge in statistics and image analysis.

Lyngby, February 2004

Rune Klarskov Jensen

Abstract

In this thesis a robust alignment of wind-speed prognosis is proposed and investigated.

The thesis presents a study of alignment in two and three dimensions through robust matching methods. The basic methods and techniques used in alignment are reviewed and discussed.

For alignment in two dimensions, the usage of robust alignment is examined. It takes advantage of gradually constraining the search range of the individual point match in the context of the EM - update. This lessens the influence of outliers.

It is investigated how to use a novel algorithm for aligning prognosis of wind-speed to observed energy produce at wind turbines. The novelty lies in the explicit minimization of wind-speed forecasts regarded as images, and making it robust by embedding it in an EM context.

The alignment problem can be addressed using well known linear methods to analyse the non linear behaviour of the problem. Aligning sets of coordinates or directly on pixels is done using different minimization methods. It is shown how the proposed methods are applicable and robust on simulations very close to that of real data.

Keywords: Alignment, Optimisation, Expectation Maximization EM, Levenberg Maquard, Image analysis.

Resumé

I denne afhandling foreslås en robust fejljustering af prognoser for vindstyrke, og denne justering undersøges.

I afhandlingen præsenteres et studie af justering/alignment i to og tre dimensioner under anvendelse af robuste match metoder. De grundlæggende teknikker og metoder gennemgåes, vurderes og diskuteres.

Justeringen studeres i to dimensioner via robuste match metoder. Herved drages fordel af en gradvis afgrænsning af søgeområdet for det individuelle punkt match i en sammenhæng med EM - opdatering. Dette mindsker indflydelsen fra uregelmæssigt placerede punkter.

Brugen af EM i to dimensioner studeres. Det undersøges hvordan, man kan bruge en ny algoritme til justering af fejlen mellem vejrudsigtens forudsigelse af vindstyrken og den faktiske produktion af vindenergi. Nyheden ligger i at justere direkte på vindstyrke data, betragtet som et billede, og at gøre det robust ved at indlejre den i en EM opdatering.

Justeringens problem kan imødegås ved hjælp af velkendte lineære metoder til at analysere problemets ikke lineære opførsel. Justering af punktmængder og direkte på pixel bliver udført med forskellige minimeringsmetoder. Det vises, hvordan de foreslåede metoder er anvendelige og robuste på simuleringer der er tæt på de virkelige data.

Acknowledgments

During the course of the last five months I have received valuable assistance, both of technical and social nature. I would like to express my gratitude to all those who have provided assistance.

First of all I would like to thank my supervisors Dr. Henrik Madsen and Dr. Torben Skov Nielsen for their guidance and for steering my work in challenging directions.

Dr. Poul Thyregod from the Mathematical Statistics Section at IMM is thanked for his good advice related to statistical issues of my work and for his discussions of my work with me.

Dr. Hans Bruun Nielsen from the numerical analysis group at IMM is thanked for his good advice related to numerical issues of my work.

Ph.D. student Martin Drews is thanked for reading and discussing my work.

Ph.D. student Jacob Viborg Tornfeldt Sørensen is thanked for discussing assimilation with me.

I thank the whole Mathematical Statistics Section at IMM, DTU for providing a pleasant and inspiring research and social environment. Especially I thank Hanne Jensen, whom I shared an office with, for fruitful discussions and moral support. Dr. Henrik Aalborg Nielsen for discussions about the thesis work and technical details.

Finally I would like to thank my friends and family for bearing with me in times of high work load.

Most of all, I would like to thank my girlfriend Mira Vidovic for her understanding, encouragement, love and support.

Contents

1 Introduction **15**

1.1 Motivation & Objectives 15

1.1.1 Existing models: 16

1.2 Thesis overview 17

1.3 Mathematical notation 18

1.4 Nomenclature 19

1.5 Acronyms 19

2 Data - Forecasts versus observations **21**

2.1 Forecasts 21

2.2 Observations. 21

2.2.1 Measured wind velocity. 23

2.2.2 Energy produced from wind. 23

2.2.3 Satellite images 23

2.2.4 Radar images 24

2.3 Patterns of wind velocity 24

3 A view on methods for the problem **25**

3.1 How is the error of the prognosis composed ? 25

3.2 What methods are raised by the problem ? 26

3.3 Dividing the tasks aimed at problem 27

3.4 Viewing methods 29

4 Alignment **33**

4.1 Overview 33

4.2 Alignment of point sets 35

4.2.1 Transformation 35

4.2.2 Minimization 36

4.3 Experimental results 38

4.4 Alignment of images 38

4.4.1 Image transformation 40

4.4.2 Image minimisation 40

4.5 Experimental results 43

4.6 Conclusion 45

4.7 Discussion 45

5 Robust alignment **47**

5.1 Overview 47

5.2 Robust Point Match RPM. 47

5.3 Experimental results 51

5.4 Mixture Point Matching MPM. 56

5.4.1 Gaussian mixture models 56

5.4.2 Density Estimation - Expectation Maximization EM 58

5.4.3 Mixture models for point matching - A maximum posterior MAP problem 60

5.4.4 EM and Complete Data Posterior Energy Function . 61
5.4.5 Problems with Mixture model and the EM algorithm 62
5.4.6 Mixture Point Matching Energy function and Algorithm 63
5.5 Experimental results 65
5.6 Modifying the point match to a vector match MPM to MVM 65
5.6.1 Integrating measures of vectors 68
5.7 Experimental results 69
5.8 Conclusion 69

6 Conclusion 71

6.1 Future works 72

A Code 79

B Point minimization. 81

C Pixel minimization. 85

D Robust Point Match. 91

E Mixture Vector Match. 93

Chapter 1

Introduction

This thesis documents the study of methods that can be used as elements in reducing the error in the prediction of wind energy.

1.1 Motivation & Objectives

The feed of electricity from wind energy to the power grid fluctuates. Knowing these fluctuations in advance carries significant economic weight.

Wind energy contribution to the electrical energy supply in Denmark is in certain situations in surplus, with low electricity demand for instance during the night. Predicting these situations will decrease losses. The development of wind turbines with more capacity is advancing, and the number of installed turbines is increasing.

The energy supply from the individual wind farm is predicted in existing models conditioned upon local measurements of energy production in the farm, incorporating numerical weather predictions produced by the Danish Institute of Meteorology .

These local predictions suggests a pattern in the error of the global/spatial prediction of wind velocity and energy production. Locally it is often observed that a rise in energy production is just prior to, or just after, what was predicted in the prognosis [30] . This error can be referred to as a phase error, a phase being the propagation delay between two waves, the wave of actual energy produced and the wave of energy production predicted.

Until now only local systems, are used to predict the energy production. This gives rise to the question about how to coordinate all the local observations of phase error between the prognosis and the production, into a global picture of error and put the error into spatial context. Identifying the spatial error pattern enables a complement to the prediction of local energy production, making it more reliable and precise. This also gives rise to the question of how much and what is to gained thereof.

The conditional expected value of the wind energy production conditioned upon a larger set of relevant informations should improve the prediction. In chapter 3 there is a further view on tasks in the problem of improving the prediction, which is what is aimed fore in the broad sense.
The more specific objectives of this thesis:

- To document and illustrate how spatial error patterns from wind prognosis phase error can be located by alignment.
- To study and develop robust alignment.

1.1.1 Existing models:

Considerable effort has been done into improving local modelling, leading to the development of several models. To mention a few:

- Predictor
  Developed at Risoe National Laboratory. This is a model that uses local measurements,also referred to as off line calibrations, and numerical weather reports, but it does not use other local measurements in spatial context. A model used in Predictor is Wasp. Wasp is a local model for use in non complex terrain. It does not predict but gives the instantaneous wind velocity elsewhere in a region from a local measurement.

- Previento
  A model like predictor, still no spatial correction of forecasts, a description can be found in [38].
- WPPT
  Developed at the Department of Informatics and Mathematical Modelling I.M.M. at D.T.U. in corporation with the power companies ELTRA and ELSAM. The model predicts wind production in a large area. Meteorological wind speed forecasts, local measurements of climatic variables and locally measured energy production from wind turbines are used. The wind farm power prediction from a local area is then up-scaled.

A reduction in a spatial context is done in [39], where spatial smoothing is used in order to assimilate the wind power production into the previento model.

1.2 Thesis overview

The thesis is divided into the following chapters:

- **Chapter 1: Introduction.**
  Introduces the issues associated with the project and presents the goals.
- **Chapter 2: Data - Forecasts versus observations.**
  A short description of the behaviour of the data at hand. Illustrates some of the qualities and properties of the methods investigated for solving the problems.
- **Chapter 3: A view on methods for the problem.**
  Deals with the main elements of the problems and specifies the tasks in the attempt to solve them.
- **Chapter 4: Alignment.**
  Introduces the method of alignment of images as an analytical problem.
- **Chapter 5: Robust alignment.**
  On top of the alignment task comes the embedding of the alignment into a robust framework. The EM approach is studied.
- **Chapter 6: Conclusion.**
  Adds up and judges results. Outlines problems and suggests future tasks.

Chapter 4 and 5 have been written with the intention that they can be read separately.

1.3 Mathematical notation

Unless otherwise stated, the following notation is used in the text.
Vectors are column vectors and typeset in non-italic lower-case boldface using spaces to separate the elements:

$$\mathbf{x} = [a\ b\ c]^T. \tag{1.1}$$

Matrices are typeset in non-italic boldface capitals:

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{1.2}$$

Identity matrices are typeset as:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \tag{1.3}$$

Inner product between two vectors is typeset as:

$$\mathbf{a}^T\mathbf{b} = \sum_{i=1}^{N} a_i b_i. \tag{1.4}$$

Sets are typeset using curly braces:

$$\{x_i\}. \tag{1.5}$$

1.4 Nomenclature

Unless otherwise stated, symbols have the meaning shown here:

$\mathbf{z}$	A 2D shape vector.
$\mathbf{Z}$	A 2D shape expressed as a matrix.
ψ	Rotation angle for a shape, $\phi \in [0; 2\pi]$.
β	Isotropic scaling of a shape, $\beta \in \mathbb{R}_+$.
$\gamma = [\gamma_x,\ \gamma_y]^T$	Translation of a shape, $\gamma \in \mathbb{R}^2$.
$\boldsymbol{\Theta}$	Vector containing the parameters for a similarity transform.
$\mathbf{J}_f$	*Jacobian matrix* containing the first partial derivatives of the function f
$f'(t)$	The first derivative of the function $f(t)$.
$f''(t)$	The second derivative of the function $f(t)$.
i	Index of points in a set.
$\boldsymbol{\Sigma}$	Covariance matrix.

1.5 Acronyms

Acronyms used in this thesis:

NWP	Numerical Weather Prognosis.
WPPT	Wind Power Prediction Tool.
HIRLAM	High Resolution Local Area Model.
IMM	Department of Informatics and Mathematical Modelling.
DTU	Technical University of Denmark.
RPM	Robust Point Matching.
MPM	Mixture Point Match.
MVM	Mixture Vector Match.
DMI	Meteorological Institute of Denmark.
SVD	Singular Value Decomposition.

Chapter 2

Data - Forecasts versus observations

The data relevant for this project is in several segments, but they can be divided into two main segments, the forecasted and the observed.

2.1 Forecasts

A forecast segment is the numerical weather prognosis NWP from the meteorological institute of Denmark DMI, which is an output from the high resolution local area model, HIRLAM. The prognosis output forecasts wind velocity in a grid with a chosen scale of 10 km in the height of 10 meters, see figure 2.1.

2.2 Observations.

There are many types of observations relevant to the forecast, all of which only fit part of the sampled format of the forecast grid. In the next four subsections some of them are introduced.

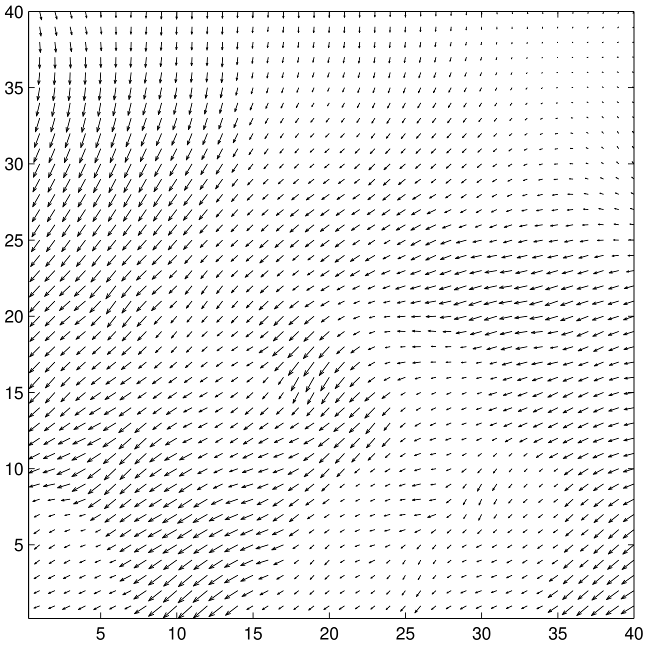


Figure 2.1: Forecasted wind velocity in the prognosis from the meteorological institute of Denmark DMI, a 40 by 40 grid.

2.2.1 Measured wind velocity.

An observation segment of data is the measured velocity at 16 wind turbine locations on a scale similar to the forecast. The observations are all within the same area as the prognosis, but they are not all located at the same grid points.

2.2.2 Energy produced from wind.

Another observation segment is the observed energy production at wind turbine farms. This measurement is somewhat noisy due to a nonlinear relationship between the energy produced and the wind-velocity. The locations of the farms are, like the measured wind velocity, all within the area of the prognosis, but they are not all at the same points as the grid points of the forecast. An advantage here is that the number of turbine parks is higher than that of the measured velocity.

