

RECONSTRUCTION AND MOTION ESTIMATION OF SPARSELY SAMPLED IONOSPHERIC DATA

Matthew Philip Foster

A thesis submitted for the degree of Doctor of Philosophy
University of Bath
Department of Electronic and Electrical Engineering

2008

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Matthew Philip Foster

Abstract

Acknowledgements

Contents

Table of Contents	i
List of Figures	iv
List of Tables	vii
Acronyms	viii
1 Introduction	1
1.1 Thesis Overview	1
1.2 Publications	2
1.3 Imaging the Ionosphere	4
1.3.1 GPS Imaging	5
1.3.2 Constructing TEC Maps from GPS Data	7
1.3.3 Ionospheric Storms	8
1.3.4 Tracking Storm Features	10
2 Interpolation of Scattered Data	11
2.1 Scattered Data	12
2.1.1 Sparsity	13
2.2 Normalised Convolution	15
2.2.1 Introduction	15
2.2.2 Zero-order Normalised Convolution	15
2.2.3 Higher Order Normalised Convolution	16
2.2.4 Zero Order Adaptive Normalised Convolution	18

2.2.5	Gradient Estimation	19
2.2.6	Structure Adaptive Normalised Convolution	21
2.3	Triangulation Based Interpolation	29
2.4	Natural Neighbour Interpolation	30
2.5	Radial Basis Function Interpolation	31
2.5.1	Biharmonic Spline Interpolation	32
2.6	Kriging	33
3	Evaluating Interpolation Performance	35
3.1	Introduction	35
3.2	Interpolation Schemes for Scattered Data	37
3.3	Quantitative Evaluation & Experimental Results	38
3.3.1	Simulated Data Results	39
3.3.2	TEC Data Results	41
3.4	Discussion and Conclusions	46
4	Interpolation Artefacts and Error Distributions	49
4.1	Artefacts	49
4.1.1	Triangulation Based Linear Interpolation	50
4.1.2	Linear RBF Interpolation	51
4.1.3	Cubic Interpolation	52
4.1.4	Natural Neighbour Interpolation	54
4.1.5	Adaptive Normalised Convolution	54
4.1.6	General Examples	55
4.2	Interpolation Error Distributions	56
4.2.1	Error Skew	57
4.2.2	Confidence Limits	59
5	Motion Estimation Techniques	60
5.1	Data Sources	60
5.2	Model Based Approaches	61
5.3	Assumptions	62
5.4	Differential Analysis	64
5.5	Optical Flow	66

5.6	Template Matching and Correlation-Relaxation Labelling	67
5.6.1	Increasing Field Density	69
5.6.2	Sub-pixel Block Matching	70
5.6.3	Similarity Measures	70
5.6.4	Relaxation Labelling	72
5.7	Boundary Tracking	74
5.7.1	Segmentation	75
5.7.2	Boundary Matching	78
5.7.3	Boundary Vector Transformations	82
5.8	Other Techniques	84
6	Motion Estimation using Optical Flow and Correlation-Relaxation Labelling	86
6.1	Motion Estimation Using Optical Flow	86
6.1.1	Results	87
6.1.2	Conclusions	90
6.2	Motion Estimation Using Correlation Relaxation Labelling	91
6.2.1	Results	92
6.2.2	Conclusions	97
6.3	Conclusions	98
7	Motion Estimation using Boundary Matching	99
7.1	Segmenting the Tongue of Ionisation	100
7.2	Motion Estimation by Boundary Context Matching	103
7.3	Increasing Field Density	109
7.4	Conclusions	112
8	Suggestions for Further Work	114
8.0.1	Interpolation and TEC Mapping	114
8.0.2	Motion Estimation	115
	References	116

List of Figures

1.1	Example electron density profiles	5
1.2	The Earth's Magnetosphere	9
2.1	The effect of resolution on image sparsity.	14
2.2	Polynomial basis functions as used by first-order NC	17
2.3	Examples of zero- and first-order NC	18
2.4	RMSE of reconstructions using varying filter dimension and NC	18
2.5	Images comparing Sobel, DoNC and first-order NC edge detection	20
2.6	Image illustrating eigenvalues and anisotropy from the GST	22
2.7	Rotated 2-D Gaussian Filter	23
2.8	A Flow diagram showing the overall ANC process.	27
2.9	ANC interpolated images	28
2.10	Voronoi diagram	30
2.11	An example semivariogram with a fitted spherical model	34
3.1	Normalised histograms of simulated data	40
3.2	Proportional RMSE from reconstructed simulated multivariate data	41
3.3	Proportional RMSE from reconstructed simulated univariate data	42
3.4	Thin shell ionosphere model	43
3.5	Example TEC data from the Halloween Storm	43
3.6	Proportional RMSE for reconstructed TEC data	45
3.7	Example interpolated TEC data	46
4.1	Close up of a single rice grain	50

LIST OF FIGURES

4.2	Interpolated images of rice grains, demonstrating artefacts	51
4.3	Example interpolated SRTM DEM data, showing artefacts	52
4.4	Examples of overshoot (and undershoot) when performing cubic interpolation.	53
4.5	Examples of over- and under-shoot in high order interpolation	54
4.6	Example interpolated pyramids, showing peak and edge artefacts	54
4.7	Various distributions with differing kurtosis values	57
4.8	An example image with its semivariogram and a histogram of interpolation errors	58
5.1	Example TEC Image showing physical location	62
5.2	Example TEC images of the 2003 Halloween Storm	63
5.3	Modelled TOI vectors	64
5.4	Differential motion examples	65
5.5	Template Matching	67
5.6	Template Matching Process	68
5.7	A diagram illustrating sub-pixel template matching	71
5.8	Relaxation Labelling Terms	73
5.9	Component Morphology Definitions	76
5.10	Area Morphology Definitions	77
5.11	Contrast Morphology	77
5.12	Contrast and Area Comparison	78
5.13	Motion estimation by boundary correspondences	79
5.14	Example cost matrix created using weighted diffusion distance	82
6.1	Optical Flow Process	87
6.2	Optical Flow Examples	88
6.3	Optical Flow Examples	90
6.4	Optical Flow Examples	91
6.5	Template Matching Process	92
6.6	Non-overlapping block matching	93
6.7	Overlapping block matching	94
6.8	Sub-pixel block matching	95
6.9	Post-filtered relaxed overlapping block matching	96

6.10 Post-filtered relaxed sub-pixel block matching	97
7.1 Curve Matching Process	99
7.2 Histogram of TEC value distributions	101
7.3 Comparing area and correlation of contrast parameters	103
7.4 Segmentation Feedback Process	104
7.5 Watershed Transform Post-processing	104
7.6 TEC images segmented using morphology	105
7.7 Vectors from shape boundaries	106
7.8 A schematic description of the retreating snout problem	107
7.9 The retreating snout problem	107
7.10 Vectors estimated from boundary matches	109
7.11 RBF interpolated vector fields	110
7.12 Examples of increasing regularisation	111
7.13 RBF regularised vector fields	111

List of Tables

2.1	GST Products	21
2.2	Example Radial Basis Functions	31
6.1	Optical flow vector entropies	89
6.2	Vector entropy (bits/vector) statistics for various kinds of block based motion estimation.	95

List of Acronyms

AC	active Contour
ANC	adaptive normalised convolution
BSI	biharmonic spline interpolation
C/A	coarse acquisition
CCC	cross correlation coefficient
CME	coronal mass ejection
CM	curve matching
DD	diffusion distance
DoNC	differential of normalised convolution
EMD	earth mover's distance
EUV	extreme ultra-violet
GNSS	global navigation satellite systems
GPS	global positioning system
GST	gradient square tensors
IGS	International GNSS Service
IIR	infinite impulse response
IMF	Interplanetary Magnetic Field
MCC	maximum cross-correlation
ME	motion estimation
MIDAS	Multi-Instrument Data Analysis System
MM	mathematical morphology

LIST OF TABLES

NC	normalised convolution
NDC	normalised differential convolution
NN	natural neighbour
NS	navigational signals
PNSR	peak signal-to-noise ratio
PRN	pseudorandom-noise
RBF	radial basis function
RL	relaxation labelling
RMSE	root mean square error
RMS	root mean square
ROI	regions of interest
SAVD	sum of absolute value of differences
SC	shape context
SED	storm electron density
SIFT	scale-invariant feature transform
SSD	sum of squared-differences
SSE	sum squared error
SVD	singular value decomposition
TEC	total electron content
TOI	tongue of ionisation
TPS	thin plate spline
TSM	thin sell model
TOA	time of arrival
VHR	very-high resolution

List of Symbols

A	Anisotropy
c	Binary confidence map
$\phi(\dots)$	A radial basis function
$f(\dots)$	An (interpolated output) image
$g(\dots)$	A Gaussian filter kernel
ρ	Cross-correlation coefficient
\otimes	Convolution
σ_u	Standard deviation along u -axis.
\mathbf{x}	Position vector (i.e. $\mathbf{x} = [x, y, \dots]^T$)

Introduction

Many fields benefit from the introduction of ideas and techniques from others, and a great deal of advances arise in the 'gaps' between areas of work. One of the aims of this thesis is bridging the gap between image processing, computer vision techniques and atmospheric research, specifically ionospheric total electron content (**TEC**) mapping. These maps show the distribution of electrons in the ionosphere, and are used in a wide range of applications, from propagation forecasting and nowcasting, to calibration of satellite navigation systems through to scientific analysis of space weather events. This work aims to aid some of these applications by introducing the state-of-the-art in interpolation and computer vision to **TEC** mapping.

1.1 Thesis Overview

This thesis is structured as follows: the remainder of this chapter describes the phenomenon of scattered data, the nature of the data on which the interpolation methods described herein operate; and an introduction to the ionosphere and sensing it using global positioning system (**GPS**). The application of interpolation methods for this take is introduced here. The magnetosphere and geomagnetic storms are then briefly described and the motivation for the second major topic, tracking storm enhancements is explained.

The following three chapters deal with interpolation, including an introduction to various methods, and two ways of evaluating performance.

Chapter 2 describes sparsity, and then introduces various interpolation methods, including normalised convolution (NC), triangulation based techniques, radial basis function interpolation and kriging.

Chapter 3 deals with how the relative performance of different interpolation methods can be examined in a quantitative fashion. The methodologies used in this chapter include simulation-validation, for evaluating interpolation on simulated data-fields, and cross-validation, for evaluation using real data. The cross-validation method is applied to TEC data collected during the geomagnetic storm event that occurred on Halloween of 2003.

Chapter 4 moves on to analysing interpolation methods in a more qualitative fashion. It examines the various artefacts produced by different methods and discusses how error distributions can be examined to provide a wealth of information on how interpolation methods perform in specific cases.

The thesis remaining chapters consider motion estimation techniques.

Chapter 5 describes the data used in the next chapters, which consists of TEC maps of the northern polar ionosphere created from data collected during the 2003 Halloween Storm and discusses methods of motion estimation, including differential analysis, optical flow, template matching, correlation-relaxation labelling and boundary tracking. This chapter also introduces attribute mathematical morphology.

Chapter 6 is an initial study on the use of optical flow, template matching and correlation-relaxation labelling based motion estimation to track the motion of an area of enhanced electron density as it moves during the storm event.

Chapter 7 discusses the use of shape boundaries to infer motion. This chapter charts the development of a two-stage approach which makes use of mathematical morphology (MM) for segmentation, and shape context (SC) matching for motion estimation.

1.2 Publications

The following publications are the result of work contained here:

- M. P. Foster and A. N. Evans. An evaluation of interpolation techniques for reconstructing ionospheric TEC maps. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2153–2164, July 2008.

1.2. PUBLICATIONS

- M. P. Foster and A. N. Evans. Performance evaluation of multivariate interpolation methods for scattered data in geoscience applications *Proceedings of the IEEE Geoscience and Remote Sensing Symposium 2008*, July 2008 *In press*
- M. P. Foster and A. N. Evans. Segmentation and tracking of ionospheric storm enhancements. *Proceedings of the XIV Conference on Image and Signal Processing for Remote Sensing*, volume 7109, page 71090O. SPIE, 2008b.
- M. P. Foster and A. N. Evans. Motion estimation of polar ionospheric TEC maps using extended correlation-relaxation labelling *IEEE Geoscience and Remote Sensing Letters*, *In preparation*

1.3 Imaging the Ionosphere

The ionosphere is the region of the atmosphere, extending from an altitude of approximately 50 km and to over 1000 km. In this region, free electrons can exist for short periods of time, and form an electrically conducting plasma. These electrons are liberated when extreme ultra-violet (EUV) light from the Sun ionises neutral atoms in the atmosphere. The free electrons then form a plasma which has various effects on electromagnetic waves, including delaying or blocking their propagation at certain frequencies.

Radio waves are affected because as they propagate through the ionosphere, they cause excitation. If the wave frequency is less than the *plasma frequency*, which is the frequency at which the electrons and ions in a slab of plasma will oscillate when perturbed, it will be re-radiated, otherwise it will pass through. The plasma frequency is given by $f_N = \sqrt{80.5N}$, where N is the electron density. The *critical frequency* of a layer is the maximum frequency which can be from it reflected at vertical incidence. The critical frequency, of a layer is given by $f_c \approx 9 \times 10^{-6} \sqrt{N_m}$, where N_m is the maximum electron density of the layer (in electrons per m^3). This is the maximum frequency at which a radio wave will be reflected at vertical incidence. Critical frequencies in the various layers are denoted f_oE , f_oF_1 and f_oF_2 . Radio waves also experience Faraday rotation when passing through ionised regions, such as the ionosphere. This is equivalent to a time delay which is proportional to the level of ionisation, and inversely proportional to the square of the signal frequency.

Photoionisation is counteracted by two main processes, acting to reduce atmospheric ionisation. In both of these cases, the rate of recombination is controlled by the number of available neutral atoms, and so varies with altitude.

- The first is the *recombination* of electrons and ions to form neutral atoms once. There are two forms of recombination known as *radiative* and *dissociative*.
 - *Radiative* recombination is most common, and occurs when an electron and an ion recombine directly.
 - *Dissociative* recombination is less common, and involves a more efficient, two stage, process. In the first stage, positive ions interact with various neutral molecules replacing one of the atoms in the molecule. In the second stage, electrons combine with the positively charged molecule just created.
- The second is *attachment*, which occurs at lower altitudes where there are more neutral atoms, and involves electrons combining with neutral atoms to form negative ions.

The electron density of the ionosphere varies with altitude. This is due to several factors, including the distribution of neutral atoms, including the fact that their densities decrease with altitude, and differences in intensity of EUV wavelengths. The change of electron density with height is known as the electron density profile. This vertical profile contains several distinct layers, known as D , E , F_1 and F_2 , in increasing altitude. During the day, all four layers are present, because of high levels of photoionisation. However, at night, recombination dominates, and the D , E and F_1 layers are almost entirely depleted, leaving only the F_2 layer. These layers are illustrated in Fig. 1.1, which shows that the F_2 region is the only layer present at night, and has the highest electron density.

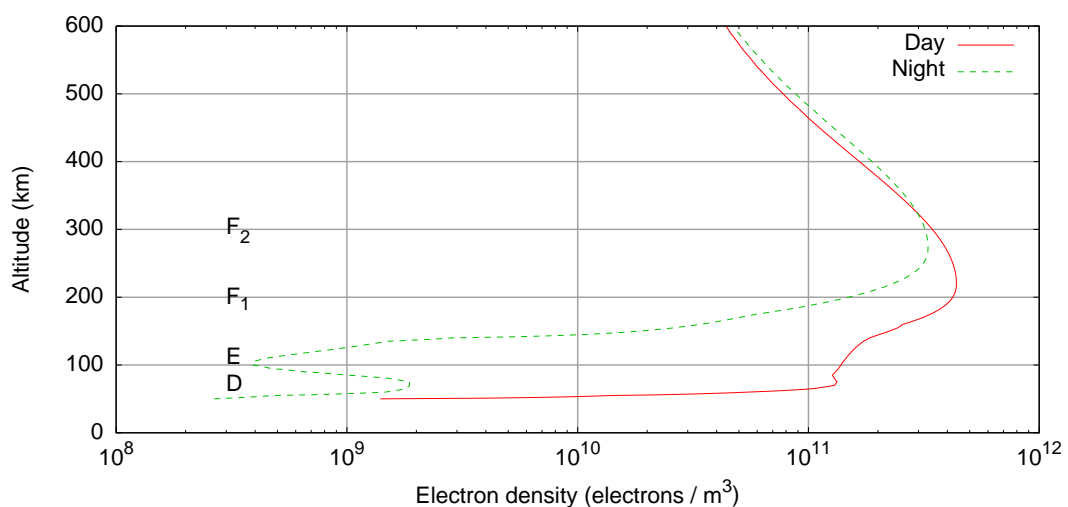


Figure 1.1: Day and night example electron density profiles, generated using IRI2001, for 30 June 2008 at noon and midnight.

1.3.1 GPS Imaging

The GPS is a timing and positioning system run by the US Department of Defence. The system is divided into three segments; known as the control, space and user segments. The *control segment* consists of various tracking stations around the world, with the main control centre at Schriever Air Force Base, Colorado, USA. These stations combine measured satellite position data with models in order to precisely compute their positions (ephemeris), and necessary clock corrections. These data are then uploaded to the satellites, for inclusion in navigational signals (NS), which are sent to receiver units.

The *space segment* consists of at least 24 satellites (currently 31) configured such that there are four satellites in six orbital planes, each inclined at 55° to the equatorial plane. This means that at least six satellites are always visible from most places on the surface of the Earth.

1.3. IMAGING THE IONOSPHERE

Each of the satellites transmits its own **NS**, containing information on the satellite, clock corrections, ephemeris and other information. This signal is created, and then added to a pseudorandom-noise (**PRN**) code known as the coarse acquisition (**C/A**) code. The resulting code is modulated on to carrier wave, known as L_1 , creating a spread-spectrum signal which can be used for ranging. A second spread spectrum signal known as L_2 is also transmitted.

Both L_1 and L_2 are modulated by a code known as the precision- (P-) code. A cryptographic key is required to allow use of the P-code. However, many modern receivers make use of L_2 code without decrypting the P-code to improve ranging performance (see below).

The *user segment* consists of **GPS** receivers and their associated operators, or users. **GPS** receivers require signals from four satellites in order to compute position in three-dimensions, and time.

The receiver creates a replica **C/A** code which it correlates with the received signal in order to find the correct time shift for the receivers clock. The receiver clock offset is known as the time of arrival (**TOA**), or the pseudorange. Once the correct offset is known, the received signal can be despread, and the **NS** demodulated.

GPS Positioning

By combining the ephemeris data from a given satellite with the pseudorange derived from the **C/A**, the receiver can fix its position to the surface of a sphere surrounding the satellite. By combining four such measurements, it is possible to fix the position to one unique point - the intersection of the four spheres.

During this process, the receiver's local clock must be continually adjusted, as clock skew can severely bias position measurements. This is done by examining the imaginary sphere intersections for systematic bias, and altering the clock according to the bias distance. Successive measurements from many satellites can reduce the clock error to negligible amounts.

Error Sources

There are several sources of ranging errors in the **GPS**. However, the influence of the ionosphere is the largest in magnitude. The ionosphere causes a frequency dependent delay in propagation of L-band signals. This delay varies according to elec-

tron concentration along the ray-path, and is typically at a minimum when the satellite is directly overhead. Current GPS handsets are able to reduce ionospheric errors to approximately 10 metres, by using models to estimate range corrections, which (assuming a maximum total electron content of 10^{18} e/m), could be as high as 26 metres for the L_2 band, and 16 metres for L_1 [pp. 294–307 Leick, 1995]. As the delay is frequency dependant, it is possible to make use of a linear combination of both L_1 and L_2 's pseudoranges to further reduce the effect of ionospheric delays.

1.3.2 Constructing TEC Maps from GPS Data

A great many fixed position GPS receivers are located around the world, collecting positional data. This can then used for a great many applications, including monitoring tectonic shift and crust strain, as well as for cartography, civil engineering and precision timing. The data recorded by these receivers can also sometimes be used to analyse the delays caused by the ionosphere and other atmospheric regions. Although measurements can be made of many different atmospheric regions and variables, in the case of ionospheric electron density, the delays can be used to derive information on electron concentration.

GPS receivers are mainly situated in areas where population density is high, and as the cost of running stations can be very high, they tend to mainly be found in more affluent countries. These factors mean that the distribution of receivers is largely random, with a considerably higher measurement density in the Northern hemisphere. The polar regions and oceans are particularly sparsely covered, making it difficult to produce TEC maps over these areas.

The two main methods of constructing maps of ionospheric TEC are interpolation [e.g. Foster and Evans, 2008a, Meggs et al., 2002] and tomography [e.g. Mitchell and Spencer, 2003].

Tomographic methods use ray-tracing to project path measurements onto a grid of voxels. Non-linear inversion techniques can then be used to reconstruct the electron content data at each voxel. Basis functions, and model based interpolation can be also used to help improve output quality. Generally, tomographic inversions can provide high resolution 3-D imagery, but require large amounts of data to do so. This means that in cases when tomography fails, standard interpolation methods must be employed.

In order to use standard 2-D interpolation methods, the data must first be converted from path measurements to spot values on a fixed height shell. This is known as the thin sell model (TSM) approach, and models the ionosphere as an infinitesimally thin

shell, at a given height, normally between 300 and 400 km [Hoffmann-Wellenhof et al., 2001, pp. 102]. The disadvantage of this approach is that information on the vertical structure is completely lost. However it has the potential to be computationally simple (depending on the interpolation method used), and allows for analysis of data which are too sparse for other methods to reconstruct. The conversion process is detailed in Section 3.3.2. Image based reconstruction methods then require these scattered data to be projected into a matrix to allow 2-D filtering or convolution to be carried out. As this topic forms a large part of this thesis, several chapters are devoted to explaining interpolation methods and its evaluating their performance.

Chapter 2 introduces the phenomenon of scattered data, and describes the property of sparsity, before introducing various command and state-of-the-art interpolation methods.

Chapter 3 then presents an analysis of these methods, using both simulated and real data. This quantitative analysis is then augmented in Chapter 4, by a qualitative analysis. This includes examinations of artefacts produced by interpolation methods, as well as an examination of error distributions, from the angle of using them to diagnose problems in interpolated outputs.

This chapter continues by discussing ionospheric storms. These events are caused by Solar interaction and can have wide reaching consequences to communications systems both on Earth and satellite based.

1.3.3 Ionospheric Storms

Much of the behaviour of the ionosphere is governed by how the Earth's magnetosphere and the Interplanetary Magnetic Field (IMF) interact and connect. The Earth's magnetosphere is a region of the atmosphere, linked to the top of the ionosphere which contains a mix of ions and electrons, held in place by the Earth's geomagnetic field and the solar wind. It consists of a long tail, about 70,000 km long, facing away from the Sun, which is swept out by the solar wind. The outer edge of the magnetosphere is known as the *magnetopause*. Outside of this is an area called the *magnetosheath*, which is bounded by the *bow shock*. This is a region where the solar wind velocity drops suddenly, and the magnetic field lines are highly compressed. Fig. 1.2 shows the positions of these regions schematically. Most high energy particles are prevented from entering lower parts of atmosphere by the magnetosphere, making its shielding effects very important for all life on Earth.

1.3. IMAGING THE IONOSPHERE

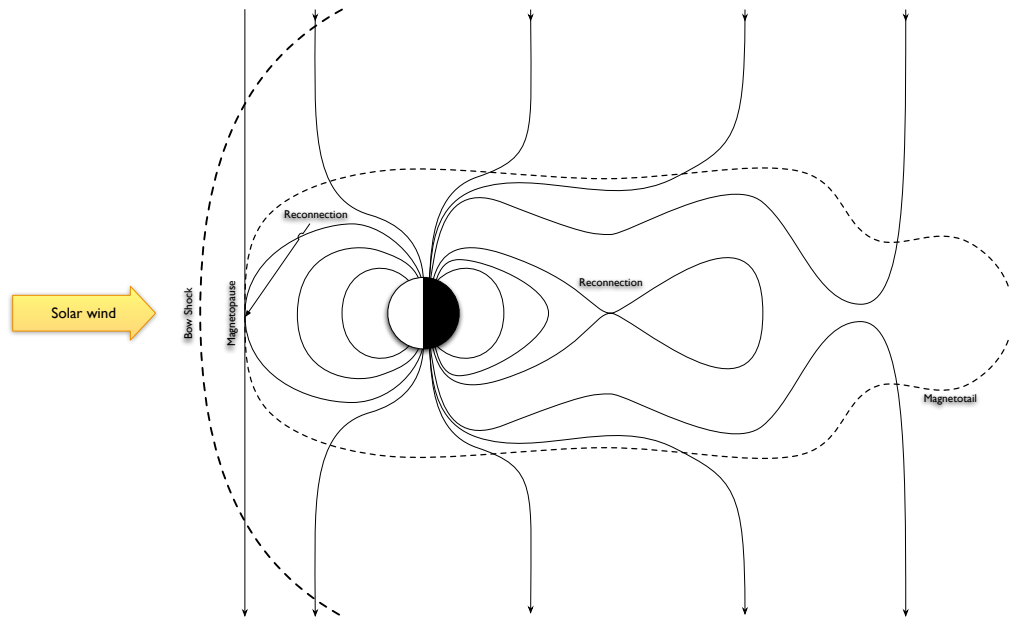


Figure 1.2: A schematic illustrating the Earth's magnetosphere. The solid lines represent magnetic field lines.

The **IMF** is formed by the steady outflow of solar wind from the Sun carrying the Sun's magnetic field. The Sun's rotation causes the **IMF** to be shaped like an Archimedean spiral¹. Intense variation in the Sun's surface magnetic field due to sunspots means that the orientation of the **IMF** varies with time. The magnitude of the 'vertical' component of the **IMF** is known as B_z , and its orientation determines whether solar wind plasma can enter the ionosphere. The polar cusps are regions which form between the sunward and tailward magnetic fields, and consist of open magnetic field lines. In the northern polar cusp, the magnetic field is directed towards Earth, and in the southern cusp, the magnetic field lines point away. When the orientation of **IMF** is southward, it is able to connect with the Earth's field, and as a result, solar wind plasma is accelerated through the magnetosphere, into the upper ionosphere. When southward B_z coincides with the Sun ejecting very large numbers of particles, due to a solar flare, or coronal mass ejection (**CME**) a geomagnetic storm can result. These storms can cause large auroral displays and disrupt power distribution and communications, making monitoring and predicting their occurrence, and and effects very important. These problems occur because the injected material consists of electrons and ions which increase the electron density of the ionosphere. Often, during large storms in the northern hemisphere, regions of enhanced electron density form in patches, known as storm electron density (**SED**), form in mid-latitudes, and then travel across the northern polar cap towards the night-side of the globe in a convective pattern known as a tongue of ionisation (**TOI**). The path taken across the polar cap depends on the direction of the **IMF**, with various different convection patterns forming under different conditions. These complex convective patterns are the subject of various scientific studies [[Foster et al., 2005](#), [Stolle](#)

¹called the Parker spiral, after Eugene Parker who predicted the Solar wind in the 1950s

et al., 2005].

There are various methods of remotely sensing the ionosphere's electron concentration, including the use of GPS Receivers and Satellites (see section 1.3.1); ionosondes (which use swept high frequency pulses); and satellite occultation by mounting GPS receivers on GPS satellites, it is possible to recover electron densities along limbs between the receiver and other GPS satellites, as they are occluded by the Earth.

The next section introduces GPS and discusses extraction of electron concentration data from signal received at GPS ground stations. It then discusses how these signals can be reconstructed into a full-image form.

1.3.4 Tracking Storm Features

One important use of TEC maps is the examination of changes in electron density during geomagnetic storms. The most obvious feature of note during large storms is the TOI. Tracking and analysing its motion allows the effects of a given storm to be evaluated, and helps to improve scientists understanding of the behaviour of the Earth's atmosphere and magnetic field under varying conditions. Whilst features are known to move across the pole empirical tracking which is not driven by models is an advancement not previously applied to TEC mapping. Chapter 5 is a broad introduction to several commonly used motion estimation techniques, two of which are then applied to ionospheric TEC data in Chapter 6.

Interpolation of Scattered Data

This chapter considers interpolating scattered data, an area which requires specially designed techniques. A great many different solutions to this problem have been proposed, and are routinely used. All of these use weighted combinations of the input data to construct output values. Methods of weighting can range from very simple cases to highly elaborate ones, depending on the technique.

An example of a *simple* case is nearest-neighbour interpolation, which chooses the closest input datum as the output for any given point (essentially setting a weight of one to the nearest input datum, and zero for all others).

Example of more *complex* cases, are weighing based on area of overlap between the cells of Voronoi diagrams, as used by natural-neighbour interpolation and the use of steered anisotropic Gaussian filters in sympathy with local image features to preserve edge information, as is used by adaptive normalised convolution (ANC).

Different methods many other aspects, including:

- Philosophy: normalised convolution (NC) based techniques use the fact that the *absence* of data, and *zero-valued* data are very different situations to improve output quality of convolution based interpolation methods. In contrast, geostatistical methods, such as kriging use the idea that data can be decomposed into a stochastic part and an autocorrelated part, and estimate these during interpolation.
- Complexity: methods range in complexity from simple convolution or filtering, through triangulation and Voronoi diagram creation, to solving very large linear systems.

- Output quality: different methods rely on different orders of fitted curves and basis functions, leading to a range of different qualities of output.
- Continuity of derivatives: closely related to complexity and output quality is the idea that interpolation methods can try to increase smoothness by ensuring continuous derivatives to different levels. Higher levels of continuity increase complexity and require more input data to compute.
- Intended use: methods which are designed to allow sensitive analysis should try and minimise artefacts in the output, whereas methods designed for real-time operation might cut corners in order to speed up operation.

Before discussing different interpolation methods in more detail, the meaning of scattered data, and how it arises must first be explained. Section 2.1 discusses scattered data and one very important property of scattered data which are projected into matrices: sparsity.

2.1 Scattered Data

Scattered data are data which are spread throughout an object or medium being measured, such that the data may not cover every part of the object evenly. Whilst scattered data-sets often occur by design, it is far more common for them to arise when measurements are made opportunistically, or by proxy. Such data-sets are common in geoscientific research.

A good example of *opportunistic* measurements leading to scattered data is the use of global positioning system (GPS) receivers to calculate electron content along paths between receivers and satellites. The number of receivers is largely random (and depends on amongst other things, population density, affluence and the presence of tectonic activity, especially fault-lines). This means that at any given moment, the number of available paths can differ greatly over different areas of the globe, and that the positions of samples is by no means regular.

A good example of *proxy* measurements leading to scattered data is meteor radar measurements of mesospheric winds. These systems use the movement of the trails of ablating meteoroids as tracers for atmospheric motion. This leads scattered measurements of the motion, as the positions of meteors entering the atmosphere is essentially random.

A good example of *intentionally* scattered data, is the use of methods such as stratified

2.1. SCATTERED DATA

random sampling [Ripley, 2004] Such schemes are used by scientists to help ease the process of data collection. This technique leads to fields of data where samples are randomly positioned throughout small areas, but the areas themselves are not random. An example where this kind of measurement system might be employed is in the measuring of tree growth [e.g. Ripley, 2004]. Other sampling methods are also available, and a large body of work from the 1950s and 60s deals with the variances introduced by these. See Ripley [2004], and references therein.

A specific case of scattered data considered in this thesis are measurements of electron content made along ray paths between GPS receivers and satellites. Data derived from measurements using GPS receivers can be used for far more than just navigation and positioning. One very important use of this data is the mapping and profiling of the ionosphere. Ionospheric delays are the main source of ranging error in GPS, so understanding how these delays change under varying ionospheric conditions is an important consideration in improving GPS accuracy. Also, the ionosphere is highly variable and heavily influenced by solar activity, and changes in conditions can have wide ranging consequences for long range radio communications and power distribution systems.

One very important property of input data which limits the quality of outputs from interpolation methods is *sparsity*. This is defined in the following section.

2.1.1 Sparsity

In terms of a matrix, sparsity, or sparseness, is a measure of how empty it is. Input data sparsity has a large effect on the fidelity of interpolated images. Karvanen and Cichocki [2003] define sparseness as the ℓ_0 norm, divided by the number of elements in the image. This is simply the proportion of non-zero values, as the ℓ_0 is defined as:

$$\|\mathbf{x}\|_0 = \frac{\#\{j, x_j \neq 0\}}{N} \quad (2.1)$$

Where \mathbf{x} is the matrix, with N elements and $\#$ is a function which counts the number of time its content evaluates to *true*. This definition leads to an entirely empty matrix having a sparseness of zero, and a full one having a sparseness of one. As this is in opposition to the definition, this thesis uses $1 - \|\mathbf{x}\|_0$ instead, and uses the term *sparsity*, as opposed to sparseness, for clarity.

Sparsity is only defined for data in matrices, as the notion of ‘elements’ or ‘pixels’ does not exist for scattered data. The sparsity of a given data-set is therefore related to the resolution of the matrix into which it has been projected. Fig. 2.1 demonstrates how

2.1. SCATTERED DATA

changing the resolution of a matrix affects the sparsity, by showing the projection of a small set of scattered data into matrices with various resolutions. The sparsity can be seen to increase as the resolution of the matrix is increased.

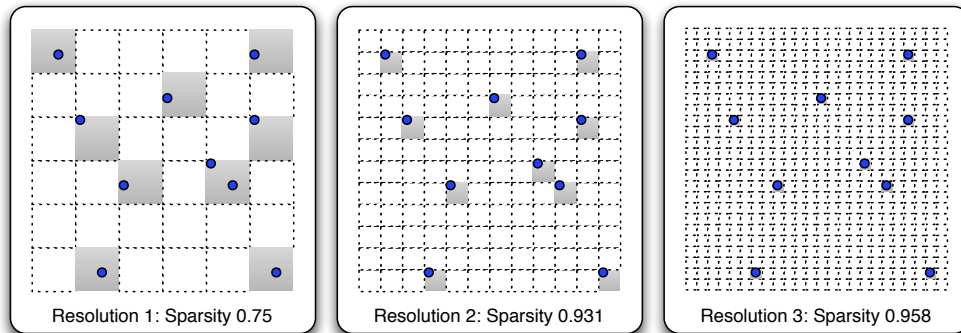


Figure 2.1: The effect that projecting scattered data into different resolution matrices has on sparsity. The small circles represent data points, and grey squares represent matrix elements which take the values of the data.

The effect of sparsity on output errors is discussed in a detailed study applied to both simulated and real data in Chapter 3.

The remainder of this chapter introduces various state-of-the-art and common methods for interpolating scattered data. These include **NC**, as well as methods based on triangulation, radial basis function (**RBF**) methods and kriging.

2.2 Normalised Convolution

2.2.1 Introduction

NC techniques are a class of methods which make use of convolution operations for the efficient interpolation and regularisation of image data; that is, data projected into matrices. They differ from other techniques in the fact that they make an implicit distinction between zero-valued and unavailable samples.

Since their initial proposal by [Knutsson and Westin \[1993\]](#), **NC** techniques have been steadily gaining in popularity, and have been applied to medical imaging see [e.g. [Estepar et al., 2003](#)], tensor field regularisation [[Westin and Knutsson, 2003](#)], motion compensation [[Farneback, 2002](#)], irregular-data fusion [[Pham, 2006](#)], and interpolation [[Pham and van Vliet, 2003](#)]. This work represents the first foray into the application of **NC** techniques to the reconstruction of geophysical data.

The most basic form of **NC**, known as zero-order **NC** is described by the following equation:

$$f = \frac{f_i \otimes g}{c_i \otimes g} \quad (2.2)$$

Where f_i is the input data, c_i is a binary map of input data positions – the ‘confidence map’, g is a kernel function and \otimes denotes the convolution operator. **NC** is therefore implemented using simple filtering operations, and so can be very computationally efficient. The term normalised comes from the division by the denominator, which serves to normalise the filtered input data.

2.2.2 Zero-order Normalised Convolution

Zero-order **NC** is similar in output to linear interpolation. The input image is convolved with a suitably sized filter kernel, in order to determine the contribution to the output by different input data. A confidence map, describing the positions of input points is then convolved with the same filter, in order to give the contribution of the filter kernel to each point in the filtered input. The filtered input is then divided by the filtered confidence map, removing the contribution from the filter, and leaving the interpolated output.

An alternative explanation is that **NC** produces a local model of the input data using projections onto a set of basis functions. The locality comes from the kernel at each pixel, and the basis function is a polynomial whose order is generally less than two,

and is usually zero.

Filters used for **NC** are usually Gaussians [e.g. [Pham and van Vliet, 2003](#), [Pham, 2006](#)], although various other kernels have been used. For example, initial studies [[Knutsson and Westin, 1993](#)] used a raised cosine.