2.2.3 Satellite images

Yet another observation segment is the images of the area of interest from a satellite. These come with a higher resolution than the forecast. The movements of the clouds between these images gives an indirect measurement of the wind velocity at 10 meters, since the clouds are situated at other heights than 10 meters. On top of this, is the uncertainty owing to whether or not clouds are actually present in the image. The movements of the clouds indicate the movements of pressure systems. In the thesis by R. K. Jensen[18] velocities in an image sequence is estimated via wavelets, which are directional multi scale filters. A key element in [18] is the use of Gabor filters to estimate dense image flow from [23]. The latter article shows how wind velocities are estimated through estimation of the image flow from satellite images. The thesis studies how the estimate can be brought to a higher precision by the use of wavelets.

2.2.4 Radar images

With a radar the height of waves on the oceans can be measured. This indicates the wind velocity with high resolution and on a broad scale.

2.3 Patterns of wind velocity

An analysis of these data starts by observing that the patterns of wind velocities are structured as a system of pressure, wherein wind gradually changes often in rotational movements see fig. 2.1. Many underlying physical measures other than pressure can be used to describe the pattern of wind movements[40]. The outer edge of the wind rotating is where the pressure has a frontier towards another system of pressure. The movements of these formations of pressure must be considered arbitrary when aiming at improving forecasts (NWP) from for instance HIRLAM. The formations are moving with velocities that are bounded, they move gradually.
Another feature of the data in this analysis is that the formations are non-rigid, their shapes are not homogeneous - they vary in many dimensions. Still the variation is generic, to a certain extend and vary only gradually with time.
An extensive modelling of these formations is done in NWP. This study aims at complementing NWP. This could be via a match between the forecast segment - predictions from NWP and the observation segment - 16 observed wind velocities. The operations on the forecast coordinates that match the 16 velocity-vectors express the spatial structure of the error. Identifying the temporal correlation of this error-structure, forms a basis for corrections of the predictions in the spatio-temporal sense. The operations that match the vector structure express a functional mapping from NWP to observations. This functional mapping in the spatial domain is usually referred to as warping (the word can be directly translated to a multi-angular bending of a piece of paper without making it deformed i-regressively).

Chapter 3

A view on methods for the problem

3.1 How is the error of the prognosis composed ?

Estimating the formations of pressure and the expectation of their movements forms a foundation for predicting the velocity of the wind. The estimated expected movements of the systems are then evaluated in the second set of data - the observations. This is done with the goal of reducing error in the prediction already available from the NWP.
It is to be achieved through correction of the patterns of movements in the groups of velocities.
To a certain extent this is achieved through a warping, although the formations of pressure only present themselves partially in this spatial domain.

Could the observations be fed into some verification package in the numerical weather prediction ?
According to the description of the HIRLAM project [40], assimilating data into the model requires more dimensions of measurements than just the wind velocities - the NWP has a high resolution and many more parameters as input.
Observations of wind velocities at the wind turbines are sparse, and sam-

ples of wind measurements at the turbines are not at the same scale as the HIRLAM model used for the NWP. The cost of purchasing other outputs and running the NWP with verifications is beyond the scope of this thesis work.

In [19] it is shown that using NWPs from HIRLAM with higher resolution will enable local predictions to take bursts of wind into account. But there are still errors caused by either, local conditions such as the coordinates of a lake and the like, or by large scale atmospheric parsings of fronts between low and high pressure systems.

In [30, 2] there are several articles dealing with the errors caused by local conditions.
This study deals with the mentioned errors caused by large scale atmospheric parsings of fronts between low and high pressure systems.

3.2 What methods are raised by the problem ?

There are numerous implementations and results of matching algorithms [11]. Making an educated choice is a task of reading through a lot of material like publicised articles and lecture notes, and consider them in the context of the problems arising from the data.
Matching two sets of vectors changes the task from what is generally known as point matching - that is warping, in the community of researchers into statistical images analysis. The matching of vectors is like matching points, only now the points have an extra feature or degree of freedom. The matched points are supposed to have a point in common from where they form a vector, a common origo. When matched, each set of two points have a common coordinate system, making up the system of change for the individual vectors. Given that the amount by which the vector can change is bounded, this system of vector change is bounded by an absolute value. One can imagine that the coordinate system of each vector is enclosed by the unit circle multiplied by a real number. This changes the structure of the known algorithms [9][7][8], which are focused on matching sets of points. Instead they must be modified according to the task of matching, not only sets of coordinates for points but also the vectors associated

with the points, in accordance with the conditions in the data studied here.

3.3 Dividing the tasks aimed at problem

The goal is to work directly with the real data sets. Building up the partial implementations of prototypes on synthetic data set and modifying the algorithm gradually, however, has proven very productive in other works. Here is a suggestion of some steps:

- Point set to point set.
  This first step consists of modifying the known algorithm from a non rigid point point matching algorithm to a that of a rigid point matching one, considering first of all the affine transformations, rotations and translations. Making it simple by keeping two sets of points, source and destination of equal size. In this step there is the minimization and making the match a robust one. This step is chosen in order to focus on what the description of the error is aimed for. The local improvements of prognosis is a kind of non rigid adaptation. Doing it on points proves that it can be done on pixels, since the point could just as well be landmark points such edges, corners - i.e. points found with features such as stationary points when partially differentiating the surface that makes up an image. It must be kept in mind here that non rigid a point match that takes account of the local deformation can be decomposed to the rigid and non rigid components.

- Image to image.
  The next step further is making the match work with sets of magnitudes of vectors. These magnitudes on the grid makes up an image. The reason for this choice is three fold. Firstly, this step is gradually increasing the complexity of the algorithm to that of an image match with vector match in sight, secondly there are more observations of energy produced than observations of wind velocity and thirdly the aim of this study is to improve the prediction of energy production. As opposed to a match of vectors this is a shortcut, but it can be the basic framework that can include more information conditioning the expected values. Simulating the error between these magnitude

images can be used as a provisional measure in the evaluation of algorithms/methods.

- Some observations to many.
  An image match is evaluated on simulated error images, paves the way for real data is to be used. Fitting a set of observations of energy produced to the grid format of the prognosis, is a matter of interpolating the values at the grid point of the prognosis, where there are not any wind turbines available. Points are inserted between the actual turbines, values at these points are approximated as expected values, most commonly as averages of neighbours. This corresponds to a low pass filtering against aliasing, known from signal processing. With an increased number of points, the values of points corresponding to grid point coordinates of prognosis is approached, and samples of the interpolated observations at these grid points is obtained. Interpolation considered as estimation is done by inferring a non stationary spatial process, called krigging in [22] [42]. Once the image of observed energy produced is interpolated, then it can be used in the image matching.
  A relatively smaller set of observed non-equidistant magnitudes of vectors can be matched to a larger set of equidistant ones, this demands a certain model- fitting. The EM-algorithm is described as especially good for fitting a model to an incomplete dataset in [25].

- Some linearised observations to many.
  There is a non linear relationship between the magnitude of the wind velocity and the energy produced from wind turbines [30][2]. Linearised energy outputs from turbines contribute information to the error correction. Using the absolute value of the wind velocity constrains the information for the match.

- An even smaller set of vectors to many.
  The smallest set of observations are the non equidistant wind velocity vectors. Although this is a small set, the fact that they are *velocities*, that is vectors, not just magnitudes, increases the amount of information in the individual observation. The prognosis is given in sets of vectors on grids. Repeating the procedure used for matching magnitudes only now with vector's can create another error correction.

- Conditioning of the expected value of the corrected prognosis.
  The conditional expectation is the optimal predictor, increasing the amount of observed information, strengthens the analysis of the error data. Evaluating the parameters found in matches of prognosis with both measured velocities and linearised energy production increases the amount of observations, and thereby improves the post processing of the error-correction. In [33] the issue of data assimilation is being dealt with. A model, e.g. HIRLAM, and the observations can both be regarded as stochastic processes. When these processes are estimated according to each other, and this is incorporated into the model, then the observations can be considered assimilated. If parameters are identified, that estimates the relationship between the wind velocity prognosis, from HIRLAM at 10 meters hight, and corresponding observations in a large area, covering Denmark and parts of neighbouring Sweeden, Norway, Germany and England, then this post processing of observations raises reasons for incorporating these parameters into the HIRLAM. Assimilation can be done.

3.4 Viewing methods

Even though it was chosen first of all to focus on the transformations, rotation and translation quite a few other transformations in various different warping algorithms have been considered. This is done in order to prepare for possible improvements in methods discussed. It takes theoretical consideration, choosing which methods to investigate further.
Considerations on what model to use for non rigid three dimensional registration is done in [4], where it is shown how a geometry constrained diffusion, invariant to similarity transforms, can be done using Gaussian convolution. In the paper [31] it is shown how a Brownian warp (analogue to Brownian motion)[3] is superior to a thin plate spline model for matching, since the problems of folding in the spline model do not occur with Brownian warps. Further-more, it is considered how the warp is made symmetric w.r.t. source and destination, i.e. the warp can go both ways. A central point in [31] is that radial basis functions is used. Spline functions [5, 29] are mathematically equivalent to the physical spline used by draught people. A physical spline is a thin flexible rod held fixed at certain points but free to move between them, subject to laws of physics. Holding some points fixed reveals a prior commitment, which is what [31] addresses,

seeking a least committed prior.
The study of warping methods and what parameters can be registered in order to transform one set of points to another is studied as point distribution models [36, 12] - also called shape models. Considering shapes see

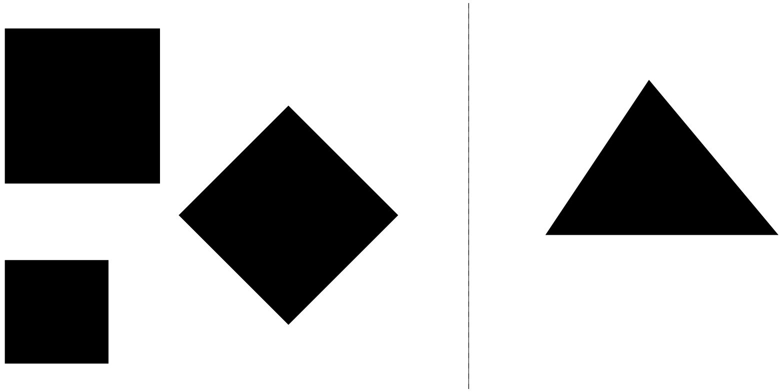


Figure 3.1: The objects on the left of the dashed line vary in size and position, but they have the same shape, different from the one to the right.

illustration 3.1 from [12].
These are usually distributions of homologous points - landmark points where similarity transformations, rotation, translation and scaling, are filtered out. When the point sets have these similar transformations done to them, they are aligned. A set of point sets aligned fall close to each other forming a distribution at each point. Some reference shape - conditional expected value of points - is then, for instance, the mean of each point distribution see illustration 3.2 from [12].

These point distributions are often described by the use of multivariate statistics such as principal component analysis, however, there are also other methods for parameterisation of shape variations - variation of point distributions. Staib and Duncan[34] uses Fourier modes, Neuman and Lorenz uses Fourier and wavelet descripters[28]. When it comes to wavelet descriptors an interesting 'best basis...' algorithm is found in [13]. Aligning shapes alone is done in shapes of biological objects[32]. A study of central moments of the order zero to infinity of point sets in the thesis by R. K. Jensen[17] covers a basic statistical approach to shapes. It shows

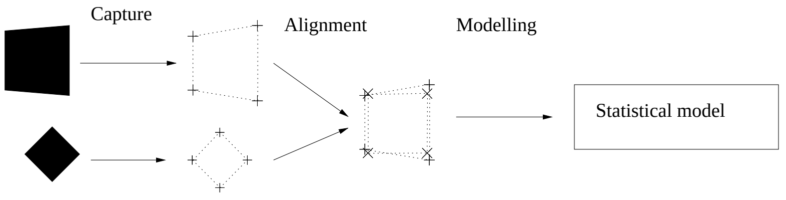


Figure 3.2: How the point distribution model is build. The three steps, capture, alignment and modelling. The two objects have similar shape - their landmark points are captured, aligned into a reference frame and statistics done to the figures in the reference frames.

how shape variation is more pronounced as the order of the moments approaches infinity. This is utillised for classifying shapes, from images that are warped/transformed. The transform is one specially developed for sidescan sonar images of the seabed.

Chapter 4

Alignment

4.1 Overview

Alignment of images have the alignment of point coordinates as a central and basic underlying issue. Many paths of compositions of basic methods can be evaluated against each other in order to reach the best possible results. This chapter will covers both the study of the coordinate and the pixel-wise alignment.

When forms of objects are measured, one often measures so called landmarks on the object, that is easily identifiable points on the objects that are somehow characteristic for them and repeat on every points set that makes up a shape. An example is the edges that are found as stationary points when differentiating an image. Stationary points carry a high content of information and appear at the edges of the surface that an image can be considered as. They are analysed and considered as singular values and used to recreate an approximation of an arbitrary image with very low error in [26].
Measurements of landmarks are prone to errors like all real world measurements. Even "worse", a landmark might be missing from an object, a missing wind mill park, or the object might have a shape that is quite different from other shape in the same class (also called outlier). The method that aligns shapes should not be greatly influenced by these sources of errors (best of all, not at all) and should be robust against them.

Another approach that is used is direct workings on the images, without reducing the data by deriving landmarks, with what noise that might bring into the analysis. A key issue here is the amount of data - pixels. More specifically in this instance, the supply of wind velocity points in the prognosis amounts to a grid with a size of only about 40 by 40, which makes it a small image considering the computing power available at present.

The most common method of aligning shapes is the Procrustes alignment[12, 35, 15]. It is often used when building point distribution models[36]. It minimises the Procrustes distance between shapes[15]. It is a least squares measure of shape similarity. It does not handle outliers and errors in a robust way. The algorithms currently used to perform Procrustes alignment are not capable (at least in a sensible way) of handling missing landmarks. Another downside to Procrustes alignment is that it tries to smooth out differences between shapes so that the real difference between them often gets obscured, that is it is not resistant to outlier shapes. Ideally the alignment methods should be robust when dealing with incorrect information (such as wrong correspondence between points) and resistant to the influence of outliers.

The algorithms presented here to align point sets are based on well understood and efficiently implemented numerical methods. They align point sets and pixel sets (images) in the sense of the l_2 norm. The formulation presented allows only to a certain extend for dealing with missing observations and the norm is only to a certain extend robust when dealing with errors and resistance to the influence of outliers. In [12] it is shown how it would be a natural extension to Procrustes alignment to use other norms for alignment, this consideration is also done in [21] with regards to optimisation. The strength of the methods derives from the fact that they are standard well understood algorithms. Many of the descriptions in the first two subsections are from [12].

4.2 Alignment of point sets

4.2.1 Transformation

The basic task of aligning, also known as superimposing, two shapes consists of rotating, translating and scaling[1]. That is estimating a similarity transform of one shape $\mathbf{z}_1$ so that it fits in some optimal way to another shape $\mathbf{z}_2$.

A similarity transformation of a point (x, y) can be expressed as

$$x' = x\beta\cos\psi - y\beta\sin\psi + \gamma_x \tag{4.1}$$
$$y' = y\beta\cos\psi + x\beta\sin\psi + \gamma_y \tag{4.2}$$

where β is the scaling, ϕ is the rotation angle and γ_x, γ_y are translation. If shapes are represented by landmark[1] coordinates as

$$\mathbf{z} = [x_1\ x_2\ \ldots\ x_N\ y_1\ y_2\ \ldots\ y_N]^T \tag{4.3}$$

then the scaling, rotation and translation of the whole shape can be expressed in matrix/vector notation as

$$\mathbf{z}' = \mathbf{Z\Theta} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ x_2 & -y_2 & 1 & 0 \\ \vdots & & & \vdots \\ x_N & -y_N & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ y_2 & x_2 & 0 & 1 \\ \vdots & & & \vdots \\ y_N & x_N & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta\cos\psi \\ \beta\sin\psi \\ \gamma_x \\ \gamma_y \end{bmatrix}. \tag{4.4}$$

Here $\mathbf{z}'$ is the transformation of $\mathbf{z}$. The rotation is around the axis of the third dimension, on the origo $(0, 0)$ of the two first dimensions, where the ordinate is normally denoted the x-axis and the abscissa the y-axis.
Rotating around a certain point is often done around the center of gravity of the point set. The center of gravity is the mean value of the individual

[1] Used in a general sense here for both anatomical landmarks, that could be manually chosen, and pseudo-landmarks, e.g. points sampled along the contour of an object.

components, mean value of x and y values. With the intention of both centered rotation and translations, the set of transformations gets divided into three. The three transforms are the translation to the origo by the coordinates that make up the center of mass $\mathbf{\Theta}_1$, the actual rotation $\mathbf{\Theta}_2$ and the return of the point set to the previous center of mass $\mathbf{\Theta}_3$:.

$$\mathbf{z}' = \mathbf{Z}\mathbf{\Theta}_1\mathbf{\Theta}_2\mathbf{\Theta}_3\mathbf{\Theta}_4 \tag{4.5}$$

Then possible transforms with other parameters like scaling and translations is done with $\mathbf{\Theta}_4$ in expression (4.5).

When the angle of the individual sinusoidal functions in $\mathbf{\Theta}$ in eq.(4.4) is not the same then a *shear* appears from the transform.

4.2.2 Minimization

The alignment of $\mathbf{z}_1$ and $\mathbf{z}_2$ can be expressed analytically as

$$\min \|\mathbf{z}_2 - \mathbf{z}_1'\| = \min_{\mathbf{\Theta}_1} \|\mathbf{z}_2 - \mathbf{Z}_1\mathbf{\Theta}_1\| \tag{4.6}$$

where $\|\cdot\|$ is a suitable norm. If the norm is chosen as the l_2 norm then the alignment is the ordinary Procrustes alignment[15]

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_2 = \sqrt{\sum_{i=1}^{N} [(x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2]}. \tag{4.7}$$

Solving a linear system of equations has undergone many investigations and many algorithms have been developed. A description from Gaussian elimination with pivoting through to QR factorization and SVD can be found in [14].

Solving an over-determined linear system of equations in the least square sense, is to find a solution $\mathbf{x}$ to the system,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.8}$$

such that the residual is minimised in the l_2 norm like in eq.(4.6).

$$\min_{\mathbf{x}} \|\mathbf{r}(\mathbf{x})\|_2 = \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2. \tag{4.9}$$

When the problem is over-determined then the QR factorization is preferable, with its orthogonalisation and upper triangulation. The QR factorization is different from LU factorization, which aims at a set of equations that makes up a quadratic matrix. The QR factorisation aimes at solving over deterministic sets of equation which makes up a rectangular matrix. When the matrix is nearly or actually rank deficient, the Singular Value decomposition becomes preferable. The problem at hand is over-determined not rank deficient, which makes the QR facorization suffice.

The solution $\mathbf{x}$ from eq. (4.9) is usually found with the help of a QR factorization of the coefficient matrix $\mathbf{A}$. Since it is used in the study of alignment, the procedure is described as follows:

Let $\mathbf{A}$ be the coefficient matrix of an over-determined system, with full rank and size $m \times n$. It has a QR factorization[14] of the form

$$\mathbf{A} = \mathbf{QR} = \mathbf{Q}\begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix} \tag{4.10}$$

where $\mathbf{Q}$ is an $m \times m$ orthogonal matrix, $\mathbf{R}_1$ is an $n \times n$ upper triangular matrix and $\mathbf{0}$ is an $(m-n) \times n$ block of zeros. Also let

$$\mathbf{Q}^T\mathbf{b} = \begin{bmatrix}\mathbf{c}_1 \\ \mathbf{c}_2\end{bmatrix} \tag{4.11}$$

where $\mathbf{c}_1$ is $n \times 1$ and $\mathbf{c}_2$ is $(m-n) \times 1$. Then

$$\begin{aligned}
||\mathbf{Ax} - \mathbf{b}||_2^2 &= ||\mathbf{QRx} - \mathbf{b}||_2^2 && (4.12)\\
&= ||\mathbf{Q}(\mathbf{Rx} - \mathbf{Q}^T\mathbf{b})||_2^2 && (4.13)\\
&= \underbrace{||\mathbf{Q}||_2^2}_{=1} || \begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{x} - \begin{bmatrix}\mathbf{c}_1 \\ \mathbf{c}_2\end{bmatrix} ||_2^2 && (4.14)\\
&= ||\mathbf{R}_1\mathbf{x} - \mathbf{c}_1||_2^2 + ||\mathbf{c}_2||_2^2. && (4.15)
\end{aligned}$$

As $\mathbf{c}_2$ is constant the optimal solution is obtained when

$$||\mathbf{R}_1\mathbf{x} - \mathbf{c}_1||_2^2 = 0 \tag{4.16}$$

that is, the solution must fulfill

$$\mathbf{R}_1\mathbf{x} = \mathbf{c}_1. \tag{4.17}$$

The problem of solving an over-determined system has now been reduced to solving a linear system with a square matrix. It is solved by back substitution as $\mathbf{R}_1$ is upper triangular.

4.3 Experimental results

The method described in the previous section, can be used directly to align shapes in the l_2 norm. Recall the linear vector function from eq. (4.6). Instead of minimising $\|\mathbf{r}(\mathbf{x})\|_2$ we minimize $\|\mathbf{F}(\phi)\|_2$[2]. Set $\mathbf{A} = \mathbf{C}$ and $\mathbf{b} = -\mathbf{d}$ and follow the same procedure.

The coefficient matrix $\mathbf{C}$ for the alignment problem is very sparse and highly structured. This could be used in an implementation to speed up the alignment. Using sparse matrices together with the backslash operator in Matlab gives quite good speedup and uses much less RAM compared to full matrices.

If any points are missing from a shape, just leave out the corresponding row in the coefficient matrix.

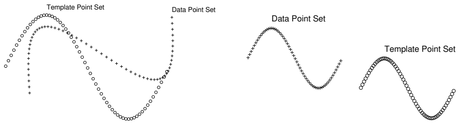
This alignment of point coordinates is working on all angles and distances for point sets, that are not deformed in their internal structure. When a point set is transformed by these affine parameters, then it is possible to align back to the original, given that the affine transforms are the only ones used. The alignment is illustrated in figure 4.1

When noise plays a role in alignment, it can be arising from the sampling of real data, for instance in the choosing of landmarks or from the detection of edges in images. The noise sensitivity of this version of point coordinate alignment has been discussed in [12]. Here it has been chosen to leave the study of noise sensitivity to section 4.5 where images are aligned.

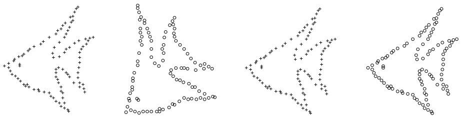
4.4 Alignment of images

In the following last two sections of this chapter there is a description of the alignment of images. This is an extension of the first two sections in this chapter, bringing the methods closer to the data and the main problem of the thesis.

[2] $\mathbf{F}(\cdot)$ is the residuals from aligning



(a) unaligned (b) aligned



(c) unaligned (d) aligned

Figure 4.1: Alignments of point sets/shapes from a sinus curve and a fish. They are transformed and then aligned by the use of the QR minimisation and re transformation. The aligned sets are displayed apart from each other, so that one can tell them apart, otherwise they would sit on top of each other. Just 30^o and a fraction of 10 percent of the scale of the point in translation, is recaptured completely for the sinus-points in subfigures a)-b). The fish points have been captured completely, this time the angle was 45^o in subfigures c)-d)

4.4.1 Image transformation

The linear transforms of images is done in a very similar way of transforming the coordinate values in 4.2, the gradual step forward here is the association of pixel values to coordinates. In this instance the pixel-value under consideration is the magnitude of the wind velocity of the prognosis. The grid points of the prognosis are denoted with coordinates in integer values but the transformed coordinates are not integer values.
This is addressed by a smoothing operation called the bicubic interpolation, wherein the transformed pixel-values of the grid are calculated by an up-sampling and a averaging operation in both the horizontal and the vertical direction. This makes it possible to have the parameters gradually changing with the resulting transformed image also gradually changing, even though its grid coordinates are integers.

4.4.2 Image minimisation

In model fitting a lot of disciplines meet, the different approaches from statistical, numerical and basic functional analysis contribute to the convergence, precision, low computational costs and speed. These different approaches are bound to crossover each other, but with all their combined work, there is the opportunity to make the contributions add up so that their information complement each other. In [6] different modelling methods are mentioned with reference to neural networks. The *Taylor expansion* is central in many minimizations as it is used to find higher order approximations using partial derivatives. This is desirable since the derivative in the first place is a linearization that cuts costs in computing, taking this to a higher order makes it possible to make linear operations although approximating to non linear behaviour of the difference between the model and the data. The latter can make the fit more precise, than the QR - minimization, which treats the problem as if it was only linear. The algorithms perform a gradient descent, using the partial derivatives to reach the argument x^* which minimizes the quantity that makes up the difference between the model and data, also referred to as the cost function $\mathbf{F}(\mathbf{x})$. Here it has been chosen to focus on the Levenberg Marquard minimization [20, 24, 27], because of its properties with respect to robustness and the stopping criteria, when a minimum has been reached. A resumé of [20] is brought for insight.

From section 4.4.1 it can be assumed that the cost function $\mathbf{F}(\mathbf{x})$ is smooth, differentiable to a high order, which makes the following *Taylor expansion* valid:

$$\mathbf{F}(\mathbf{x}+\mathbf{h}) = \mathbf{F}(\mathbf{x}) + \mathbf{h}^{\mathbf{T}}\mathbf{g} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{h}^{\mathbf{T}}\mathbf{H}\mathbf{h} + \mathbf{O}(||\mathbf{h}||^{\mathbf{3}}) \tag{4.18}$$

where $\mathbf{g}$ is the gradient,

$$\mathbf{g} \equiv \mathbf{F}'(\mathbf{x}) \equiv \begin{bmatrix} \frac{\partial F}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial F}{\partial x_n}(\mathbf{x}) \end{bmatrix}, \tag{4.19}$$

and $\mathbf{H}$ is the *Hessian* matrix,

$$\mathbf{H} \equiv \mathbf{F}''(\mathbf{x}) \equiv \left[\frac{\partial^2 F}{\partial x_i \partial x_j}(\mathbf{x})\right], \tag{4.20}$$

where

$$F(x) = \frac{1}{2}\sum_{i=1}^{m}(f_i(x))^2 = \frac{1}{2}||\mathbf{f}(\mathbf{x})||^2 = \frac{1}{2}\mathbf{f}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) \tag{4.21}$$

The *Hessian* matrix of F is also needed with the elements in position (j,k),

$$\frac{\partial^2 F}{\partial x_j \partial x_k}(\mathbf{x}) = \sum_{i=1}^{m}\left(\frac{\partial f_i}{\partial x_j}(\mathbf{x})\frac{\partial f_i}{\partial x_k}(\mathbf{x}) + f_i(\mathbf{x})\frac{\partial^2 f_i}{\partial x_j \partial x_k}(\mathbf{x})\right) \tag{4.22}$$

showing that

$$\mathbf{F}''(\mathbf{x}) = \mathbf{J}_f(\mathbf{x})^T\mathbf{J}_f(\mathbf{x}) + \sum_{i=1}^{m} f_i(\mathbf{x})\mathbf{f}_i''(\mathbf{x}). \tag{4.23}$$

The *Taylor expansion* can now be used for the following approximation, where $\mathbf{h}$ is the step to minimize $\mathbf{f}$,

$$\mathbf{f}(\mathbf{x}+\mathbf{h}) \backsimeq l(\mathbf{h}) \equiv \mathbf{f}(\mathbf{x}) + \mathbf{J}_f\mathbf{h}. \tag{4.24}$$

Inserting this into (4.21) of F one sees that,

$$\begin{aligned} F(\mathbf{x}+\mathbf{h}) \backsimeq L(\mathbf{h}) &\equiv \frac{1}{2}l(\mathbf{h})^T l(\mathbf{h}) \\ &\vdots \end{aligned} \tag{4.25}$$

$$(\mathbf{J}_f^T \mathbf{J}_f)\mathbf{h}_{GN} = -\mathbf{J}_f^T \mathbf{f} \tag{4.26}$$