Simple zero-order **NC** is highly efficient, only requiring operations that can be executed extremely quickly with modern computing hardware. It is also intuitively simple, and requires no triangulation or calculation of derivatives. However, filter sizes must be chosen such that they are as small as possible whilst encompassing all input data, or gaps will appear in the output. Using a filter which is larger than necessary will result in over-smoothing in areas where data are closely spaced. In effect, this is discarding data, a proposition which seems counter-intuitive when dealing with irregularly sampled data.

In order to improve output quality of zero-order **NC**, the filter size can be adapted, and a suitable filter size for each pixel chosen. The simplest possible adaptation of this kind is to vary the filter radii so that the filter used at any given point is related to the distance to the nearest sample. This, and other adaptations are known as **ANC**.

2.2.3 Higher Order Normalised Convolution

So far, discussion has been limited to **NC** using constant bases. This section discusses higher order **NC**, where projection onto higher order bases is carried out, using matrix inversions. Due to the irregularity of sample positions, inversions must be carried out at every output position. For this reason, first- and higher-order **NC** methods are more computationally expensive than zero-order **NC**.

First order **NC** is the simplest improvement over zero-order **NC**, and uses bi-quadratic basis functions, where zero-order **NC** uses a constant. Nine-convolutions are required in all: six for the numerator, and three for the denominator. The following equation describes the process mathematically:

$$\begin{bmatrix} f_1 \\ f_x \\ f_y \end{bmatrix} = \left(\begin{bmatrix} g & g \cdot x & g \cdot y \\ g \cdot x & g \cdot x^2 & g \cdot xy \\ g \cdot y & g \cdot xy & g \cdot y^2 \end{bmatrix} \otimes c_i \right)^{-1} \times \left(\begin{bmatrix} g \\ g \cdot x \\ g \cdot y \end{bmatrix} \otimes (c_i \cdot f_i) \right), \quad (2.3)$$

Where g , $g \cdot x$, $g \cdot y$, $g \cdot x^2$, $g \cdot xy$, and $g \cdot y^2$ are filter kernels (g) that have been multiplied by surfaces of various orders, this can be see in [Fig. 2.2](#). The form is similar to [\(2.2\)](#),

but involves an inversion, instead of a division, because of the introduction of multiple basis functions. It also outputs f_x and f_y , the first derivatives in the x - and y -directions. The overall process of first-order NC is therefore one of fitting basis functions, and then normalising the output.

Orders of higher than one are possible, but are again more computationally expensive. However, second-order NC would yield second derivatives, which could be useful in some situations.

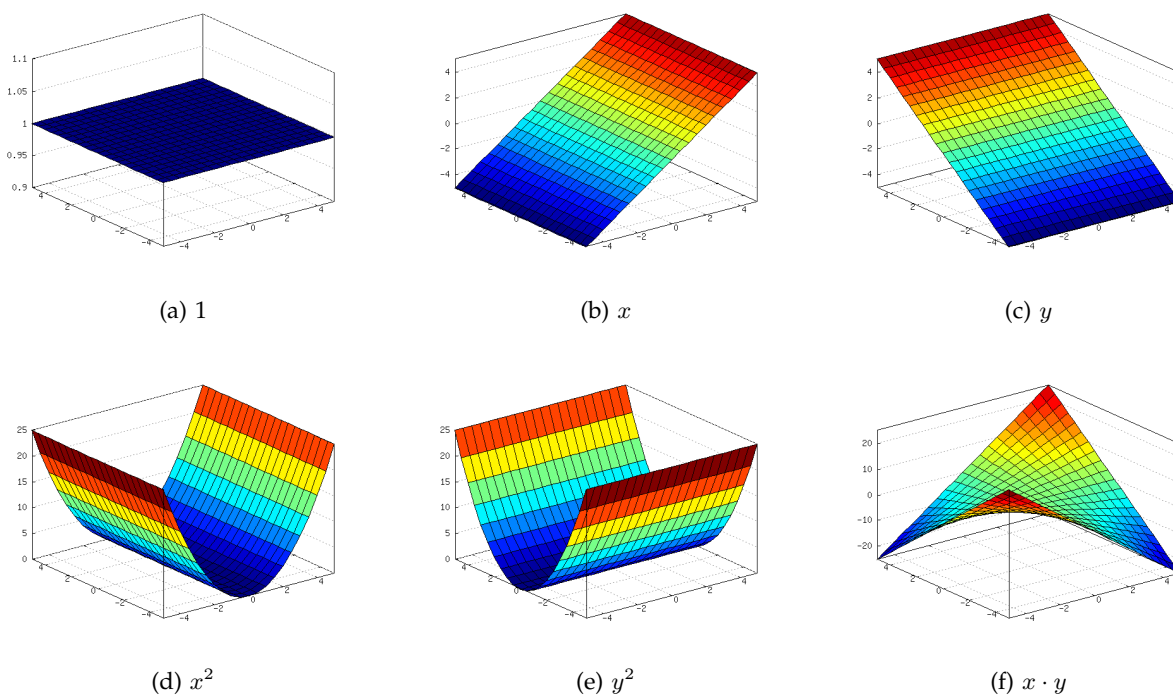


Figure 2.2: Polynomial basis functions as used by first-order NC

First-order NC provides little improvement over zero-order NC. Both of which exhibit performance similar to linear interpolation (but without a faceted appearance) when filters are sized such that there are no output gaps. The main advantage is the fact that gradients are returned in addition to the interpolated image.

First order NC is more computationally expensive than zero-order NC because of the use of multiple basis functions, and matrix inversions. However, tests on the crane image (Fig. 2.3 indicate that first-order NC is more sensitive to filter size than zero-order NC. This is shown in Fig. 2.4, in which a strong negative gradient is visible in the left-hand section of the first-order NC graph. This section coincides with small filter sizes, suggesting that first-order NC is highly sensitive to changes in the size of small filters.

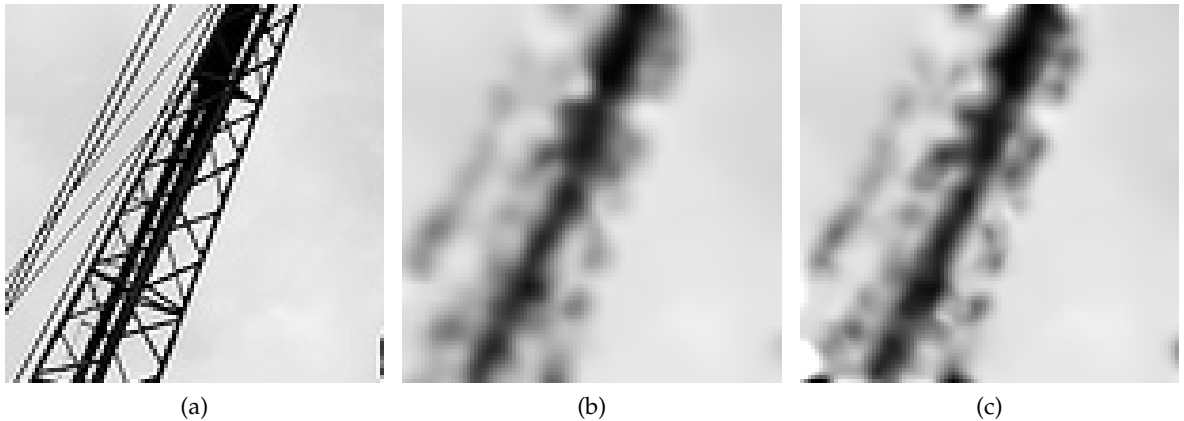


Figure 2.3: (a) Small section of an image of a dockside crane. This was sampled to $\approx 90\%$ sparsity, and reconstructed using (b) zero-order NC, and (c) first-order NC.

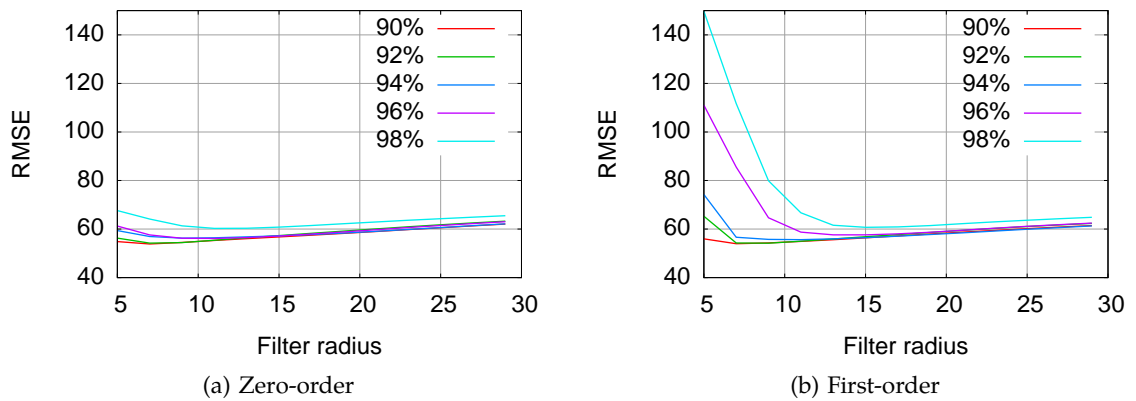


Figure 2.4: Graphs showing RMSE of reconstructions of the crane image from various sparsities and varying filter dimension, using (a) zero- and (b) first-order NC.

2.2.4 Zero Order Adaptive Normalised Convolution

The most simple form of ANC, known as *size adaptive NC* adapts the filters such that the kernel standard deviation at any given point is specified by the Euclidean distance transform [e.g. Hlavac et al., 1999] (or other distance transform) of the input samples.

Alternatively, a more computationally expensive scale-space based method can be used [e.g. Pham, 2006]. This method makes use of a scale-space pyramid to allow estimation of the filter dimension at which a given certainty constant (C) is found. The pyramid is constructed by filtering the confidence map with non-normalised Gaussians of exponentially increasing scales ($\sigma_i = 2^i$, $i = -1, 0, 1, 2, \dots$). Recursive Gaussian filtering [Young and van Vliet, 1995] can be for high-speed operation. The pyramid values at each scale will then increase quadratically, due to the non-normalised filters, and a quadratic regression can be used to estimate the scale at which the desired certainty constant is found. This scale is then used to set the filter sizes at each point, in order to minimise

smoothing.

Unfortunately, size adaptation alone does not provide much of an improvement over simple zero-order **NC**, and in order to further boost performance, alternative methods of adaptation are needed.

As well as adjusting the filter sizes in sympathy with input sample positions, the filters can be adapted according to properties of the image being reconstructed. Using 2-D Gaussians gives three variable parameters, the standard deviations along two axes and filter orientation. By lining filters up with the edges and other features, the output image should contain more similar properties to the input image. However, this requires properties of the sampled image (specifically edge information) to be known, and in non-simulated situations original input images are very rarely available.

For this reason, there are methods of gradient calculation which are based on **NC**, and are designed to provide edge information in irregularly sampled images. These are described in the following section.

2.2.5 Gradient Estimation

Estimation of gradient in irregularly sampled images requires the image to be interpolated and then differentiated, usually along both the x - and y -axes.

There are two **NC** based methods of estimating gradient. This first, *first-order NC* was described in section 2.2.3, and produces gradients as a product of its normal process. The second, *differential of normalised convolution (DoNC)* uses differentiated zero-order **NC**, and is described below.

Differential of Normalised Convolution

The **DoNC** method is formed by applying derivative operators to the **NC** equation (2.2), in two directions. This gives the following equations for the x -axis:

$$\Delta_x \left(\frac{D(x, y)}{N(x, y)} \right) \equiv \frac{D_x(x, y) \times N(x, y) - N_x(x, y) \times D(x, y)}{N^2(x, y)} \quad (2.4)$$

where:

$$D_x(x, y) = x.g(x, y) \otimes f_i(x, y), \quad (2.5)$$

and:

$$N_x(x, y) = x \cdot g(x, y) \otimes c_i(x, y). \quad (2.6)$$

In the above equations (2.5) and (2.6), $x \cdot g(x, y)$, is an edge enhancement filter which could be any arbitrary (normally Gaussian) filter multiplied by a variable x . This effectively tilts the filter relative to the x -axis.

The same process is also extended to the y -axis, and both processes are applied to sampled inputs, they can be used to provide the edge vector $[\Delta_x, \Delta_y]^T$.

As discussed above, the filters used for DoNC should be chosen such that there are no gaps in the output, but whilst attempting to minimise over-smoothing. Fig. 2.5b shows an example of DoNC and first order NC applied to an image sampled to $\approx 90\%$ sparsity, both used the same size Gaussian filters (11 with $\sigma \approx 3$). For comparison, the Sobel operator has also been applied to the original image.

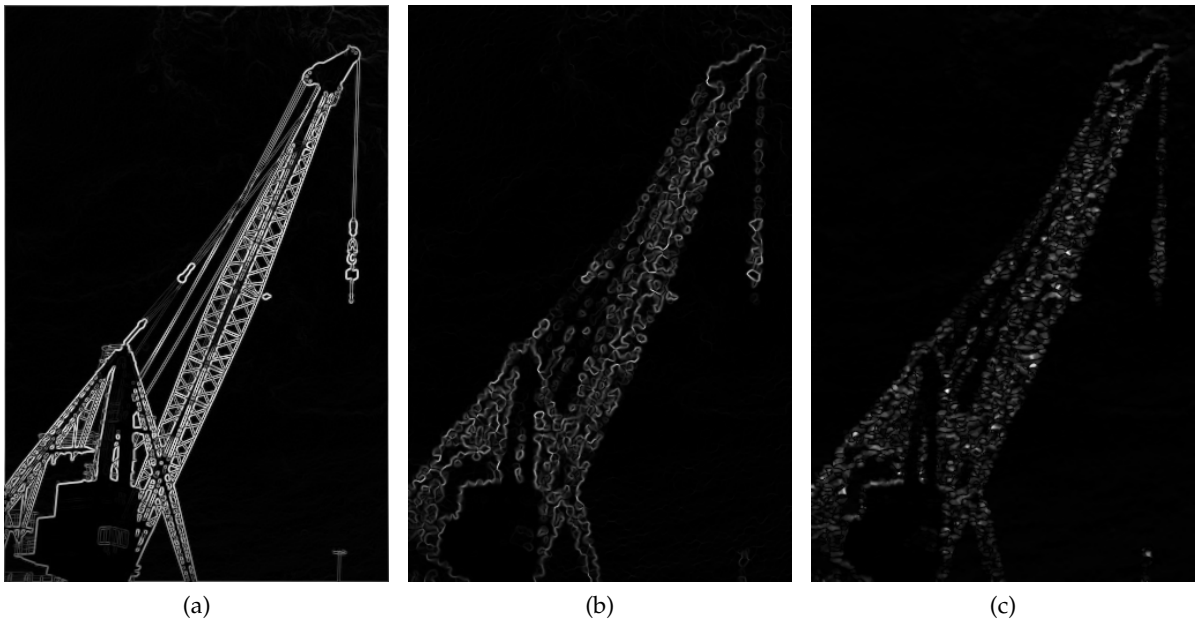


Figure 2.5: Images illustrating the difference between (a) Sobel edge detection on an unsampled original image, (b) DoNC and (c) first-order NC. Both images were reconstructed using 11 filters with $\sigma \approx 3$.

Gradients from First-order NC

Gradients calculated using first-order NC are similar to those produced by DoNC, but can be slightly more accurate for some images, such as those with a large amount of high-frequency content [de Jong et al., 1998]. The lower computational complexity, and similar performance of DoNC suggest that it is a sensible choice in practical situations

where speed is important.

2.2.6 Structure Adaptive Normalised Convolution

Once estimates of image gradients are available, they can be analysed in order to provide information on image structure, including local energies, orientations and anisotropies. These properties describe gradient, edge direction and how non-uniformly varying a give image is.

These properties can then be used to set the size and orientation of the **NC** filters in order to maintain features, and high frequency detail, whilst ensuring that no gaps appear in the output, a process which has been used in the past by **Nitzberg and Shiota [1992]**, **Almansa and Lindeberg [2000]**, and specifically adapted for use in **ANC** by **Pham and van Vliet [2003]**.

Kass and Witkin [1987] describe that fact that anisotropy can be detected by examining local power spectra, and noting that high-frequency energy will tend to lie perpendicular to the direction of flow. It is then suggested that orientation-selective linear filters can be used to detect this clustering energy, and hence the anisotropy of the local area. **van Vliet and Verbeek [1995]** develop this idea by using smooth local-gradient measures to form matrices known as gradient square tensors (**GST**), which are approximately equivalent to covariance matrices of the gradients, and are given by:

$$GST = \begin{bmatrix} g_x^2 & g_{xy} \\ g_{xy} & g_y^2 \end{bmatrix} \quad (2.7)$$

The effect of smoothing is the localisation of the matrix, so that when its eigenvectors are calculated, they correspond to the local area. Each pixel will have an associated **GST**, and calculating these gives the various pieces of information, which are summarised in Table 2.1.

λ_1	Smallest eigenvalue
λ_2	Smallest eigenvalue
$A = 1 - \frac{\lambda_2}{\lambda_1}$	anisotropy (consistency of local orientation)
φ_1	local gradient direction (direction associated with λ_1)
φ_2	local orientation (direction associated with λ_2)
$\lambda_1 + \lambda_2$	local energy

Table 2.1: GST Products

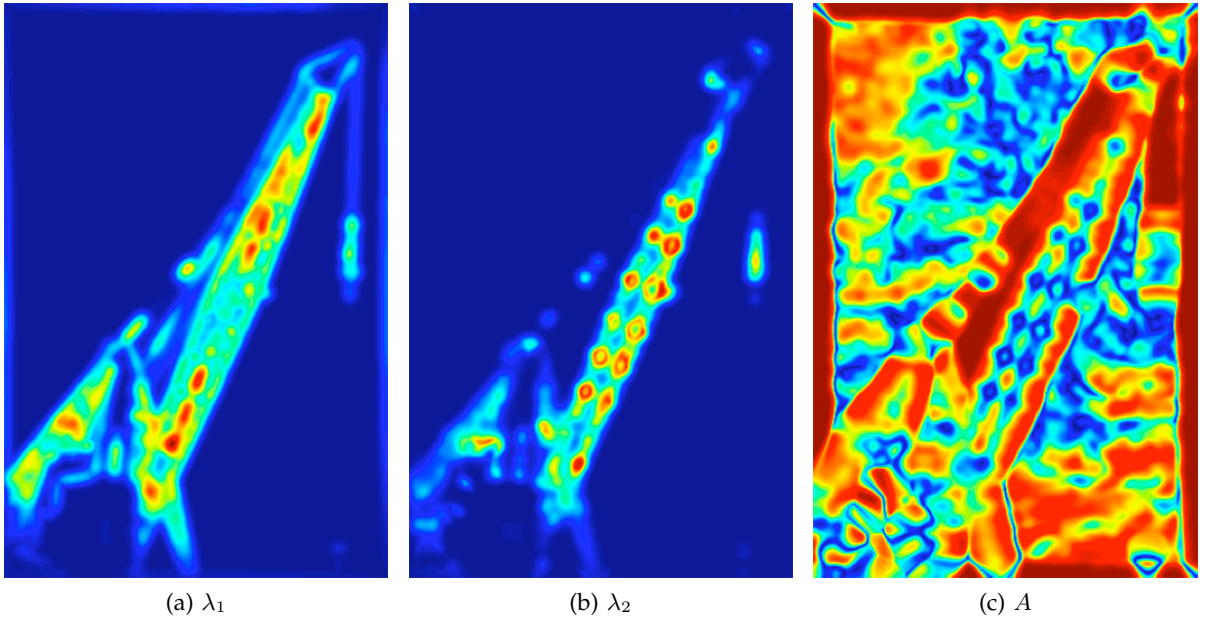


Figure 2.6: Eigenvalues and anisotropy. (a) and (b) the largest and smallest eigenvalues of the crane image and (c) the anisotropy. Images were generated using smoothed gradient vectors from Sobel operators.

The two most useful of these products are the anisotropy (A , see Table 2.1 and Fig. 2.6), and φ_2 , the ‘local direction’, which is given by:

$$\varphi_2 = \tan^{-1} \left(\frac{g_{xy}}{\lambda_2 - g_y^2} \right). \quad (2.8)$$

These measures are then used to set the size of the filters at any given point using:

$$\sigma_u = C(1 - A)^\alpha \sigma_a \quad (2.9)$$

and

$$\sigma_v = C(1 + A)^\alpha \sigma_a \quad (2.10)$$

Where u and v denote filter axes, and σ_u and σ_v correspond to Gaussian standard deviations along those axes. These three parameters ($\sigma_u, \sigma_v, \varphi_2$) can then be used to set the size and rotation of the filter used at any given output point (Fig. 2.7 shows how these parameters relate to a rotated Gaussian kernel).

The GST process can also be used to extract information on local curvature, which can then be used to warp filters [van Ginkel et al., 1999].

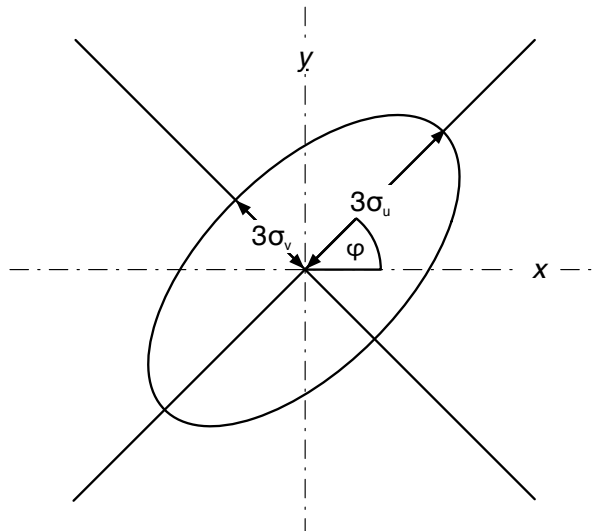


Figure 2.7: A schematic diagram showing shape, and salient parameters of a rotated 2-D Gaussian. The oval indicates 3σ confidence limits in each axis, which corresponds to a 99.7% confidence interval, and is a good place to truncate filters.

Implementation

Various points in the above processes require filtering with Gaussian kernels for either smoothing, or in the case of **NC** localisation of output contributions. Multiple filtering stages can quickly lead to high computation times, making it prudent to investigate methods of speeding up these processes.

Two types of filtering process are heavily used by **ANC**. These are isotropic and anisotropic filtering, and both have their own sets of speed improvements. These are dealt with separately below.

Isotropic Filtering Generation of 2-D Gaussian filter kernels can be sped up appreciably by using the property of separability, which allows symmetrical functions to be split down into orthogonal components. This is possible when a filter kernel can be written as the output product of two vectors. For example, consider the Sobel kernel:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.11)$$

2.2. NORMALISED CONVOLUTION

Using this, one can quickly generate 2-D Gaussian filters by convolving two vectors:

$$g(u, v; \sigma_u, \sigma_v) = \frac{1}{\sqrt{2\pi}\sigma_u} \exp\left\{-\frac{1}{2} \frac{u^2}{\sigma_u^2}\right\} \otimes \frac{1}{\sqrt{2\pi}\sigma_v} \exp\left\{-\frac{1}{2} \frac{v^2}{\sigma_v^2}\right\} \quad (2.12)$$

Alternatively, since convolution is associative, this property can be used to quickly filter images by using two 1-D processes, instead of a single 2-D process.

The image is first filtered in one dimension, using one kernel, to give an intermediate image. This image is then filtered in the other dimension, using the other kernel. The output is exactly the same as if a more complex 2-D filtering process had been used, but is much faster thanks to the lower complexity.

Complexity of filtering operations which can be decomposed drops from M^2P^2 multiplies and additions to $M^2(2P)$. Where M is the image dimension, and P is the filter dimension. Therefore, the relative speedup is $\frac{M^2}{2P}$.

A great many filters can be separated into orthogonal components, using the following algorithm:

1. Perform the singular value decomposition (**SVD**) of the filter kernel. This will give three values: U (output basis vectors), S (singular values) and V (input basis vectors). The can be recombined to reconstruct the original kernel using:

$$K = USV^* \quad (2.13)$$

Where $*$ denotes the conjugate matrix transpose.

2. Take the rank of the singular values. This corresponds to the number of independent rows in S . If $\text{rank}(S) = 1$, then the filter kernel is separable.
3. Two separate the kernel, two 1-D vectors are formed by multiplying U and V by the square root of the non-zero value from S :

$$\begin{aligned} K_u &= U\sqrt{s}, \\ K_v &= V\sqrt{s} \end{aligned} \quad (2.14)$$

Gaussian filtering can also be carried out using recursive [infinite impulse response (**IIR**)] filters, for further speed improvements [Young and van Vliet, 1995].

Anisotropic Filtering Filtering images using steered, anisotropic filters is a computationally expensive operation, as convolution with a different filter kernel is required at each output point. As with isotropic filtering, speed improvements can be made to anisotropic filters by using approximate filter separability, as described by Geusebroek et al. [2002].

This process uses two filtering stages, one parallel to the x -axis, and another along the direction of rotation. The first filter stage uses the following filter:

$$g_x(x, y) = w_o f(x, y) + \sum_{i=1}^{\lfloor N/2 \rfloor} w_i (f(x - i, y) + f(x_i, y)) \quad (2.15)$$

Where N is the size of the Gaussian filter, whose weights are given by $w_o \dots w_N$. The filter standard deviation is:

$$\sigma_x = \frac{\sigma_u \sigma_v}{\sqrt{\sigma_u^2 \cos^2 \theta \sigma_v^2 \sin^2 \theta}} \quad (2.16)$$

In this equation, θ is the filter orientation and σ_u and σ_v are the standard deviations of the filter before rotation. The second pass of the filter operates on the intermediate output of the first, using linear interpolation to calculate off-grid values, and has the following form:

$$\begin{aligned} g_\theta(x, y) = & w_o g_x(x, y) + \\ & \sum_{j=1}^{\lfloor M/2 \rfloor} w_j \{ a(g_x(\lfloor x - j/\mu \rfloor, y - j) + \\ & g_x(\lfloor x + j/\mu \rfloor, y + j) + \\ & (1 - a)(g_x(\lfloor x - j/\mu \rfloor - 1, y - j) + \\ & g_x(\lfloor x + j/\mu \rfloor + 1, y + j)) \} \end{aligned} \quad (2.17)$$

Where $\mu = \tan \varphi$ is the direction along which this filtering operation occurs. This term arises from the equation for the Gaussian standard deviation for this pass, which is:

$$\sigma_\phi = \frac{1}{\sin \phi} \sqrt{\sigma_u^2 \cos^2 \theta \sigma_v^2 \sin^2 \theta} \quad (2.18)$$

φ (and μ) are found using:

$$\mu = \tan \varphi = \frac{\sigma_u^2 \cos^2 \theta \sigma_v^2 \sin^2 \theta}{(\sigma_u^2 - \sigma_v^2) \cos \theta \sin \theta} \quad (2.19)$$

As is the case for other separable filters, the overall complexity depends on the size of the filter being used. This process can also be extended to use recursive filters for further speed increases [Geusebroek et al., 2002].

Fig. 2.8 diagrammatically shows the whole ANC process, including various filtering stages which can be used in order to tune the process for different types of image. Finally, Fig. 2.9 shows an example image, including before and after sampling, and after reconstruction using ANC.

2.2. NORMALISED CONVOLUTION

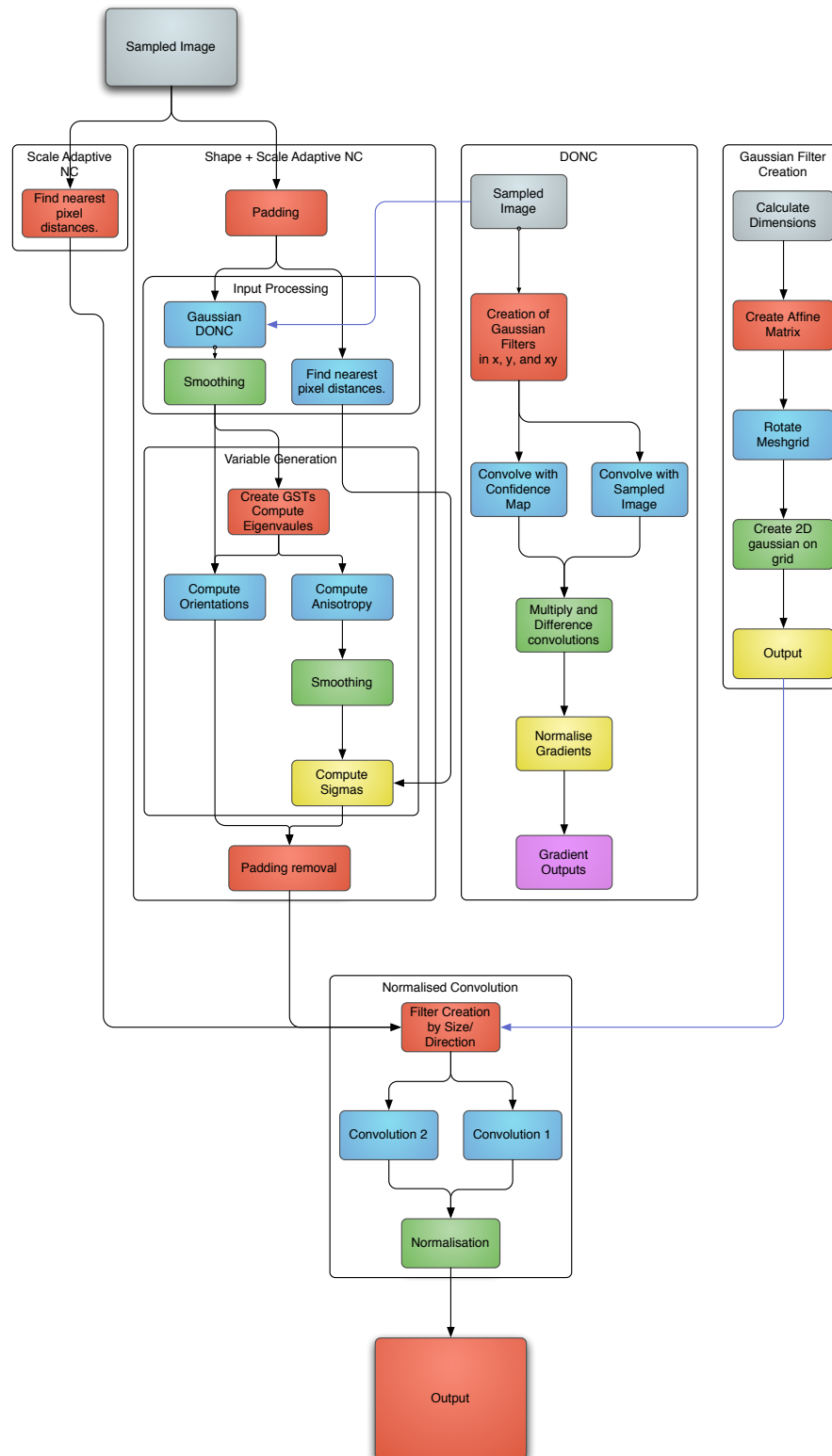


Figure 2.8: A Flow diagram showing the overall ANC process.

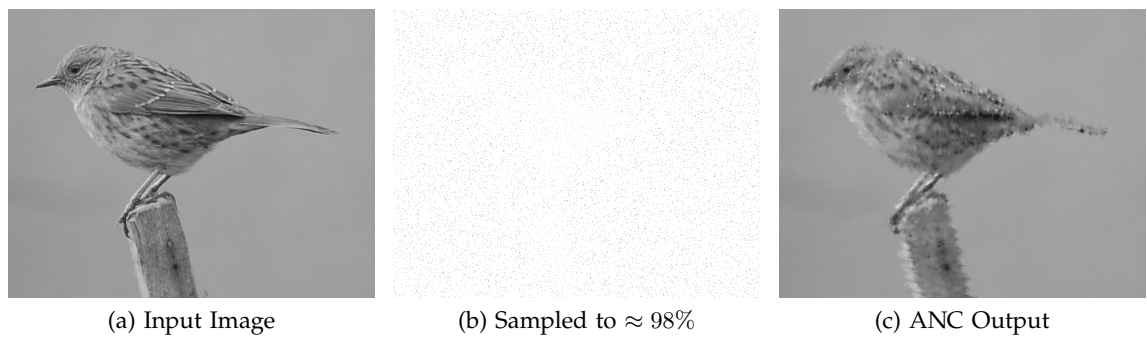


Figure 2.9: A photograph of a dunnock (a), which has been randomly sampled to $\approx 98\%$ (b), and then reconstructed using ANC (c).

2.3 Triangulation Based Interpolation

Triangulations are often used as the basis for interpolation of irregular data – the Delaunay triangulation in particular has some properties which make it particularly useful, and so most triangulations are of this type.

The result of a Delaunay triangulation is a set of vertices and edges (a graph) which conveniently defines the convex-hull of the data. It also gives a simple definition of locality. The local neighbours of any given point are those which are connected to it by triangulation edges.

The geometric dual of the Delaunay triangulation, known as the Voronoi diagram, also has many useful properties. Its edges form cells around each point, such that the area within each cell is closest to the point at its centre. Cells at the edges of Voronoi diagrams are unbounded. Voronoi edges are always perpendicular to Delaunay edges.

Once a data-set has been triangulated, each point in the data-set will be connected to several others by triangle vertices. Given the values of a triangle's nodes (f_i , where $i = 1, 2, 3$), the interpolated value of any point within the triangle, can be found using

$$f(x, y) = \sum_{i=1}^3 \phi_i(x, y) f_i \quad (2.20)$$

where $\phi_i(\mathbf{x})$ is the interpolating *basis function*, which weights the contributions of the inputs. For a simple case, linear interpolation, the basis function can be replaced by a simple first order polynomial,

$$f(x, y) = c_1 x + c_2 y + c_3. \quad (2.21)$$

The coefficients $\mathbf{c} = (c_1, c_2, c_3)$ can then be found by solving $\mathbf{A}\mathbf{c} = \mathbf{f}$ where $\mathbf{f} = (f_1, f_2, f_3)^T$ and \mathbf{A} is a 3×3 matrix of rows with the form $(x_i, y_i, 1)$, where i is the row number.

Higher (and lower) order basis functions can also be used, but require larger numbers of input samples. Zero order triangulation-based interpolation is known as nearest neighbour interpolation. Other commonly used schemes include quadratic and cubic interpolation. This process can also be generalised to higher-dimensions.

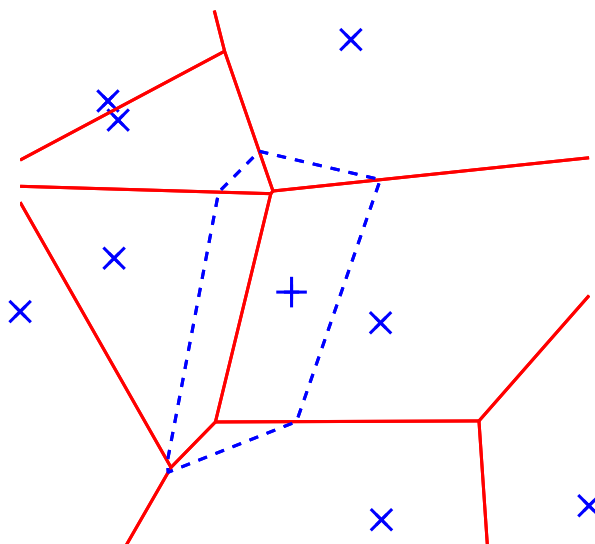


Figure 2.10: Voronoi diagram showing new cell (dotted line) overlapping cells from the original tessellation.

2.4 Natural Neighbour Interpolation

Watson [1985] defines natural neighbours as

“points which share a common interface, or region, that is equally close to each of the pair, and all other neighbours are no closer”

This means that if circles (or n -spheres in n dimensions) are drawn such that their circumferences pass through $n + 1$ or more data points, no data points will be within any of the n -spheres. This is related to the Delaunay triangulation which can be formed by linking the data at points which are on the circumference of each n -sphere. The Delaunay triangulation is not unique when more than $n + 1$ points lie on a sphere edge (i.e. when four or more points are coplanar in 2-D).

Once the natural neighbours have been established, the interpolated output value at any point can be determined using a weighted sum of the values at its neighbours. The way in which the weights are determined is best described in terms of Voronoi tessellations, the geometric dual of the Delaunay triangulation, see Fig. 2.10.

For each point where a value is required:

1. Assume the data are already tessellated
2. Re-tessellate the data to include the output point. This adds a new Voronoi cell which overlaps the cells of the natural neighbours of the output point.

3. The contribution from each neighbour is given by the ratio of the area of overlap to the total area of the new cell. These ratios form the basis function $\phi_i(x, y)$ in (2.20).

In terms of the Delaunay triangulation, the basis function $\phi_i(x, y)$ is only non-zero within the circum-circles which pass through the natural neighbour nodes. This means that the operation is local, in the sense that only neighbouring values are used in the interpolation.