The significant change from the Gauss-Newton to the Maquardt algorithm is the introduction of the damping parameter μ,

$$(\mathbf{J}_f^T \mathbf{J}_f + \mu \mathbf{I})\mathbf{h}_M = -\mathbf{g} \text{ with } \mathbf{g} = \mathbf{J}_f^T \mathbf{f} \text{ and } \mu \geq 0 \tag{4.27}$$

Here, $\mathbf{J}_f = \mathbf{J}_f(\mathbf{x})$ and $\mathbf{f} = \mathbf{f}(\mathbf{x})$ The damping parameter μ has several effects:

- For all $\mu \geq 0$ the coefficient matrix is positive definite, and this ensures that $\mathbf{h_M}$ is a descent direction
- For large values of μ the following is obtained,
  $\mathbf{h_M} \backsimeq -\frac{1}{\mu}\mathbf{g} = -\frac{1}{\mu}\mathbf{F}'(\mathbf{x})$
  a short step in the descent direction.
- If μ is very small, then $\mathbf{h_M} \backsimeq \mathbf{h_{GN}}$ which is a good step in the final stages of the iteration, when $\mathbf{x}$ is close to $\mathbf{x}^*$. If $F(\mathbf{x}^*) = 0$, or very small, then quadratic convergence is approached.
  The damping parameter influences both the direction and the size of the step, and this leads to the making of a method without a specific line search. The choice of initial μ-value should be related to the size of the elements in $A_0 = \mathbf{J}_f(\mathbf{x}_0)^T \mathbf{J}_f(\mathbf{x}_0)$, for instance by letting

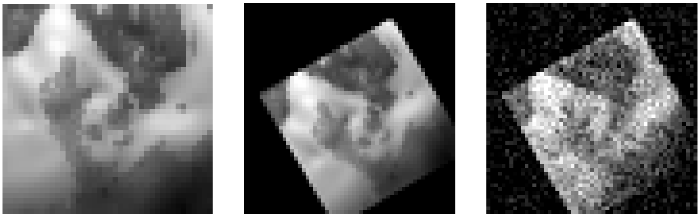
  $$\mu_0 = \tau \cdot \max x_i a_{ii}^{(0)} \tag{4.28}$$

  where τ is chosen by the user. During iteration the size of μ is controlled by the *gain ratio*,

  $$\varrho = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h_M})}{L(\mathbf{0}) - L(\mathbf{h_M})} = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h_M})}{\frac{1}{2} h_M^T(\mu \mathbf{h_M} - \mathbf{g})} \tag{4.29}$$

  Large value of ϱ indicate a that $L(\mathbf{h_M})$ is a good approximation to $F(\mathbf{x} + \mathbf{h_M})$, and μ can be decreased so that the next step is closer to a Gauss-Newton step. If ϱ is small, even negative, then $L(\mathbf{h_M})$ is a poor approximation and μ should be decreased with the twofold aim of getting closer to the steepest descent direction and reducing the step length.

The Levenberg Maquardt algorithm is a step towards robustness of minimization.



(a) Original wind-speed prognosis.

(b) Windspeed prognosis rotated and translated.

(c) Transformed prognosis with Gaussian noise.

Figure 4.2: From subfigure a) to b) there is an angle of 30^o and 4 points translation in the both the x and the y-direction. Gaussian noise with 0.1 in standard deviation is added to the image in subfigure b) and shown in subfigure c). A close look at the image in subfigure a) reveals how the image of wind speed is brighter the areas on the sea. The contours of Denmark shows as the darker area, also parts of Norway, Sweeden and Germany is depicted.

4.5 Experimental results

An image is randomly transformed, Gaussian noise is added to it and it is aligned using the alignment described in the previous subsections 4.4.1 and 4.4.2. A square of 24 by 24 is cut out with corresponding centroids for the alignment the two images. The difference between these two squares is used for the least squared minimization. This cutting ensures that the alignment is based upon the pixels and not the edges created by the template introduced in figure 4.2. The minimization is sensitive to the initial guess at the corresponding centroid point mentioned. This sensitivity is addressed by carrying out more guesses. In this test it is necessary to look at 9 different initial points, as they are conditional to the degree of rotation. As it can be seen at the rotated image in figure 4.2, the frame is extended according to the angle of rotation.

A quantitative measure of the modelling algorithms ability to identify the global transformations, described in article [7], is used with a few modifications. The image in subfigure c) is aligned to the original in subfigure a) in

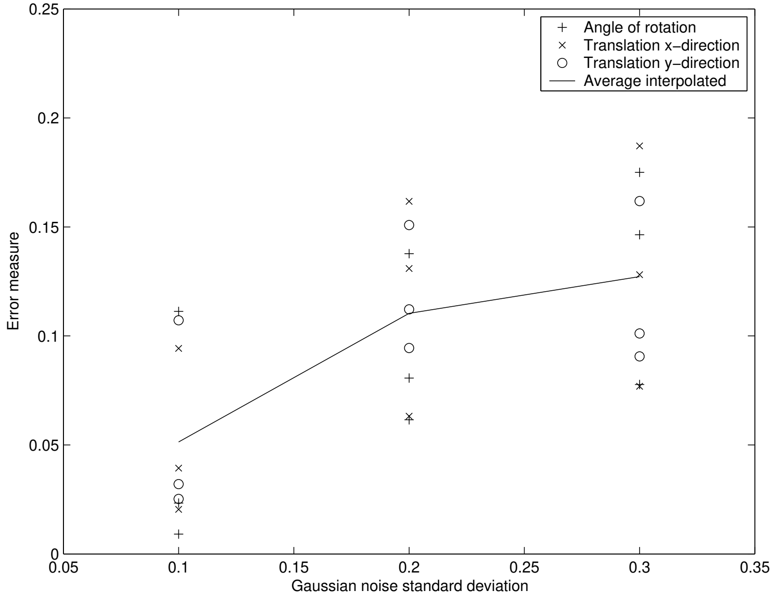


Figure 4.3: Above is a test of the image alignment. Images of windspeeds from forecasts for 3 different days in september, has had 20 alignments done to them with 3 different noise levels. Each outcome plotted is the mean of the 20 different error measures for the individual parameter. That gives 3 parameter mean values for each noise level.

figure 4.2. The true rotation θ^{true} carried out on the original image in subfigure a) is sampled from a uniform distribution within $[0, r_\theta]$ with $r_\theta = 40^0$ just like the translation t^{true} is sampled from the uniform distribution $[0, r_t]$, $r_t = 6$. The randomly transformed original image is then aligned. This is repeated numerous times. The error between the true transformation parameters $(t^{true}, \theta^{true})$ and those recovered by the algorithm is computed as $error = mean(mean(\frac{abs(\theta^{true} - \theta)}{r_\theta}) + mean(\frac{abs(t^{true} - t)}{r_t}))$. The latter is the interpolated line illustrated in figure 4.3.

4.6 Conclusion

The alignment task has been described and it has been shown that it works on alignment of sets of coordinates for sets of points. Other studies described in cited articles have shown that, within certain limits the alignment works on noisy sets of points. In order to be able to align in the l_2 norm, it has to be assumed that the residuals in the alignment are distributed in Gaussian distributions.
It has also been described how alignment directly on images can be carried out.
The alignment of images has been tested on images that are made up from the magnitude of the forecasted velocities of wind. The test is carried out on samples of forecasts, with a range of transformations similar to that of real error in forecasts. The test of the alignment suggests that it is applicable to the improvement of forecasts on wind energy. The test also suggests that this application has its limits in terms of how much noise that the alignment can cope with.

4.7 Discussion

The levels of wind speed is modelled with systematic difference between the area on land and at sea in the forecasts as it can be seen in figure 4.2. The difference between sea and land has to be taken into account, when the alignment is used for the correction of the forecast.
A central question in this chapter is about the choice between aligning directly on images or aligning on points from edge detection, i.e. stationary points from partial derivatives. There is noise arising from this edge detection, but there is also noise from aligning directly on images. The question is then which noise is preferable. It may be that the noise from working directly on images is more in accordance with the real data, where noise from edge detection has different properties more like processing noise.

A central idea was not reducing the data - including as much information as possible. Minimizing directly on images is including partial derivatives in a very different way, compared to the data reduction of edge detection. Clustering in a way so that global minima are found more robustly, could be useful on top of this. There is still a lot to investigate in this field.
Some elements in the chapter addresses the issue of robustness, changing

the norms and the use of different minimization methods like Levenberg Maquardt, has to do with this. This motivates the next chapters focus on robustness.

Chapter 5

Robust alignment

5.1 Overview

When aligning point set coordinates and the pixel values associated with them, then the key issue arises, which variations is the alignment sensitive to ? The point of making the algorithm robust, is to direct the algorithm towards beeping sensitive to only the relevant variation. The term robust is sometimes described as, an estimate insensitive to small variation. When variation is estimated in some parameter, it is considered more regular - describable analytically. The study of robustness can be regarded as a study of how to make the algorithm converge in a regular way.

In the CPA-group at IMM DTU, robust point matching algorithms from H. Chui and A. Rangarajan [10, 8, 9, 7] has been considered for the analysis of spots on the electrophoresis gel pictures. This chapter has been written with the purpose to boil down the most important parts of their papers on Robust Point Matching RPM. Introducing the algorithm for evaluation, modification and production of implementations.

5.2 Robust Point Match RPM.

The RPM algorithm [8] is introduced in order to solve the problem of finding the correspondence from one point, in one point-set x_a, to one point in an other point-set v_a. Since the mapping is of a non-rigid nature, there is the need to reject certain outliers, i.e. points with no homology/no direct 'one to one' match, and there is a need to minimise both an affine and a non-affine transformation matrix. Euclidean transformation of scale, angle and translation, can be made affine or not.

Thin plate splines TPS can be decomposed to both affine and non-affine subspace, which is why they are chosen.
The energy function of the TPS:

$$E_{TPS} = \sum_{a=1}^{K} ||x_a - f(v_a)||^2 + \lambda \int \int [(\frac{\partial^2 f}{\partial x^2})^2 + (\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2] \quad (5.1)$$

In the first part of expression (5.1) there is an infinite number of matches. The second part of expression (5.1) comes into play to regularise the energy function which in this instance means, filtering the points more according to the structure of the template. The regularisation is weighted less as the algorithm steps moves towards an exact match.

A Transformation function $f(v)$ of the points in the data set to the template set, is made up of the affine Transformation matrix d and the non-affine warping coefficient matrix $\phi(v)$

$$f(v) = d \cdot v + \phi(v) \cdot w \quad (5.2)$$

The warping is described by:

$$\phi_a(v) = c||v - v_a||^2 logc||v - v_a|| \quad (5.3)$$

Where c is a constant in 5.3. This is further described in ref. [41]

Substituting the transformation/deformation expression (5.2) into the energy-functional (TPS) (5.1) leads to the following:

$$E_{TPS}(d, w) = ||X - Vd - \Phi W||^2 + \lambda trace(w^T \Phi w) \quad (5.4)$$

The basic idea is as follows: The canonical solution to (5.1) [i.e. without the data term] is called a Greens function. In 2-D, the solution is $f(r) = c \cdot r^2 log(r)$. Using Greens function, one can plug in the solution and reduce the variational problem in (5.1) into an algebraic problem in (5.4).

This is also further described in ref. [41]

The resulting transformation from the TPS is always decomposed into a global affine transformation and a local warping component.

The progression from (5.1) to (5.4) mirrors a transition, from the standard TPS functional to an energy function defined on the affine (d) and warping (w) parameters.

Joint optimisation and correspondence is done by setting up a correspondence matrix M configured so that every row and column sums up to the value one - they are normalised. The number of rows and columns is equal to the number of points plus one for outliers. Incorporating this into the energy function gives:

$$E_{TPS}(M, d, w) = \sum_{a=1}^{K} \sum_{i=1}^{N} m_{ai} ||x_i - v_a d - \phi(v_a) w||^2 + \lambda trace(w^T \Phi w) \quad (5.5)$$

This expression cannot be minimised simultaneously w.r.t. both M, d and w. The energy function must be minimised alternating between the following two steps until the energy converges to a match: First M is held fixed while the $E_{TPS}(d, w)$ is fitted between the point sets v_a and

$$y_a = \sum_{i=1}^{N} m_{ai} x_i \quad (5.6)$$

Then d and w is kept fixed while the match matrix M is calculated between the transformed point set $\{v'_a = f(v_a|d, w)\}$ and $\{x_a\}$.

Soft assign and deterministic annealing is applied in order to avoid local minima and this makes the algorithm converge toward the optimal solution.

The basic idea of soft assign is to relax the correspondence variable in the match matrix to be continuous in the interval [0, 1], this is while maintaining the constraint, that each row and column being normalised. The energy function becomes better behaved because the correspondences are able to improve gradually.

The correspondences are always constrained to be fuzzy and doubly sub stochastic. Deterministic annealing, as opposed to simulated annealing, deals with this fuzziness. An additional temperature parameter T is introduced with the entropy term:

$$T\sum_{ai} m_{ai}log(m_{ai}) \tag{5.7}$$

This expression (5.7) specifies the degree of fuzziness of the correspondence matrix: Higher temperature T, greater fuzziness. At each temperature the initial conditions from the previous temperature is used, and a straight forward deterministic descent is performed on the energy function. As temperature decreases, the doubly sub stochastic matrix approaches a permutation matrix. Introducing soft assign and deterministic annealing into the energy function:

$$\begin{aligned}E_{TPS}(M,d,w) = &\sum_{a=1}^{K}\sum_{i=1}^{N} m_{ai}||x_i - v_a d - \phi(v_a)w||^2\\ &-\zeta\sum_{a=1}^{K}\sum_{i=1}^{N} m_{ai} + T\sum_{a=1}^{K}\sum_{i=1}^{N} m_{ai}logm_{ai}\\ &+\lambda_1 trace(w^T\Phi w) + \lambda_2 trace[((d-I)^T(d-I))]\end{aligned}$$

Where $m_{ai} \in [0,1]$ satisfies:

$$\sum_{a=1}^{K+1} m_{ai} = 1, \text{and} \sum_{i=1}^{N+1} m_{ai} = 1 \tag{5.8}$$

Where the second term with the parameter ζ is there to guard against null matches. The fifth term is introduced to constrain the affine mapping d

by penalising the part of d that differs from an identity matrix. λ_1 and λ_2 follows the above mentioned linear annealing ($\lambda_i = \lambda_i^{init} T$, $i = 1, 2$). And, in order to provide more freedom for the affine transformation λ_1^{init} is set for less than λ_2^{init}. When updating the correspondence the closed form solution for M can be obtained just differentiating the energy function in (5.8) w. r. t. M and setting the result to zero:

$$m_{ai} = e^{-\frac{||x_i - v_a d - \phi(v_a) w|| - \zeta}{T}} \tag{5.9}$$

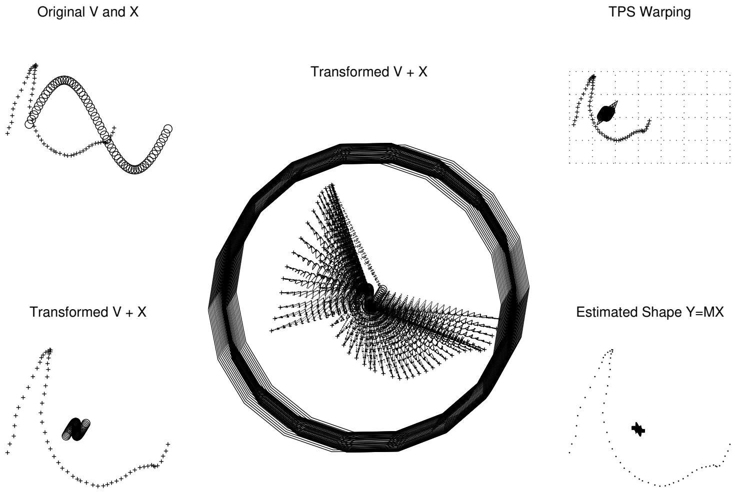
When correspondence is held fixed, the least squares approach is used to solve for the TPS parameters d and w. This can be implemented with QR decomposition described in ref. [41]. All this adds up to a unifying pseudo code:

Algorithm 1 TPS-RPM Algorithm pseudo code

Initialise parameter M, d and w.
Initialise parameter λ_1, λ_2 and T.
Begin A Deterministic annealing
    Begin B Alternating update
    Update M: Soft assign.
    Update (d, w): Least squares solution to TPS.
    End B
Decrease T, λ_1 and λ_2.
End A

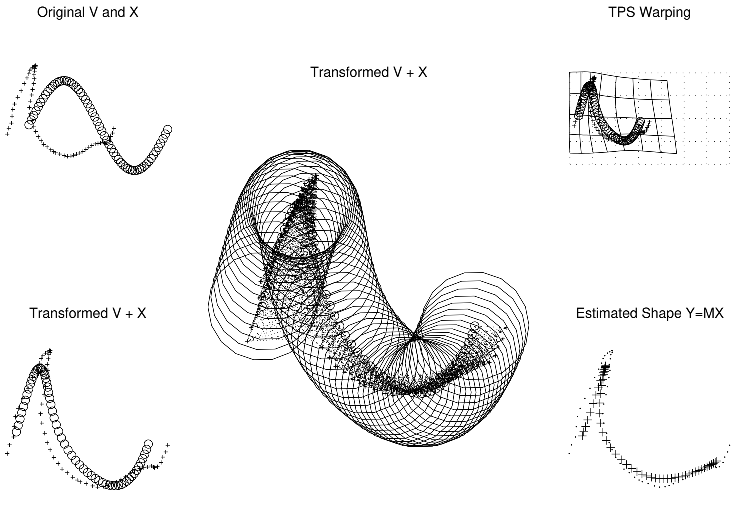
5.3 Experimental results

From [8] a demo package has been produced, it is written under GNU public license for matlab. This package has been used to produce figure 5.1 and 5.3. As opposed to the point sets in figure 4.1, which have a rigid deformation as there deviation between them, these point sets have non rigid deformations.

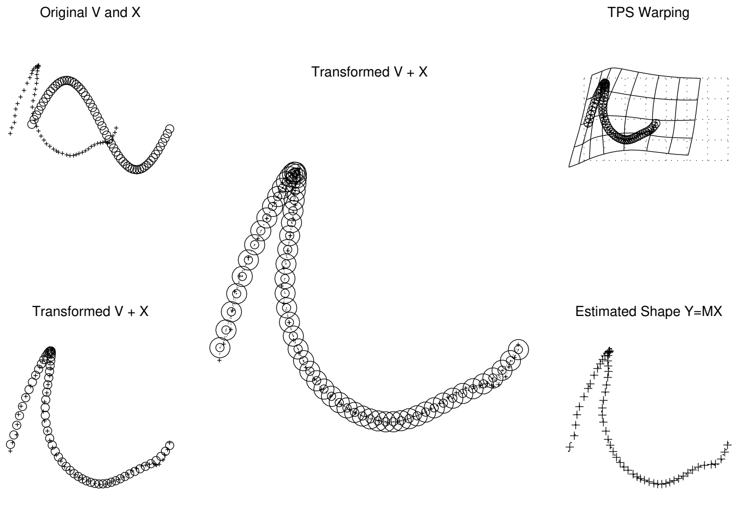


(a) Early stage

Figure 5.1: Here are snapshots at three stages of the non rigid point match RPM, see also figure 5.2. A point set - samples from a sinusoidal curve - is matched to an irregular set with similar shape. The big circles in the early stage illustrate the search range that includes all the points of the other point set and then it decreases. A dotted line marks the correspondence between points from one set to the other, at the early stage a lot of these lines illustrates that a point in one set has many corresponding points in the other set. As the process progresses - anneals - the number of correspondences decreases approaching a one to one match.

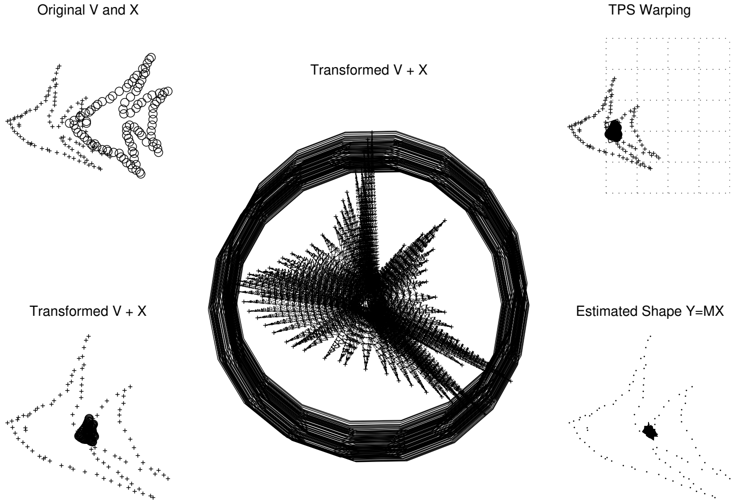


(a) Intermediate stage



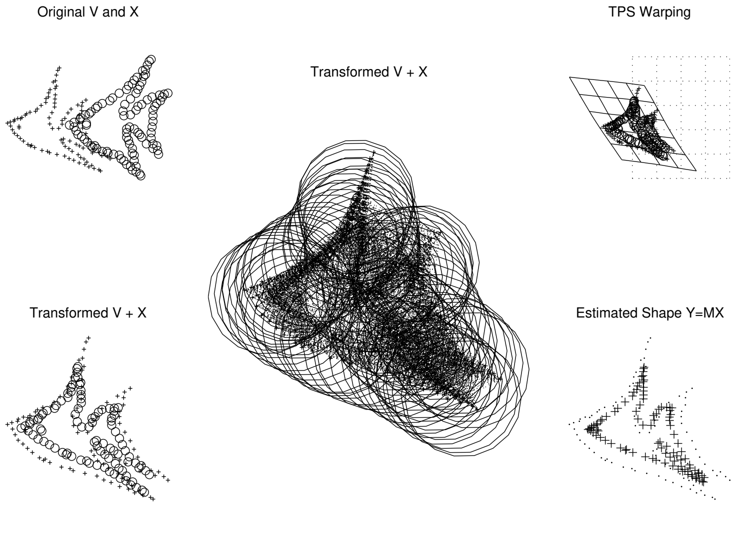
(b) Final stage

Figure 5.2: Last stages from figure 5.1

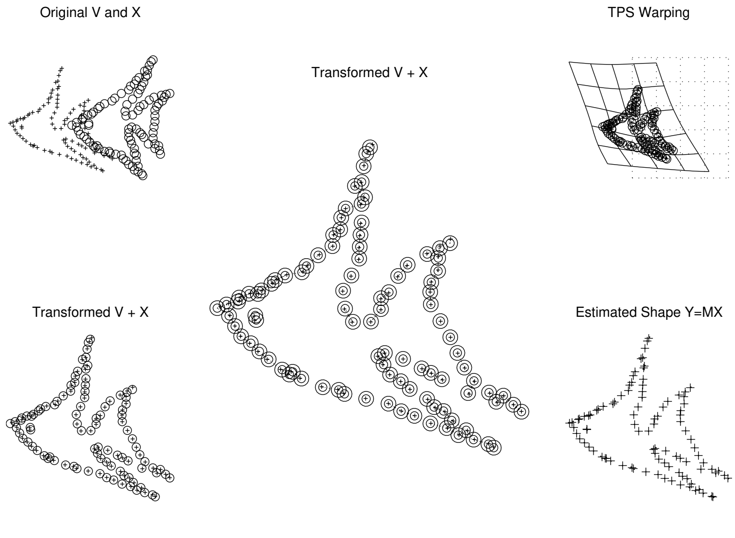


(a) Early stage

Figure 5.3: Here are three other snapshots at the stages of the non rigid point match RPM, see also figure 5.4. A point set - samples from a fish shape - is again matched to an set with similar shape, but where the positions have been deformed in an irregular way - a non rigid way. As the process progresses - anneales - the number of correspondences decreases approaching a one to one point match.



(a) Intermediate stage



(b) Final stage

Figure 5.4: Last stages from figure 5.3.

5.4 Mixture Point Matching MPM.

RPM is constrained to match a point set to an other one,on a one to one basis, although it is done with the allowance for certain outliers. It is of interest to investigate the possibility of matching point sets of different size. There are results regarding this in ref. [9], in which it is called Mixture Point Matching MPM. In [9] point matching is re-interpreted as a mixture density problem that can be solved using an Expectation Maximization EM algorithm [25, 37], explaining the change of title from RPM to MPM.

5.4.1 Gaussian mixture models

In order to deal with this algorithm it is necessary to describe the derivation of it from [6]. Starting with the mixture distribution:

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x}|j)P(j) \tag{5.10}$$

The model of density in (5.10) is made up of a linear combination of component densities $p(\mathbf{x}|j)$. These are weighted by the coefficients $P(j)$ called the prior probabilities, which satisfies the constraints:

$$\{\sum_{j=1}^{M} P(j) = 1\} \wedge \{0 \leq P(j) \leq 1\} \tag{5.11}$$

The 'priors' are normalised just like the component density functions are:

$$\int p(\mathbf{x}|j)d\mathbf{x} = 1 \tag{5.12}$$

A data point can be generated from the probability distribution (5.10), one of the components j are chosen at random with the probability $P(j)$ and then the point is generated from the corresponding distribution $p(\mathbf{x}|j)$.

Given a sufficiently large number of components and correctly chosen parameters, one can approximate any continuous density.
There is not any labelling to tell which class was responsible for generating each data point $\mathbf{x}$, like there would have been in true classification. This is caused by incomplete training data, the samples used fore setting up mixture models are scarce resulting in an approximation to the distribution of the density - EM on mixture models is suitable for incomplete sets of data (ref'Modelling nonlinear and non stationary time series'). Estimation of density can be considered an exercise in approximation. This mixture modelling of the mixture distribution can be used to separately model each class C_k in a true classification. In which case the class-conditional density $p(\mathbf{x}|C_k) = \sum_{j=1}^{M} p(\mathbf{x}|j)P(j)$. Bayes' theorem puts the prior probability $P(j)$, the posterior probabilities $p(j|\mathbf{x})$ and the conditional densities $p(\mathbf{x}|j)$ in this relation:

$$p(j|\mathbf{x})p(\mathbf{x}) = p(\mathbf{x}|j)p(j) \Rightarrow p(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)p(j)}{p(\mathbf{x})} \tag{5.13}$$

with $p(\mathbf{x})$ given in (5.10) given the constraints in (5.11) the posterior probabilities satisfy:

$$\sum_{j=1}^{M} P(j|\mathbf{x}) = 1 \tag{5.14}$$

In the case of using Gaussian radial basis functions with the covariance matrix $\mathbf{\Sigma}_j$ made up by some scalar σ_j^2 multiple of an identity $\mathbf{I}$ so that $\mathbf{\Sigma}_j = \sigma_j \mathbf{I}$, then the conditional density is made up by:

$$p(\mathbf{x}|j) = \frac{1}{(2\sigma_j^2)^{d/2}} e^{\{-\frac{||\mathbf{x}-\mu_j||}{2\sigma_j^2}\}} \tag{5.15}$$

The negative log-likelihood for the data is then given by:

$$E = -\ln\mathcal{L} = -\sum_{n=1}^{N} lnp(\mathbf{x}^n) = -\sum_{n=1}^{N} \ln\{\sum_{j=1}^{M} p(\mathbf{x}^n|j)P(j)\} \tag{5.16}$$

In (5.16) maximizing $\mathcal{L}$ is the same as minimizing E

5.4.2 Density Estimation - Expectation Maximization EM

In order so avoid some of the complexities of non-linear optimisation methods this is a description of the special version of the EM - algorithm.
In (5.16) maximizing the likelihood $\mathcal{L}$ is minimizing E. With this consideration an update of parameters is sought by setting up the difference between the probability density evaluated with the old parameters E^{old} and the density with the new parameters E^{new} as:

$$E^{new} - E^{old} = -\sum_{n=1} \ln\left\{\frac{p^{new}(\mathbf{x}^n)}{p^{old}(\mathbf{x}^n)}\right\} \tag{5.17}$$