2.5 Radial Basis Function Interpolation

RBF interpolation approximates a field of data using a weighted sum of radially symmetrical functions, known as basis functions [e.g. Carr et al., 1997]. One basis function is centred on each input sample, so that any given output point is composed of contributions from each input point. **RBF** interpolation is therefore considered a global technique. The output at any given point \mathbf{x} is given by

$$f(\mathbf{x}) = p_m(\mathbf{x}) + \sum_{j=1}^N \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (2.22)$$

where p_m is a low-order polynomial surface with coefficients c_0, c_1, \dots, c_n which has been fitted to the data and is only used during linear and thin-plate spline interpolation, ϕ is the basis function whose form is fixed across the field and λ_i is the weight for input \mathbf{x}_i . Many different basis functions can be used, with differing performance and order of continuity. Some common functions are shown in Table 2.2 [e.g. Light, 1992, Powell, 1990].

Table 2.2: Example Radial Basis Functions

Name	Equation
Linear	$\phi(r) = r$
Thin-plate Spline	$\phi(r) = r^2 \log r$
Multiquadratic	$\phi(r) = (r^2 + c^2)^{0.5}$
Inverse Multiquadratic	$\phi(r) = (r^2 + c^2)^{-0.5}$
Gaussian	$\phi(r) = e^{-ar^2}$
Biharmonic Spline	$\phi(r) = r ^2(\log r - 1)$

To find values for the weights λ and coefficients c , the linear system

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad (2.23)$$

must be solved, where A is a matrix composed of evaluated basis function values for every possible pair of input values, P is a matrix of homogeneous input coordinates (coordinates with leading ones),

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix} \quad (2.24)$$

and $[\lambda \ c]^T$ and $f = (f_1, f_2, \dots, f_n)^T$ are column vectors of weights and input values respectively.

When a polynomial is not being fitted, the output system reduces to

$$A\lambda = f. \quad (2.25)$$

The calculation of the matrix A and solving the linear system described by (2.23) are computationally expensive operations and this has motivated work aimed at decreasing the overall complexity, using techniques such as domain decomposition [e.g. [Beatson et al., 2001](#)]. In addition to the basis functions given in Table 2.2, some other interpolation methods can be formulated in terms of RBF interpolation. A well-known example is biharmonic spline interpolation and this is described in more detail below.

2.5.1 Biharmonic Spline Interpolation

A special case of RBF interpolation, which has been used since before RBF interpolation was available, is biharmonic spline interpolation (BSI). This is a method of finding the minimum curvature surface which passes through a set of points [[Sandwell, 1987](#)]. In practice, this method produces results which are indistinguishable to RBF interpolation using thin plate spline (TPS). BSI can be numerically unstable for large numbers of points and, like other cubic methods, has a tendency to drastically overshoot when points are close together. This problem occurs because of the imposed continuity in the surface's derivatives, which makes smoothly varying curves preferable around data points. Therefore, BSI is better suited to the interpolation of highly sparse data. The phenomenon of overshooting is discussed in more detail in Chapter 4.

2.6 Kriging

Kriging was first suggested and developed in the 1960s by D. G. Krige, a South African mining engineer. It was originally developed as a technique for estimating yields of ore deposits from sparsely distributed core samples, but has now been widely applied to many different fields and scenarios (for example, mining, mathematics and classification [Boucher et al. \[e.g. 2006\]](#)), as discussed in [Cressie \[1990\]](#). One of the main attractions of kriging is its ability to provide a variance estimate for each output point. Kriging and all geostatistical methods operate under the assumption that a process being interpolated or analysed consists of a *stochastic part* and an underlying trend [[Matheron, 1973](#)]. The trend may consist of both local and global components. This is Matheron's "Theory of Regionalized Variables" [[Matheron, 1971](#)]. The stochastic component is comprised of both random and autocorrelated parts, where the degree of autocorrelation is a function of distance. This means that points in close proximity are more closely correlated than distant ones.

The first step in interpolation using kriging is the formation of a semivariogram [e.g. [Omre, 1984](#), [Cressie, 1991](#)]. This is a diagram of the spatial dependence of samples and is a function of all possible separations (or lags) and semivariance. The semivariance is defined by

$$\gamma(\mathbf{h}) = 0.5(f(\mathbf{x}) - f(\mathbf{x} + \mathbf{h}))^2, \quad (2.26)$$

where $f(\mathbf{x})$ contains the point values at a given location (\mathbf{x}) and $f(\mathbf{x} + \mathbf{h})$ is the point value at a point separated from \mathbf{x} by the lag vector \mathbf{h} . For the isotropic two-dimensional case, it is simplest to calculate lags by using a distance metric between points. However, the number of combinations of sample positions which must be compared quickly becomes very large. For this reason, input coordinates are often binned to reduce the total number of lags.

After the semivariogram has been created, a model must be fitted to it. One example, the spherical semivariogram model is a curve of the form

$$\gamma_{\text{sph}} = \begin{cases} c \left(1.5 \left(\frac{h}{a} \right) - 0.5 \left(\frac{h}{a} \right)^3 \right) & \text{for } 0 \leq h \leq a \\ c & \text{for } h > a, \end{cases} \quad (2.27)$$

where, in geostatistical parlance, c is the 'sill', the value that the semivariogram reaches after its initial rise, and a is the 'range' or length of spatial dependency. This sill is generally close to the variance of the input values. Details of this and other commonly used models can be found in [Cressie \[1991\]](#) and [Trauth \[2006\]](#). [Fig. 2.11a](#) shows a typical semivariogram with a fitted spherical model.

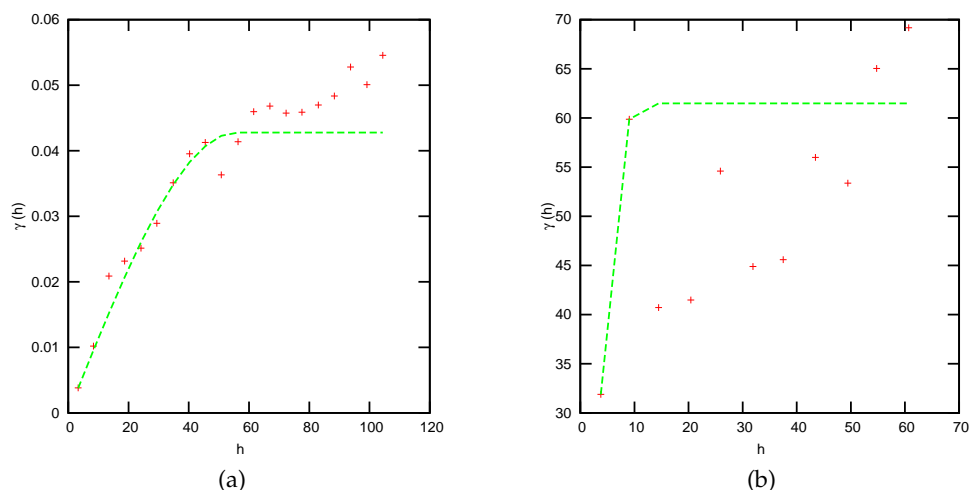


Figure 2.11: Semivariograms (+) and automatically fitted spherical model of (2.27) (dashed lines). (a) Simulated correlated data demonstrating a good fit and (b) degenerate TEC data showing a poor fit.

The next, and final, step is the actual kriging process. Kriging uses a weighted average of input points to estimate the value any given output point. The weights are found by minimising the *kriging variance* – the difference between the estimate and the actual input value. An output variance is also directly calculable. As it minimises the variance of the output, kriging is often called the “best linear unbiased estimator”. However, the variance which is minimised is relative to the semivariogram model, so the results will only ever be as good as the model and, therefore, the semivariogram. For a good, concise description of the kriging process, see [Blanch et al. \[2002\]](#). It should also be noted that other authors have heavily criticised some of the underlying assumptions behind geostatistics [e.g. [Philip and Watson, 1986](#)] and these issues should be borne in mind when using kriging and related techniques.

Chapter 3 examines the performance of the algorithms described in this chapter, using both simulated and real data, and quantitative techniques known as simulation- and cross-validation.

Evaluating Interpolation Performance

This chapter is based on material from [Foster and Evans \[2008a\]](#), and details methods for analysing the performance of interpolation methods using simulated and real data (in the form of ionospheric total electron content ([TEC](#)) measurements).

3.1 Introduction

The distribution of electrons in the ionosphere is of interest to scientists and also to engineers working on applications such as Earth-space communication systems, which must transmit through the ionosphere, and skywave systems which make use of ionospheric refraction. Electron content is commonly examined using [TEC](#) mapping. This mapping finds use in other applications, such as studying the evolution of magnetic storms which have, in the past, had profound effects on satellite communication systems and on other critical ground based systems, such as the US power grid. Information on the electron content of the ionosphere can be collected using the global positioning system ([GPS](#)), by examining the phase and amplitude changes which occur in paths between transmitting satellites and ground based receivers. These data can then be processed in order to create maps of the ionospheric [TEC](#).

As the number of paths between [GPS](#) ground stations and satellites is relatively low, producing [TEC](#) maps is an exercise in reconstruction from sparse data. Recent research has mainly focused on methods such as tomography that provide time-dependent volu-

metric reconstructions [Mitchell and Spencer, 2003, Pallares et al., 2005]. However, when the data points are too sparsely distributed these techniques are under-constrained and do not produce meaningful results. In ionospheric studies, problems relating to sparsity are especially prevalent in historic data-sets. For example, in 1992 there were only 25 receiver sites operated by the International GNSS Service (IGS) in the USA [Brockmann and Gurtner, 1996], by 1996 there were over 75, and now there are over 500. Therefore, whilst the issues due to undersampling have largely disappeared for TEC imaging systems utilising modern GPS data, they still remain for older data, and regularly arise in other geoscience applications [Liao et al., July 2007, Gianinetta and Villa, Oct. 2007]. Consequently, interpolation methods still have an important role to play in ionospheric studies. The most commonly used interpolation technique for TEC mapping studies is kriging [Blanch et al., 2002, Stanislawska et al., 2002, Wielgosz et al., 2003]. Although these studies have generally found kriging to perform satisfactorily, in other geophysical applications numerous problems with the kriging method have been reported [Philip and Watson, 1986]. In addition, there many other interpolation methods for geophysical data that have received little recent attention from the ionospheric imaging community.

The overall aim of this chapter is to assess the performance of currently available multi-variate interpolation techniques for ionospheric TEC mapping. The need to establish the relative performance of scattered data interpolation schemes has been recognised and partly addressed in the past, see for example Franke [1982]. However, only very specific case studies exist involving more up to date methods [Blanch et al., 2002, Mahdian et al., 2001, Rauth, 1998]. This study considers, for the first time, the specific application of such techniques to TEC mapping and presents the results of a comprehensive quantitative evaluation, using both simulated data and real ionospheric electron content measurements. Schemes evaluated include those previously used for TEC mapping (e.g. kriging), interpolation methods commonly used in other fields (such as interpolation based on Voronoi tessellations and radial basis functions), and schemes in use for other geophysical applications (such as natural neighbour interpolation). We further propose the application of adaptive normalised convolution (ANC) to the problem of TEC mapping and quantify its performance in comparison with extant techniques. ANC is a recently proposed interpolation scheme that has found application to the reconstruction of data with varying spatial frequency content, orientation and anisotropy. As these properties are also found in TEC images, their reconstruction using ANC appears an attractive proposition. The proposed objective evaluation scheme enables the benefits conferred by ANC to be quantitatively assessed.

The quantitative evaluation methodology is outlined in Section 3.3 and used to determine the performance of the interpolation schemes described in sections 3.2 and 2.2. Finally, discussion and conclusions are presented in Section 3.4.

3.2 Interpolation Schemes for Scattered Data

Scattered data interpolation has been studied for many years and in many fields – because of this, it has been given many names. The term scattered, for example, is also referred to as “spatial”, or “multivariate”, and the term interpolation is often called “reconstruction” or, less formally, “approximation”. An interesting history of interpolation from ancient times is provided by Meijering [2002]. Although its fundamental concepts do not differ, multivariate interpolation is a more recent development. In their study of the mathematical development of multivariate interpolation up to the second half of the 20th century, Gasca and Sauer cite the first modern literature on multivariate interpolation as the work of Borchardt and Kronecker, that appeared in 1860 and 1865 respectively [Gasca and Sauer, 2000].

Sparsity is a term that is often associated with scattered data, but which can only be defined relative to the desired resolution of reconstruction. Empirically, if there are not enough data points to fully reconstruct every point in the field at all desired spatial frequencies, then the data are undersampled or sparse. Altering the resolution of the reconstruction changes the relative sparsity by changing the number of spatial frequencies which are harmonics of the field size. Although no single definition of sparsity exists, in this paper we consider data to be sparse if values exist at fewer than 5% of the discrete elements present in the reconstructed field.

Interpolation methods can be divided into two categories, *local* and *global*, depending upon the locality of the points which are used to derive a given output point. Local techniques make use of a definition of *locality* to compute output values; only data which fall within a given point’s local neighbourhood are used to calculate output values. Global techniques use a weighted sum of *all* data to compute output values and for large numbers of input points an approximation is generally used. When a new datum is added to a globally interpolated field the whole field must be re-calculated whereas, for a locally interpolated field, only those positions within the neighbourhood of the added datum need to be re-calculated. These two points tend to favour the use of local techniques.

The interpolation schemes evaluated in this study represent a broad cross-section of those in common use. Specifically they are:

- Triangulation based (nearest neighbour [Sugihara et al., 2000], linear [Watson and Philip, 1984] and cubic [Watson, 1992]), section 2.3;
- Natural neighbour [Sibson, 1981], section 2.4;

- Radial basis function [Carr et al., 1997, Light, 1992, Powell, 1990], section 2.5;
- Biharmonic spline [Sandwell, 1987], section 2.5.1;
- Ordinary kriging [Cressie, 1991, Trauth, 2006], section 2.6.

Of the list above, only ordinary kriging, radial basis function (RBF) interpolation and biharmonic spline interpolation (BSI) are considered truly global techniques. Natural neighbour, nearest neighbour and triangulation based interpolation all use a neighbourhood defined by the Delaunay triangulation of the input data coordinates. Chapter 2 introduced the interpolation methods used in this study.

3.3 Quantitative Evaluation & Experimental Results

The interpolation methods tested and examined in this study are a mixture of commonly available implementations and custom written code (see chapter 2). From the triangulation-based class of techniques, the nearest neighbour, linear and cubic interpolation available in MATLAB 2007a's `griddata` function were selected [The Mathworks, Inc, 2007]. The natural neighbour interpolation code was that available from Sakov [2005]. The RBF interpolation of Carr et al. [1997] was implemented using both linear and multiquadratic bases. No domain decomposition was used due to its complexity and the fact that the fields being interpolated were relatively small. The biharmonic spline algorithm used was the v4 algorithm in MATLAB's `griddata`, which uses the algorithm of Sandwell [1987]. The kriging method used is known as *ordinary kriging* and works with isotropic, normally distributed data. The implementation evaluated was based on code given in Trauth [2006], with some modifications. In particular, a spherical model was chosen as it represents a good trade-off between the complexity associated with models with a high degree of freedom and the poor performance exhibited by simpler functions such as the linear model. In tests the spherical model was found to perform well with both simulated and TEC data. To enable the unsupervised reconstruction of TEC fields the spherical model was automatically fitted to the semivariogram using a least-squares method. The ANC interpolation technique implemented was the zero-order scheme of Pham [2006]. The kernel used was a two-dimensional Gaussian, whose size and orientation were set using (2.8)–(2.10). gradient square tensors (GST)s were constructed using gradients obtained from normalised differential convolution (NDC). To reduce the complexity, the efficient decomposition technique that provides a close approximation for rotated Gaussians proposed by Geusebroek et al. [2002] was used in the final filtering stage. The Euclidean distance was used for all techniques requiring a distance metric.

The techniques to be evaluated were applied to both simulated and real **TEC** data. As simulated data provides ground truth values, it has the advantage of allowing analysis of residual errors to be calculated at every point in the output field. In addition, parameters such as the of the input sparsity can be carefully controlled. It should be noted, however, that the performance of any interpolation method can vary considerably with the statistics of the input data, and therefore the results gained through simulation are not necessarily indicative of the general performance. Therefore, the ultimate test of reconstruction techniques remains their application to real data. To this end, the interpolation methods are applied to **TEC** data from the much studied October 2003 ionospheric storm.

3.3.1 Simulated Data Results

Following the methodology detailed by Omre [1984], two kinds of simulate test data were generated. The first type were produced by generating fields of normally-distributed random data, which were then filtered using a pillbox. The filtering process introduces autocorrelation with a lag distance dependent on the filter radius. This data has a multivariate Gaussian distribution which is considered ideal for ordinary kriging. The second method generates univariate data with an approximately log-normal distribution by filtering fields of uniformly distributed random data. The multi-normality is then removed by examining 5×5 neighbourhoods, around each point, and randomly selecting from the ten highest values. Finally, the natural logarithm of the each data point is calculated. Histograms illustrating typical distributions generated by these two methods can be seen in Fig. 3.1.

The interpolation techniques were used to reconstruct each type of the simulated data from sparsities ranging from 95% to 99% in steps of 1%, and 99% to 99.9% in steps of 0.2%. The sampling was carried out by thresholding uniform pseudo-random numbers, so the percentage of remaining samples is not necessarily the same as the requested value. Plotted results show the actual sparsity obtained. Each generated data field was sampled 30 times at each sparsity, and then reconstructed with each interpolation method. The number of reconstructions was set to 30 to minimise computation time, whilst ensuring the statistical significance of the results.

The **RMSE** between the original and reconstructed data outputs were calculated and averaged over the 30 reconstructions, see Figs. 3.2 and 3.3. In both of these figures, the **RMSE** values were normalised by dividing by the average value of the data being interpolated to give the **RMSE** as a proportion in which, for example, a value of

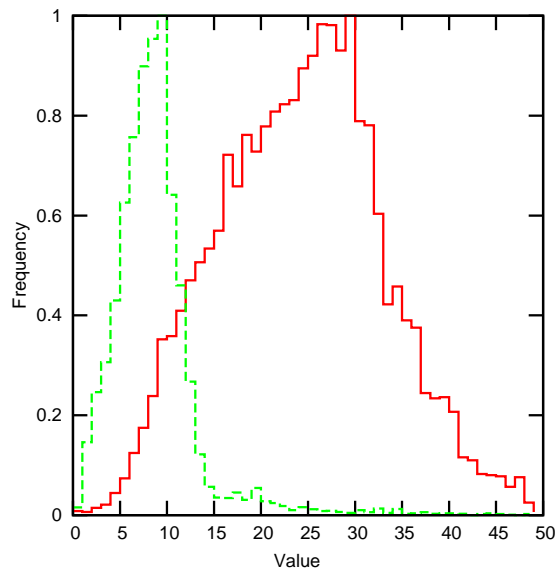


Figure 3.1: Normalised histograms of the two types of simulated correlated data described in section 3.3.1. Multivariate (solid) and univariate (dashed).

0.1 corresponds to a 10% error. For clarity, the results for the two worst performing techniques, linear and nearest neighbour interpolation, were removed.

The **RMSE** performance of all interpolation techniques increases with sparsity for both types of simulated data. Overall, the **RMSE** for the multivariate data increases from around 10% at a sparsity of 95% to 25-30% at a sparsity of 99.6%. The performance for the univariate data at the corresponding sparsities is better, increasing from approximately 7% to 15% at sparsities of 95% and 99.6% respectively.

BSI is the worst performing technique for both the univariate and multivariate simulated data, with a consistently higher **RMSE** than other schemes. Although kriging is the best performer for many sparsities, its error is dramatically increased at sparsities $> 99.3\%$ and at certain other lower sparsities, probably as a result of failing to correctly fit to the semivariogram. This is significant as the errors in these cases are up to 4 times those of the other techniques. The **ANC** performance at all sparsities is 1-2% worse than the best performing techniques. Cubic interpolation generally performs well and the overall best performer is natural neighbour interpolation, which exhibits no anomalous behaviour, whilst maintaining good performance throughout.

This process described above was repeated whilst altering the size of the pillbox used to impose auto-correlation of the simulated data from 10 to 50 in steps of 10. Overall, this has little effect on the **RMSE** of the reconstructions, with the exception of kriging, whose implicit assumptions about data auto-correlation are violated when the lag distance is small. In both the univariate and multivariate cases natural neighbour interpolation performed best with respect to changing radius of correlation. This is because

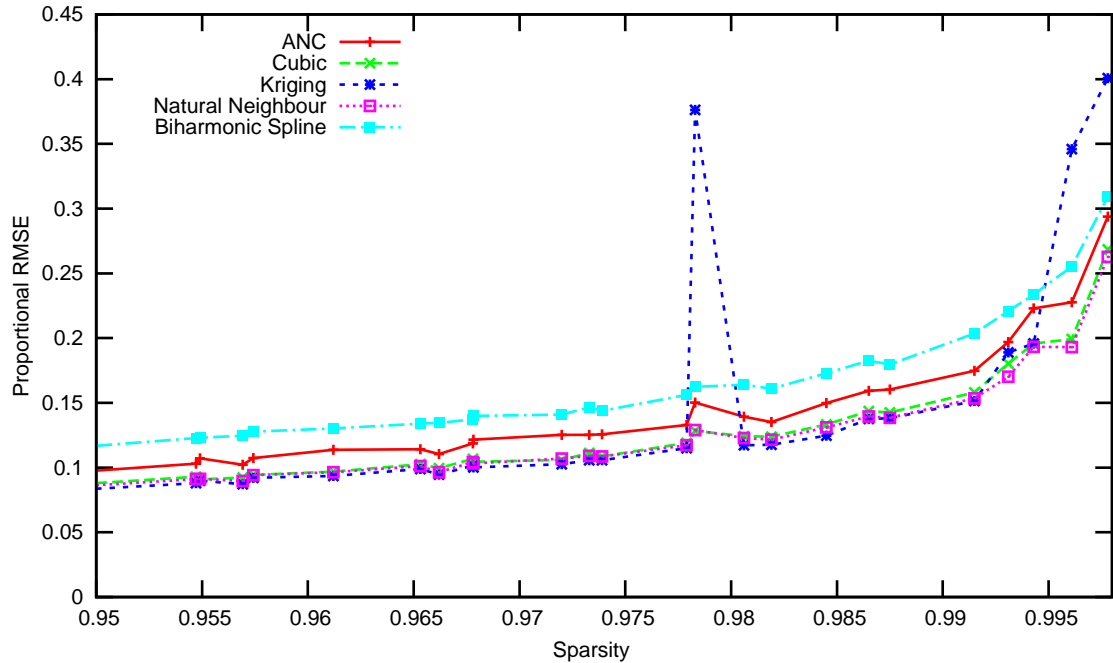


Figure 3.2: Proportional **RMSE** as a function of sparsity for simulated multivariate correlated data reconstruction.

its performance is based on data position rather than value.

In addition to providing overall error values, simulated data allows for analysis of residual errors at every point in the field. In all cases the residual errors exhibited Gaussian distributions with means very close to zero, showing the interpolation techniques have negligible bias.

3.3.2 TEC Data Results

The **TEC** is defined as the line integral of the electron content over a path between two points, usually a satellite and a receiver. Various methods have been developed for extracting **TEC** information from the amplitude and phase of **GPS** signals, e.g. [Warnant and Pottiaux \[2000\]](#), [Arikan et al. \[2007\]](#). The data used in this study were processed and calibrated using the MIDAS tomographic inversion software from the University of Bath [Mitchell and Spencer \[2003\]](#). MIDAS calculates the **TEC** biases by analysing the differences in length between the measured and inverted paths [Meggs and Mitchell \[2006\]](#). Whilst other methods for removing biases are available [Mannucci et al. \[1999\]](#), [Center for Orbit Determination in Europe, Astronomisches Institut Universität Bern \[2005\]](#), the aim here was to provide representative **TEC** data for the evaluation of the interpolation techniques.

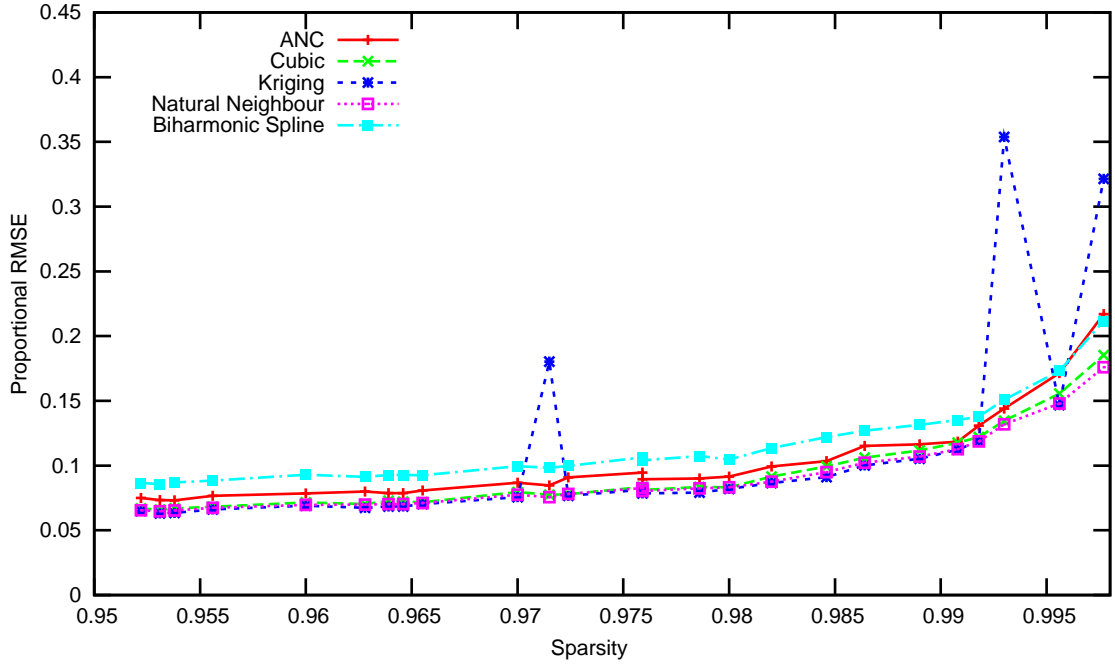


Figure 3.3: Proportional **RMSE** as a function of sparsity for simulated univariate correlated data reconstruction.

The sources of data used to test **TEC** reconstruction are approximately 80 **GPS** measuring stations lying within 20–70° N, and 70–130° W. This corresponds to a coverage of most of North America. Whilst more sites were available at that time, not all sites were used, as the main aim of this chapter is to examine interpolation during high sparsity cases. The time period over which data were drawn was from noon to midnight on October 30th, 2003 – the peak of the much studied “Halloween Storm” [Hernandez-Pajares et al., 2005]. Data were considered stationary within 15 minute intervals, and projected onto a ‘thin shell’ for reconstruction [Mannucci et al., 1999]. The thin shell used covered the same area as the ground stations and had latitudinal and longitudinal resolutions of 0.5°, giving rise to fields of size 101 × 121 pixels. As each ground-based receiver station can see approximately 6 satellites at any one time, there are around 500 paths associated with the 80 measuring stations. Each path’s **TEC** values were projected onto an infinitesimal shell at a fixed height by calculating the ray’s intersection with the shell. With reference to Fig. 3.4, the function which maps from slant to vertical **TEC** is then given by

$$F(z) = \left(1 - \frac{R_e \cos(90 - z)}{R_e + H}\right)^{-0.5} \quad (3.1)$$

where z is the satellite elevation angle (in degrees), H is the height of the shell (400 km in this case) and R_e is the radius of the Earth. Paths with elevation angles $< 20^\circ$ were removed because of high levels of error associated with low angles [Mannucci et al., 1999]. The projected **TEC** values at 3 hourly intervals over the 12 hour storm period

3.3. QUANTITATIVE EVALUATION & EXPERIMENTAL RESULTS

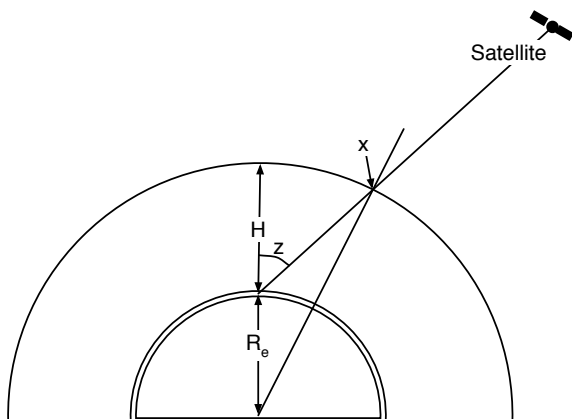


Figure 3.4: Thin Shell Ionosphere Model. R_e is the radius of the Earth, z is the elevation angle from the ground station to the satellite, x is the point at which the path between the satellite and ground station intersects the shell, and H is the height of the shell.

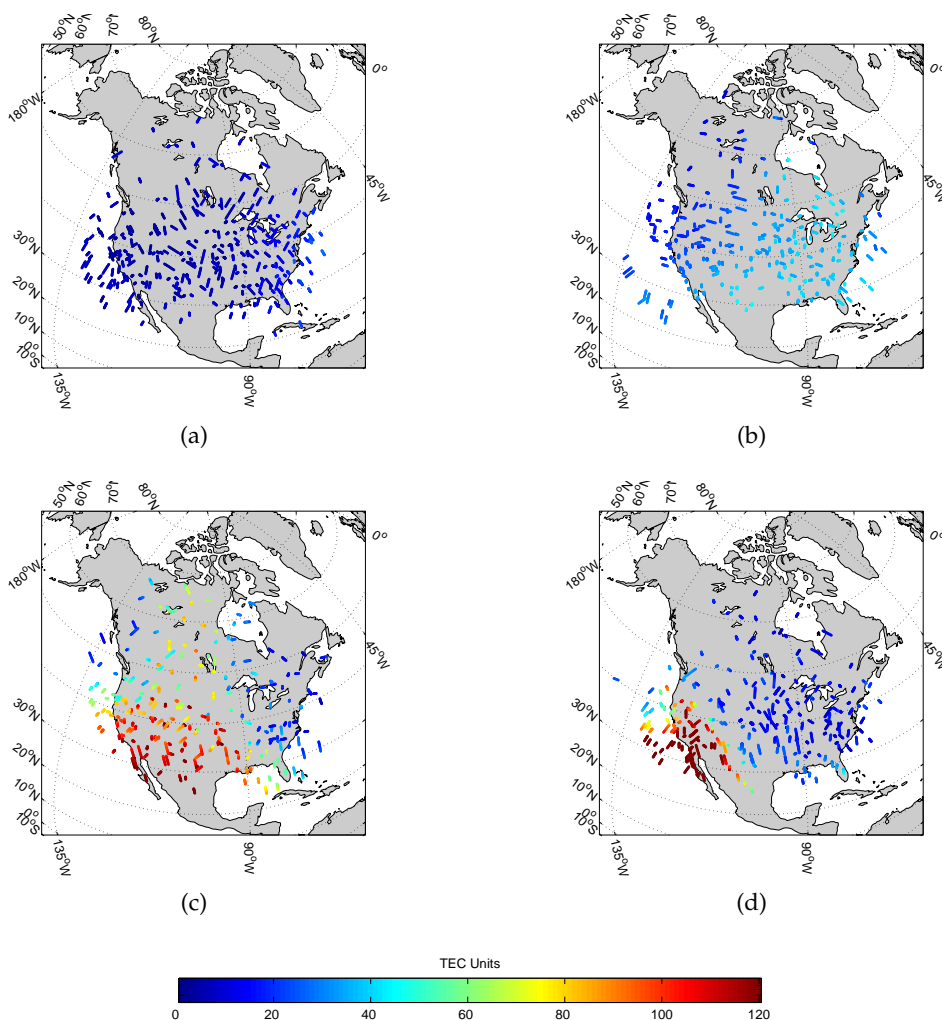


Figure 3.5: Example **TEC** input data for the cross-validation test in Section 3.3.2. All data are from October 30th, 2003 for the time periods (a) 1200-1215 UT, (b) 1600-1615 UT, (c) 2000-2015 UT and (d) 0000-0015 UT on the 31st.

are shown in Fig. 3.5.

As there is no ground truth data against which interpolated TEC fields can be evaluated accurate testing is more difficult. Previous approaches have used models as a basis for comparison (e.g Samardjiev et al. [1993], used the CCIR model to compare f_0F_2 results¹). However, in many cases modelled data are far smoother than the actual phenomena being modelled, which leads to anomalous results. In particular, results tend to be biased towards favouring techniques which produce artificially smooth outputs. Alternatively, partitioning the data-set into two classes, the testing and reconstruction data, allows the testing of output against real data which that were not used in the reconstruction. This technique is known as cross-validation and is often used for testing the accuracy of classification systems where no ground truth data are available [Kohavi, 1995].

The specific method used was k -fold cross-validation in which the list of all ray paths is randomised and then partitioned into several blocks. Values corresponding to the first block are interpolated using the schemes under test, and the values which belong to the unused blocks are used to compute differences with the output value at corresponding positions. These can then be used to calculate the RMSE and other difference metrics. Both the first and second blocks are then used for the reconstruction, and the errors for ray paths in the remaining blocks found. The process is repeated until only one block remains. This approach has the added advantage of being able to produce input data fields with varying degrees of sparsity. To ensure the significance of the results, validation results should only be used where a reasonable number of validation positions are available. In this study, only cases with upwards of 30 positions were used. Care was also taken to ensure that the average TEC values of the points used for validation were similar in magnitude to those being used for the reconstructions, to avoid biasing the output values.

Fig. 3.6 shows the electron content results binned into 40 sections, and averaged across each section for the interpolation techniques evaluated in the previous section. As before, the RMSE have been divided by the average field value to produce proportional RMSE results that are more directly comparable with the simulated results. The lower sparsity limit of 0.9825% is higher than for the simulated data as this is the greatest density that can be achieved with the available data points. Compared to the simulated data, the increase in error with sparsity is less marked for all techniques.

Once again, BSI was the overall worst performing technique. The inconsistency of the errors across the range of sparsities associated with kriging that was exhibited in the simulated results is also evident. For example, at a sparsity of 99.3% its RMSE is over 4

¹The CCIR model is now part of the International Reference Ionosphere (IRI) model.

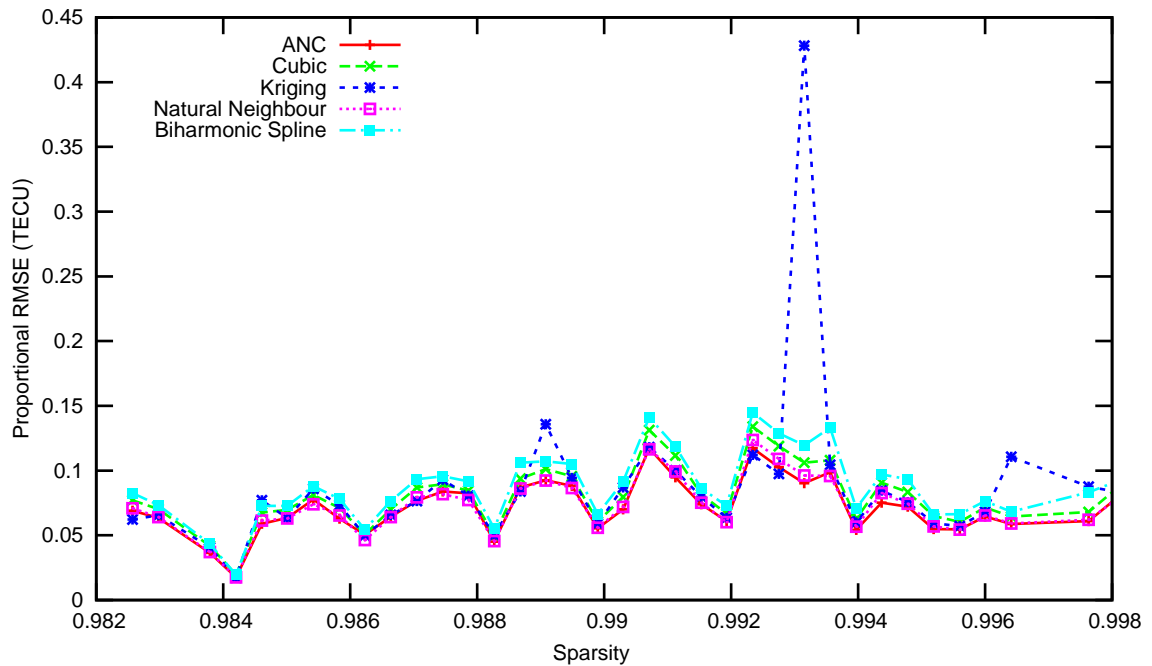


Figure 3.6: Proportional **RMSE** as a function of sparsity for **TEC** data from **GPS** path measurements. Errors were calculated using the cross-validation method described in the text (section 3.3.2).

times that of the best performing technique. Cubic interpolation performed consistently with approximately average results.

Natural neighbour interpolation again performed well but, unlike the results for simulated data, its performance is matched by **ANC**'s. In fact, over the range of sparsities the proportional **RMSE** produced by **ANC** is, on average, 0.08% less than the equivalent natural neighbour results, and also showed a slightly lower variance. In comparison, the average **RMSE** produced by cubic and kriging interpolation were 0.91% and 1.57% worse than **ANC**, respectively. The inconsistency of kriging is reflected by a variance approximately 10 times higher than the other techniques.

To illustrate the results produced by some of the different interpolation techniques, Fig. 3.7 presents example reconstructions produced by **ANC** and kriging for two sets of input data from Fig. 3.5c. The input data in each case was 25% of the available **GPS** path signals, giving a sparsity of approximately 98.8%. For both sets of input data, kriging and **ANC** have produced slightly different results. The proportional **RMSE** for each reconstruction can be found using the remaining 75% of the data. For this case the average proportional **RMSE** is 0.0524 **TEC** Units and the difference between the kriging and **ANC** errors is less than 1%. Fig. 3.7 also shows that the set of input data used produces more significant differences in the output fields than the choice of interpolation method. This observation underlies the benefit of the cross-validation evaluation procedure which removes any sensitivity to choice of input data by the

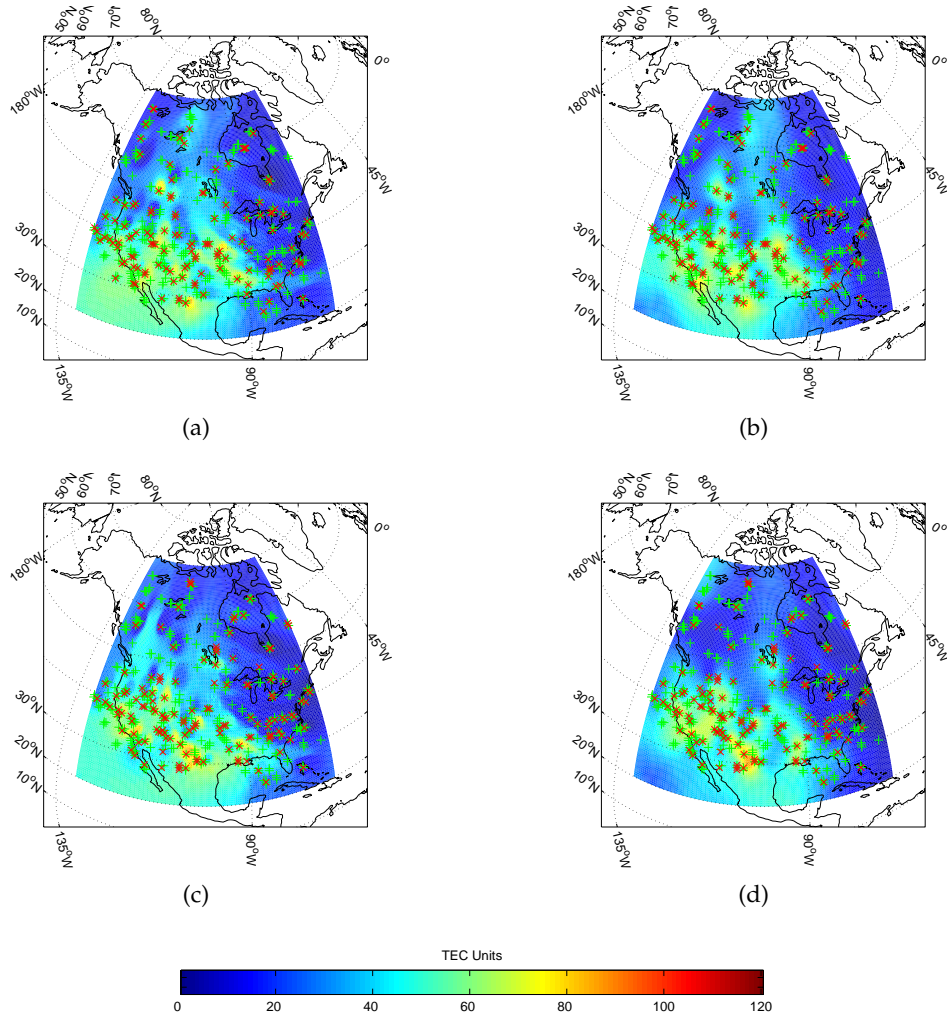


Figure 3.7: Example output images produced by **ANC** and kriging using two sets of the input data from Fig. 3.5c. Each set consisted of 25% of the available data (\times) and the remaining data ($+$) used to calculate the proportional **RMSE**. (a) and (b) Set 1 results (sparsity 98.94%) produced by **ANC** and kriging respectively. (c) and (d) Set 2 results (sparsity 98.79%) produced by **ANC** and kriging respectively. The proportional **RMSE** values are (a) 0.0640, (b) 0.0533, (c) 0.0503, and (d) 0.0421 **TEC Units**.

averaging the results within a given range of sparsities.

3.4 Discussion and Conclusions

In the literature, kriging has been the interpolation method of choice for producing **TEC** maps of the ionosphere. However, to date there has been little in the way of evidence to support its adoption over other interpolation schemes. This chapter has sought to address this issue by performing a comprehensive quantitative evaluation of kriging and a selection of other interpolation methods currently in use. To this end, an evaluation methodology that uses both simulated and **TEC** data has been proposed.

3.4. DISCUSSION AND CONCLUSIONS

With simulated data, error values can be calculated at all output points. For **TEC** reconstructions this is not possible and, instead, an evaluation using cross-validation was performed. Considering the overall performance for both simulated and **TEC** data, the following remarks about the individual interpolation techniques can be made.

Triangulation-based techniques are widely used in computer graphics applications. The best performing of these, cubic interpolation, has a relatively low complexity and its performance is, in many cases reasonably close to that of the best performing, more complex methods. Therefore, for ionospheric applications where a small loss in accuracy can be sacrificed for a faster run-time, it is a reasonable choice of technique.

Although the kriging scheme used in this investigation performs well at many sparsities, it exhibits a very large variance for both the simulated and **TEC** data. This variance is due to spikes where the proportional **RMSE** is excessively high. Two main stages of kriging-based interpolation are the construction of the semivariogram and the fitting of a suitable model, and both of these are sensitive to the settings of their various parameters. This is one of the main reasons why it is often recommended that kriging is implemented as an interactive process, as opposed to an automatic one. When the semivariogram model being used fails to accurately fit the experimental semivariogram, the output of the interpolation will be poor. Fig. 2.11b shows an example, degenerate semivariogram where the data have a high variance at all lags, a breach of the fundamental assumption of high autocorrelation at low lags. In these cases the fitted model poorly matches the actual data, resulting in an output field with a high error. Although, in theory, it may be possible to automatically detect degenerate semivariograms and try to find a more suitable one to fit to, the procedure is very complex and, for the tests performed here, the level of sophistication required would be far beyond that required by the other techniques being evaluated.

Natural neighbour interpolation performs well across all data types and sparsities. It is the best performing method for both types of simulated data and is only surpassed by **ANC** on the **TEC** data. The main drawback of natural neighbour interpolation is that it is complicated to implement, and there are few modern reference implementations available. However, if it were more widely known, and its performance recognised, this situation could change.

While there are many performance features that are common for both the simulated and **TEC** reconstructions there are also some significant differences, such as the change in **RMSE** with sparsity, and the relative performance of individual interpolation techniques. This suggests that in testing interpolation methods simulations should only be used if they are demonstrably very similar to the real data to be interpolated. If this is not the case, the data-driven cross-validation methodology demonstrated here is ideal

3.4. DISCUSSION AND CONCLUSIONS

for testing the performance of interpolation schemes using only real data. The major difference between the simulated and **TEC** data is that of anisotropy and this helps explain why the relative performance of **ANC**, a technique that copes well with anisotropy, was dramatically improved for the **TEC** reconstruction. Indeed, given that **ANC** was the best performing technique for the **TEC** data, these results suggest that **ANC** and natural neighbour interpolation should be the methods of choice for ionospheric reconstructions as they offer an error performance that is better and more consistent than kriging.

Interpolation Artefacts and Error Distributions

4.1 Artefacts

Artefacts are a very interesting and important aspect of interpolation. They can differ greatly across different methods, and appear in widely varying situations that cause them to appear. This section examines different methods with a view to identifying different artefacts, and where they appear.

Knowledge of when, where and how different artefacts occur, as well as which techniques produce them can be very useful in a wide variety of situations, such as when using sensitive analysis methods, or when closely examining interpolated outputs.

From an alternative angle, knowledge of different artefacts can aid in gaining an intuitive understanding of how different interpolation methods work, and therefore, should help prospective users to choose the technique most suited to their data-sets, and applications.

In this case, the word *artefact* can refer to any characteristic of an interpolated output field which has been introduced by the interpolation technique used to create it. Examples of artefacts commonly seen are peaks, concave slopes and overshooting edges. This definition is deliberately loose, as interpolation is arguably the introduction of artefacts into a sparse set of data points. There is an infinite space of available outputs, all of which are interpolations of the input data, the vast majority of which are completely

inappropriate. Of the remaining (tiny fraction) of outputs deemed acceptable only some will be appropriate in any given case. For this reason, any specific notable feature of any interpolation method can be considered an *artefact*, and so the following discussion will be kept fairly broad in scope.

This section examines artefacts produced by various different interpolation methods, as well as the situations in which they occur, and attempts to explain the reasons behind their different properties.

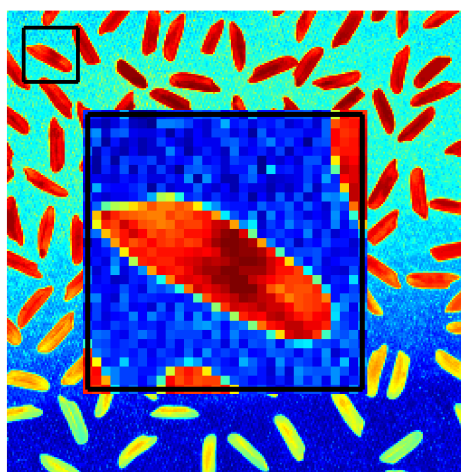


Figure 4.1: A greyscale image of rice grains, displayed using false colour, with the single grain used in Fig. 4.2.

Fig. 4.1 shows an image of rice grains, and illustrates the section which was sampled and interpolated to produce Fig. 4.2. In this figure, various interpolated versions of (a) (which was first randomly sampled to $\approx 99\%$) are shown. Whilst there are no *major* differences between the surfaces, they do illustrate the subtle differences between the different methods.

4.1.1 Triangulation Based Linear Interpolation

Linear triangulation based interpolation operates by fitting flat surfaces to Delaunay triangulated x - and y -coordinates, with heights defined by the z -coordinates. For this reason, data interpolated by this kind of interpolation can appear to be highly faceted, especially when interpolating particularly sparse data.

Linear interpolation based on triangulation is analogous to fixing triangular plates to-

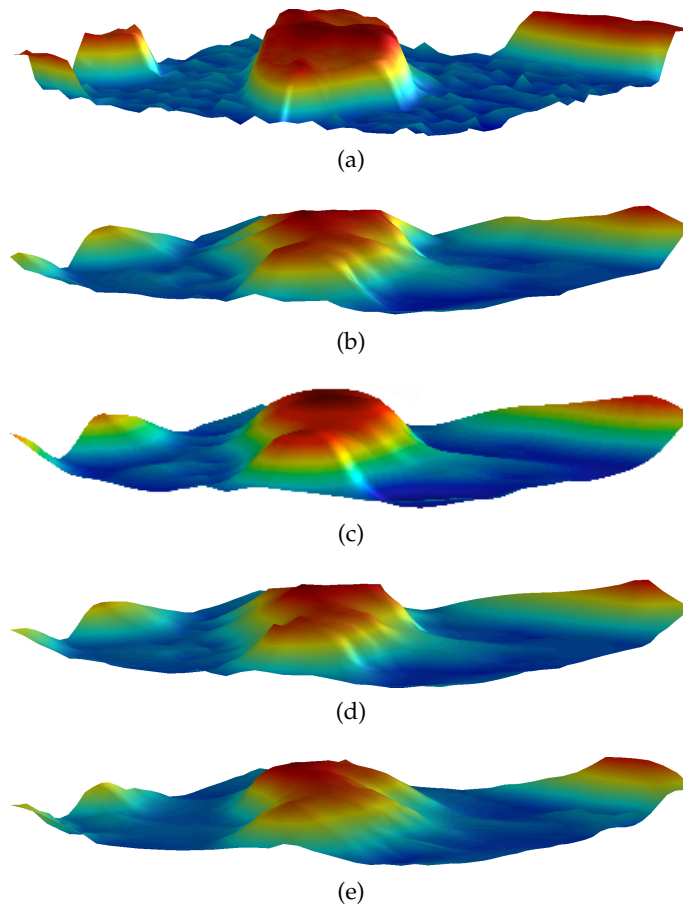


Figure 4.2: Examples of a section of an image of rice grains. The original image, displayed as a displacement map (a) was sampled to a sparsity of $\approx 99\%$, and is followed by linear (b), RBF TPS (c), NN (d) and ANC (e) interpolated versions of the sampled image.

gether over a frame whose vertices are the data points, and with edges following those of the triangles. There is not necessarily any continuity across any derivatives of the edges, and therefore the surfaces can appear highly jagged. However, because no continuity of derivatives is enforced all points in the interpolated output will lie within the surfaces defined by the triangulation. This means that linear interpolation of this kind never contains overshoots, which can be an advantage in some situations, and absolutely essential in others.

4.1.2 Linear RBF Interpolation

Linear RBF interpolation uses a basis function which varies linearly with the distance input datum. This yields results which are similar to those produced by linear triangulation-based interpolation, but which do not contain the faceted appearance. The trade-off here is that RBF interpolation is a global interpolation method, making it computationally expensive when compared to triangulation based methods. For small data-sets where overshoots are undesirable, this interpolation method is ideal.

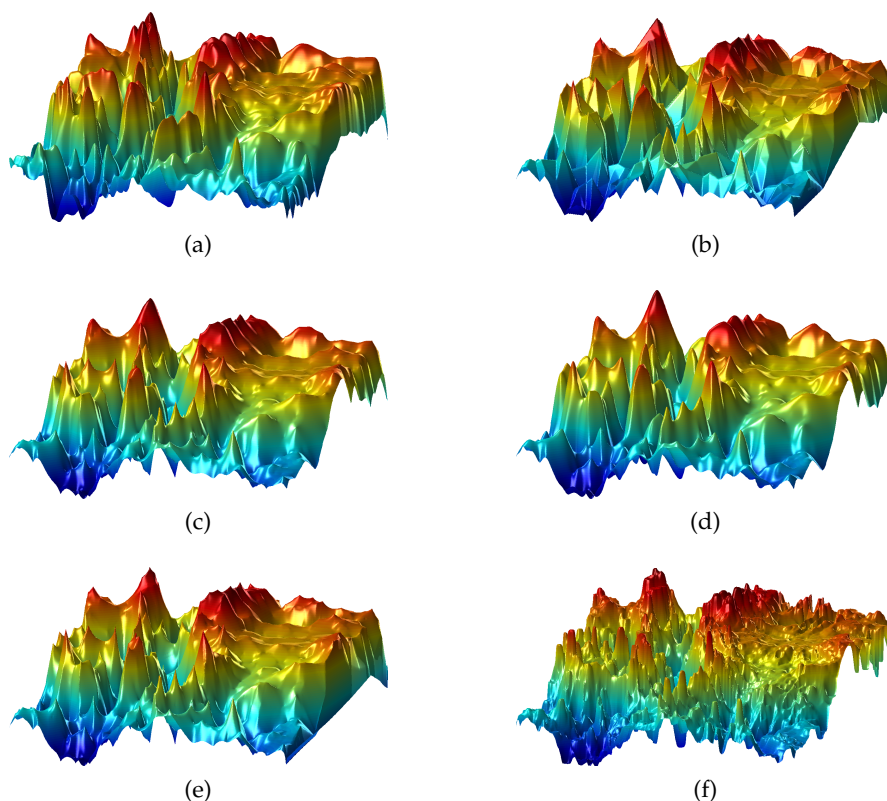


Figure 4.3: Example elevation data from the SRTM, shown as a false colour surface. (a) is the original input data, which was sampled to $\approx 99\%$ sparsity, and reconstructed by (b) was interpolated using linear triangulation based interpolation and (c) was interpolated using linear RBF interpolation., (d) TPS RBF interpolation, (e) NN interpolation, (f) ANC.

4.1.3 Cubic Interpolation

Cubic polynomial surfaces are commonly used in interpolation, as they produce smooth surfaces, and are less computationally expensive to compute whilst requiring less input datum than other higher order surfaces. These factors mean that cubic interpolation methods are among the most commonly used of all interpolation methods, and are the default in many software packages, making knowing about their potential artefacts very important.

Figs. 4.2c and 4.3d were interpolated using TPS RBF. This produces results which are almost identical to biharmonic spline interpolation (BSI), and is characterised by smooth, isotropic surfaces. TPS is a kind of cubic spline, and as such produces similar outputs to triangulation based cubic interpolation. These outputs are all characterised by smooth surfaces, and a tendency to produce values which overshoot data points. This is because cubic interpolation methods attempt to maintain continuous first and second derivatives at all points. Fig. 4.4 shows a minor example of the kind of overshoot and undershoot that can be caused by cubic interpolation. The area in the top left shows a small and steep peak, caused by the two close points after the leading low valued

point. On the right, a similar undershoot with a larger horizontal extent appears for the same reason.

Overshoot is a well documented and analysed problem, [Fried and Zietz \[1973\]](#), [Maeland \[1988, see e.g.\]](#), although despite the large volume of work devoted to providing interpolation methods which are free from such problems cubic methods are still very widely used.

A good physical analogy to spline based interpolation is imagining the image having been interpolated using metal sheets, which are able to bend a certain amount, and whose joins with other plates must be continuous to at least the first derivative.

Other popular interpolation methods such as the venerable Akima method [Akima \[1978\]](#), [Ripley \[2004\]](#), aim to reduce these overshoots and succeed in suppressing them, however algorithms such as this are no longer commonly used in modern software and tend only to be used in applications where legacy code is heavily relied upon, such as in the interpolation of precipitation information. [Chen et al. \[2008\]](#), for example compares three commonly used interpolation methods for interpolation global rainfall. The methods tested include [Shepard \[1968\]](#), which is based on a modified inverse distance weighting function, designed when computational methods for irregular interpolation was first being heavily investigated, in the late 1960s.

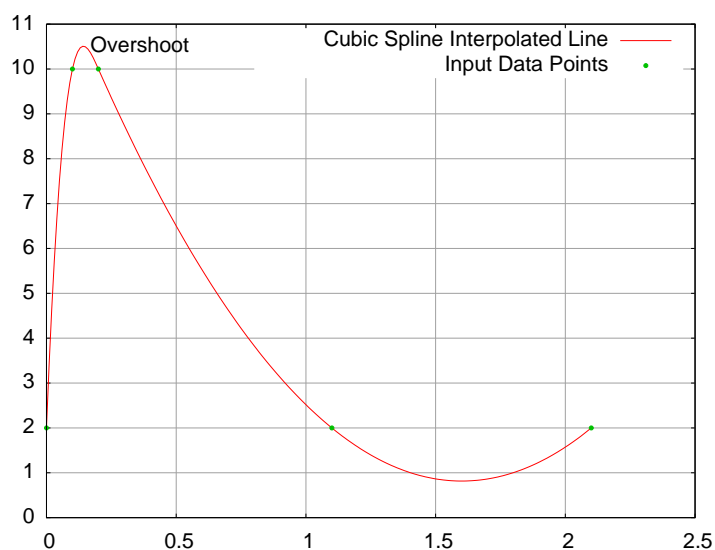


Figure 4.4: Examples of overshoot (and undershoot) when performing cubic interpolation.

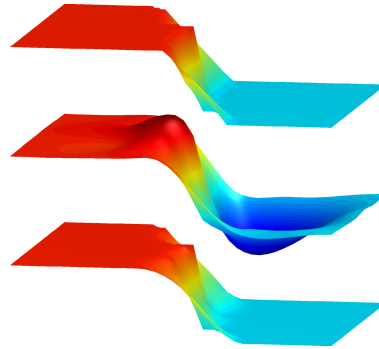


Figure 4.5: Examples of overshoot and undershoot in interpolated simulated edges. Images were reconstructed using (bottom) **NN** (middle) **TPS RBF** and (top) Cubic interpolation.

4.1.4 Natural Neighbour Interpolation

NN, which again uses the Delaunay triangulation, but uses the ratio of overlapping areas to determine the weighting of data points. This results in surfaces which vary more smoothly than linear interpolation. The most obvious artefact caused by **NN** is the fact that it tends to produce sharp points near the input points. The surfaces produced are similar in appearance to a heavy rubber sheet, stretched over and attached to input points. The sheet is more heavily stretched near input point, because **NN** creates output values based on the area of overlap between Voronoi cells before and after the input point is added to the image.

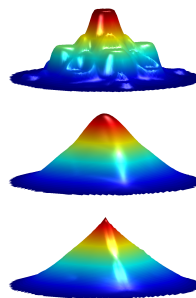


Figure 4.6: Example interpolated pyramids, demonstrating the pointy effect that **NN** causes (bottom), along with the same data interpolated with cubic interpolation based on triangulation (middle), and **ANC** with undersized filters (top).

4.1.5 Adaptive Normalised Convolution

ANC [Foster and Evans, 2008a], which uses convolution with rotated and scaled filters to perform interpolation. It produces outputs which includes anisotropy where necessary, and in too case produces outputs which are somewhere between natural-neighbour and **TPS** interpolation. The main artefact that **ANC** produces is a kind of stepping effect,

which occurs when the filters are too small to accurately capture the spatial variation of the image. In this case, the output images appear similar to the topmost image of Fig. 4.6, which has a stepped appearance. This kind of artefact can be prevented by increasing the minimum size of the filters used. Errors such as this are fairly common in adaptive techniques where the adaptation algorithm is unable to handle data whose properties do not match those for which it was designed.

4.1.6 General Examples

Knowing the artefacts that different interpolation methods can cause, as well as when, where and how they occur can be very useful in situations where analysis techniques are sensitive to specific effects. In these cases, the ability to choose the interpolation method best which best suits the application, or analysis is very important. For these reasons, this section has described and introduced some artefacts produced by several interpolation techniques.

4.2 Interpolation Error Distributions

One very effective way of examining interpolation methods for possible problems is to create a histogram of the errors between an interpolated output, and a simulated full-field input. This can also be attempted using a cross-validation style method (see Chapter 3), although often this will not yield enough error measurements for the creation of a full histogram.

The histogram should describe an approximate Gaussian distribution centred on zero. If the input data are reasonably dense, then the bins around zero will likely have considerably higher values than would be expected in a standard Gaussian, due to large numbers of zero (and close to zero) valued errors. Long tails are also fairly common in real data due to noise which cannot be effectively interpolated.

Once the distribution of errors has been calculated, the first three statistical moments of the distribution can be used to examine the output for biases and skewed output, as well as the calculation confidence limits on the errors.

In ascending order the moments can be used as follows:

1. The first standardised moment of a distribution is its mean. If this is not zero, then there is a systematic bias in the reconstruction being examined. These results should almost certainly be discarded. Fortunately, seriously biased interpolated outputs only rarely occur in practice, making testing common interpolation methods largely unnecessary.
2. The second standardised moment is the variance. This is useful for characterising the spread of error values. Taking the value of the distribution at the positions indicated by the standard deviation gives a 68.26% confidence limit. Confidence limits are discussed in more detail below.
3. The third moment describes the 'skewness', or asymmetry of the distribution. A non zero skewness, is indicative of a tendency for the interpolation method to under-, or over-estimate the output values which do not lie on input datum. Asymmetry in distributions is known as 'skew', and is defined as the third-standardised moment of a given distribution:

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \quad (4.1)$$

Where μ_3 is the third moment about the mean, and σ is the standard deviation.

4. The fourth moments is the ‘kurtosis’, which describes how outlier-prone a distribution is. Standard Gaussian distributions have a kurtosis of 3; distributions which are more outlier prone have higher kurtosis values, and those which are less outlier prone have lower values. Many definitions of kurtosis subtract 3 to make the kurtosis of the normal distribution equal to zero. This is known as *excess kurtosis*. High kurtosis values in error distributions are an indicator that there may be fairly small numbers of large errors. If this is the case, an examination of confidence bounds to ensure that the performance is acceptable should be carried out. Fig. 4.7 shows plots of three distributions: Gaussian, in red, which has an excess kurtosis of 0, Logistic, in green, which has an excess kurtosis of 1.2, and Wigner semicircle which has an excess kurtosis of -1. These curves show that the kurtosis measures how much of the distribution lies close to the mean.

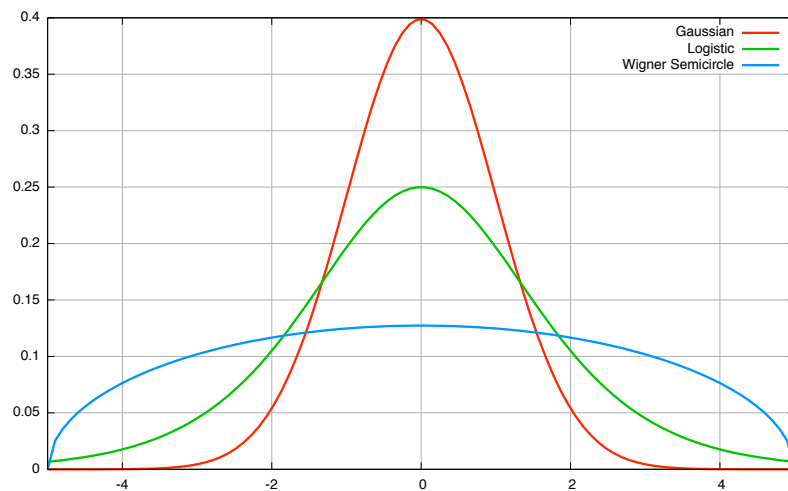


Figure 4.7: Various distributions with differing kurtosis values, including Gaussian (excess kurtosis 0), Logistic (excess kurtosis 1.2) and Wigner semicircle (excess kurtosis -1).

4.2.1 Error Skew

As a further example of the skew problem mentioned above in point 3, Fig. 4.8 shows a sample image and its reconstruction with kriging (a), the semivariogram (b), and the error histogram (c). This histogram shows that the technique is slightly skewed in this case, indicating that output values tend to be under-estimates.

To examine why this is, consider how kriging operates. Kriging works in several stages, the first of which is the estimation of an *experimental semivariogram*, which describes the spatial autocorrelation of the data being interpolated. This then has a model fitted to it, which is then used as a basis function for a global interpolation. In this case, the image has patches with high levels, followed closely by patches with low levels (the patches

4.2. INTERPOLATION ERROR DISTRIBUTIONS

are actually grains of rice), because each grain of rice has a different orientation, the level of autocorrelation in the image varies significantly across the image, which is also highly anisotropic. This means that the semivariogram is unable to fully capture the spatial variation of the image, which in turn leads to a poorly fitting semivariogram model, which culminates in a poor interpolated output. The main consequence of the poorly fitting model is that the basis function chosen for the interpolation leads to concave surfaces in the output, which causes a large proportion of the output positions to contain under estimates than overestimates, this is evidenced by the longer left hand tail in Fig. 4.8c. Problems with kriging are also discussed in section 3.4.

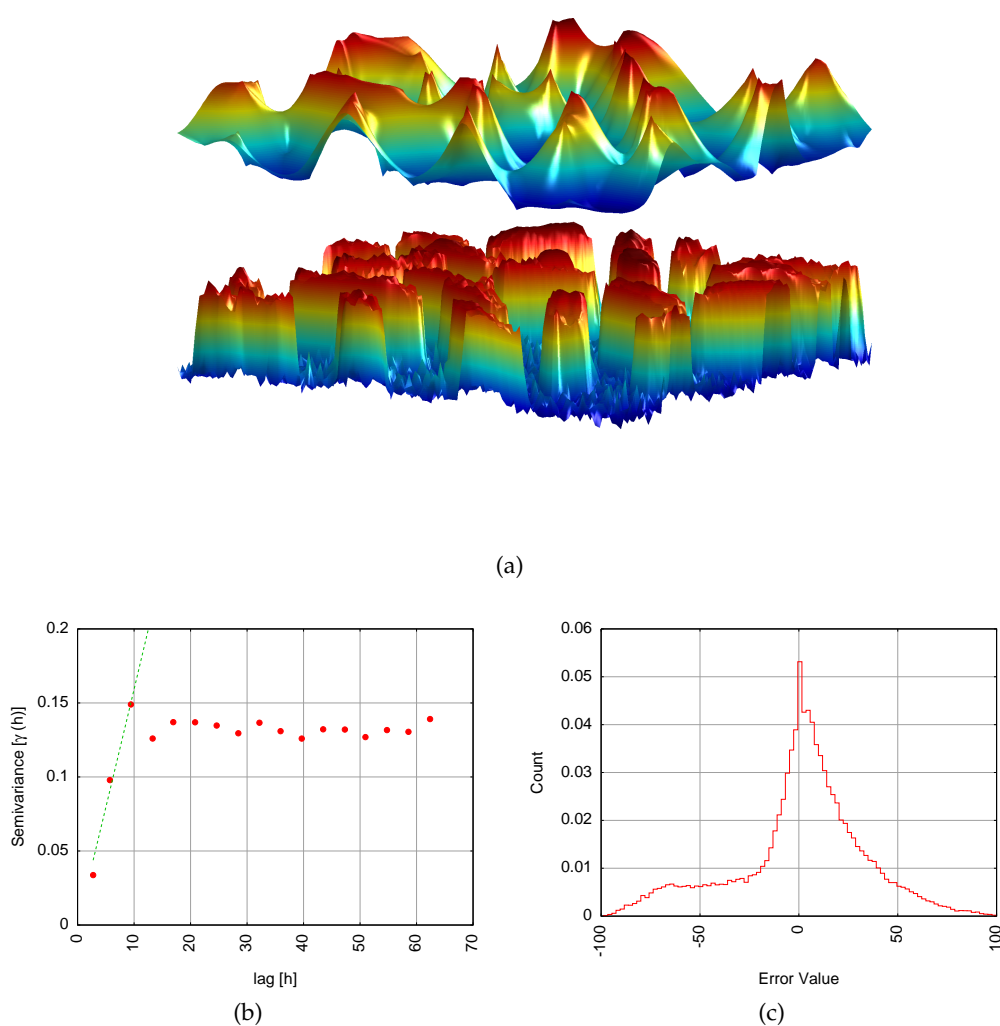


Figure 4.8: (a) (top) a reconstructed version of the sampled image (bottom), (b) the semivariogram for the sampled version of the sampled version of the bottom image in (a). (c) a histogram showing the distribution of error values between the images in (a).

4.2.2 Confidence Limits

Error distributions can be used to calculate confidence limits on output errors. These can be calculated using the standard distribution of errors, or as absolute error bounds are probably more useful, the histogram of absolute errors could be used to calculate the limits.

Motion Estimation Techniques

This chapter describes the most common motion estimation techniques and their application to features in polar ionospheric total electron content (TEC) data. First, in Section 5.1, the data being examined are described, example frames and a schematic diagram are presented to illustrate the spatial extent of the data. Section 5.3 then introduces the assumptions which are common to all motion estimation techniques. Section 5.4 describes *differential analysis*, one of the most fundamental techniques, and the precursor of the subject of Section 5.5, *optical flow*. Section 5.6 introduces *template matching*, and *correlation-relaxation labelling*, two related techniques. (These are examined in more detail in Section 6.2.) Section 5.7 discusses *boundary tracking* methods. A class of techniques which make use of segmentation and shape boundaries in order to infer motion. This section includes discussions on morphological segmentation, (Section 7.1) and shape context matching (Section 5.7.2). Finally, feature detection based techniques are discussed in Section 5.8.

5.1 Data Sources

The chief source of data considered in this, and later chapters on motion estimation (6 and 7), was gathered by global positioning system (GPS) receivers during the “Halloween Storm”, which occurred during the 29—31 October 2003. This was a large geomagnetic storm, with peak activity that occurred between about 20:00 and 23:00 UTC, over North America. During this time, a large region of storm electron density (SED) developed over mainland USA, and moved northward over Canada and through the polar region to the night-side of the Earth. This storm caused a great many oper-

ational problems and failures for satellites as well as many problems for land-based communications and power systems.

The available data consist of images created using the Multi-Instrument Data Analysis System (MIDAS) software from the Department of Electronic and Electrical Engineering, at the University of Bath. These images were generated by extracting integrated electron measurements from paths between GPS receivers and satellites, and performing a 4-D tomographic pseudo-inversion. The maximum resolution of an output field from any given inversion algorithm is heavily dependent on the number of available path measurements, thus the number of GPS receivers is an important factor in determining output image resolution. There is a strong correspondence between population density and the number of GPS receivers in a given area, such that the resolution of the TEC images that can be created varies with geographic location. For example, over North America, there are over 500 GPS receivers operated by the IGS, whilst in the northern polar region (considered here), the number of receivers is considerably lower and so the resolution of the TEC maps that can be produced is also low. The region covered by the inversion is a 3-D grid extending radially in an approximate square of dimension 96° around the north geographic pole, such that the voxels have a horizontal resolution of 4° and vertical resolution of 40 km, from an altitude of 100 to 1600 km. TEC maps were created by taking radial line integrals through the grid, giving a 2-D image with a resolution of 25×25 pixels. Fig. 5.1 shows the extent and location of the mapped area, as well as an example of the actual data.

The low resolution of this data is the chief source of difficulty in estimating the motion of the tongue of ionisation (TOI). The images contain data ranging in value from close to zero to ≈ 250 TECU¹, meaning that a compact representation using 8-bit unsigned integers is possible. This is advantageous because many image processing techniques, such as mathematical morphology are only able to operate on integer classes.

5.2 Model Based Approaches

Traditionally, analysis of TOI motion has been carried out using models [e.g. Alexeev et al., 2007], as opposed to making use of image-derived vectors. For example, Fig. 5.3 shows plots of the data being considered here with overlaid modelled $E \times B$ vectors. These vectors were calculated using the Weimer [1995] electric field and IGRF magnetic field models and show the modelled plasma flow. The images clearly show a simple

¹1 TECU or TEC unit, corresponds to an electron concentration of 10^{16} electrons per m^2

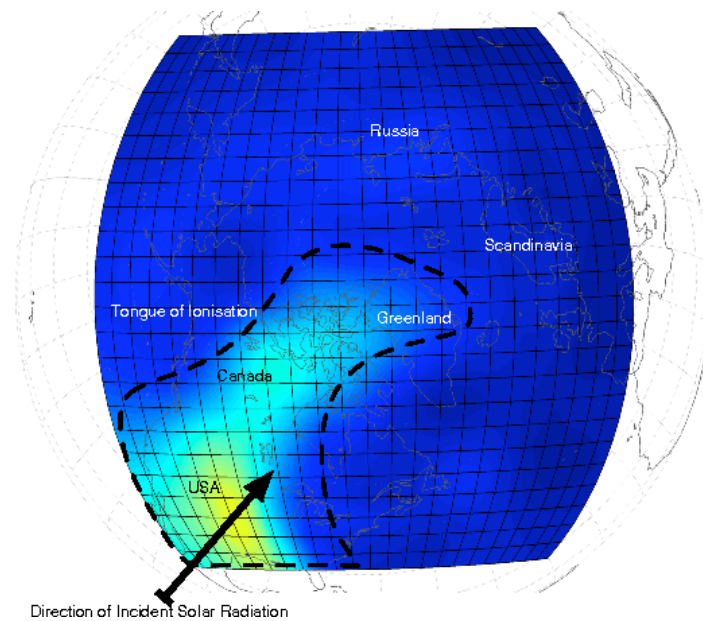


Figure 5.1: The physical location and coverage of the **TEC** data considered here. The dashed line marks the **TOI** and various regions and countries are marked, along with the direction of incident solar radiation and material.

precessing two-cell convective pattern. Various different convection patterns can occur, depending on the orientation and strength of the components of the Interplanetary Magnetic Field (**IMF**) [e.g. [Cowley, 1998](#)], although this pattern is one of the most common.

5.3 Assumptions

As with most practical systems, assumptions must be made in order to allow object motion to be estimated and analysed using images. The main assumptions made by most motion estimation systems are given by [Hlavac et al. \[1999\]](#), and state that:

- A given object's motion does not exceed some *maximum velocity*. This means that if the object's position in one frame is known, it must be within a circle of radius $v_m \cdot dt$, which is centred on the last known position in the next. Where v_m is the maximum velocity of the object, and dt is the time between frames.
- The acceleration of objects is small.
- The motion of objects is *similar*, meaning that points belonging to one object move similarly.
- There is *mutual correspondence* between object points in each frame. The exceptions are object occlusion and rotation, which can mean that parts of an object present

5.3. ASSUMPTIONS

in one frame may not be present in following frames.