Inserting the mixture density defined in (5.10) into (5.17) and an identity element of $\frac{p^{old}(j|\mathbf{x}^n)}{p^{old}(j|\mathbf{x}^n)}$ to the right:

$$E^{new} - E^{old} = -\sum_{n=1} \ln\left\{\frac{\sum_{j=1}^{M} p^{new}(\mathbf{x}^n|j)P^{new}(j)p^{old}(j|\mathbf{x}^n)}{p^{old}(\mathbf{x}^n)p^{old}(j|\mathbf{x}^n)}\right\} \tag{5.18}$$

Jensens inequality: Given a set of numbers λ_j that meets the criteria: $\lambda_j \geq 0 \& \sum_j \lambda_j = 1$

$$\ln\left(\sum_j \lambda_j x_j\right) \geq \sum_j \lambda_j \ln(x_j) \tag{5.19}$$

Since $p^{old}(j|\mathbf{x})$ sums to unity, then jensens inequality (5.19) can be applied to (5.18):

$$E^{new} - E^{old} = -\sum_n \sum_j p^{old}(j|\mathbf{x}^n) \ln\left\{\frac{p^{new}(\mathbf{x}^n|j)P^{new}(j)}{p^{old}(\mathbf{x}^n)p^{old}(j|\mathbf{x}^n)}\right\} \tag{5.20}$$

Minimizing E^{new} w.r.t. its parameters from the use of the difference to the old density (5.20) can be done just minimising numerator from (5.20):

$$Q = -\sum_n \sum_j p^{old}(j|\mathbf{x}^n) \ln\{p^{new}(\mathbf{x}^n|j)P^{new}(j)\} \tag{5.21}$$

In this case it is the *Gaussian* mixture model under consideration substituting Gaussians into (5.21)

$$Q = -\sum_n \sum_j p^{old}(j|\mathbf{x}^n)\{\ln P^{new}(j) - d\ln\sigma_j^{new} - \frac{||\mathbf{x}^n - \mu_j^{new}||^2}{2(\sigma_j^{new})^2}\} + const. \tag{5.22}$$

Minimizing the mixing parameter $P^{new}(j)$ can be done taking into account that $\sum_j P^{new}(j) = 1$ by introducing a Lagrange multiplierλ rewriting (5.22):

$$Q + \lambda\left(\sum_j p^{new}(j) - 1\right) \tag{5.23}$$

The derivative of (5.23) w.r.t. $P^{new}(j)$ is sat to zero obtains:

$$0 = -\sum_n \frac{p^{old}(j|\mathbf{x}^n)}{p^{new}(j)} + \lambda \tag{5.24}$$

The value of λ c an be found multiplying both sides of (5.24) by $P^{new}(j)$ summing up over j. Using that $\sum_j P^{new}(j) = 1$ and $\sum_j P^{new}(j|\mathbf{x}^n) = 1$ obtaining $\lambda = N$. This leads to the update of $P^{new}(j)$:

$$p^{new}(j) = \frac{1}{N}\sum_n P^{old}(j|\mathbf{x}^n) \tag{5.25}$$

Likewise are the updates for μ_j^{new} and σ_j^{new} found from (5.22)

$$\mu_j^{new} = \frac{\sum_n p^{old}(j|\mathbf{x}^n)\mathbf{x}^n}{\sum_n p^{old}(j|\mathbf{x}^n)} \tag{5.26}$$

$$(\sigma_j^{new})^2 = \frac{1}{d}\frac{\sum_n p^{old}(j|\mathbf{x}^n)||\mathbf{x}^n - \mu_j^{new}||^2}{\sum_n p^{old}(j|\mathbf{x}^n)} \tag{5.27}$$

This can be extended to Gaussians with full covariance matrices

5.4.3 Mixture models for point matching - A maximum posterior MAP problem

Returning to the MPM from [9] it is now the purpose to bring the general description of the derivation of the EM on Gaussian mixture models in to the specific concept of point matching.
The transformation function $f(v)$ of the points in the data set to the template set, made up of the affine Transformation matrix d and the non-affine warping coefficient matrix $\phi(v)$ represented abstractly by, $f(v,\alpha)$ with α denoting both the affine d and non-affine warping $\phi(v)$ parameters in the arguments for the transformation $f(v,\cdot)$. Model point(s) v_a is transformed to $u_a = f(v_a,\alpha)$ in order to find the best corresponding point(s) in the set of data x_i. The data set X contains an object to be recognised amongst some outlier points. The model V is to be transformed to get the best possible fit to the part of the data that truly corresponds to it. Modelling an incomplete data set makes up an a-symmetry that justifies the choice of the EM-algorithm on mixture models. The likelihood for the whole data set is formulated as:

$$p(X|V,\alpha,\Sigma,\pi) = \prod_{\mathbf{i=1}}^{\mathbf{N}} \sum_{\mathbf{a=1}}^{\mathbf{K}} \pi_a p(x_i|f(v_a,\alpha),\Sigma_a) \tag{5.28}$$

The percentage of the data that belongs to a certain cluster named the occupancy denoted π composed of π_a is called the prior probabilities in (5.10). The covariance matrices are now denoted Σ composed of Σ_a Extending the Gaussians from (5.15),

$$p(x_i|f(v_a,\alpha).\Sigma_a) = \frac{1}{2\pi|\Sigma_a|^{\frac{1}{2}}} e^{-\frac{(x_i - f(v_a,\alpha))^T \Sigma_a^{-1}(x_i - f(v_a,\alpha))}{2}} \tag{5.29}$$

To cater for spurious point that are outliers to existing clusters, an extra cluster is added as the $(K+1)th$. Modifying (5.28)

$$p(X|V,\alpha,\Sigma,\pi) = \prod_{\mathbf{i=1}}^{\mathbf{N}} \sum_{\mathbf{a=1}}^{\mathbf{K+1}} \pi_a p(x_i|f(v_a,\alpha),\Sigma_a) \tag{5.30}$$

Different problems require different transformations as functional mappings the choice of transformations reflect the prior knowledge about the generic problem. The transformations are therefor not completely arbitrary. For instance to prevent non rigid mappings from behaving to arbitrary smoothness regularizers are often used. The operator L represent this regularization on the mapping as $||Lf||^2$ and is regarded as the prior distribution of the parameters α.

$$p(\alpha) = \frac{e^{-\lambda||Lf||^2}}{Z_{part}} \tag{5.31}$$

where Z_{part} is the partition/normalization function for the prior density $p(\alpha)$, λ is a weight parameter. This prior is used to construct the posterior by use of Bayes rule like in (5.13):

$$p(\alpha|X, V, \Sigma.\pi) = \frac{p(X|V, \alpha, \Sigma.\pi)}{p(X)} \tag{5.32}$$

The term $p(X)$ is omitted because it does not depend on α. The best fit is found solving the maximum *a posterior* (MAP) from (5.32). As the maximizing $\mathcal{L}$ is minimizing E in (5.16), solving the MAP problem is equivalent to minimizing the log posterior energy function:

$$E_1(\alpha, \Sigma, \pi) = -\log p(X|V, \alpha, \Sigma, \pi) - \log p(\alpha) \tag{5.33}$$

5.4.4 EM and Complete Data Posterior Energy Function

In [16] it is shown that the minimization of the original log posterior in (5.33) can be turned into a minimization of the complete data posterior energy function. This is possible by introducing the intermediate binary variable denoted Z composed of zai as latent data. Z represents the classification of data points i that belongs to the ath cluster. The incomplete posterior in (5.33) converted to a minimization of the complete data posterior energy function:

$$E_2(M,\alpha,\Sigma,\pi) = \sum_{i=1}^{N}\sum_{a=1}^{K+1}\frac{m_{ai}[x_i - f(v_a,\alpha)]^T\Sigma_a^{-1}[x_i - f(v_a,\alpha)]}{2}$$
$$+\sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}\log m_{ai} - \sum_{i=1}^{N}\sum_{a=1}^{K+1}\frac{m_{ai}}{2}\log|\Sigma_a| \quad (5.34)$$
$$-\sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}\log\pi_a + \lambda||Lf||^2$$

Where m_{ai} satisfies $\sum_{a=1}^{K+1} m_{ai} = 1 \ \forall \ i \in \{1,2,...,N\}$ For this complete posterior data there exists a dual step update that guarantees the algorithm to converge to a local minimum of the original incomplete posterior energy function in (5.33). The dual steps are called the 'E' and the 'M' step. The update steps where derived in (5.25),(5.26) and (5.27) by differentiating w.r.t. the parameter in question. Similarly is the is 'E' step found by differentiating (5.34) w.r.t. 'm_{ai}' setting the result to zero following update equation is achieved:

$$m_{ai} = < z_{ai} > = \frac{\pi_a p(x_i|f(v_a,\alpha),\Sigma_a)}{\sum_{b=1}^{K+1}\pi_b p(x_i|f(v_b,\alpha),\Sigma_b)} \quad (5.35)$$

The notation '$< z_{ai} >$' is used to denote that m_{ai} is the *conditional expected value* of z_{ai} at each step. The matrix M composed of m_{ai} is considered the fuzzy estimator of the binary correspondence matrix Z. In the M step M is kept fixed while minimizing the complete data in order to update the rest of the parameters. Differentiating the complete data in (5.34) w.r.t. α, Σ & π and setting to zero, leads to the equations for the M step. Alternations between the dual steps are done until convergence.

5.4.5 Problems with Mixture model and the EM algorithm

In article [9] this algorithm is confronted with a test, in which a point set is used as template to estimate the free parameters, without any transformations or outliers. Symmetry breaking random noise is added to M. The same data is then used with the same initial setup. The final configuration found by the EM algorithm shows that groups of points matched cross each other. This is dealt with by simplifying the mixture model. The simplification builds upon observing that the magnitude of the covariance matrix is proportional to the 'range of search'. $\{m_{ai}\}$ is approaching a binary matrix with the magnitude of Σ_a getting smaller. $\{m_{ai}\}$ is approaches a uniform, with values close to $\frac{1}{K+1}$, with larger magnitudes of Σ_a. In the latter case the Mahanalobis distance between point set $\{x_i\}$ and $\{u_a = f(v_a, \alpha)\}$ is small. Small covariance magnitudes leads to the cluster centers looking very locally for candidates. Making the covariance matrix invariant to rotation isotropic and controlling its magnitude by a temperature $\{T\}$ parameter introduces a match strategy from global search to local. According to this observation the Mahanalobis distance measure is simplified to the Euclidean:

$$\frac{[x_i - f(v_a, \alpha)]^T \Sigma_a^{-1} [x_i - f(v_a, \alpha)]}{2} \mapsto \frac{||x_i - f(v_a, \alpha)||^2}{2T} \tag{5.36}$$

The outlier cluster with index $K + 1$ has to be able to take any points into account. Therefore this cluster is assigned a center at the point sets center of mass (center of gravity - with equally distributed mass) x_0 and the temperature T_0 is kept fixed and large. The occupancy parameter is disregarded since all clusters is considered to be able to explain equal amount of the data. With these modifications to the energy function E_2 it is put to the same test. clusters found are then more evenly distributed an more consistent, with variation due to the symmetry breaking perturbations.

5.4.6 Mixture Point Matching Energy function and Algorithm

When the temperature becomes low the transformations has to be influenced more by the data, as the prior vanishes and visa versa, the regularization of the transformation is therefore weighted by the temperature

$\lambda T||Lf||^2$. The final energy function:

$$\begin{aligned} E_3(M,\alpha) = \sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}||x_i - f(v_a,\alpha)||^2 \\ +T\sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}\log m_{ai} - T\sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{ai}\log T \\ -T\sum_{i=1}^{N}\sum_{a=1}^{K+1} m_{K+1,i}\ \log T_0 + \lambda T||Lf||^2 \end{aligned} \tag{5.37}$$

Where m_{ai} still satisfies $\sum_{a=1}^{K+1} m_{ai} = 1\ \forall\ i \in \{1,2,...,N\}$ with $m_{ai} > 0$ The mentioned dual update embedded in an annealing scheme:

E step: Update the correspondence

$$m_{ai} = \frac{p(x_i|f(v_a,\alpha),\Sigma_a)}{\sum_{b=1}^{K+1} p(x_i|f(v_b,\alpha),\Sigma_b)} \tag{5.38}$$

For the clusters $a = 1,2,...,K$,

$$p(x_i|f(v_a,\alpha).\Sigma_a) = \frac{1}{2\pi T} e^{-\frac{(x_i - f(v_a,\alpha))'(x_i - f(v_a,\alpha))}{2T}} \tag{5.39}$$

And for the out lier cluster $a = K+1$,

$$p(x_i|f(v_{K+1},\Sigma_{K+1}) = \frac{1}{2\pi T_0} e^{-\frac{(x_i - x_0)'(x_i - x_0)}{2T_0}} \tag{5.40}$$

Normalisation of m_{ai} over all clusters, that is the original template including the out lier cluster introduces competition. Meaning that unless a point shows strong evidence that it belongs to a certain cluster it is rejected as an out lier. The evidence is evaluated automatically. In order to break symmetry , perturbations (zero mean Gaussian noise with very small standard deviation) is added to m_{ai} after normalisation.