The third of these is known as the *velocity smoothness constraint* and is the cornerstone of optical flow techniques. It is a very reasonable assumption (similar to the theory of regionalised variables, see Section 2.6) which arises from the fact that images tend to have regional autocorrelation (characterised by an autocorrelation distance). If this autocorrelation is not present, an image is essentially random noise. This does not necessarily hold where multiple objects are moving in different directions, which would be expected to cause discontinuities in the motion field.

It should be noted that the final two assumptions do not necessarily hold unless the objects being analysed are rigid. Nevertheless, these are the fundamental assumptions upon which techniques such as template matching (see below) are based.

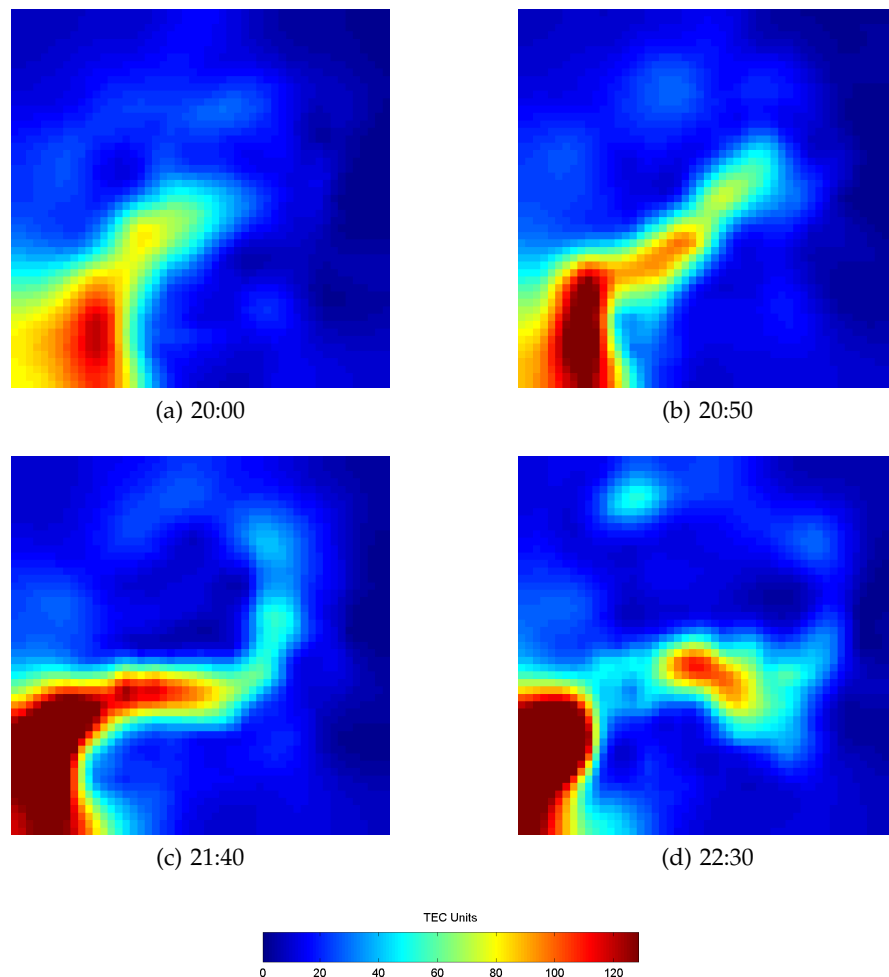


Figure 5.2: Example false-colour frames from the images sequence from Halloween 2003. Images (a)—(d) are each separated in time by 50 minutes, and are up-sampled by two. The colour-scale is shown below.

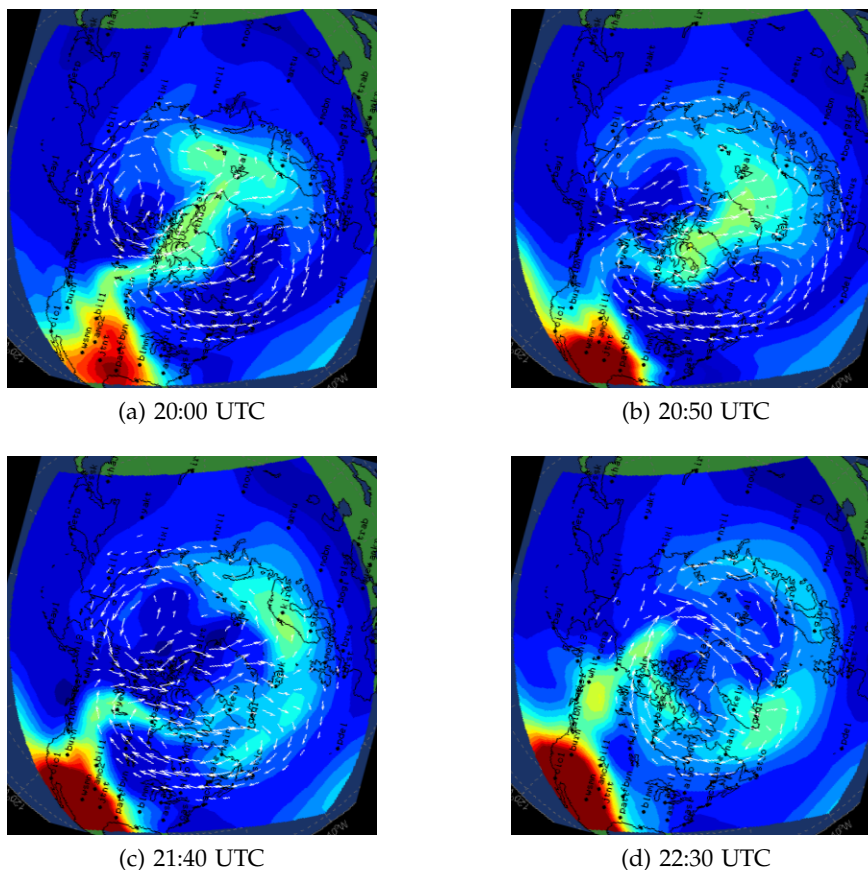


Figure 5.3: Plots of the storm data with overlaid modelled $E \times B$ vectors. Images are rotated so that they are similarly oriented to the previous figures. After Dr P. Yin.

The two main approaches to motion estimation are optical flow, of which differential analysis is a precursor, and correspondence based methods, in which similarity metrics are used to identify objects across as they move across frames. Methods of this type include template matching and boundary correspondence based techniques. The remainder of this chapter discusses these methods, starting with differential analysis in Section 5.4.

5.4 Differential Analysis

Subtracting images acquired at different times allows motion to be detected in a very simple and direct fashion. Given two frames $f_1(x, y)$ and $f_2(x, y)$, motion between them can be detected to give a binary image $d(x, y)$, by subtracting the two images and thresholding the result:

$$\begin{aligned}
 d(x, y) &= 0 \quad \text{if } |f_1(x, y) - f_2(x, y)| \leq \epsilon \\
 &= 1 \quad \text{otherwise.}
 \end{aligned}
 \tag{5.1}$$

Where ϵ is a small positive constant. This allows detection of any motion relative to the background (provided images are registered correctly) but cannot distinguish between different motion directions. Cumulative difference images (essentially moving averages) can be used to acquire more information, but require a stationary ‘background’ to allow for its proper removal.

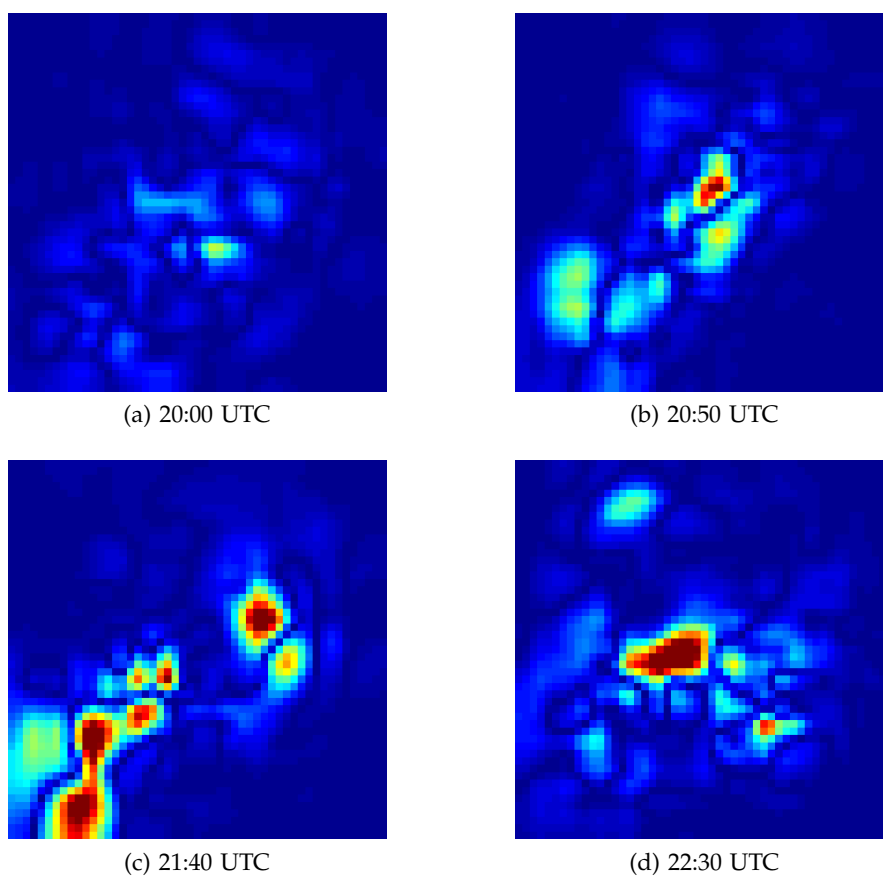


Figure 5.4: Absolute differences between frames at different times in the TEC data sequence.

Fig. 5.4 shows an example of differential motion applied to four frames of TEC data (shown in Fig. 5.2). For purposes of illustration these have not had a threshold applied and have a scaled colour map. In all of these cases, whilst some motion *would* be detected, this method cannot accurately capture the motion of the TOI and gives the highest values in regions outside of the areas where motion can most easily be seen. This is because of the level of change in noise and background between the images. Similarly, there is a change in average value through the sequence, making a static threshold ineffective.

This technique is commonly improved by detecting moving edges. However, in this case edge detection operators, such as the Sobel operator, which typically uses a 3×3 (or larger) kernel, are too large relative to the scale of the image to provide meaningful results. Applying a simple Sobel magnitude edge detector to the the early TEC frames results in two lines, each approximately half the width of the main features, running up either side. This is in contrast to the desired output, which would ideally form a line around the feature.

The use of both spatial and temporal gradients allows differential analysis to reliably detect slow moving edges and some weak edges. This is the conceptual leap used by optical flow techniques, which are described in Section 5.5, below.

5.5 Optical Flow

An optical flow field is a velocity field which represents the motion of points across an image. Optical flow techniques aim to calculate these fields by differentiating in time and space, and applying smoothness constraints on the produced flow fields. The main additional assumption on which optical flow computation is based is that an object point's brightness is constant over time. This is the *constant brightness* criterion. Optical flow's derivation [see e.g. Nixon and Aguado, 2008] uses this to give the motion constraint equation:

$$u \nabla_x + v \nabla_y + \nabla_t = 0 \quad (5.2)$$

Where $\nabla_x = \frac{\partial \mathbf{P}}{\partial x}$, $\nabla_y = \frac{\partial \mathbf{P}}{\partial y}$, $\nabla_t = \frac{\partial \mathbf{P}}{\partial t}$, u and v are the horizontal and vertical *optical flow* components, and \mathbf{P} is an input image image.

This equation describes how an image changes in time or *moves*, and shows that 'optical flow' or the motion of each pixel, plus the rate of spatial intensity produce this effect together. To extract the optical flow, this equation can be solved by estimating ∇_x , ∇_y and ∇_z using image derivatives. However, this problem is ill-posed, since many values of u and v will satisfy (5.2).

Instead, solutions which minimise the error (ec) and the integral of the rate of change of flow along both axes (es) must be sought. In addition, a smoothness constraint can be applied to ec , because of the assumption that neighbouring points move with similar velocities. This leads to the minimisation of:

$$e = \lambda \times ec + es \quad (5.3)$$

Where λ is a regularisation parameter, or smoothness constraint. This can then be

implemented in discrete form using first-order differences and solved in an iterative process. Gradients can be calculated by examining 2×2 pixel, or larger neighbourhoods (or using scale-space methods). However, in the context of these small images, larger neighbourhood sizes will not be as effective. Also, using even-sized neighbourhoods will result in a slight offset to the estimates.

Altering λ , the regularisation parameter, in (5.3) from small to large values increases the weighting of the smoothness constraint over the brightness constraint. Tuning this parameter can increase field smoothness, although care must be taken not to over-smooth the output field, as this will remove detected motion. This is especially true in cases where the motion is expected to be small. The equation, (5.3), can be modified to include other terms as required.

5.6 Template Matching and Correlation-Relaxation Labelling

Template and block matching arose from the problem of finding *template* images, or small objects, in other images [Nixon and Aguado, 2008, Gonzalez and Woods, 2001].

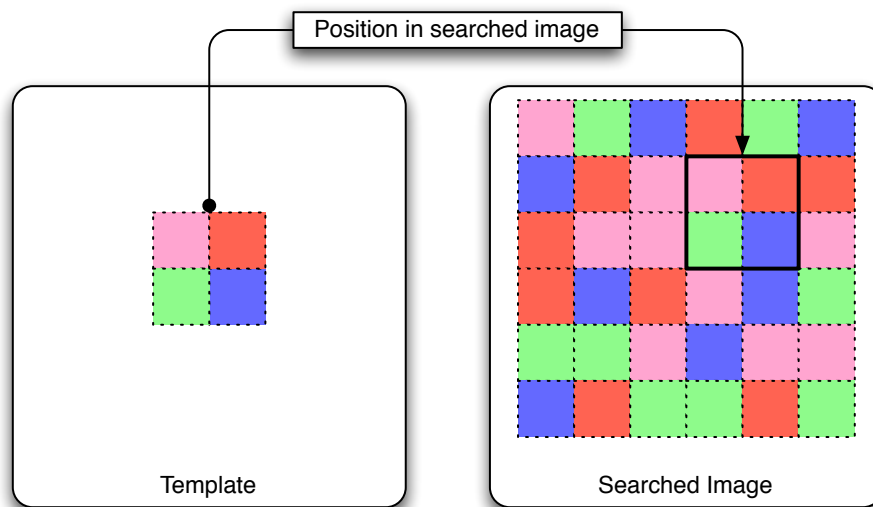


Figure 5.5: A schematic example of template matching

This can be carried out by sliding the template image throughout the source and using a similarity or dissimilarity measure (such as cross-correlation, or sum of squared-differences (SSD)) to measure compatibility between the two. In a more general case, rotation and scale must often also be considered, although in motion analysis these are not usually necessary.

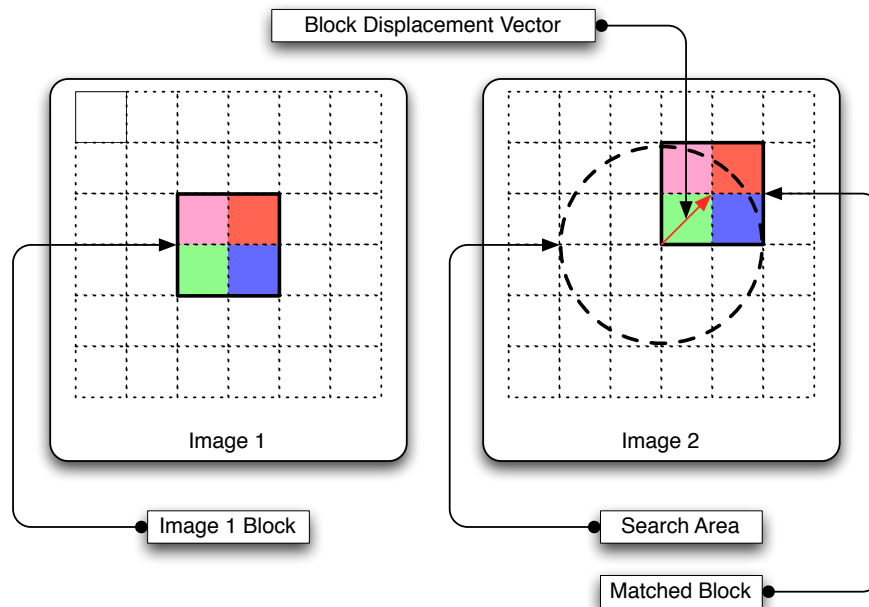


Figure 5.6: The template matching process. Image two is “searched” for values similar to those within the current template in image one. The result is a vector describing the displacement necessary to shift the template in image one to its position in image two.

As described above, template matching is a technique for estimating motion between images by comparing blocks in one image, with blocks at various different offsets in a second image. A block is simply a square patch of pixels, and the offsets and comparisons are essentially searching for patches of pixels in the second image which are similar to the block in the first. Once this has been carried out, the motion estimate is simply the block-offset which is deemed to most appropriately represent the apparent motion between the image. This is often simply the shift to the best matching block, but can also be adjusted using some additional filtering or regularisation.

In this section, the first image will be referred to as the *source* image, and the second will be known as the *destination*. Fig. 5.6 illustrates the (simplified) process of template matching.

Correlation-relaxation labelling is the extension of template matching by the addition of a relaxation-labelling stage. This is an iterative vector regularisation step, which increases the smoothness of a vector field using a smoothness constraint and comparisons with local neighbouring vectors.

This type of motion estimation is commonly used for video coding, although it has also found use in many diverse areas, especially in cases where the temporal resolution between images is too low for optical flow techniques to be effective. Template matching

and correlation-relaxation labelling have been widely applied for estimating the motion of non-rigid remotely sensed data. For example, Dransfeld et al. [2006] applied maximum cross-correlation based motion estimation to ocean surface imagery, and found that alone it was unable to provide locally consistent fields which were representative of small-scale ocean currents. Bellerby [2006] used template matching with a scaled correlation measure as the first stage in constructing a mesh to represent a topological transformation. Evans [2006] applied correlation-relaxation labelling to multichannel MSG cloud imagery and found that it gave locally consistent dense vector fields. Evans [1999], also showed the effectiveness of relaxation labelling when used to ordinal measures (based on rank-statistics). These previous results suggest that template matching and correlation-relaxation labelling may be effectively applied to TEC images.

Template searching is done by measuring the similarity between the source template and various different destination positions. The measurements can be done using several measures, and various different structures and similarity measures. These are described below.

5.6.1 Increasing Field Density

Template matching aims to find parts in a destination image which match a given template from a source image. For the purposes of motion estimation, it is simple and convenient for this template to be a square block of pixels in the source image which are searched for in the second image. This process is known as block matching and is a subset of the more general technique of template matching.

As described above, the block matching procedure aims to find matches between blocks in two images. There are several ways of laying out the blocks in the source image, including *non-overlapping* and *overlapping* blocks.

Performing motion estimation with *non-overlapping* blocks gives one motion vector for each block. This can lead to a motion field with few vectors if the images being examined are small, or the blocks large.

Denser motion fields can be estimated by either decreasing block sizes, or by overlapping the image blocks instead of tiling them. A disadvantage of using smaller blocks is that it increases the output noise in the vector field, as smaller blocks reduce the statistical significance of matches. Blocks can be overlapped by shifting adjacent blocks by an amount which is less than the block dimension, with the natural limit being a one pixel shift. Using *overlapping* blocks gives more vectors and a smoother output field than the non-overlapping case due to increased correlation between neighbouring blocks. The

disadvantage is a correspondingly higher computational cost due to a larger number of comparisons.

The main weakness of correlation-based methods is their poor performance in situations where texture is lacking. This is a particular problem in the sequence being considered here, as the vast majority of the area of the images corresponds to regions with very low energy and texture. Additionally, a large proportion of each image is background noise. As a result, a large proportion of the vectors are essentially useless. To help mitigate this problem, threshold values can be applied to block statistics, disabling them when the conditions are not met and so only estimating motion where appropriate.

5.6.2 Sub-pixel Block Matching

In general, motion between frames does not necessarily occur in integer pixel increments. This is because the images being examined are sampled representations of a continuous object or area. For this reason, it is useful to be able to measure motion with *sub-pixel* accuracy. This is done by up-sampling the images being examined, before performing block matching. However, up-sampling the images can result in the introduction of interpolation artefacts. To reduce these, the up-sampling factor should be kept low (< 4 , if using bi-cubic interpolation). Care should also be taken when up-sampling to ensure that the desired values lie on the same grid as the input samples. This will ensure that the actual input values are included in the output. Additionally, there is no need to up-sample the source frame, which can instead be padded with zeros. Fig. 5.7 shows the process diagrammatically. The output of sub-pixel accurate motion estimation systems is a vector field where the vector-resolution is $\frac{1}{S}$ (where S is the up-sampling rate), and where the maximum displacement is equal to the search radius.

5.6.3 Similarity Measures

Similarity measures establish the correspondence (or correlation) between source and destination blocks. The choice of measure depends on computational cost, and performance, and, as with most problems, some trade-offs must be made. The most common similarity and dissimilarity measures are described below, in approximate order of complexity:

- The sum of absolute value of differences (**SAVD**) is a *dissimilarity* measure, defined as the sum of the absolute value of differences between corresponding pixels in a source block f and destination block g .

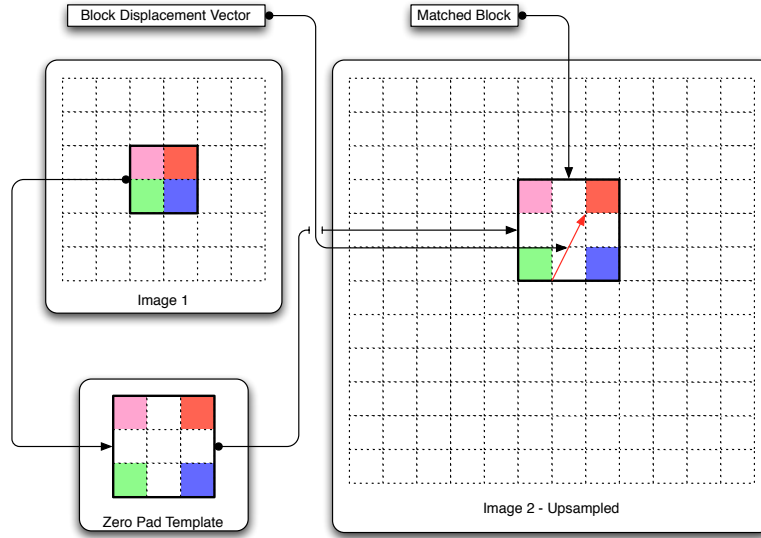


Figure 5.7: A diagram illustrating sub-pixel template matching

$$SAVD(f, g) = \sum_i \sum_j |f(i, j) - g(i, j)| \quad (5.4)$$

SAVD can be made invariant to changes in average intensity, by subtracting the mean of from each block before performing the subtraction.

- The **SSD** is also a dissimilarity measure, and is very similar to the **SAVD**. It is defined as:

$$SSD(f, g) = \sum_i \sum_j (f(i, j) - g(i, j))^2 \quad (5.5)$$

Although the mean intensity is commonly subtracted, as described above.

- The cross correlation coefficient (**CCC**) is widely regarded as the most effective measure, and is a *similarity* measure, returning 1 when blocks are identical, and -1 when they are *orthonormal*. The **CCC** is invariant to changes in mean intensity and standard-deviation and is fairly resistant to noise.

$$\rho(f, g) = \frac{\sum_i \sum_j ((f - \bar{f}) \times (g - \bar{g}))}{\sqrt{(f - \bar{f})^2} \times \sqrt{(g - \bar{g})^2}} \quad (5.6)$$

Where \bar{f} and \bar{g} are the mean values of f and g respectively.

- Ordinal measures [Evans, 1999, 2000b, Bhat et al., 1998] are similarity metrics based on relative rank of intensity values. They were not examined in this study, as using **CCC** based template matching is generally more effective.

A good comparison of these measures was made by Wu et al. [1997], who concluded that whilst the SSD and SAVD are much faster to compute than the CCC, the CCC produces better quality vectors. If these vectors are to be used in a relaxation labelling stage, the number of iterations necessary is lower than when using other similarity measures. For these reasons, and the fact that no real-time processing was necessary, the CCC was used for all template based motion estimation in this thesis.

After the searches and correlations have been performed for a given block, the output will be a set of displacement vectors with a associated CCC for each vector. The most simple systems use the vector associated with maximum CCC as the motion estimate for a given block, a process known as the maximum cross-correlation (MCC) method. However, when using additional smoothing stages, a more sophisticated approach is to use the top n vectors. As an additional analysis step, the CCC values can be plotted against the vector components as a 2-D surface. These *correlation surfaces* are often noisy multi-modal, and have indistinct peaks [Anandan, 1989, Evans, 2000a], which can make choosing the best vectors difficult. This is especially true when using remotely sensed data which is characterised by a lack of texture and contrast. Therefore, relaxation-labelling is used to regularise the motion fields in this work.

5.6.4 Relaxation Labelling

The MCC method has the tendency to produce noisy vector fields because the correlation surfaces it returns are often multi-modal (as described above). To mitigate this problem, additional smoothness constraints are required. This section introduces an effective vector-regularisation method known as relaxation labelling (RL), which is based on probabilistic scene labelling [§8.5 Hlavac et al., 1999].

Relaxation labelling, or probabilistic relaxation, is a method of regularising or smoothing vector fields whilst ensuring spatial consistency. It is commonly used when examining remotely-sensed images of non-rigid physical phenomena and where inconsistently moving fields would be unphysical [Evans, 2006, 2000a, Wu et al., 1997], impossible, or inconvenient. In video compression for example, a smooth vector field compresses better than a disparate one, because the entropy is lower. Similarly, when tracking clouds, a smooth vector field may be more realistic than a noisy one, as small scale motion is often the result of mismatched areas.

The relaxation process works as follows. Some of the terms used below are illustrated graphically in Fig 5.8:

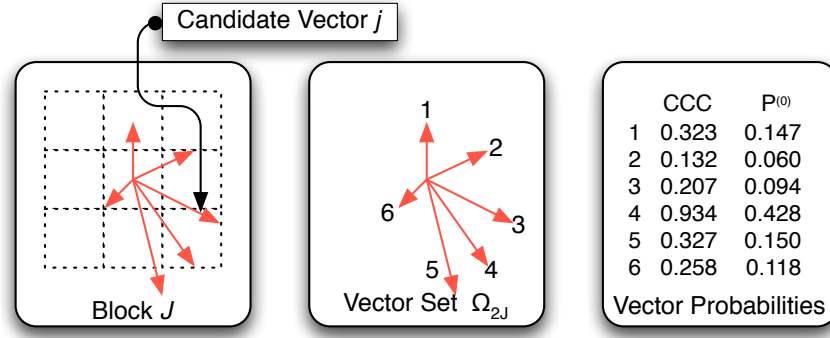


Figure 5.8: Terms used in the relaxation labelling equations (5.7)—(5.10).

1. Each block (J) will have various vectors (j) associated with it, and each pair of vectors will have an associated **CCC**. Vectors which have a low **CCC** should be discarded.
2. The remaining vectors' **CCC**s should be normalised, such that the sum across any given blocks **CCC** values is one. These are now considered to be probabilities. The initial probability for block J , and vector j , is $P^{(0)}(J \rightarrow j)$, and is calculated as follows:

$$P^{(0)}(J \rightarrow j) = \frac{\rho(J \rightarrow j)}{\sum_{\lambda \in \Omega_{2J}} \rho(J \rightarrow \lambda)} \quad (5.7)$$

Where Ω_{2J} is the set of all vectors for a block J .

3. The probabilities are updated using the non-linear relaxation formula (5.8). This formula uses several nested support functions to judge a given vector's similarity with its neighbours in an iterative fashion. These similarity measures are then used to update the probability attached to each vector. The vector from each block with the highest probability attached is chosen as the output.

$$P^{(n+1)}(J \rightarrow j) = \frac{P^{(n)}(J \rightarrow j)Q(J \rightarrow j)}{\sum_{\lambda \in \Omega_{2J}} P^{(n)}(J \rightarrow \lambda)Q(J \rightarrow \lambda)} \quad (5.8)$$

Where $Q(\dots)$ is the following compatibility measuring function, which judges a vector j 's compatibility against those in neighbouring blocks.

$$Q(J \rightarrow j) = \prod_{I \in G_j} \sum_{i \in \Omega_{2I}} P^{(n)}(I \rightarrow i)R(I, J, i, j) \quad (5.9)$$

The output from this will depend on both the probabilities of the neighbouring block's vectors, and the output of the function $R(\dots)$, which measures vector

similarity, and has the following form:

$$R(I, J, i, j) = \exp \left[\frac{\Delta x_{I,i} - \Delta x_{J,j}}{\sigma} \right] \exp \left[\frac{\Delta y_{I,i} - \Delta y_{J,j}}{\sigma} \right] D(I, J) \quad (5.10)$$

Where $x_{I,i}$ and $x_{J,j}$ are the x -components of the displacement vectors i and j in blocks I and J respectively. σ is a parameter that controls the convergence during the iterative procedure of updating the probabilities. In practice, it is usually set to 1. $D(I, J)$ is a function which returns the relative distance between the blocks I and J , given by:

$$D(I, J) = \max(D_0 - D_{I,J}). \quad (5.11)$$

In (5.11), $D_{I,J}$ is the sum of the horizontal and vertical distances between the blocks, normalised to the block dimension.

4. This process is repeated, until the probabilities are stable to within a given absolute value, or for a given number of iterations.

The output field is then generated by choosing the vector with the highest attached probability from each block, and should be a fairly smooth and consistent field.

As an additional step, a conditional vector-median post filter can be applied to remove vectors which the relaxation labelling stage could not regularise. This works by switching vectors with the median of their local neighbours if the magnitude differs from it by more than a set amount.

In Section 6.2, correlation-relaxation labelling is applied to the TEC data described in Section 5.1, and the advantages of overlapping blocks and smoothing using relaxation labelling are examined using vector entropy.

5.7 Boundary Tracking

Boundary tracking is a broad class of techniques which aim to estimate motion by segmenting object boundaries and then solving correspondences between boundary points across frames. They typically require two stages, *segmentation* and *boundary matching*.

5.7.1 Segmentation

Segmentation can be carried out using various different methodologies, from simple thresholding to more complex methods such as the watershed transform, or graph connectivity [e.g. Soille, 2008], and new methods are continually being developed. The main segmentation framework considered here is mathematical morphology and more specifically, attribute morphology, which is described below.

Mathematical morphology provides a robust toolkit for segmentation and feature extraction based tasks. The extension of traditional morphology for processing greyscale images and later into more general *attribute* morphology has paved the way for the development of a great many different types of segmentation system. Greyscale morphological operators, such as opening and closing by reconstruction, can be used with marker image and thresholding to create simple, shape-preserving systems as-is, and attribute morphology can be used for segmentation in situations where specific knowledge about features being extracted (such as contrast, area or statistical moments) are available.

Morphological Segmentation

Greyscale morphology is a branch of mathematical morphology [Hlavac et al., 1999, Acton, 2001] which operates on greyscale images, and works by considering the connectivity of level-sets. For example, imagine the grey-level of a greyscale image to be a vertical displacement. A level-set is a horizontal slice through the image, at a specific value. This means that a given pixel value can be a member of several level-sets (up to 256 for a typical, 8-bit greyscale image). Fig. 5.9 shows an example image cross-section, illustrating the concepts of level-sets and local maxima. For example, in Fig. 5.9, the set A is a subset (\subset) of D , which, in turn, is a subset of G , which is a subset of I , which, finally, is a subset of J . The hierarchy of sets can conveniently be represented using a directed graph structure, where the root set (J in this case) is the base node of the tree. As the tree represents connectivity between sets, it is known as a *connected component graph*. Formally, a set (C_1) in a connected component graph is linked by an edge to another set (C_2), only if C_2 is a superset of C_1 (written as $C_1 \subset C_2$). The direction of connectivity is always towards the larger set. In this example, the edge ($\mathbf{E}(\dots)$) between sets is written as $\mathbf{E}(C_1 \rightarrow C_2)$ [Meijster, 2001, Salembier, Apr 1998]. The right hand panel of Fig. 5.9 shows the connected component graph for the example image on the left.

This structure is useful in many situations, as the graph nodes can be used to represent

many different image components and properties. This could include lists of pixels in the sets or measures such as area or contrast (see below), perimeter or moment of inertia.

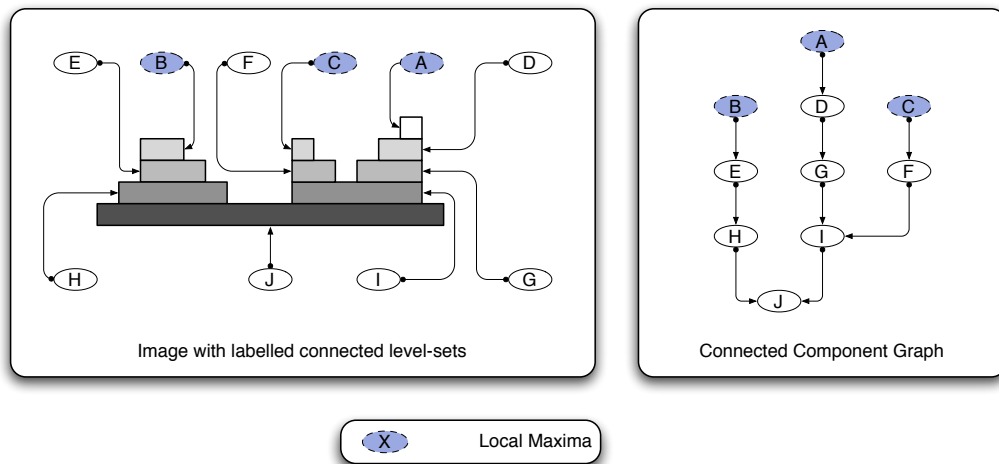


Figure 5.9: An example greyscale image cross-section with labelled connected level-sets, and connected component graph showing the relationship between the various level-sets.

One situation in which this structure is particularly useful is area morphology, where the graph nodes (also known as vertices) relate to the areas of the level-sets. The two main operators available in area morphology are *closing* and *opening*. These modify an image by removing features based entirely on their area. This means that (unlike traditional morphological operators) there is no structuring element, and so none of the associated artefacts. As the parameter of interest is component area, area morphology lends itself well to scale-space applications, [Acton and Mukherjee](#) [see e.g. 2000], [Mukherjee and Acton](#) [see e.g. 2002].

Area opening removes bright or ‘ascending’ objects which do not meet a specified minimum area. Area closing removes ‘descending’ objects which do not meet the specified criterion. Area opening is equivalent to moving down from the top of the connected component graph until the area is exceeded, and area closing is the equivalent to moving up from the bottom until the criterion is exceeded. Fig. 5.10 shows how an area opening using an area of three would alter an example image, and its connected component graph. In the image, A, B, C, D, and F would be removed, as they all correspond to components with areas of less than three.

The connected component graph is also useful for describing and implementing *contrast morphology*. This is similar to area morphology, except that instead of using the area to decide what to remove (which is purely a property of a single level-set), it makes use of the set’s contrast. In the case of opening, the contrast is the difference between

5.7. BOUNDARY TRACKING

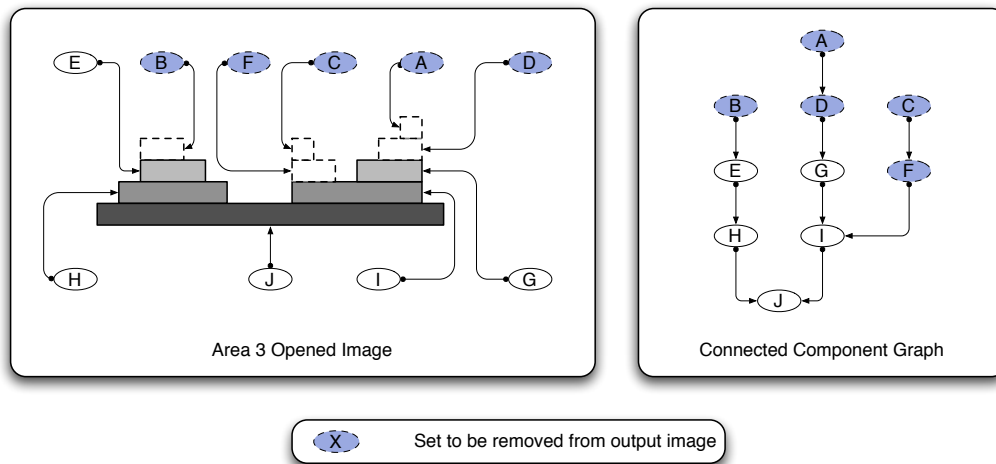


Figure 5.10: The example cross-sectional image from 5.9 showing the effect of an area closing with an area of three (assuming the image depth is only one pixel).

the highest connected local maxima of which the set is a subset, and the set's value. Sets with contrasts below a certain threshold will be removed, whilst others will be left intact. Fig. 5.11 shows how a contrast opening would alter an example image, and its connected component graph. In this case, A, B, C, D, E and F would be removed, as all of these are components whose contrasts are less than three.

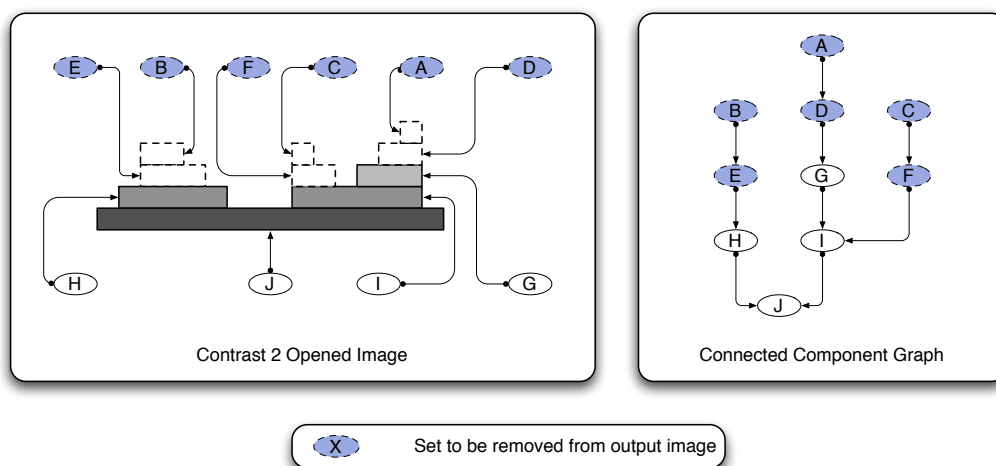


Figure 5.11: The example cross-sectional image from 5.9 showing the effect of a contrast closing with a contrast of two.

Using contrast as the morphological attribute has several advantages over areas based approaches in images where high gradients are present. Where the gradient is high, a small range of areas can correspond to a large vertical extent, and in these cases using contrast instead of area measures could be beneficial. This is illustrated in Fig. 5.12, which shows how a large contrast can be found across an object which only has small

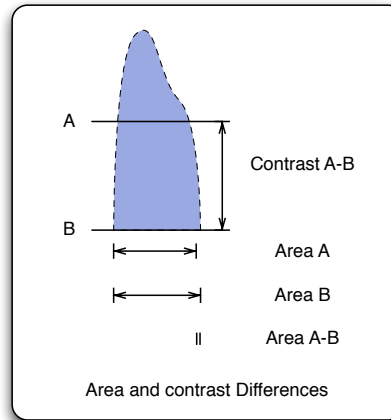


Figure 5.12: The difference between contrast and area parameters. The area can be very insensitive to height, leading to a small change in area consuming a large vertical section of image.

change in area across it.

Snakes

Snakes, or active contours [Kass et al., 1988] are a method of iteratively finding objects with an image. A common analogy is with the deflation of a balloon around an object, once a certain amount of air has been let out of the balloon, its boundary will describe a hull around the object [e.g. Nixon and Aguado, 2008]. Snakes model this process mathematically, using a cost function (known as an energy functional) to try and match image features, such as edge intensity and brightness, whilst minimising length and curvature. Snakes are commonly used to find and track shape boundaries and are usually represented as splines. Splines are good boundary descriptors, which can be also used to infer motion. However, when applied to the TEC images used here, snakes were found to be unsuitable. This is because of the lack of texture in the images, and their low resolution.

5.7.2 Boundary Matching

After boundaries have been extracted and described using chain code or coordinates, it is necessary to compute the correspondences between boundary points. It may also be appropriate to resample them so that the number of points in each boundary is equal, a process which can be made easier by converting the boundaries to a spine-based representation. Spines are piecewise parametric polynomials which are an extremely powerful tool for compactly and smoothly representing complex curves. Using splines

allows smoothing of shape boundaries as well as arbitrary resampling along the parameter of the boundary. These factors make splines ideal for describing object boundaries in both low and high-resolution images.

Following appropriate resampling, methods such as shape context matching [Belongie et al., Apr 2002] or the alignment method of Sebastian et al. [2003]² can be used to determine how boundary points in each frame relate to one another. The coordinate of the corresponding boundary points is then subtracted to give a set of motion vectors, describing how one boundary can be warped into the other. Fig. 5.13 illustrates how boundary correspondences can be used to extract motion estimates. Finally, vector density and smoothness can be increased using interpolation or appropriate models. The shape context method is discussed in more detail below.

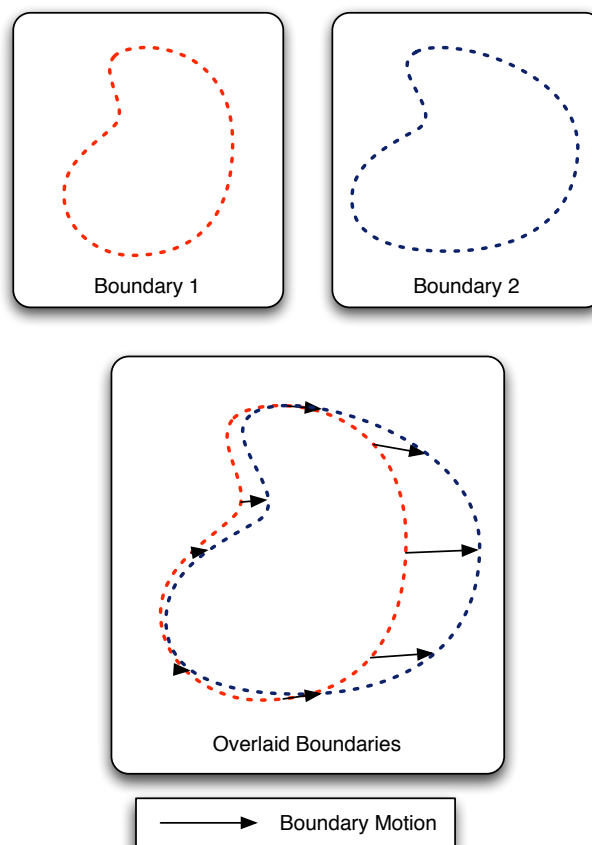


Figure 5.13: A schematic illustrating motion estimation by boundary correspondences

²The latter is similar to the edit-distance for strings (a description of how many insertions, changes and deletions are needed to convert from one string to another) [e.g. Klein et al., 2001].

Shape Context Matching

The method known as shape context (SC) matching was developed by [Belongie et al. \[Apr 2002\]](#) for measuring similarity between shape boundaries. This is done in three steps, the first two of which can be used to estimate the relative motion of the two shapes.

Given boundary points for two shapes, the three steps required to measure their similarity are:

1. Solve correspondences between points on their boundaries;
2. Use these to estimate a boundary alignment;
3. Sum matching errors to calculate similarity.

The output of step 2 is a permutation $\pi(i)$ which maps points on the first shape to corresponding points on second. Differencing the coordinates of these corresponding points gives the vectors which are required to warp the boundary of the first shape into that of the second. If both boundaries represent the same object, and the fundamental assumption is that they *do*, then these vectors will represent the relative motion between frames.

SC matching uses properties of shape boundaries known as *contexts* to calculate correspondences between shapes' boundaries. Further stages can then be used to calculate similarity metrics if desired.

A shape context is histogram which describes the distribution of boundary points relative to an origin point (also part of the boundary). A shape context is created in the following manner:

- choose an origin on the shape boundary;
- subtract the origin coordinates from the other boundary coordinates;
- convert the new coordinates into polar form to get r and θ ;
- create a 2-D histogram by binning $\log(r)$ and θ ([Belongie et al. \[Apr 2002\]](#) uses five bins for $\log(r)$ and 12 bins for θ)

In order to establish point correspondences between boundaries, shape contexts must be created for every possible origin point for each shape. This allows a matching cost,

5.7. BOUNDARY TRACKING

$C_{ij} = C(p_i, p_j)$ to be computed for every possible pair of points (p_i, p_j) , where p_i is a point on the boundary of the first shape, and p_j is a point on the second. The cost of matching can be calculated using a variety of methods, such as the χ^2 cost, which is given by:

$$C_{ij} \equiv \chi^2(p_i, p_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \quad (5.12)$$

As this only compares histogram bins with corresponding bins in the second histogram, other methods, such as the earth mover's distance (EMD) [Ling and Okada, May 2007] or diffusion distance (DD) [Ling and Okada, 2006] (a dissimilarity measure found to be more robust and accurate than the EMD under certain conditions) can be used. These metrics include cross-bin comparisons and so are more robust to problems such as boundary occlusion and noise. The DD is particularly attractive due to its simplicity (it makes use of a Gaussian pyramid and simple filtering operations).

The DD is defined by:

$$K(h_i, h_j) = \sum_{l=0}^L |d_l(\mathbf{x})|, \quad (5.13)$$

where,

$$d_0(\mathbf{x}) = h_1(\mathbf{x}) - h_2(\mathbf{x}) \quad (5.14)$$

$$d_l(\mathbf{x}) = [d_{l-1} * \phi(\mathbf{x}, \sigma)] \downarrow_2 \quad l = 1, \dots, L \quad (5.15)$$

Where l represents the current layer of the Gaussian pyramid with L layers, \downarrow_2 denotes downsampling to half-size. σ is the standard deviation of the Gaussian filter $\phi(\dots)$.

Once matching costs have been calculated — the result of which will be a cost matrix, C_{ij} — the task is to minimise the total cost of matching across the two boundaries, given by:

$$H(\pi) = \sum_i C(p_i, p_j), \quad (5.16)$$

subject to π being a permutation. This means that a one-to-one match is required. This problem is known as a *weighted bipartite matching* problem, which can be solved in a variety of ways, such as the Hungarian method. Which attempts to find an optimal path through the matrix.

The result of minimising equation 5.16 is a permutation $\pi(i)$, describing the optimal mapping between shape boundary points. Constraints and extras costs can be added to this technique by modifying the cost matrix to include them as necessary. For example,

a distance weighting could be added to help ensure that the minimisation returns points close in the boundary order. Metrics relating to image properties could also be added, although finding suitable metrics is problematic, as many, such as curvature, can be very noisy and introduce ‘pits’, very low values, into the cost matrix. These serve to pull the match towards them, which can give poor vectors as it can lead to bad boundary correspondences.

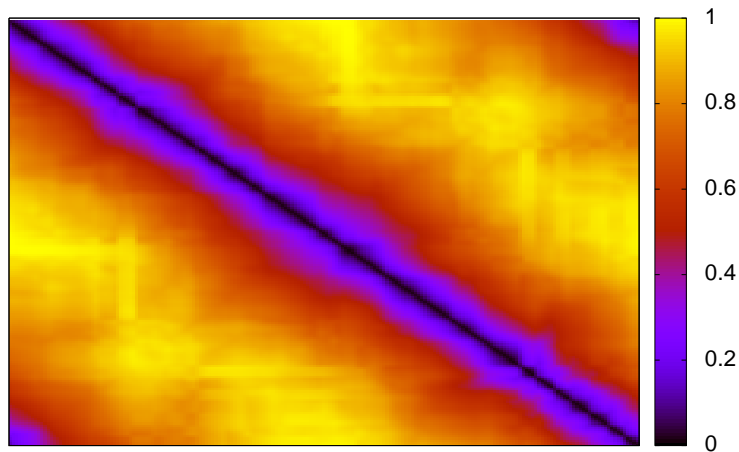


Figure 5.14: An example cost matrix created using the diffusion distance histogram metric, weighted by the distance between matching points.

The permutation can then be used to calculate the distances between corresponding boundary points on the two shapes, by subtracting the coordinates of the first shape’s boundary points from the corresponding points on the second shape. The result of this will be the set of vectors required to warp the first shape into the second, that is their relative motion.

Chapter 7 applies this process to the **TEC** data described in Section 5.1.

5.7.3 Boundary Vector Transformations

An additional step which can be performed on estimated boundary vectors is estimating the plane transformation required to warp one shape into the other. This can be done in a variety of ways, and results in a 2-D warping transformation T which allows arbitrary points from one shape to be mapped into the other. This process allows denser vectors

fields to be estimated. For more detailed information on image warping, see [Wolberg \[1994\]](#).

The two most commonly used methods of choosing T are least-squares fitting a standard affine model, and the estimation of warping planes using radial basis function (RBF) interpolation [[Belongie et al., Apr 2002](#)]. This interpolation method is described in more detail in Section 2.5. Regularisation can also be applied to smooth the warping planes as needed, reducing the noise in the vectors.

Affine Model

The affine method attempts to fit a standard affine model of the form

$$T(\mathbf{x}) = A\mathbf{x} + o, \quad (5.17)$$

where A is a matrix describing rotation and scaling, and o is a translation vector. The least squares solution is denoted $\hat{T} = (\hat{A}, \hat{o})$. \hat{o} can be found by taking the mean distance between corresponding boundary points, and \hat{A} can be found using:

$$\hat{A} = (Q^+P)^t, \quad (5.18)$$

P and Q contain the homogeneous coordinates of the boundaries described by p and q , in the same form as equation 2.24. Q^+ is the pseudo-inverse of Q . The outputs, \hat{A} and \hat{o} can then be used in place of A and o in equation 5.17.

Typical fields produced by affine models and are characterised by an area where the vectors are very small (an origin). Affine modelled fields tend to be have insufficient degrees of freedom to accurately describe complex motion, such as exhibited by the TOI in the TEC images.

RBF Fitting and Regularisation

[Belongie et al. \[Apr 2002\]](#) uses two separate thin plate spline (TPS) surfaces (one for the x -axis mapping and one for the y -axis), fitted using RBF, giving a model of the form:

$$T(x, y) = (f_x(x, y), f_y(x, y)) \quad (5.19)$$

The fitting is carried out by considering the boundary positions of the first shape $p_i = (x_i, y_i)$ as the input coordinates, and then taking the x -component second shapes' coordinate as z_i . RBF interpolation is then performed to give f_x . The same process is

then performed using the y -component of the displacement to get f_y .

If the coordinate mapping is considered noisy, it is possible to relax the interpolation condition during the RBF fitting. This process is called regularisation, and allows a smooth surface to be fitted to the data.

This is done by modifying A in equation 2.23, by replacing it with:

$$A + \lambda I \tag{5.20}$$

Where λ is a scale dependent *regularisation parameter* which controls the amount of smoothing, and I is an identity matrix. Setting $\lambda = 0$ corresponds to interpolation, and setting λ to large values creates outputs which are similar to those from a fitted affine model.

The scale dependence can be removed by replacing λ with $\alpha^2 \lambda_0$, where α is the scale of the inputs points, and λ_0 is normalised λ . This can be estimated by taking the mean edge length between input points. λ_0 can now be varied between zero and approximately one. This makes choosing a value for λ much easier.

5.8 Other Techniques

Some common techniques for estimating motion have been described in the previous sections. This section describes other techniques which have not been further considered in this thesis.

Sparse velocity fields can be estimated by using the correspondence of *points of interest*. This technique is useful in large images where the time interval is too long for optical flow techniques to be effective.

This process works by detecting points of interest using operators such as the Moravec operator or scale-invariant feature transform (SIFT) features [Lowe, 1999]. These operators generally search for corners (or intersecting straight edges). The points returned from two or more adjacent frames are then used to estimate inter-frame motion, using, for example a probabilistic relaxation framework [e.g. Hlavac et al., 1999]. The result of this process will be a set of correspondences between detected points of interest which can be converted to motion vectors by simply subtracting the vectors.

Due to the fact that most feature detectors, such as SIFT, require multiple filtering operations at different scales, this method is unsuitable for use on the small images

5.8. OTHER TECHNIQUES

used in the studies in this thesis.

Motion Estimation using Optical Flow and Correlation-Relaxation Labelling

During Ionospheric storms, the concentration of electrons in the atmosphere can increase dramatically. These electron concentration enhancements, known as storm electron density (**SED**), manifest as *blobs*, which move across, or around, the northern polar region under convection [e.g. [Foster et al., 2005](#), [Mitchell et al., 2005](#)]. During certain conditions [when the z -component of the Interplanetary Magnetic Field (**IMF**) is negative], the plasma convection takes on a certain pattern known as a tongue of ionisation (**TOI**).

This chapter discusses the application of optical flow and template matching based techniques to estimating the motion of the **TOI** in total electron content (**TEC**) images. The image sequence being used was described in Section 5.1 (page 60), and model based vectors were shown in Fig. 5.3. The following sections will examine the performance of optical flow and template matching, respectively, and will assess their relative performances by comparing vector entropies. It will also examine the resulting vector fields relative to the modelled $E \times B$ fields.

6.1 Motion Estimation Using Optical Flow

This section examines the application of simple gradient based optical flow (see Section 5.5, page 66) to estimating the motion of ionospheric storm enhancements, as

described in Section 5.1, on page 60.

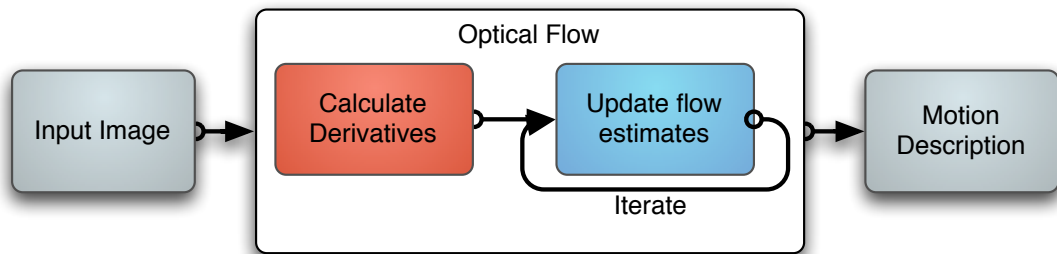


Figure 6.1: The processing pipeline for the optical flow process.

The implementation used in this chapter was based on a modified version of the Matlab code given in Nixon and Aguado [2008] and as such these results are representative of a modern optical-flow algorithm. Fig. 6.1 shows the optical flow process diagrammatically.

6.1.1 Results

To examine the performance of the gradient based optical flow technique, the TEC image sequence was processed using various different regularisation parameters (λ), ranging from 0.1 to 10. Ten iterations were used in all cases to ensure stable results.

Figs. 6.2 – 6.4 show the results from processing the images using regularisation parameters of 0.1, 1 and 5 respectively. The main thing to note, with regards to these images, is their high degree of similarity.

Following the method used in Section 6.2, entropy values were calculated for the produced vector fields using the encoding method of H.263. These are shown in Table 6.1, which shows the average vector entropy values. The first set shows the median, mean and standard deviations of field entropy values and the second shows the result of masking the same vector field using a thresholded a standard deviation filter ($\sigma = 4$), as in Section 6.2 These values are similar in magnitude to those in Table 6.2 (on page 95), although generally higher in this case. This indicates that the vector fields generated are less smooth than those produced by correlation-relaxation labelling.

As table 6.1 shows, changing the regularisation parameter has little effect on the entropy values, and results in an increase in entropy of approximately 30% from the lowest to

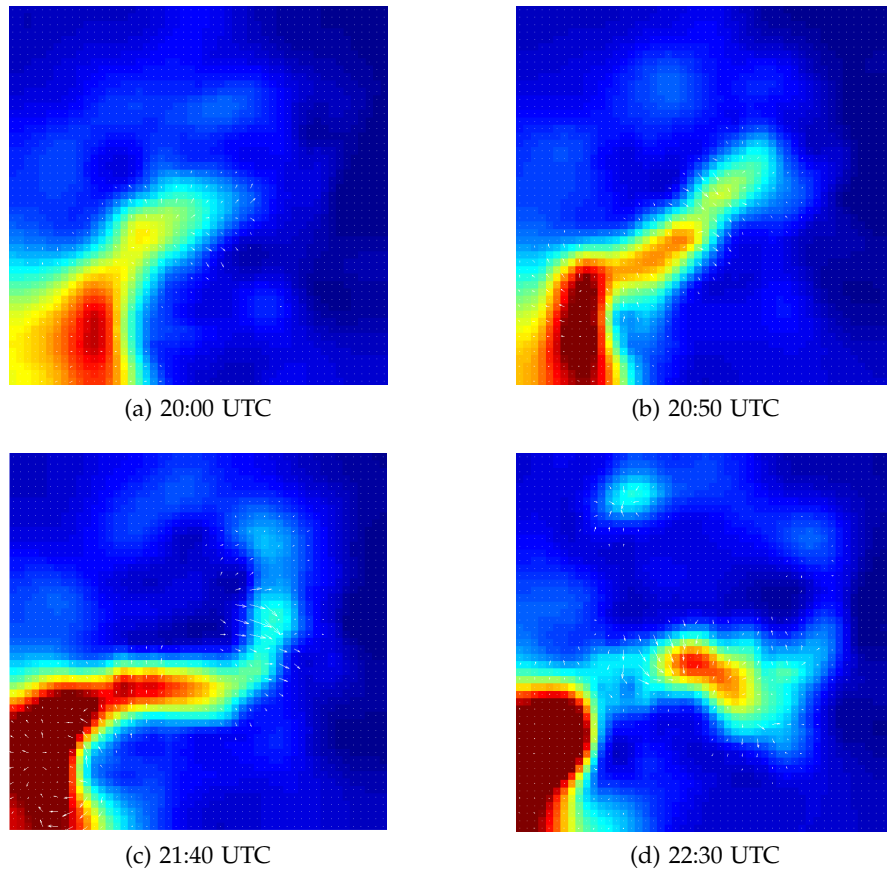


Figure 6.2: Vectors produced using optical flow processing on the TEC image sequence, with $\lambda = 0.1$ and ten iterations.

the highest in the masked case, and less in the standard.

Masking the vector field decreases the entropy values by approximately 2.5 times and decreases the standard deviations of the entropies by a factor of approximately 5. This suggests that removing the ‘background’ regions is an effective addition to the technique.

Visual comparison of these results with Fig. 5.3 (on page 64) shows similar vectors at the start of the sequence, although once again, the vectors towards the end are somewhat confused. Also notable is the fact that in the last frame of each set (d), the general direction of flow is downwards. This is in contrast to the modelled vectors which show an upward motion in that part of the frame.

Table 6.1: Optical flow vector entropies

	λ	median H	mean H	σ
Standard	0.1	5.759	5.765	0.017
	0.2	5.773	5.778	0.018
	0.3	5.763	5.771	0.018
	0.4	5.782	5.791	0.030
	0.5	5.763	5.772	0.023
	0.6	5.769	5.778	0.026
	0.7	5.777	5.783	0.022
	0.8	5.774	5.781	0.024
	0.9	5.779	5.784	0.021
	1.0	5.782	5.785	0.020
	2.0	5.775	5.783	0.026
	5.0	5.783	5.800	0.045
	10.0	5.803	5.813	0.037
Masked	0.1	3.088	3.088	0.003
	0.2	3.226	3.227	0.004
	0.3	3.180	3.180	0.003
	0.4	3.176	3.174	0.008
	0.5	3.168	3.168	0.005
	0.6	3.229	3.228	0.007
	0.7	3.216	3.216	0.005
	0.8	3.266	3.266	0.006
	0.9	3.290	3.292	0.006
	1.0	3.314	3.312	0.006
	2.0	3.308	3.308	0.010
	5.0	3.290	3.291	0.007
	10.0	3.333	3.333	0.012

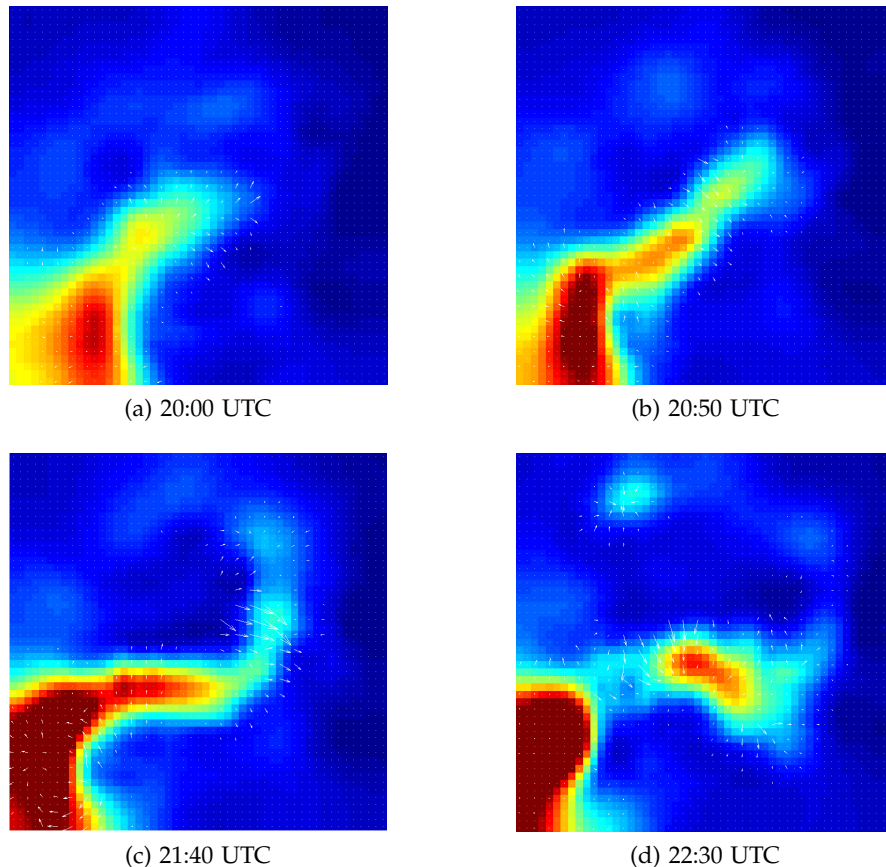


Figure 6.3: Vectors produced using optical flow processing on the TEC image sequence, with $\lambda = 1$ and ten iterations.

6.1.2 Conclusions

The gradient method of calculating optical flow has been shown to perform adequately for estimating the motion of TEC enhancements. Unfortunately, the results for the last section of the sequence have proven to be complex and lacking in consistency. This is exemplified by Fig. 6.2d – 6.4d.

In general, the calculated entropy values are higher for optical flow than for the correlation-relaxation labelling results. This suggests that the fields are less smooth, something which visual comparisons corroborate. However, it should also be noted that optical flow techniques produce vector fields with floating-point magnitudes, a factor which will definitely increase entropy.

Future work in this area would benefit from the application of more recent advancements to optical flow techniques, such as scale-space estimation. This would allow the detection of larger displacements which might more accurately capture the motion of the features in the end of the sequence. However, care would need to be taken not to

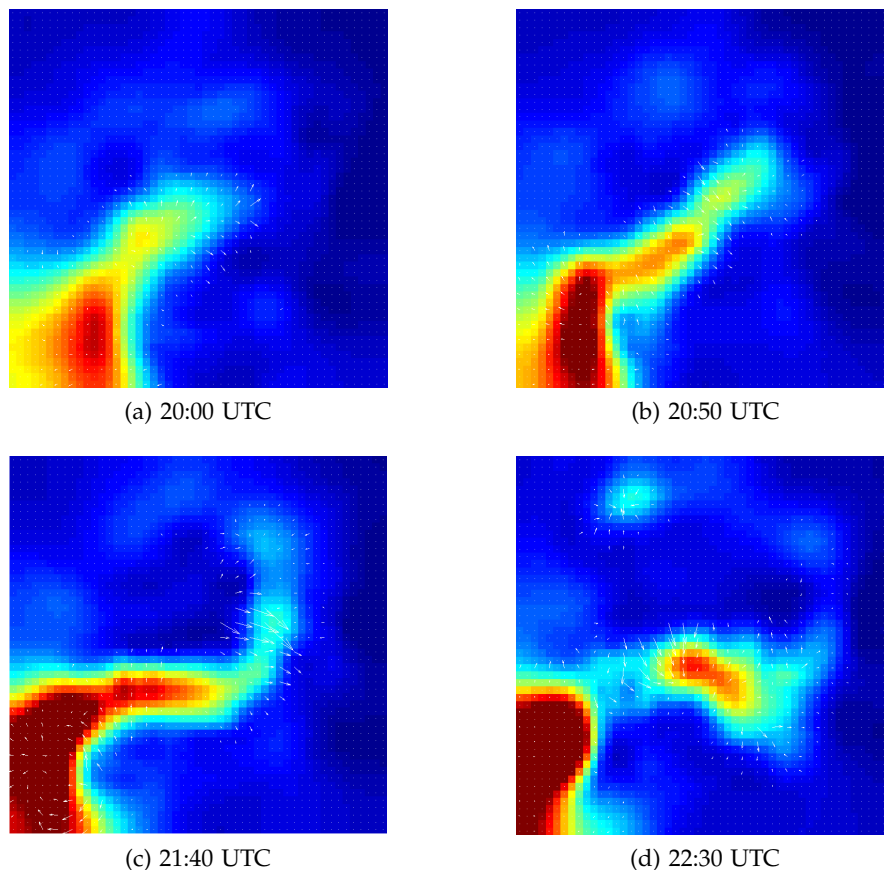


Figure 6.4: Vectors produced using optical flow processing on the TEC image sequence, with $\lambda = 5$ and ten iterations.

use filters which were too large, due to the small size of the images.

6.2 Motion Estimation Using Correlation Relaxation Labelling

This section is concerned with the development of data-driven approaches to estimating the motion of such regions, using template matching techniques. As these are data-driven, they are largely de-coupled from the underlying physics and have the potential to provide a set of techniques for analysing specific storm events that is not reliant on the available physical models [Weimer, 1995, Bilitza, 2001] and instead allows actual events to be examined. These results could then be used for the validation of physical models, analysis of specific storm events, or tracking such as might be used in a forecasting system. Section 6.2.1 shows motion fields calculated by applying the various methods introduced in Section 5.6, and Section 6.2.2 examines these results using comparisons with modelled vectors and entropy values, before drawing conclusions about their relative performance.

6.2.1 Results

The techniques described in Section 5.6 were applied to the TEC data sequence (described in Section 5.1) in order to evaluate the quality and accuracy of the motion vectors they produce in this context.

The result of applying traditional non-overlapping block matching to four frames of the sequence is shown in Fig. 6.6. These vectors were generated using block matching with 5×5 blocks, and a search radius of five pixels, with non-overlapping blocks. Fig. 6.7 shows the result of applying the same process, using overlapping blocks.

As discussed in Section 5.6.1, the estimated TEC image motion can be upon improved by masking the vector field. In this case a mask was applied based on the standard deviation of overlapping blocks (using a threshold condition of $\sigma > 4$). All output images this chapter, from Fig. 6.7 onward use this masking process. Similarly, the images displayed show the vector-medians calculated from non-overlapping 3×3 neighbourhoods of the vectors field to increase image clarity.

Sub-pixel Block Matching

Sub-pixel accurate block matching was introduced in Section 5.6.2 as a method of improving the accuracy of motion vectors. As the vectors shown in Fig. 6.7 do not appear to accurately represent the motion of the sequence, sub-pixel template matching was tested in order to assess its ability to produce a more accurate vector field.

Fig. 6.8 shows the result of sub-pixel block matching on some example frames from the TEC image sequence, 5×5 blocks were used, along with a ten pixel search radius. The same frames are shown here as in Figs. 6.7 and 6.6

These figures clearly show the improvement in field density gained by using overlapping

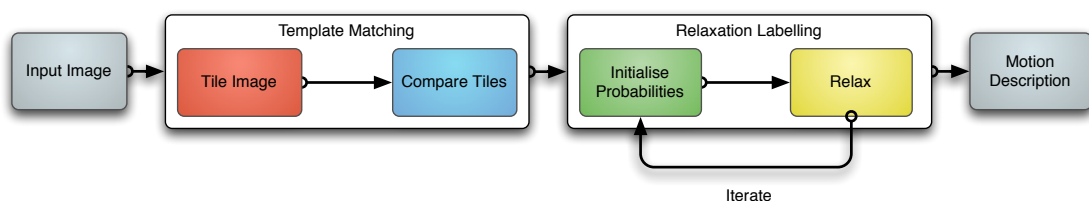


Figure 6.5: The processing pipeline for the correlation-relaxation labelling process.

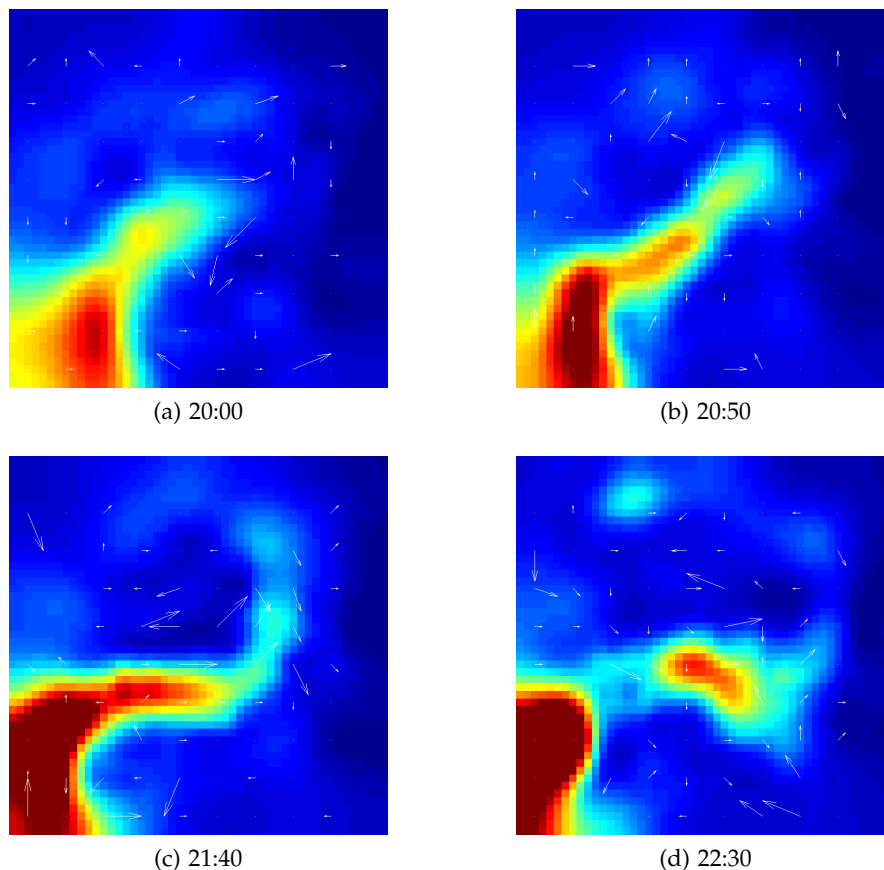


Figure 6.6: Example outputs created using traditional template matching with non-overlapping blocks on the TEC images, up-sampled by a factor of two.

blocks and hint at the improvement in accuracy gained by using sub-pixel estimation. However, in both the standard and overlapping case, and to a lesser extent the sub-pixel cases, the vectors are lack smoothness. For this reason, the use of a relaxation-labelling stage was investigated for improving field smoothness.

This study used the relaxation labelling scheme described in [Evans \[2000a\]](#), which does not include a no-match category. This ensures that each block has an output vector, even when the output field is less consistent than is ideal. Ten iterations were run, using a neighbourhood size of 3×3 (equivalent to 9-nearest neighbour connectivity), and only considering vectors with cross correlation values of greater than 0.5.

Fig. 6.9 shows the result of using overlapping block matching with relaxation-labelling on the TEC image data. The vector field is masked using the standard deviation filter mentioned earlier. The vector fields still contain some areas lacking in smoothness which do not appear to match the observable motion of the TOI. For example on the tip of the TOI in Fig. 6.9a.