M step: Update the transformation Solve the following least square problem:

$$\min_{\alpha} E_4(\alpha) = \min_{\alpha} \sum_{a=1}^{K} ||y_a - f(v_a,\alpha)||^2 + \lambda T||Lf||^2 \tag{5.41}$$

Where,

$$y_a = \sum_{i=1}^{N} m_{ai} x_i \tag{5.42}$$

Utilizing that $\sum_{a=1}^{K+1} m_{ai} = 1$. y_a can be regarded as the new cluster centers.
Annealing: The annealing scheme is deterministic, the temperature T is gradually reduced according to the annealing rate r

Algorithm 2 MPM Algorithm pseudo code

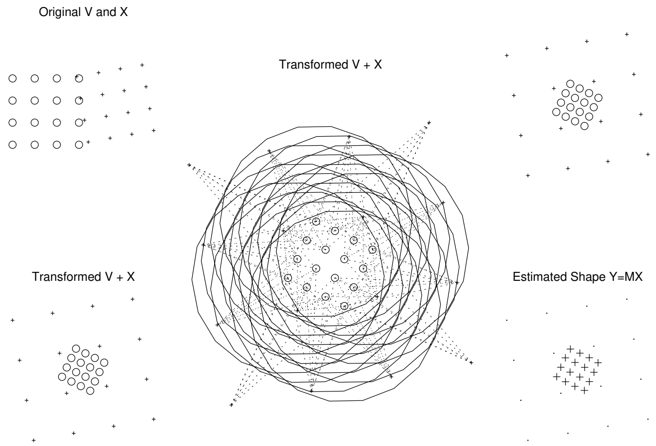
Initialise parameter T and α.
Alternating update
  -E-step update m_{ai}.
  -M-step update α.
Lower $T = T \cdot r$ until T_{final} is reached.

5.5 Experimental results

The considerations here are aimed at incorporating the Mixture point match in the alignment of point grids deformed by the affine transformations introduced in section 4.2.1, which gives the deformations rigid behaviour. The demonstration package is modified to do only the affine transformations results shown in figure 5.5. The pitfall, mentioned in the caption, appears because there is no constraint guiding the algorithm towards one of the four solutions to the problem of aligning a shape of a square with the right angle. The constraint that would adress this would be, relating the transformation to the value of the pixel. This increases the number of dimensions of the clustering. Before this kind of alignment can be tried some theory has to be dealt with.

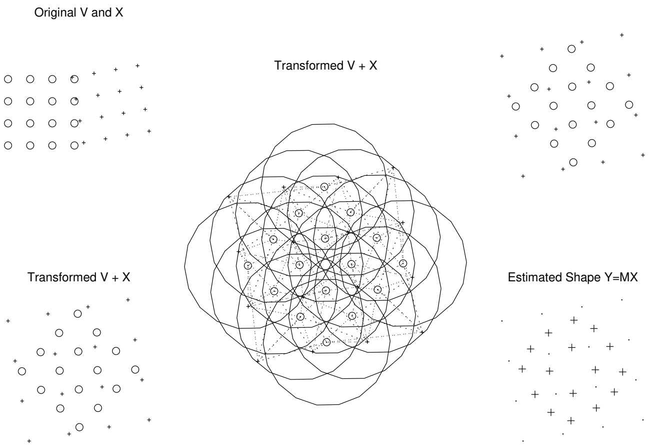
5.6 Modifying the point match to a vector match MPM to MVM

The wind data is formatted as vectors not as points. The difference between the vector sets and the sets of points is that, there are two additional

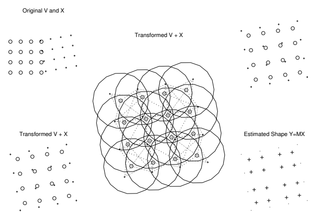


(a) Early stage

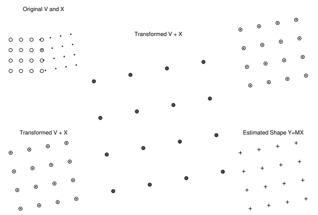
Figure 5.5: Stages of rigid point match RPM, see also figure 5.6. A point set of equidistant points is brought for the simplicity. The big circels still illustrate search range like in figure 5.1 and 5.3. As the process progresses corrospondences approaches a one to one match, but there is the pitfall of a multiple of 90^o rotation, since the shape is symmetric



(a) Intermediate stage



(b) Intermediate stage



(c) Final stage

Figure 5.6: Last stages of figure 5.5.

features to every point namely the length and the angle of the vector associated to it.
The task is to identify the spatial information about the error between the forecasted and the measured wind velocity. The spatial informations about the error consists of global and local transformations that will map the sets of vectors to each other. Keeping in mind that the goal is to find the relationship, that correlates the mappings at each time slot, making the analysis spatio-temporal. Training the model supervised by the observed wind data versus the forecasted makes the estimate of conditional expectation, that should improve the prediction of wind data. The global transformations translation, rotation and scaling of the starting points of the vectors means that location of the vectors are moved regarding the set of vectors x_i and v_a as rigid bodies. Expressing the transformation from section 4.2.1 w.r.t. the points:

$$f(v_a, \theta, t) = R(\theta) \cdot v_a + t \tag{5.43}$$

θ is the angle of rotation used in the two dimensional rotation matrix $R(\theta)$ and $t = (t_x, t_y)$ is the two dimensional global translation. The regularization term in equation (5.41) can be omitted since the transformations are rigid.

5.6.1 Integrating measures of vectors

The measure of difference between the point sets in both equation (5.41) and (5.38) need to be modified. The angle ϕ and magnitude or distance I that composes each vector g makes the sets x_i and v_a four dimensional. But as in section 4.4 the main focus of this study is on treating the sets of magnitudes of the vectors I as sets of intensities of pixels in images.

Integrating the magnitudes I into the update of the transformations and the correspondence, the **E** and **M** steps in equation (5.39),(5.40) and (5.41):

$$p(I(x_i)|I(f(v_a, \alpha)), \Sigma_a) = \tag{5.44}$$

$$\frac{1}{2\pi T} e^{-(I(x_i)-I(f(v_a,\alpha))'(I(x_i)-I(f(v_a,\alpha))\frac{(x_i-f(v_a,\alpha))'(x_i-f(v_a,\alpha))}{2T})} \tag{5.45}$$

The outliers mentioned in section 5.4.5, where x_0 is assigned to the center of mass(mean value of all coordinates), has to be adressed here as well. Following the same idea here, x_0 is replaced by $I(0)$ and assigned the intensity mean $(I(x))$.

$$p(I(x_i)|I(f(v_{K+1},\Sigma_{K+1})) = \frac{1}{2\pi T_0} e^{-\frac{(I(x_i)-I(0))'(I(x_i)-I(0))}{2T_0}} \tag{5.46}$$

$$\min_\alpha E_4(\alpha) = \min_\alpha \sum_{a=1}^{K} ||I(y_a) - I(f(v_a,\alpha))||^2 \tag{5.47}$$

Where,

$$I(y_a) = \sum_{i=1}^{N} m_{ai} I(x_i) \tag{5.48}$$

Utilizing that $\sum_{a=1}^{K+1} m_{ai} = 1$. $I(y_a)$ can be regarded as the new cluster centers, but as opposed to in (5.42) the centers signifies values of magnitude/distance I.

5.7 Experimental results

Some very limited tests shows that this framework converges as should be expected. An extensive statistical test like the one in section 4.5 should show improvement in the results from the alignment experiments in section 5.3. This can be expected from the limited tests of this mixture vecture match mentioned and from the experiments from the 'point coordinate level'.

5.8 Conclusion

It has been described how robustness can be introduced in a statistical framework using the EM update and annealing on points and pixels. And it has been shown that it works on alignment of sets of coordinates for sets of points, that have deformation of both rigid and non rigid nature. Preliminary tests have also shown that the robust alignment of images directly works, in the sense that it converges and that it can be used. A thorough statistical test can be done when setting up the implementation using real data.

Chapter 6

Conclusion

The main objective of the thesis set forth in chapter 1.1 where:

- To document and illustrate how spatial error patterns from wind prognosis phase error can be located by alignment.
- To study and develop robust alignment.

The basic well known methods for alignment has been studied. Issues associated with alignment of simulated spatial error in wind speed prognosis, has been revealed. The considerations and evaluations of the alignment of images shows, that the spatial error of wind speed prognosis can be aligned, given that the assumptions on how the error is appearing are true. It has been shown how the scope of the alignment is bounded by the noise added to the image. One way to explain this bound is by considering that the adaptation of the model gets caught in a local minima of the cost function. Considering the local versus the global perspective and all the intermediate perspectives leads to the study of robustness. Some methods in this field have been studied and one of them has been developed to address the robustness of alignment of forecast images. Limited tests shows that the methods works and can be applied for improving the prognosis in a spatial context.

6.1 Future works

Considerations on future works comes out of setting the view on the data from chapter 2 and the methods from chapter 3 up against the further study of methods in the chapters 4 and 5 in the thesis.
First and foremost is the task of implementing the robust alignment studied in this thesis, using real observations of wind energy production. One central task in implementing the alignment is adapting the observations of energy production. As mentioned in section 3.4, the grid of the prognosis will have missing observations, where there are no wind turbines and values at these points have to be interpolated via estimation, one of the methods referred to is Krigging.
Once the alignment is implemented a task of time series analysis appears, because the alignments of the individual prognosis instances should reveal a pattern in the error that covers a certain time span. From that time series analysis comes the improvement in prediction. The robust alignment has to be 'fine tuned' with regards to the annealing in the EM part of the implementation.
Once the alignment is implemented then there is room for further improvements:

- Incorporating more observations.
  As mentioned in section 2.2 there are quite a few other sources of information relevant to the improvement of prediction. The 16 local measurements of wind velocity can be used for the alignment, it is described in section 3.3 how this is a natural extension of the alignment recommended for implementation.
  In section 2.2.3 there is a short descriptions of how velocity of wind from satellite images can be estimated with reference to a thesis by me[18]. Estimating the wind velocity from satellite images should further improve the prediction of wind energy production. Using the radar images mentioned in section 2.2.4 also involves statistical image analysis methods mentioned in section 3.4 with special attention to the other thesis by me [17] dealing with a problem similar to that of a radar. These different estimates all taking into the data, upon which the conditional expectation is estimated, makes it optimal.
  The more information used for the alignment, the more robust it becomes.
- Improving the methods.
  Often when studying methods many more ideas and questions are

raised, than is answered, this is a fortune and it makes it interesting. In section 3.4 there is a discussion about spline models versus Brownian warps. This issue can be explored with special attention to the question about, how non rigid matching can improve the alignment. Alignment can be done first and then the non rigid part of the match. But there is also the other option of doing the non rigid point matching and then decomposing it into the rigid and non rigid parts. These two approaches could be compared and maybe other combinations might improve the model.

Another field that could be investigated is deformable models like active shape models, where the typical wind velocity patterns are stored like active shape models uses point distribution models.

When describing distributions like in Gassian mixture models a set of Gaussian basis function are used, but there might be other basis functions that are useful.

Two aspects of the matching problem is, firstly regarding the correspondence problem, which has been addressed to a degree in this thesis, and secondly the aperture problem, which I have addressed in [18]. The idea of addressing these aspects using geometry constrained diffusion mentioned in section 3.4 looks promising.

Issues like Bayes relation, taking partial derivatives through convolution with a differentiable functions, defining (best) basis functions and setting up frameworks for statistical analysis has to be addressed when attempting to improve the methods.

Bibliography

[1] *Interactive Computer Graphics: A top down approach with OpenGL*, chapter 4. Addison Wesley Longmann, Inc., 1997.

[2] *Using Meteorological Forecasts in Online predictions of Wind Power*, chapter 6.5, pages 65–68. IMM/Institute of Informatik og matematisk modellering at D.T.U., 1999.

[3] P. Johansen A. D. Jackson, B. Lautrup and M. Nielsen. Products of random matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 66, 2002.

[4] Per Roensholt Andresen and Mads Nielsen. Non-rigid registration by geometry-constrained diffusion. *Medical Image Analysis*, Vol.5 Issue.2:81–88, 2001. Provider elsevier ISSN 13618415.

[5] C.Deboor. *A practical guide to splines.* Springer Verlag, 1978.

[6] Chistopher M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press Oxford, Department of Computer Science and Applied Mathematics Aston University Birmingham, 1997. 482 pp.

[7] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA) 2000, (in press).*

[8] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2000, (in press).*

[9] H. Chui and A. Rangarajan. Non-rigid point matching: a new framework and algorithm. In *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[10] Knut Conradsen and Jan Pedersen. Analysis of two-dimensional electrophoretic gels. *Biometrics*, 48:1273–1287, 1992.

[11] Gianluca Donato and Serge Belongie. Approximate thin plate spline mappings. pages pp. 21–31. ECCV 2002, 2002.

[12] Hrafnkell Eiríksson. Shape representation alignment and decompostion. Master's thesis, IMM/Institute of Informatik og matematisk modellering at D.T.U, 2001. nr.25.

[13] *An Introduction to Wavelet Analysis*. Birkhauser Boston, 2002. ISBN 0-8176-3962-4.

[14] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[15] Colin Goodall. Procustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society*, 53(2):285–339, 1991.

[16] Richard J. Hathaway. Another interpretation of the em algorithm for mixture distributions. *Statistics and Probability Letters*, 4(2):53–56, 3 1986.

[17] Rune KLarskov Jensen. Automatic image analysis, 1999.

[18] Rune KLarskov Jensen. Estimation of velocity via wavelets. Master's thesis, IMM/Institute of Informatik og matematisk modellering at D.T.U, 2001. nr.34.

[19] Jess U Jørgensen. On the accuracy of land cover data in nwp forecasts for high resolution wind energy prediction, 2001.

[20] Hans Bruun Nielsen Kaj Madsen and Ole Tinglev. Methods for non linear least squares problems. IMM/Institute of Informatik og matematisk modellering at D.T.U., 1999.

[21] Per Christian Hansen Kaj Madsen, Ole Tingleff and Wojciech Owczarz. Robust subroutines for non-linear optimisation. Technical report, Institute for Numerical Analysis, Technical University of Denmark, april 1990. report no 90-06.

[22] D. G. Krige. *J. Inst. Mine Surveyors S. Afrika*, (12):45–84,95–136, 1962.

[23] Rasmus Larsen, Knut Conradsen, and Bjarne Kjær Ersbøll. Estimation of Dense Image Flow Fields in Fluids. *grs*, 36(1):256–264, jan 1998.

[24] K. Levenberg. A method for solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[25] Henrik Madsen and Jan Holst. *Modelling Non Linear and Non Stationary Time Series*. IMM/Institute of Informatik og matematisk modellering at D.T.U., Lyngby, Denmark, 2000.