Fig. 6.10 shows the result of using overlapping block-matching with sub-pixel accu-

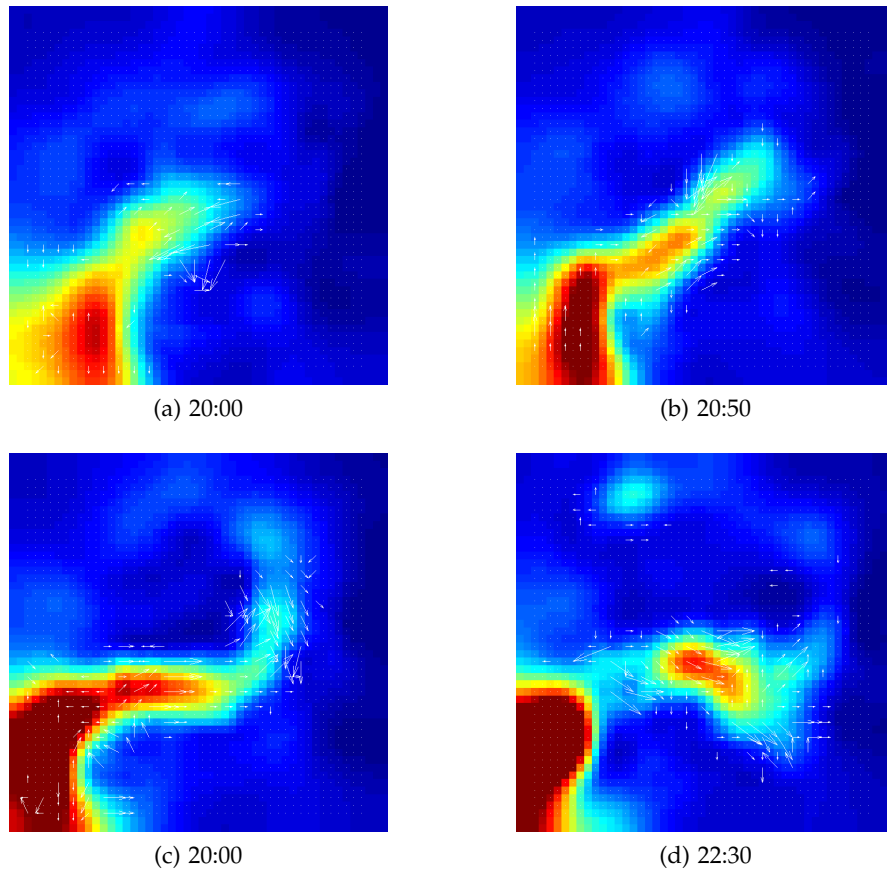


Figure 6.7: Example outputs created using traditional template matching with overlapping blocks on the TEC images, up-sampled by a factor of two.

rate estimation and relaxation-labelling. The vector fields produced using this method are much smoother than in the previous examples and show better agreement with observable motion.

Model Comparisons and Vector Entropy

Comparison of the results with the modelled vectors in Fig. 5.3 shows a good level of agreement, especially with Fig. 6.10. The sub-pixel method shows the best agreement with Fig. 5.3 and the general motion, although the motion in the later frames is somewhat confused, and the large area of enhancement is outside of the plotted vectors. The early frames of the sequence capture the TOI moving through the centre of the two-celled pattern in Fig. 5.3, a features which can be seen in both Fig. 6.9 and Fig. 6.10. Fig. 5.3c–d share some important features with Fig. 6.10c–d, most notably the reversal in direction around the edges of the TOI.

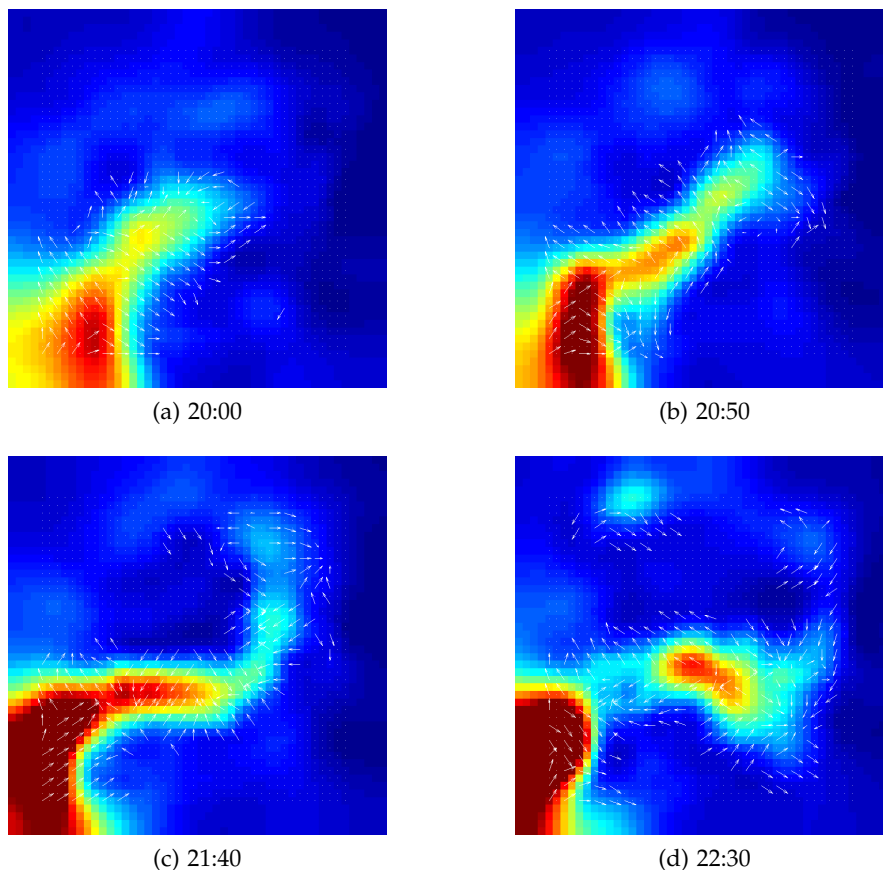


Figure 6.8: Example outputs creating using sub-pixel block matching with overlapping blocks, on the up-sampled TEC data sequence.

Table 6.2: Vector entropy (bits/vector) statistics for various kinds of block based motion estimation.

Stage	median H	mean H	σ
Standard	4.370	4.474	0.474
Overlapping	3.359	3.379	0.206
Overlapping with RL	2.612	2.623	0.130
Masked Overlapping	2.638	2.647	0.100
Masked Overlapping with RL	2.315	2.315	0.073
Sub-pixel	3.546	3.560	0.093
Sub-pixel with RL	3.494	3.491	0.095
Masked Sub-pixel	2.924	2.891	0.164
Masked Sub-pixel with RL	2.864	2.832	0.164

Whilst visual examination is a useful and important way of checking the consistency of motion fields, quantitative methods are able to provide additional validation and lend weight to conclusions drawn by eye. For this reason, table 6.2 shows average the vector-field entropy for different block matching methods. Entropy values (H) were calculated using the encoding scheme of the H.263 video compression scheme [see Rijkse, 1996], as applied by Evans [2006]. A lower entropy implies a smoother, more compressible motion field which is likely to be more realistic than a less smooth, more noisy field. Medians,

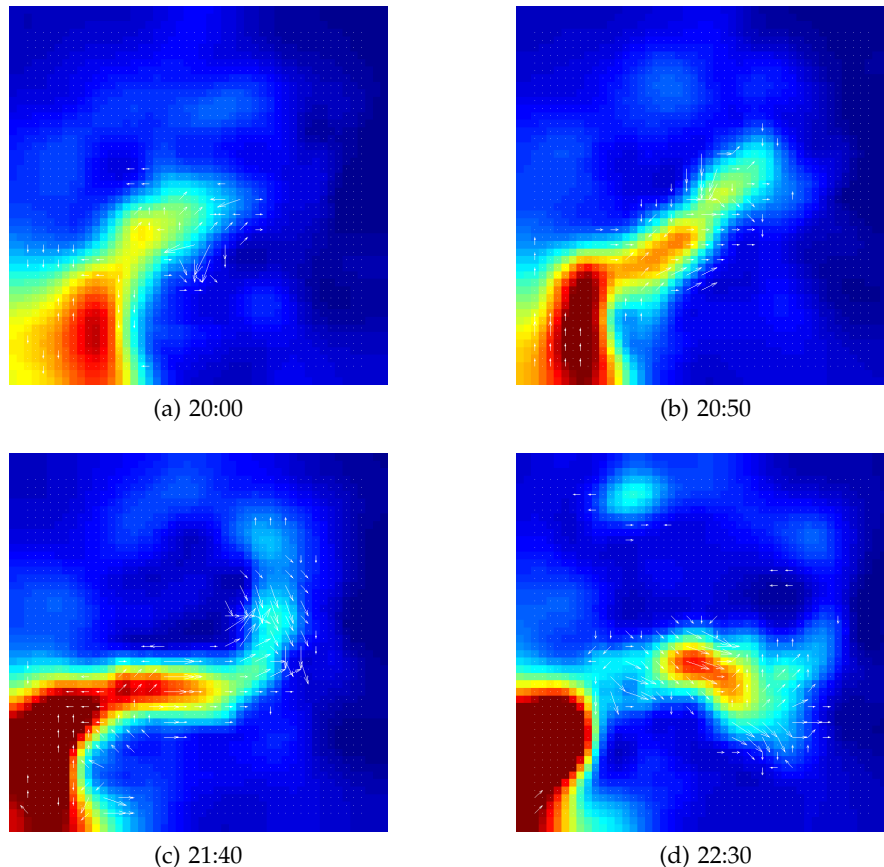


Figure 6.9: Example outputs created using block matching with overlapping blocks and relaxation labelling, followed by post-filtering, carried out on the up-sampled TEC data sequence.

means and standard deviations are shown, and were calculated after processing 35 frames. This table clearly shows the *added value* gained by using relaxation labelling, and also demonstrates that for this data sequence, sub-pixel estimation increases the entropy of the vector field. This is to be expected, as sub-pixel accuracy increases the number of states to be encoded. Visual examination of the results confirms that in general, the frames generated using sub-pixel estimation with overlapping blocks and relaxation labelling show smooth motion vector fields which appear to agree with the image motion. However, towards the end of the sequence the features become more complex in shape and the motion estimation methods are unable to capture the motion as effectively.

As seen in Table 6.2 the entropy values were lowered significantly by masking the output vector fields using block standard deviations, this reflects the additional smoothness gained by this, as well as the fact that a large number of the vectors are set to zero.¹

¹The masked entropy values were calculated by setting vector displacements in disabled blocks to zero, and then calculating the entropy of the entire field. Averages were calculated using the number of vectors in the mask.

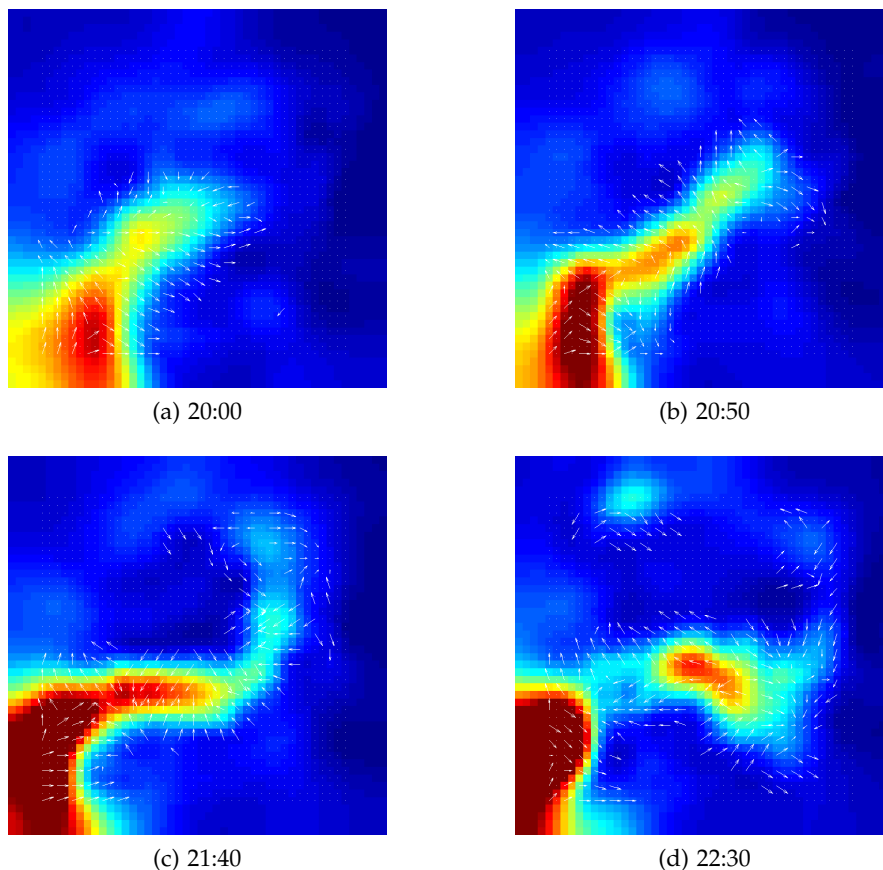


Figure 6.10: Example outputs created using sub-pixel block matching with overlapping blocks and relaxation labelling, followed by post-filtering, carried out on the up-sampled TEC data sequence.

6.2.2 Conclusions

A number of observations can be made regarding the data in Table 6.2. For example, the saving in entropy in moving from standard non-overlapping blocks to overlapping blocks with relaxation labelling is significant at 67%. The sub-pixel approach was found to increase the vector fields entropy by 25% after relaxation labelling, and compared to overlapping blocks with relaxation labelling. Sub-pixel estimation with relaxation results in entropy values 25% lower than the standard technique. In both the overlapping and sub-pixel case, adding relaxation labelling was found to lower entropy, although the improvement was much greater in the overlapping case (29%), than the sub-pixel case (2%). All tested methods were found to produce inconsistent vectors in the background parts of the images, which are very noisy and contain no discernible motion. This situation was largely rectified by selectively disabling blocks. Applying this technique to the estimated vectors was found to further lower entropy by approximately 21% in the sub-pixel case.

Visual comparisons of correlation-relaxation labelling estimated and model based vec-

tors show good agreement, with clear correlation in early parts of the sequence and some similarity in the more-confused later parts. These results suggest the sub-pixel correlation-relaxation labelling with overlapping blocks is a good method to apply to low resolution images containing non-rigid object motion.

Overall, sub-pixel correlation-relaxation labelling is a good choice here, despite the confused motion at the end of the sequence and the need for a vector field mask.

The next chapter will examine motion estimation using boundary tracking, to see if any improvements can be made to vector accuracy and aesthetics.

6.3 Conclusions

This chapter has described the application of two traditional methods of motion estimation to the TEC images described in Section 5.1. These methods are correlation-relaxation labelling and optical flow. Of these two techniques, correlation-relaxation labelling using overlapping blocks and sub-pixel estimation produces fields with the best agreement with the modelled vectors, and with lower entropies than fields produced using optical flow. This suggests that the correlation-relaxation labelling method is better suited, than gradient based optical flow, to estimating the motion of non-rigid features in low-resolution remotely sensed images.

The next chapter will examine a less traditional approach to motion estimation, based on matching shape boundaries, to see what further improvements can be made to the accuracy of the vector outputs.

Motion Estimation using Boundary Matching

Chapter 5 introduced the problem of the estimating motion of storm enhancements in low-resolution total electron content (TEC) imagery and presented various methods which are typically used to do this. Chapter 6 then examined template matching and correlation-relaxation labelling based methods, and also looked at optical flow techniques. This chapter continues along the same vein, by examining how motion can be estimated by matching shape boundaries. This somewhat less traditional approach has various useful features, such as the ability to resample boundaries, giving more vectors, thus increasing field resolution. One downside is the fact that vectors are estimated along shape boundaries, a problem can be mitigated by estimating warping transforms by using interpolation or regularisation, both of which can increase vector field density.

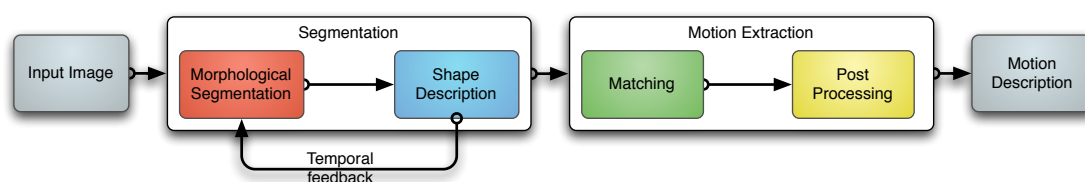


Figure 7.1: The processing pipeline for the curve matching process.

The motion estimation technique described here is based on a two stage approach, illustrated in Fig. 7.1. These two stages were introduced in Section 7.1, on page 100 and Section 5.7.2, on page 80 respectively.

This work was presented in SPIE Europe Remote Sensing 2008, and published as [Foster and Evans \[2008b\]](#).

7.1 Segmenting the Tongue of Ionisation

This section describes the contents of the ‘segmentation’ box in Fig. 7.1. The overall aim of the segmentation stage is to provide accurate descriptions of regions of interest with the frames in order to allow the second motion extraction stage to proceed.

The **TEC** images can be considered to be composed of three main components: regions of storm electron density (**SED**) corresponding to significant regional maxima [assumed to be the tongue of ionisation (**TOI**)]; noise and artefacts corresponding to other maxima (and minima), and the remainder of the image which can be considered background. The aim of the segmentation stage is to correctly discriminate between the **TOI** and the other image components. Once the images have been segmented, it is possible to extract information about the enhancements, including their boundaries which can then be used for estimating the motion of the **TOI**, and hence studying its evolution.

A simple technique for separating objects from background or clutter is *optimal thresholding* [[Nixon and Aguado, 2008](#)]. This approach is appropriate when the object of interest has a different range of values than the background, as is the case with the **TOI**. It is also designed for use on data with a bi-modal intensity distribution. However, histograms of the **TEC** data for the four images from Fig. 5.2 (on page 63), shown in Fig. 7.2, show that the **TEC** data is characterised by a distribution with only a single peak, which corresponds to the background value. The **TEC** values which correspond to the **TOI** occur in the tails (on the right hand side) of the illustrated distributions. This, coupled with the presence of noise, and the fact that the maximum **TEC** value changes over time means that simple thresholding is doomed to failure. Other segmentation methods, such as those based on image texture are, also inappropriate due to the low resolution and smoothness of the data.

To overcome these problems, a segmentation scheme based on mathematical morphology (**MM**) is proposed (see Section 5.7.1, page 75). Morphological filters have a number of advantages over traditional filters, most importantly shape preservation and idempotence. **MM** provides a number of tools for feature extraction and segmentation and many of these have been applied to remotely sensed data. For example, [Soille and Pesaresi \[2002\]](#) provide a topical review of **MM** applications in remote sensing, ranging from segmentation to skeletonisation. [Pesaresi and Benediktsson \[2001\]](#) propose a morphological segmentation method using the derivative of the morphological profile with

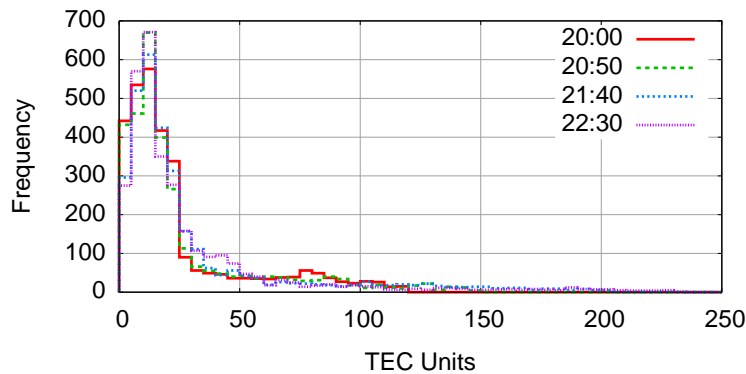


Figure 7.2: Histograms showing the distribution of TEC values in the data. The peaks on the left represent background data (low values), and the area on the right represents both noise and the TOI.

residuals from opening and closing by reconstruction. This method is extended to multiple scales by using increasingly large structuring elements. More recently Akçay and Aksoy [2007, 2008] designed a hierarchical MM segmentation technique for remotely sensed imagery. This method creates morphological profiles by applying structuring elements of increasing size to each spectral band. Image patches extracted at different scales are then merged to create meaningful objects using spectral homogeneity and connectivity to optimise the groupings. More information on attribute morphology can be found in Section 5.7.1.

Whilst the segmentation schemes proposed in Soille and Pesaresi [2002], Pesaresi and Benediktsson [2001], Akçay and Aksoy [2007, 2008] were developed for application to very-high resolution (VHR) images, the images in this study are characterised by extremely low resolution. In these images, the lack of texture and small size mean that multi-scale segmentation methods are unsuitable. To help overcome these problems a segmentation scheme based on attribute morphology is proposed. Attribute morphology is the extension of traditional greyscale morphology to use more general feature information in the opening and closing operations. Typical attributes include area (the most commonly used feature), moments, orientation, contrast and many other measurable properties of image structure. Here, attribute morphology is advantageous when compared with standard morphology as it does not require the use of structuring elements, all of which would be large when compared to the dimensions of the images, and so introduce significant regions of noise. It also allows the filtering action to be finely controlled by attributes other than shape and size, producing segmentation results that can be tuned according to the properties of the features under consideration. Another useful property is that of shape preservation, a property shared with reconstruction based methods.

Contrast was chosen as the most suitable attribute for segmenting the TEC data due to its low sensitivity to gradient relative to other attributes, such as area. Whilst the peak

SED value within the **TOI** shows considerable variation over the duration of the storm, the background **TEC** value is much more stable. Performing an attribute closing with a suitable contrast limit essentially flood-fills any basins with a contrast value less than the chosen attribute limit.

In order to choose a suitable contrast value with which to operate the morphological filter, ground-truths were generated by segmenting the images by hand. These were necessary due to the noisy nature of the frames which meant that inter-frame differences were not necessarily consistent and the sudden appearance and disappearance of small features was common. This process gave rise to a set of binary images, one for each frame, in which the salient regions were marked.

To find the contrast value for each frame which gave rise to the results most closely matching the corresponding ground-truth images, the data were filtered using a range of contrasts, and the outputs compared to the ground truths. The optimal value found in this fashion was then used as the starting point for the segmentations. Fig. 7.3 shows two image maps illustrating the best contrast parameter for each frame, using two different criteria – area (darker means a lower difference between the test and hand-segmented frames), and correlation (brighter means a higher correlation coefficient between the test and hand-segmented frames). In both cases, similar behaviour can be seen; on average, the contrast value which gave rise to results closest to the hand-segmentations was 33 **TEC** units. Fig. 7.3 shows that the performance of the attribute-based segmentation is relatively insensitive to the choice of contrast value. Therefore, although the contrast value here was selected using hand-segmented **TEC** images, the availability of more data would enable the starting contrast value to be estimated using training data with little overall loss in performance.

To improve the temporal consistency of the segmentation results over the image sequence, a temporal feedback system was used, along with the ground-truth derived contrast value. This system operated by performing an initial segmentation using a contrast closure, followed by a small-scale area-opening to remove small peaks which were considered clutter. This was then thresholded to give a binary mask identifying the **TOI**. The feedback system operated by calculating the change in foreground area between each frame and the frame preceding it, and then increasing or decreasing (as appropriate) the contrast level in small increments if the area differed by more than 10%. If the contrast was altered, the image was re-segmented. This process was repeated in an iterative fashion, until the segmented area was stable or until 10 iterations had occurred. This helped ensure the correspondence of objects between frames, leading to a smoother motion description. A flow diagram illustrating this process is shown in Fig. 7.4.

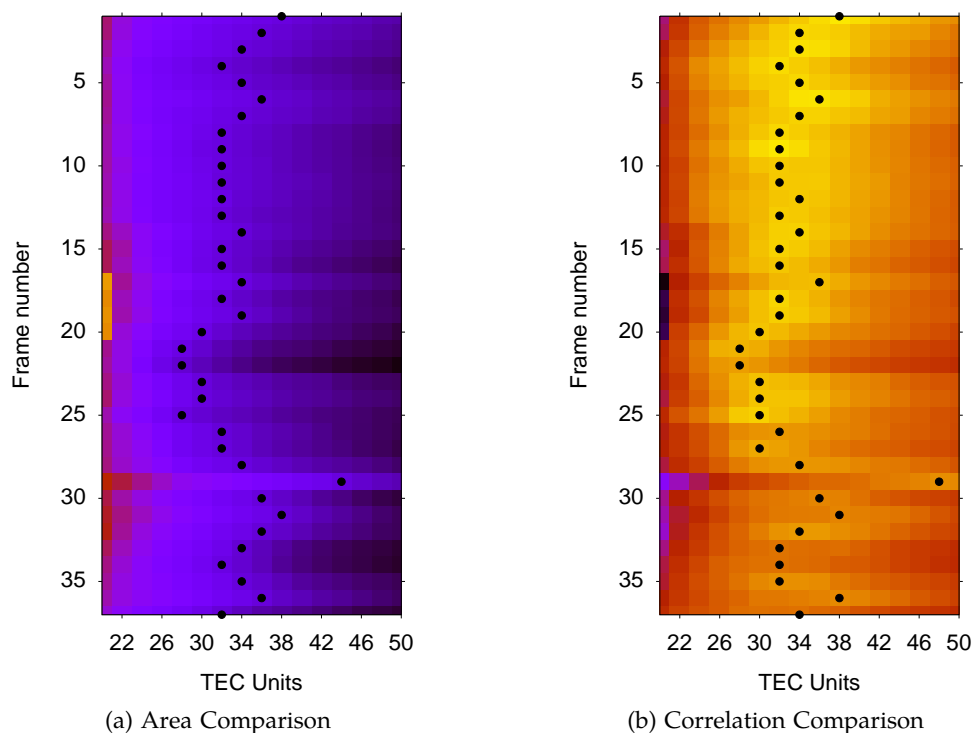


Figure 7.3: Comparisons of area difference (a) and correlation (b) between frames segmented at various contrasts, and hand segmented frames. In both cases the contrast with the closest matches to the hand segmented frames was 33 TEC Units.

The above process yielded results which are similar to the ground truth images whilst maintaining area consistency between frames. However, in a few cases the segmentation was too generous, meaning that some areas which were separated in the hand-segmented images were left joined by the feedback process. This occurred when saddle-shaped structures joined features which would otherwise be separate. To mitigate this problem, a further stage was added to the process. By inverting the input images, masked by the segmented images, and then applying the watershed transform, thin saddle points joining areas were severed, as demonstrated in Fig. 7.5. The segmentation results for the images from Fig. 5.2 are shown in Fig. 7.6. To obtain final TOI representations in a form which is smoothed and can be re-sampled, the image boundaries were traced and then converted to a smoothing-spline representation.

7.2 Motion Estimation by Boundary Context Matching

After boundaries have been extracted, the problem of estimating motion reduces to one of shape correspondence, i.e. given the boundaries for the TOI in two adjacent frames,

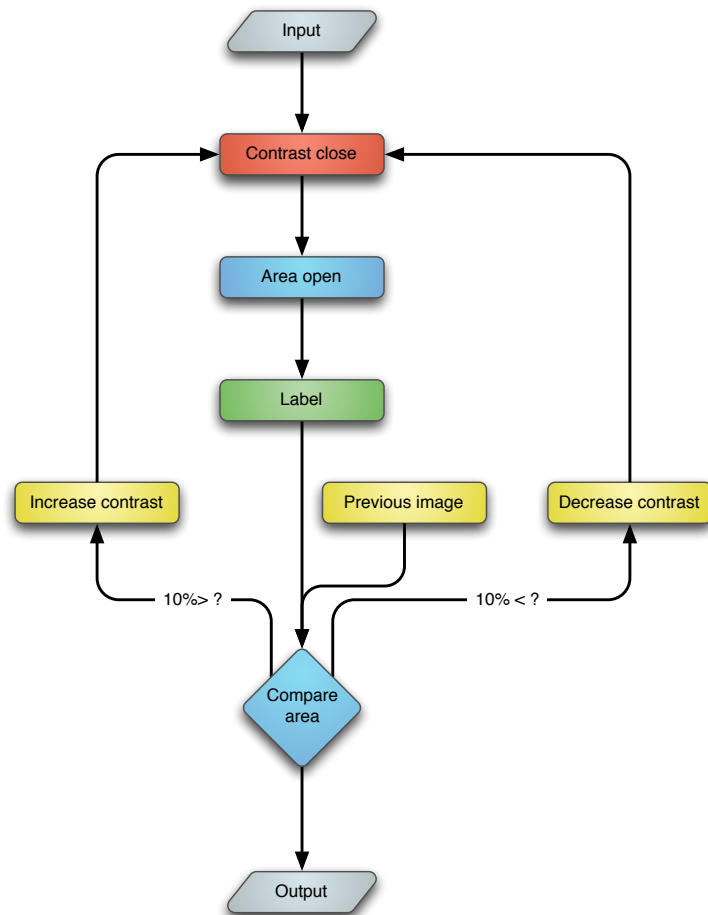
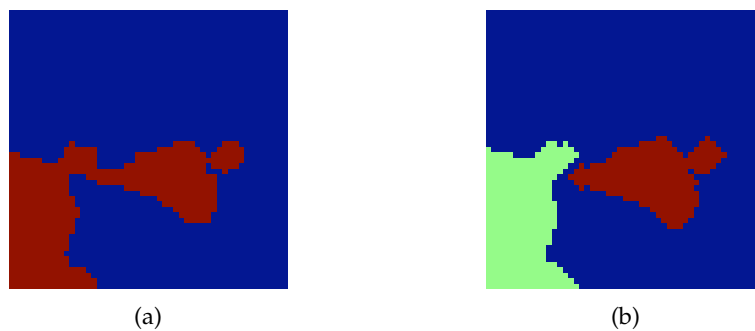


Figure 7.4: Segmentation process showing area based feedback.

Figure 7.5: Reducing under-segmentation of **SED** islands using the watershed transform. Example TEC image from 22:35 UTC on 30th August 2003 (a) before and (b) after watershed transform post-processing.

the aim is to discover how one shape must be deformed in order to produce the other. The end result of this process is a series of vectors showing the correspondences between samples on each boundary, which can then be taken to represent cross-frame motion.

The method used to estimate the cross-frame motion was shape context (**SC**) matching, a technique for measuring similarity between shapes, using their boundaries **Belongie**

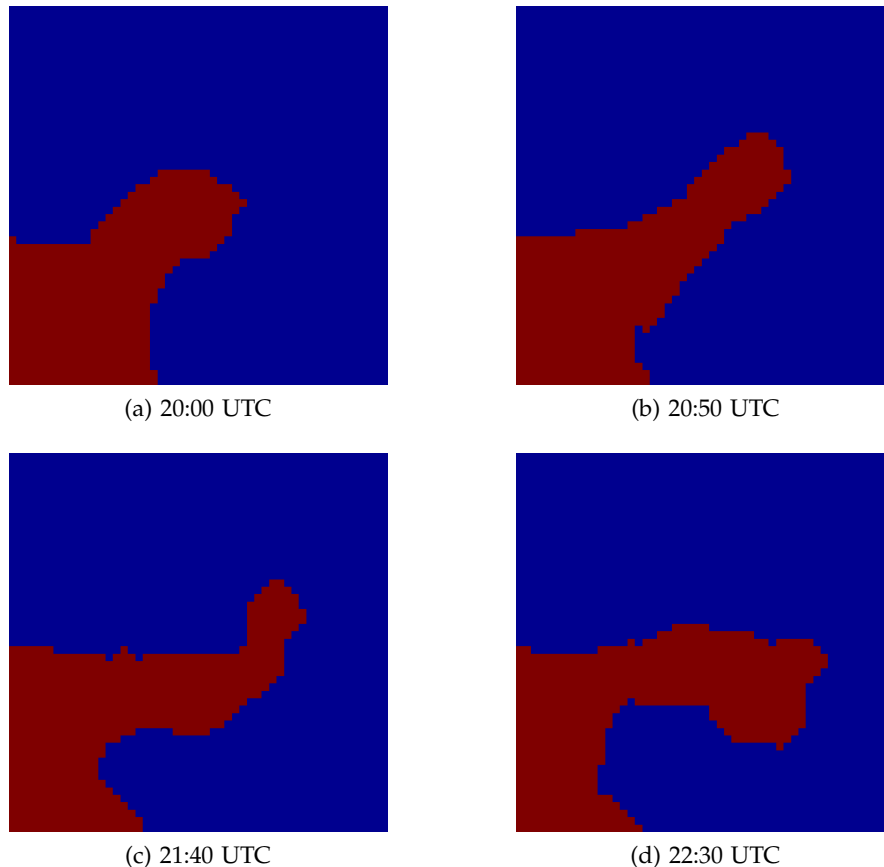


Figure 7.6: Images from Fig. 5.2 segmented using morphological segmentation process described in the text.

et al. [Apr 2002]. This process was fully described in Section 5.7.2 and was applied to the smoothed spline boundaries from the segmentation method described in the previous section. This resulted in images such as those shown in Fig. 7.9.

Applying SC based motion estimation to the segmented boundaries provides estimated vectors at the shape boundaries in each frame. In general, this proved to be successful and as shown in Fig. 7.7, it captured the motion in the the beginning of the sequence particularly well. In this section of the data-set, the motion is highly constrained, and the TOI is easily distinguished from the background. However, at other points in the sequence this is not always the case.

As described in section 5.1, on page 60, the image sequence corresponds to a patch over the polar cap. One consequence of this is that one side of the image is always in sunlight, and one is in darkness. Because of the nature of the ionosphere, the side in sunlight is continually injected with electrons and ions, and the side in darkness undergoes depletion due to recombination effects. This results in the TOI being drained at its tip, in a fashion similar to the snout of a glacier, where ice melts, and drains away. If the rate of melting increases, the snout will appear to retreat whilst water will always flow

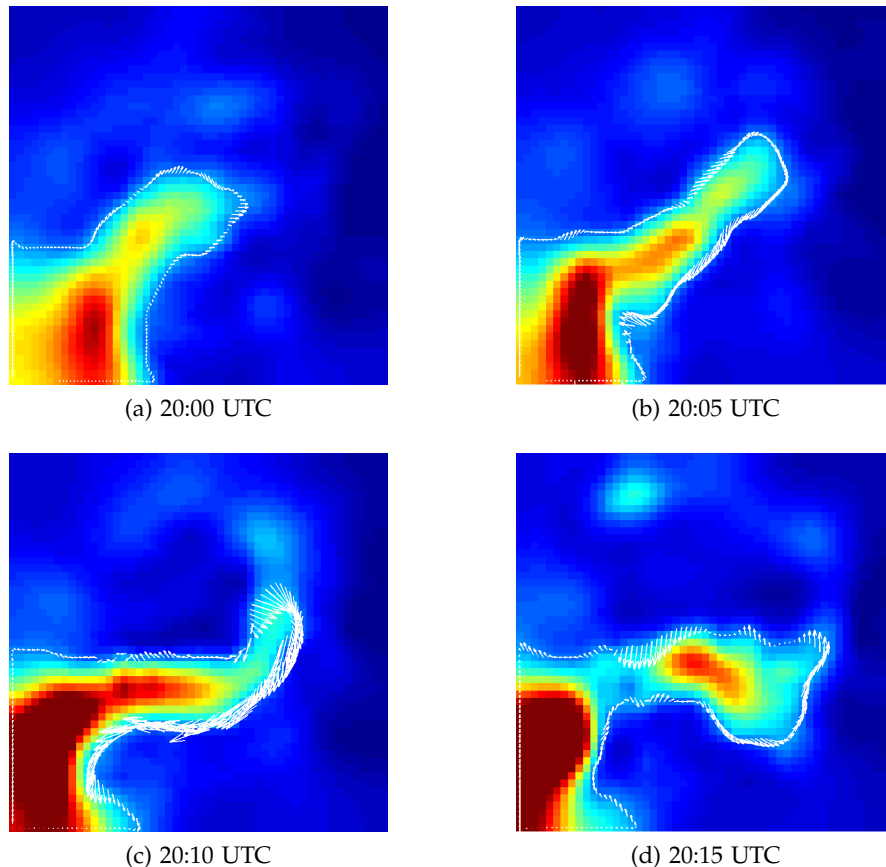


Figure 7.7: Example false-colour frames from early in the sequence. Vectors are the raw outputs from the segmentation and shape context matching method.

away. If only the position of the snout was being measured, the glacier would appear to be flowing backwards, which is clearly never the case. This effect is illustrated in Fig. 7.8.

This problem occurs towards the end of the image sequence under examination. Fig. 7.9 shows an example of this problem occurring due to a slight drop in value at the very tip of the TOI. Unfortunately, these retreating boundaries cannot be detected by simply examining the vector directions relative to other vectors in the same frame, as frames where this problem manifest itself tend to have very noisy vectors in general. For this reason, several techniques to detect and replace these retreating vectors were evaluated for suitability, and effectiveness.

Tested detection methods included:

- Marking vectors by thresholding the magnitude of the current frame's vectors was examined as a simple method of detecting anomalies. However because no other

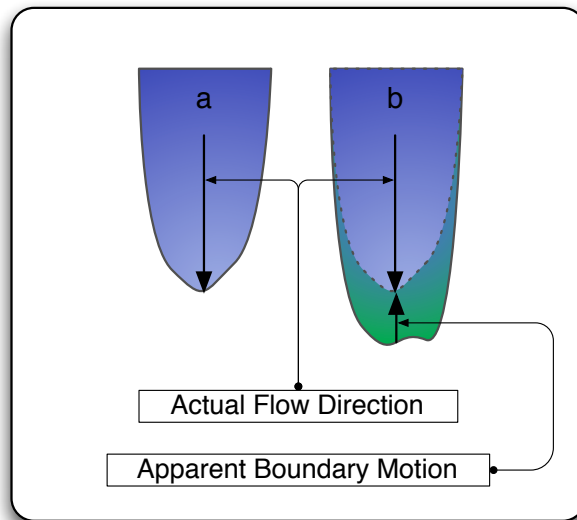


Figure 7.8: A diagram showing how a change in depletion rate can alter the apparent direction of flow.