[26] Stephane Mallat. *a wavelet tour of signal processing*. Academic Press, 2nd edition, 1999.

[27] D. Maquardt. An algorithm for least squares estimation on non linear parameters. *SIAM J. of Appl. Math.*, 1963.

[28] A. Neuman and C Lorenz. Statistical shape model based segmentationof medical images. *Computerized Medical Imaging and Graphics*, 22:233–243, 1998.

[29] Hans Bruun Nielsen. Cubic splines. Technical report, IMM, Dpartment of mathematical modelling, 1998.

[30] Torben Skov Nielsen. *Online prediction and control in nonlinear stochastic systems*. IMM/Institute of Informatik og matematisk modellering at D.T.U., Lyngby, Denmark, 2002.

[31] Jackson A.D. Nielsen M., Johansen P. and Lautrup B. Brownian warps: a least committed prior for non-rigid registration. pages Page no. 557–64 Publ. pages xxix+693. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002. 5th International Conference. Proceedings, Part II (Lecture Notes in Computer Science Vol.2489), Publisher Springer-Verlag, 2002. ISBN 3540442251.

[32] A. F. Siegel and R. H. Benson. A robust commparison of biological shapes. *Biometrics*, 38:341–350, 1982.

[33] Jacob Viborg Thornfeldt Sørensen. *Data assimilation in hydrodynamic models of continental shelf seas*. PhD thesis, IMM/Institute of Informatik og matematisk modellering at D.T.U., 2004. nr 126.

[34] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. In *IEE Transactions on Pattern Analysis and machine intelligence*, volume 14, pages 1061–1075. 1992.

[35] Mikkel B. Stegmann. Active Appearance Models: Theory, Extensions and Cases. Master's thesis, Kgs. Lyngby, 2000.

[36] D.H.Cooper T.F.Cootes, C.J.Taylor. active shape models their training and application. *Compuer Vision, Graphics and Image Processing*, 61(1):38–59, 1995.

[37] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, chapter 8.5. Springer, 2001.

[38] Hans-Peter waldl Ulrich Focken, Mattias Lange. Previento - a wind power prediction system with an innovative upscalinc algorithm. Technical report, Department of Energy and Semiconductor Research, Faculty of Physics Carl von Ossietsky University of Oldenburg,Germany.

[39] Hans-Peter waldl Ulrich Focken, Mattias Lange. Reduction of wind power prediction by spatial smoothing effects. Technical report, Department of Energy and Semiconductor Research, Faculty of Physics Carl von Ossietsky University of Oldenburg,Germany.

[40] Per Unden. Hirlam-5 scientific documentation, December 2002.

[41] G. Wahba. *Spline models for observational data.* SIAM, USA, Philadelphia, PA, 1990.

[42] E.H.T. Whitten, R.G.Craig, and M.L.Labowitz. *Future trends in Geomathematics*, pages 48–61. 1981.

Appendix A

Code

There exists code written for the testing of theory in chapter 4 and 5. The code can be divided into four parts:

1. Minimization of point sets in app. B.
2. Minimization of pixel sets in app. C.
3. Robust point matching in app. D.
4. Mixture vector matching app. E.

All functions and scripts are available from author or supervisor. Only the most important files are shown in the appendix, in order for someone to be able to pick up the essential parts, then one is of to a good start to carry out an implementation. A further amount of guidelines are in the 'testscript.m' available from author or supervisor.

Appendix B

Point minimization.

This Matlab file does the affine transforms of a points set consisting of coordinates.

```
% The source code (M-files) are provided under the
% terms of the GNU General Public License with an explicit
% clause permitting the execution of the M-files from within
% a MATLAB environment. See the LICENSE file for details.

function y1=cv_rt(x1,raw)
%function y1=cv_rt(x1,a,tx,ty)
%This function rotates and translates 'round the center of mass

mid=size(x1,1);

scl=[(max(x1(:,1))-min(x1(:,1))) (max(x1(:,2))-min(x1(:,2)))];
[x1c,com]=central(x1);
y1c=(homogen([x1c(1:mid),x1c((mid+1):(mid*2))]))*raw;
y11=decentral([y1c(1:mid),y1c((mid+1):2*mid)],com);
y1=[y11(1:mid),y11((mid+1):(2*mid))];


function vectordecentered=decentral(vector,com)
vectordecentered=(homogen(vector))*[1;0;com(1);com(2)];
return
```

```
function [vectorcentered,com]=central(vector,com)
if (nargin == 1)
com=[sum(vector(:,1))/(size(vector,1))...
 sum(vector(:,2))/(size(vector,2))];%center of mass
end
vectorcentered=(homogen(vector))*[1;0;-com(1);-com(2)];
return
function homf=homogen(vector)
homf=[[vector(:,1);vector(:,2)] [-(vector(:,2));vector(:,1)] ...
        [ones(1,size(vector,1)),zeros(1,size(vector,1))]'...
 [zeros(1,size(vector,1)),ones(1,size(vector,1))]'];
return
```

This Matlab file does the minimization between two point sets consisting of coordinates w.r.t. the affine transform parameters.

```
% The source code (M-files) are provided under the
% terms of the GNU General Public License with an explicit
% clause permitting the execution of the M-files from within
% a MATLAB environment. See the LICENSE file for details.

function x=cv_rtmin_v2(x1,vy)
%This function minimizes w.r.t. rotation and
%translation 'round the center
%of mass of x1

scl=[(max(x1(:,1))-min(x1(:,1)))...
 (max(x1(:,2))-min(x1(:,2)))];
[x1c,com]=central(x1);
[vyc,com]=central(vy,com);


x=homogen([x1c(1:(size(x1,1))),x1c((size(x1,1)+1)...
:(size(x1,1)*2))])\vyc;
%x(1)=acos(x(1))*360/(2*pi);
%x(2)=asin(x(2))*360/(2*pi);
%x(2)=(x(3)/scl(1))*100;
%x(3)=(x(4)/scl(2))*100;

function vectordecentered=decentral(vector,com)
```

```
vectordecentered=(homogen(vector))*[1;0;com(1);com(2)];
return
function [vectorcentered,com]=central(vector,com)
if (nargin == 1)
com=[sum(vector(:,1))/(size(vector,1))...
 sum(vector(:,2))/(size(vector,2))];%center of mass
end
vectorcentered=(homogen(vector))*[1;0;-com(1);-com(2)];
return
function homf=homogen(vector)
homf=[[vector(:,1);vector(:,2)] [-(vector(:,2));vector(:,1)] ...
        [ones(1,size(vector,1)),zeros(1,size(vector,1))]'...
 [zeros(1,size(vector,1)),ones(1,size(vector,1))]'];
return
```

Appendix C

Pixel minimization.

This Matlab file does the affine transforms of a an image so a template image is made adds optional noise to it and recovers the image by minimization.

```
function par=testvoptimtrnsfrm_v3(angle,txx,tyy,var)

rangle=30;
ub=45;
lb=0;%18;

load ~/data/mwind.mat

global mwindT;
global mwind;
global trcbweight;
global p;
global cost;
global midguess
global debug
debug=0;
p=[];
cost=[];

[trcbweight,dummy]=vtricube(23);
```

```
mwind=mwind(1:40,:);


a=0;tx=-20;ty=-20;
mwindR = imtransform(mwind,maketform('affine',...
[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);

a=(angle/360)*2*pi;
tx=0;ty=0;
[mwindR,x,y] = imtransform(mwindR,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);
a=0;tx=0;ty=0;
[mwindR,x,y] = imtransform(mwindR,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);

tx=txx;ty=tyy;
[mwindT,x,y] = imtransform(mwindR,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),...
'XData',[x(1) x(2)],'YData',[y(1) y(2)],'fill',0);
if(debug)
size(x)
disp(size(mwindT));
figure(87);imshow(mwind);
figure(88);imshow(mwindT);

figure(89);imshow(mwindT);
end

mwindT=mwindT+var.*randn(size(mwindT));

offset=(size(mwindT,2)-40)/2;

tx=0;ty=0;a=0;

[mwindT,x,y] = imtransform(mwindT,maketform...
```

```
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),...
'XData',[(x(1)+offset) (x(2)-offset)],'YData',...
[(y(1)+offset) (y(2)-offset)],'fill',0);
mwindTT=mwindT;


mguess=0:1:8;%2:8;
xguess=[];
for i=1:9;
midguess=mguess(i);

mg=midguess;
x=[1,(40+2*mg)];y=[1,(40+2*mg)];

mwindT=mwindTT;
a=0;tx=0;ty=0;
mg
[mwindT,x,y] = imtransform(mwindT,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'XData',...
[(1-mg) (40+mg)],'YData',[(1-mg) (40+mg)],'fill',0);

b=inline('testvdifferimtrnsfrm_v3(x)');
%[x,resnorm]=lsqnonlin(b,[lb 0 0],[lb 0 0],[ub 7 7]);

[x,resnorm]=lsqnonlin(b,[lb 0 0],[-inf -inf -inf],...
[inf inf inf],optimset('LargeScale','off','MaxFunEvals',1000) );
xguess=[xguess [x resnorm]'];

end


[val,indeksss]=min(xguess(4,:));
     par=xguess(:,indeksss);
%disp(xguess(:,indeksss))
disp(xguess)
resnorm
par(1)
```

```
a=(par(1)/360)*2*pi;tx=0;ty=0;
[Tmp,x,y] = imtransform(ones(40),...
maketform('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'))...
,'FillValues',0);


offset=(size(Tmp,2)-40)/2;

mwindT=mwindTT;

mg=offset;%8.5;


size(mwindT);
offset=(size(mwindT,2)-(40+mg*2))/2;
a=0;tx=0;ty=0;
[mwindT,x,y] = imtransform(mwindT,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'XData',...
[(1+offset) (40-offset)],'YData',[(1+offset) (40-offset)],'fill',0);

     %[x,resnorm]=lsqnonlin(b,[par(1) 0 0],[lb 0 0],[ub 7 7]);
[x,resnorm]=lsqnonlin(b,[par(1) 0 0],[-inf -inf -inf],...
[inf inf inf],optimset('LargeScale','off','MaxFunEvals',3000) );
disp(x)
     par=x;
```

This Matlab file does the affine transforms of a an image and returns the difference from another image to the minimizer function in matlab file above.

```
function difference = testvdifferimtrnsfrm_v3(par)

global mwindT;
global mwind;
global trcbweight;
global p;
global cost;
```

```
global midguess
global debug
midmwT=midguess+(size(mwind,2))/2;%round(size(mwindT,2)/2);
vrtr=vcalc_trnsform(par);



midvrtr=midmwT;%round(size(vrtr,2)/2);

difference = mwindT((midmwT-11):(midmwT+12),...
(midmwT-11):(midmwT+12))-vrtr((midvrtr-11):...
    (midvrtr+12),(midvrtr-11):(midvrtr+12));
if(debug)
figure(55);imshow(mwindT((midmwT-11):...
(midmwT+12),(midmwT-11):(midmwT+12)))
figure(56);imshow(vrtr((midvrtr-11):...
    (midvrtr+12),(midvrtr-11):(midvrtr+12)));
end

ss=sum(sum(difference.^2));
cost=[cost ss];
p=[p par(1)];

    function vrtr=vcalc_trnsform(par)

        global mwind;
        global debug;
        %vrtr=imrotate(iwind(8:34,8:32),par,'nearest','crop');

a=0;tx=-20;ty=-20;
mwindR = imtransform(mwind,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);

a=(par(1)/360)*2*pi;
tx=0;ty=0;
[mwindR,x,y] = imtransform(mwindR,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);
```

```
a=0;tx=0;ty=0;
[mwindR,x,y] = imtransform(mwindR,maketform...
('affine',[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'FillValues',0);

tx=par(2);
ty=par(3);
vrtr = imtransform(mwindR,maketform('affine',...
[cos(a) sin(a) tx;-sin(a) cos(a) ty;0 0 1]'),...
    makeresampler({'cubic','cubic'},'fill'),'XData',...
[x(1) x(2)],'YData',[y(1) y(2)],'FillValues',0);


if(debug)
figure(9); imshow(vrtr);
end
```

Appendix D

Robust Point Match.

Articles and matlab files for the demonstration of Robust point match, are available at the following home page:

http://noodle.med.yale.edu/ chui/tps-rpm.html

The code is covered by GNU Public License and is also available from the author of this thesis or supervisor.

Appendix E

Mixture Vector Match.

This little sample of code is just here to illustrate the calculation of a version of the correspondence matrix m. The rest of the functions and scripts are combinations of files for RPM and the alignment directly on pixel from chapter 4.4

```
elseif strcmp (m_method, 'mixture')
    vxcoor = [im2col(ones(1,40)'*(1:40),[40 40],'distinct') ...
    im2col((1:40)'*ones(1,40),[40 40],'distinct')];

    ycoor = [im2col(ones(1,40)'*(1:40),[40 40],'distinct') ...
    im2col((1:40)'*ones(1,40),[40 40],'distinct')];

  % Given v=tranformed(x), update m:
  y_tmp = zeros (xmax, ymax);

  for it_dim=1:dim
    y_tmp = y_tmp + (vx(:,it_dim) * ones(1,ymax)...
 - ones(xmax,1) * y(:,it_dim)').^2;
  end;

  y_coortmp = zeros (xmax, ymax);

  for it_dim=1:dim
    y_coortmp = y_coortmp + (vxcoor(:,it_dim) * ...
```

```
ones(1,ymax) - ones(xmax,1) * ycoor(:,it_dim)').^2;
  end;
    y_coortmp = y_coortmp./(39^2);
  une=ones(xmax, ymax);
  m_tmp = 1/sqrt(T) .* exp (-y_tmp.*(une+y_coortmp./T));

  %m_tmp = m_tmp + randn(xmax, ymax) * (1/xmax) * 0.001;

  m = m_tmp;

  % normalize accross the outliers as well:
  %sy          = sum (m) + m_outliers_row;
  %m           = m ./ (ones(xmax,1) * sy);

  % sx = sum(m')' + m_outliers_col;
  % m2 = m ./ (sx * ones(1,ymax));
  % m = (m+m2)/2;
```