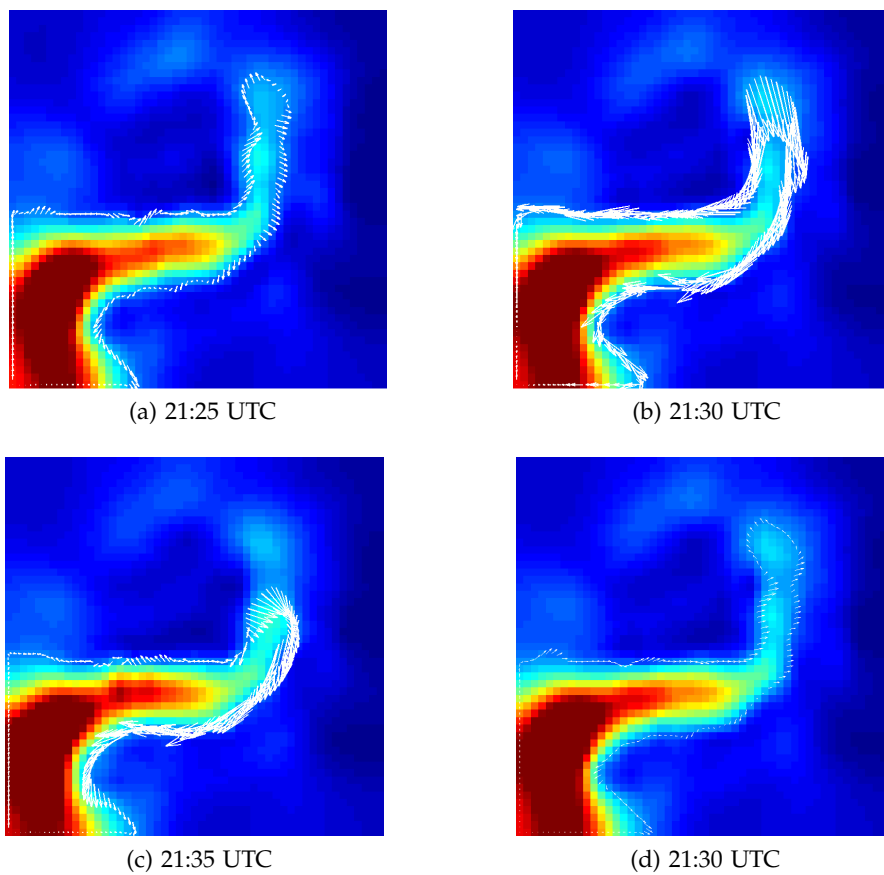


Figure 7.9: Example frames illustrating the retreating snout problem. (a)–(c) Frames from before, during and after the problem occurs, respectively, and (d) the result of detecting the depletion and replacing all the vectors with vectors interpolated from those from the previous frame.

frames are examined it is unable to detect retreating vectors. For this reason it was deemed unsuitable.

- Marking retreating vectors by first subtracting each given frame's vectors from the previous frame's vectors. These are interpolated to the points at which the current vectors lie. These new difference vectors are then filtered to remove those outside of a 95% confidence limit (two standard deviations either side of the mean) for magnitude, and finally thresholded. This method detects vectors which are significantly different from those in the previous frame, but is very sensitive to the threshold parameter chosen as the histograms of displacements vectors tend to be very flat.
- Marking vectors in a manor similar to the above method, but making use of angle instead of magnitude. This method was found to be unreliable because of the wide variety of vector directions between frames.
- Marking entire frames based on the mean and standard deviation of the vectors differenced with those from the previous frame. Using a combination of a mean value of 5, and standard deviation of 3 was found to be effective.

Following detection of the retreating vectors, the selected vectors must be replaced. Vector replacement methods which were tested included:

- Replacing the marked vectors with the vector-median [Astola et al., 1990] of the remaining vectors. As the frames containing the retreating boundary problem tend to contain noisy vectors, this was found to be unpredictable.
- Replacing the detected vectors with new vectors generated using an affine model of the previous frame's vectors. Whilst this tends to give more suitable magnitudes than the previous method, the affine fit often ends up with vectors which seem to point *wrong* direction.
- Replacing the detected vectors with new vectors generated by interpolating the previous frame's vectors onto the marked vector positions. This was carried out using both standard interpolation and regularisation. This gives good results provided the previous frame's vectors are reasonable. For this reason, processing should be done on a frame by frame basis.
- Replacing all of the vectors in the frame with the previous frame's interpolated vectors. This is an effective way of getting smooth vectors over the entire boundary, provided the previous frame's vectors are smooth.

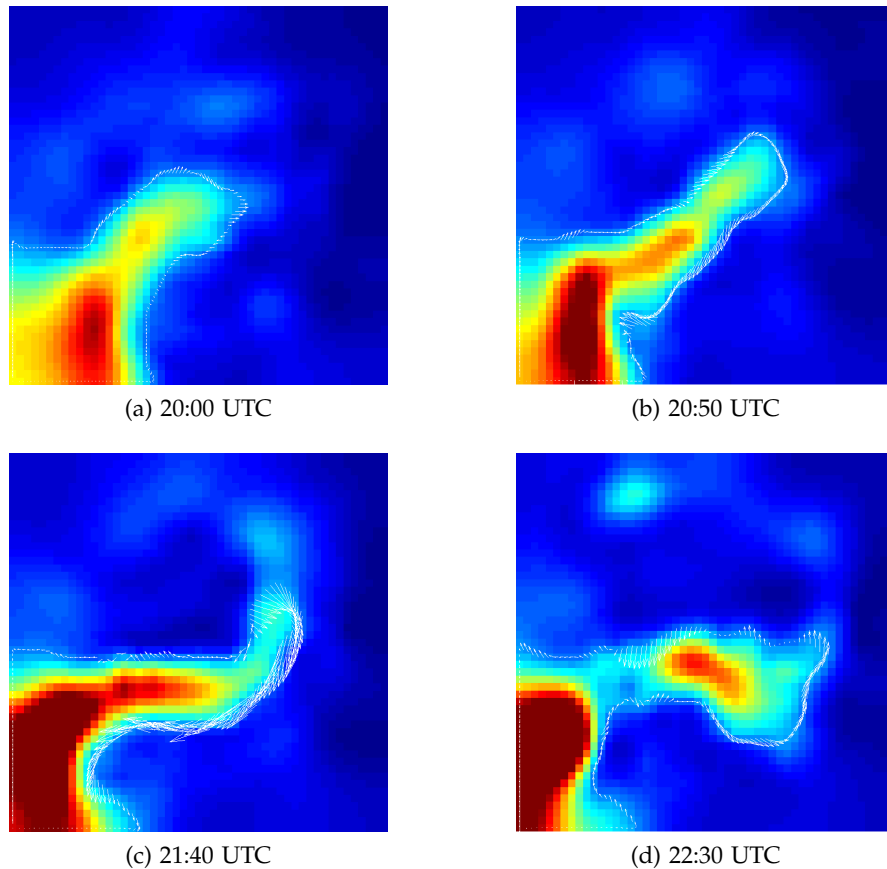


Figure 7.10: Example false-colour frames from Fig. 5.2. Vectors are shown after post-processing by replacing all vectors in frames with detected retreating areas.

Of the detection and replacement methods tested, replacing all vectors based on measuring vector magnitude statistics was found to be most effective due to the overall smoothness of the output. From an aesthetic point of view, it is advantageous because replacing the entire frame does not result in the discontinuities which are present in the other partial replacement methods at the boundaries of regions where vectors are being replaced.

Fig. 7.10 shows example frames with boundary motion vectors plotted at the boundary points (where they were estimated). The next section discusses methods of increasing the density and smoothness of the vector fields produced using this technique.

7.3 Increasing Field Density

As an additional processing step, the output vectors can be interpolated to cover the area within the boundaries (and extrapolated to cover the background, if necessary). This could be very useful for interpretation, as large scale trends are easier to observe

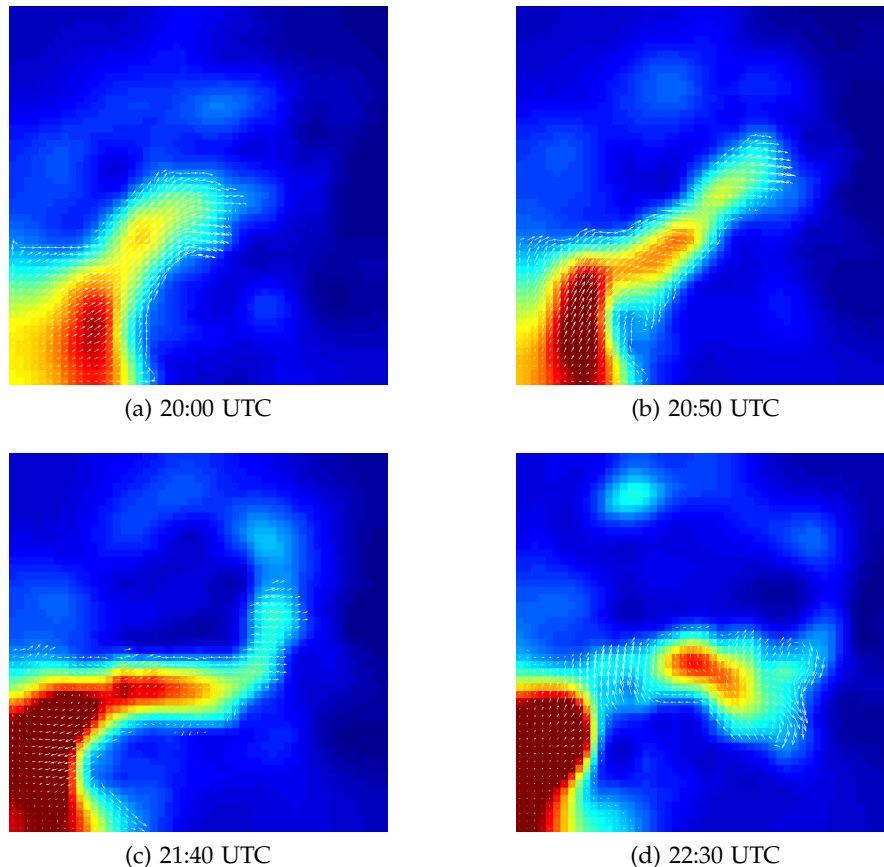


Figure 7.11: Frames from the TEC sequence showing vectors which have been interpolated throughout segmented objects.

than those occurring only on boundaries.

Fig. 7.11 show some example frames from the TEC sequence which have been interpolated using radial basis function (RBF) thin plate spline (TPS) interpolation. Other interpolation methods could also be used, to similar effect, although using linear methods results in far less smooth outputs.

Regularisation can also be used to smooth these vector fields, thus removing noisy and inconsistent vectors. This is similar to the use of relaxation-labelling (see Section 5.6.4, on page 72), although it does not pick from a discrete set of available vectors, but uses fitted smooth surfaces. The regularisation is also part of the ‘interpolation’ method, rather than a add-on step. Fig. 7.12 shows the how changing the scale-dependent regularisation parameter, λ changes the smoothness of a surface being reconstructed. Fig. 7.13 then shows the same frames as Fig. 7.11, but regularised using a λ of 0.5. This produces smoothed vectors, with a tendency for the field to be less curved than in the non-regularised case.

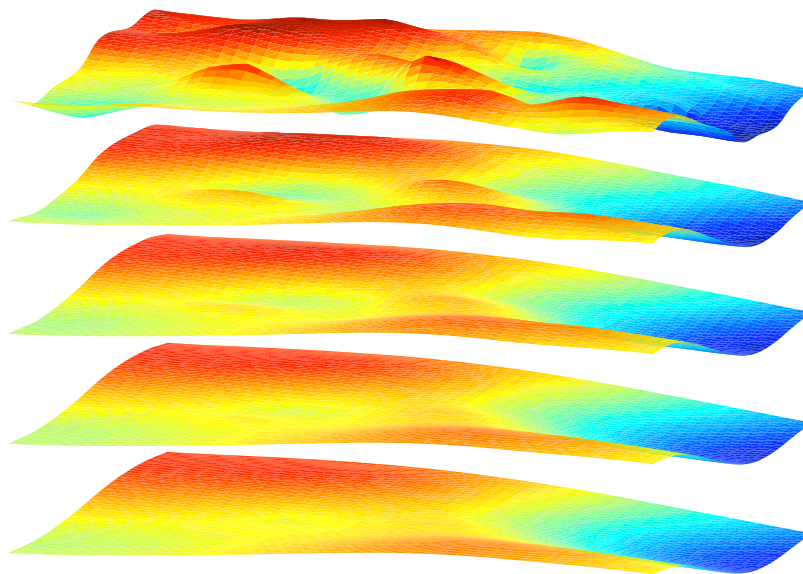


Figure 7.12: The effect of adjusting the regularisation parameter (λ) in steps of 0.25, starting at zero (top), and ending at one (bottom).

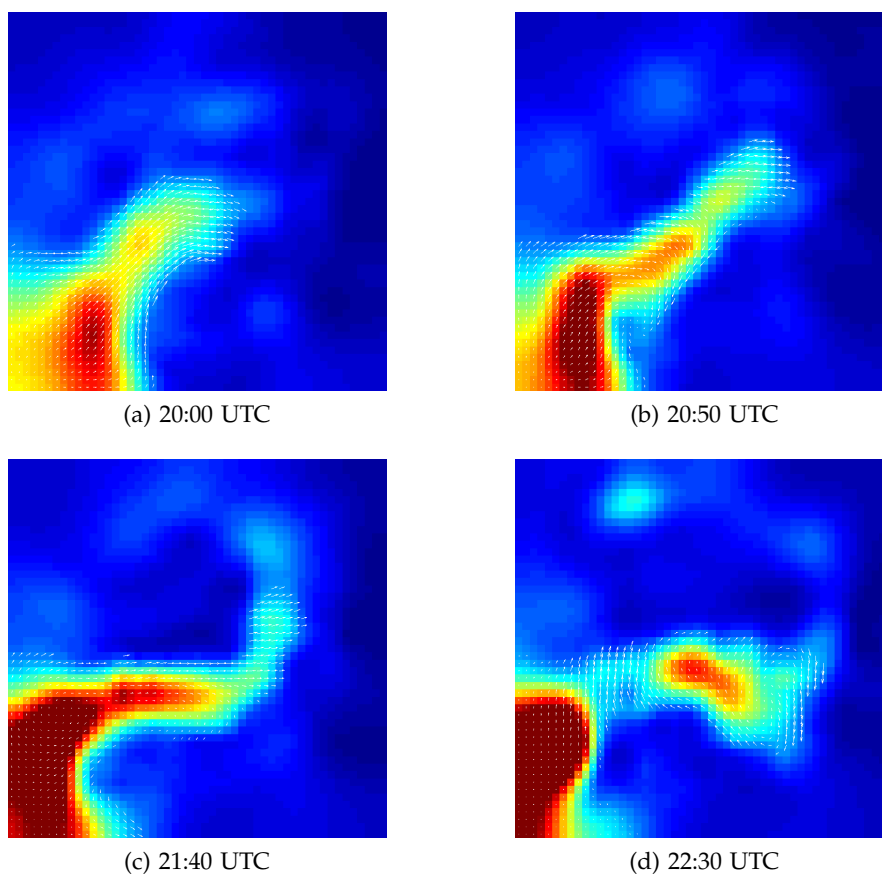


Figure 7.13: Frames from the TEC sequence showing vectors which have been in-filled throughout segmented objects using regularisation, with $\lambda = 0.5$.

Comparing Figs. 7.11 and 7.13 with Fig. 5.3 show similar results to the other tested motion estimation techniques. In the beginning of the sequence the behaviours are fairly similar, with a tendency for motion towards the top-right of the images. As the sequence progresses, however, for the right-hand side of Fig. 5.3 feature to appears to show clockwise rotation – a feature which can also be seen in both Figs. 7.11 and 7.13. This suggests that this technique is well suited for capturing large-scale movement, including rotation, something for which the other tested techniques (see Sections 6.2 and 5.5) have proven ill-equipped.

7.4 Conclusions

A new method for segmenting and estimating the motion of features in TEC images of the polar ionosphere during storm conditions has been presented. Unlike other applications of greyscale morphology to remotely sensed data, where the current trend is towards using VHR imagery, the application here is characterised by very low resolution images. Other major challenges result from the non-rigid motion of the TOI and the recombination effects. To overcome these problems an attribute morphology-based segmentation scheme has been proposed. This method uses the contrast attribute and produces segmentation results that are robust to changes in the maximum SED value. In addition, temporal feedback is used to improve the consistency of the segmentation results by automatically varying the attribute limit through the sequence. The results achieved by this approach are temporally consistent and show a good correspondence with hand-segmented ground truths.

To estimate the motion of the TOI an approach was developed which applied shape contexts to track boundary changes in the segmented TOI between successive frames. This method makes use of the shape contexts of smoothed boundaries (with equal numbers of samples) to estimate correspondences which can then be converted into motion vectors. Although this was found to be an effective approach when the TOI feature was stable, it gave erroneous results when the TOI boundary appears to retreat as a result of electron depletion. To mitigate this effect, a post-processing step was proposed in which the mean magnitude and standard deviations of a frame's vectors were compared with those from the previous frame. When certain thresholds were exceeded, the frame's vectors were entirely replaced by vectors from the previous frame, interpolated to the correct positions.

Finally, interpolation and regularisation were used to increase the density and smoothness of the output vector fields, in order to make observations easier. Comparison with modelled vectors suggest that this technique captured large scale motion and rotation

7.4. CONCLUSIONS

very well, especially when coupled with this vector up-sampling method.

Suggestions for Further Work

This chapter briefly list some ideas for further work that could be carried out. Projects related to interpolation re listed first, followed by projects relating to motion estimation.

8.0.1 Interpolation and TEC Mapping

A large number of older interpolation methods exist which have not been tested in recent studies [e.g. [Akima, 1978](#), [Shepard, 1968](#), [Ripley, 2004](#)]. Many of these are also implemented in languages such as FORTRAN (or older languages). Porting these methods to C or C++ and then including them in the studies in Chapter 3 would be a useful benchmark. It would show how well these methods can cope with highly-sparse data (something which seems to have only been given attention recently), as well as how well they compare to more modern methods.

Another useful advancement would be improving the speed of the radial basis function (RBF) interpolation methods by implementing domain decomposition. This was briefly mentioned in Chapter 3. The result in Chapter 4 show that these methods are largely free from the artefacts which manifest in triangulation based methods, meaning that a very fast implementation would be an excellent tool for day-to-day interpolation.

Work on examining how sampling patterns alter interpolation outputs should also prove very interesting and informative. This should concentrate on the spatial frequency content that can be reconstructed for different sampling methods and sparsities, and might pave the way for a universal description of the limits of reconstruction quality at differing sparsity. A good sampling point for this work would be [Ripley \[2004\]](#) and

references therein.

Finally, with regards to total electron content (TEC) mapping: these maps do not *necessarily* need to be created using interpolation. An evaluation of regularisation methods (such as those described in Chapter 7) would be a useful evolution of the work on evaluating interpolation performance. The resilience to noise would also be an excellent property of these methods.

8.0.2 Motion Estimation

Future work in the area of motion estimation of storm enhancements should focus on the extension of these techniques into an unsupervised system which does not require the manual creation of ground truths. This will require more data which have been gathered during storms.

A suitable training set could then be created which would allow contrast attributes to be chosen. Extending this to be proportional to the maximum data value would be more general and allow the use of the same parameter on many different storms.

Independent data sources such as vector data from the SuperDARN or EISCAT radars could also be used both for validation of results and perhaps more interestingly, the production of improved motion fields by combining them estimates of boundary positions.

Implementation of a tracking system, using for example, a Kalman filter [Hlavac et al., 1999] would be a useful extension of the work, which could allow for forecasting of storms motion they progress. These results could also be useful for scientific reasons.

Exhaustive testing of the different motion estimation methods examined here could be carried out by simulating data with known motion, thereby proving a ground truth. This data would have to be generated in such a way as to ensure realism, whilst ensuring confidence in the estimated vectors. Examining warping methods [e.g. Wolberg, 1994] which could be modified for image generation would be a good first-step towards implementing this study. Although it should be noted that simple warping models, such as the affine model, can not appropriately capture the complex motion of the tongue of ionisation (TOI).

References

- S. T. Acton. Fast algorithms for area morphology. *Digital Signal Processing*, 11(3):187–203, 2001.
- S. T. Acton and D. P. Mukherjee. Scale Space Classification Using Area Morphology. *IEEE Trans. Image Process.*, 9(4):623, 2000.
- H. G. Akçay and S. Aksoy. Automatic detection of geospatial objects using multiple hierarchical segmentations. *IEEE Trans. Geosci. Remote Sens*, 46(7):2097–2111, July 2008. ISSN 0196-2892. doi: 10.1109/TGRS.2008.916644.
- H. G. Akçay and S. Aksoy. Morphological segmentation of urban structures. *Urban Remote Sensing Joint Event, 2007*, pages 1–6, April 2007. doi: 10.1109/URS.2007.371765.
- H. Akima. A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points. *ACM Transactions on Mathematical Software*, 4(2):148–159, 1978.
- II Alexeev, ES Belenkaya, S.Y. Bobrovnikov, VV Kalegaev, JA Cumnock, and LG Blomberg. Magnetopause mapping to the ionosphere for northward IMF. *Ann. Geophys*, 25:2615–2625, 2007.
- A. Almansa and T. Lindeberg. Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection. *IEEE Trans. Image Process.*, 9(12):2027–2042, 2000.
- P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.
- F. Arikan, O. Arikan, and C. B. Erol. Regularized estimation of TEC from GPS data for certain midlatitude stations and comparison with the IRI model. *Advances in Space Research*, 39(5):867–874, 2007.
- J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4), 1990.

REFERENCES

- R. K. Beatson, W. A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal on Scientific Computing*, 22:1717–1740, 2001.
- TJ Bellerby. High-Resolution 2-D Cloud-Top Advection From Geostationary Satellite Imagery. *IEEE Trans. Geosci. Remote Sens*, 44(12):3639–3648, 2006.
- S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr 2002. ISSN 0162-8828. doi: 10.1109/34.993558.
- D. Bhat, N. Nayar, and K. Shree. Ordinal measures for image correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(4):415–423, April 1998.
- D. Bilitza. International Reference Ionosphere 2000. *Radio Sci*, 36(2):261–275, 2001.
- J. Blanch, T. Walter, and P. Enge. Application of spatial statistics to ionosphere estimation for WAAS. *Proceedings of ION NTM*, 2002.
- A. Boucher, K. C. Seto, and A. G. Journel. A novel method for mapping land cover changes: Incorporating time and space with geostatistics. *IEEE Trans. Geosci. Remote Sens*, 44(11):3427–3435, 2006. ISSN 0196-2892.
- E. Brockmann and W. Gurtner. Combination of GPS solutions for densification of the European network: Concepts and results derived from 5 European associated analysis centers of the IGS. *EUREF workshop, Ankara, May, 1996*.
- J. C. Carr, W. R. Fright, and R. K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Trans. Med. Imag.*, 16:96–107, 1997. ISSN 0278-0062.
- Center for Orbit Determination in Europe, Astronomisches Institut Universität Bern. Differential GPS code biases (DCBs), 2005. URL http://cmslive2.unibe.ch/unibe/philnat/aiub/content/research/gnss/code___research/index_eng.html.
- M. Chen, W. Shi, P. Xie, V. Silva, V. E. Kousky, R. W. Higgins, and J. E. Janowiak. Assessing objective techniques for gauge-based analyses of global daily precipitation. *J. Geophys. Res*, 113, 2008.
- S. W. H. Cowley. Excitation of Flow in the Earth’s Magnetosphere-Ionosphere System: Observations by Incoherent-Scatter Radar. In J. Moen, A. Egeland, and M. Lockwood, editors, *Polar Cap Boundary Phenomena*, 1998.
- N. Cressie. The origins of kriging. *Mathematical Geology*, 2:239–252, 1990.
- N. A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, Inc., 1991. ISBN 0-471-84336-9.

REFERENCES

- F. de Jong, L. J. van Vliet, and P. P. Jonker. Gradient estimation in uncertain data. 1998.
- S. Dransfeld, G. Larnicol, and P.-Y. Le Traon. The potential of the maximum cross-correlation technique to estimate surface currents from thermal AVHRR global area coverage data. *Geoscience and Remote Sensing Letters, IEEE*, 3(4):508–511, 2006. ISSN 1545-598X.
- R. S. J. Estepar, M. Martin-Fernandez, C. Alberola-Lopez, J. Ellsmere, R. Kikinis, and C.-F. Westin. Freehand ultrasound reconstruction based on roi prior modeling and normalized convolution. *Lecture Notes In Computer Science*, pages 382–390, 2003.
- A. N. Evans. Cloud tracking using ordinal measures and relaxation labelling. *IEEE International Geoscience and Remote Sensing Symposium*, 2:1259–1261, 1999.
- A. N. Evans. Cloud motion analysis using multichannel correlation-relaxation labeling. *IEEE Geoscience and Remote Sensing Letters*, 3(3):392–396, July 2006.
- A. N. Evans. Glacier surface motion computation from digital image sequences. *IEEE Trans. Geosci. Remote Sens*, 38(2), 2000a.
- A. N. Evans. On the use of ordinal measures for cloud tracking. *International Journal of Remote Sensing*, 21(9):1939–1944, 2000b.
- G. Farneback. *Polynomial Expansion for Orientation and Motion Estimation*. PhD thesis, Linköping University, Sweden, 2002.
- J. C. Foster, A. J. Coster, P. J. Erickson, J. M. Holt, F. D. Lind, W. Rideout, M. McCready, A. van Eyken, R. J. Barnes, R. A. Greenwald, et al. Multiradar observations of the polar tongue of ionization. *Journal of Geophysical Research*, 110, 2005.
- M. P. Foster and A. N. Evans. An evaluation of interpolation techniques for reconstructing ionospheric TEC maps. *IEEE Trans. Geosci. Remote Sens*, 46(7):2153–2164, July 2008a. ISSN 0196-2892. doi: 10.1109/TGRS.2008.916642.
- Matthew P. Foster and Adrian N. Evans. Segmentation and tracking of ionospheric storm enhancements. volume 7109, page 71090O. SPIE, 2008b. doi: 10.1117/12.799806. URL <http://link.aip.org/link/?PSI/7109/710900/1>.
- R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.
- J. Fried and S. Zietz. Curve fitting by spline and akima methods: possibility of interpolation error and its suppression. *Physics in Medicine and Biology*, 18(4):550–558, 1973.
- M. Gasca and T. Sauer. On the history of multivariate polynomial interpolation. *Journal of Computational and Applied Mathematics*, 122(1-2):23–35, 2000.

REFERENCES

- J. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Trans. Image Process.*, 12(8):938–943, August 2002.
- M. Gianinetto and P. Villa. Rapid response flood assessment using minimum noise fraction and composed spline interpolation. *IEEE Trans. Geosci. Remote Sens.*, 45(10):3204–3211, Oct. 2007. ISSN 0196-2892. doi: 10.1109/TGRS.2007.895414.
- R. C. Gonzalez and R. E. Woods. *Digital Image Processing 2/E*. Prentice Hall, 2001. ISBN 0-13-094650-8.
- M. Hernandez-Pajares, J. M. J. Zornoza, J. S. Subirana, R. Farnworth, and S. Soley. EG-NOS test bed ionospheric corrections under the October and November 2003 storms. *IEEE Trans. Geosci. Remote Sens.*, 43(10):2283–2293, 2005.
- V. Hlavac, M. Sonka, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing Company, 2 edition, 1999. ISBN 0-534-95393-X.
- B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS Theory and Practice 5/e*. Springer Wein New York, 2001. ISBN 3-211-83534-2.
- J. Karvanen and A. Cichocki. Measuring sparseness of noisy signals. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, Nara, Japan, April 2003.
- M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, 37(3):362–385, 1987.
- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- P. N. Klein, T. B. Sebastian, and B. B. Kimia. Shape matching using edit-distance: An implementation. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01)*, pages 781–790, New York, January 2001. ACM Press.
- H. Knutsson and C.-F. Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 515–523, New York City, USA, June 1993.
- R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 2(12):1137–1143, 1995.
- A. Leick. *GPS Satellite Surveying 2/e*. John Wiley & Sons, Inc., 1995. ISBN 0-471-30626-6.
- M. Liao, T. Wang, L. Lu, W. Zhouzhou, and D. Li. Reconstruction of DEMs from ERS-1/2 tandem data in mountainous area facilitated by SRTM data. *IEEE Trans. Geosci. Remote Sens.*, 45(7):2325–2335, July 2007. ISSN 0196-2892. doi: 10.1109/TGRS.2007.896546.

REFERENCES

- W. A. Light. *Approximation Theory, Spline Functions and Applications*, chapter Some aspects of radial basis function approximation, pages 163–190. Kluwer Academic Publishers, Boston, MA, 1992.
- H. Ling and K. Okada. An efficient earth mover’s distance algorithm for robust histogram comparison. *Pattern Analysis and Machine Intelligence, IEEE Trans.*, 29(5):840–853, May 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1058.
- H. Ling and K. Okada. Diffusion distance for histogram comparison. In *Proc. CVPR*, pages 246–253, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: <http://dx.doi.org/10.1109/CVPR.2006.99>.
- D. G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 2:1150–1157, 1999.
- E. Maeland. On the comparison of interpolation methods. *IEEE Trans. Med. Imag.*, 7(3): 213–217, Sep 1988. ISSN 1558-254X. doi: 10.1109/42.7784.
- M. H. Mahdian, E. Hosseini, and M. Matin. Investigation of spatial interpolation methods to determine the minimum error of estimation: Case study, temperature and evaporation. In *GeoComputation*, 2001.
- A. Mannucci, B. Iijima, U. Lindqwister, L. Sparks X. Pi, and Wilson B. D. *Review of Radio Science 1996 - 1999*, chapter 9 GPS and Ionosphere. Oxford University Press, 1999. ISBN 0198565712.
- G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5:439–468, December 1973. ISSN 00018678.
- G. Matheron. The theory of regionalized variables, and its applications. *Centre de Geostatistique, Fontainebleau, Paris*, 1971.
- R. W. Meggs and C. N. Mitchell. A study into the errors in vertical total electron content mapping using GPS data. *Radio Science*, 41(1), 2006.
- R. W. Meggs, C. N. Mitchell, and P. S. J. Spencer. Simulations of thin shell and 4-d inversion techniques for mapping of total electron content. In *7th General Assembly of the Union of Radio Science International (URSI)*, volume 15, page 20, Maastricht, Netherlands, August 2002.
- E. Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, 2002.
- M. H. F. Meijster, A.; Wilkinson. Fast computation of morphological area pattern spectra. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 3:668–671 vol.3, 2001. doi: 10.1109/ICIP.2001.958207.

REFERENCES

- C. N. Mitchell and P. S. J. Spencer. A three-dimensional time-dependent algorithm for ionospheric imaging using GPS. *Annals of Geophysics*, 46(4):687–696, 2003.
- C. N. Mitchell, L. Alfonsi, G. De Franceschi, M. Lester, V. Romano, and .A. W Wernik. GPS TEC and scintillation measurements from the polar ionosphere during the october 2003 storm. *Geophys Res Letters*, 32(L12S03):1–4, 2005.
- D. P. Mukherjee and S. T. Acton. Cloud tracking by scale space classification. *IEEE Trans. Geosci. Remote Sens.*, 40(2):405–415, 2002.
- M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(8):826–833, 1992.
- M. Nixon and A. Aguado. *Feature extraction & image processing*. Academic Press, 2 edition, 2008.
- H. Omre. The variogram and its estimation. *Geostatistics for Natural Resources Characterization, Part, 1*:107–125, 1984.
- J. M. Pallares, G. Ruffini, and L. Ruffini. Ionospheric tomography using GNSS reflections. *IEEE Trans. Geosci. Remote Sens*, 43(2):321–326, 2005.
- M. Pesaresi and J. A. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Trans. Geosci. Remote Sens*, 39(2):309–320, Feb 2001. ISSN 0196-2892. doi: 10.1109/36.905239.
- T. Q. Pham. Robust fusion of irregularly sampled data using adaptive normalized convolution. *EURASIP Journal on Applied Signal Processing*, 2006, 2006.
- T. Q. Pham and L. J. van Vliet. Normalized averaging using adaptive applicability functions with applications in image reconstruction from sparsely and randomly sampled data. *Image Analysis, Proc.*, 2749:485–492, 2003.
- G. M. Philip and D. F. Watson. Matheronian geostatistics – quo vadis? *Mathematical Geology*, 18(1):93–117, 1986.
- M. Powell. The Theory of Radial Basis Function Approximation in 1990. *Advances in numerical analysis. Vol. 2: Wavelets, subdivision algorithms, and radial basis functions, Proc. 4th Summer Sch., Lancaster/UK*, page 2, 1990.
- M. Rauth. *Gridding of geophysical potential fields from noisy scattered data*. PhD thesis, University of Vienna, May 1998, 1998.
- K. Rijkse. H.263: video coding for low-bit-rate communication. *Communications Magazine, IEEE*, 34(12):42–45, Dec 1996. ISSN 0163-6804. doi: 10.1109/35.556485.
- B. D. Ripley. *Spatial Statistics*. Wiley-Interscience, 2004.

REFERENCES

- P. Sakov. Natural neighbour interpolation software, 2005. URL <http://www.marine.csiro.au/~sakov>.
- A.; Garrido L. Salembier, P.; Oliveras. Antiextensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, 7(4):555–570, Apr 1998. ISSN 1057-7149. doi: 10.1109/83.663500.
- T. Samardjiev, P. A. Bradley, L. R. Cander, and M. I. Dick. Ionospheric mapping by computer contouring techniques. *Electronics Letters*, 29(20):1794–1795, 1993.
- D. T. Sandwell. Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data. *Geophysical Research Letters*, 14(2):139–142, 1987.
- T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
- D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968.
- R. Sibson. A brief description of natural neighbour interpolation. *Interpreting Multivariate Data*, pages 21–36, 1981.
- P. Soille. Constrained connectivity for hierarchical image decomposition and simplification. 30(7):1132–1145, July 2008.
- P. Soille and M. Pesaresi. Advances in mathematical morphology applied to geoscience and remote sensing. *IEEE Trans. Geosci. Remote Sens*, 40(9):2042–2055, Sep 2002. ISSN 0196-2892. doi: 10.1109/TGRS.2002.804618.
- I. Stanislawska, G. Juchnikowski, L. R. Cander, L. Ciralo, P. A. Bradley, Z. Zbyszynski, and A. Swiatek. The kriging method of TEC instantaneous mapping. *Advances in Space Research*, 29(6):945–948, 2002.
- C. Stolle, S. Schluter, S. Heise, C. Jacobi, N. Jakowski, S. Friedel, D. Kurschner, and H. Luhr. GPS ionospheric imaging of the north polar ionosphere on 30 october 2003. *Advances in Space Research*, 36(11):2201–2206, 2005.
- K. Sugihara, A. Okabe, and B. Boots. Spatial tessellations: Concepts and applications of voronoi diagrams. *Probability and Statistics*, 2000.
- The Mathworks, Inc. griddata, 2007. URL <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/griddata.html>.
- M. H. Trauth. *MATLAB Recipes for Earth Sciences*. Springer, 2006.
- M. van Ginkel, J. van de Weijer, L. J. van Vliet, and P. W. Verbeek. Curvature estimation from orientation fields. In P. Johansen B. K. Ersboll, editor, *Proc. 11th Scandinavian Conference on Image Analysis*, pages 545–551, Kangerlussuaq, Greenland, June 1999.

REFERENCES

- L. J. van Vliet and P. W. Verbeek. Estimators for orientation and anisotropy in digitized images. In *Proceedings of the first annual conference of the Advanced School for Computing and Imaging*, pages 442–450, Heijen, NL, 1995.
- R. Warnant and E. Pottiaux. The increase of the ionospheric activity as measured by GPS. *Earth, Planets and Space*, 52(11):1055–1060, 2000.
- D. F. Watson. *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon Press, 1992.
- D. F. Watson. Natural neighbour sorting. *Aust. Comput. J.*, 17(4):189–193, 1985.
- D. F. Watson and G. M. Philip. Triangle based interpolation. *Mathematical Geology*, 16(8):779–795, 1984.
- D. R. Weimer. Models of high-latitude electric potentials derived with a least error fit of spherical harmonic coefficients. *J. Geophys. Res.*, 100(19,595), 1995.
- C-F. Westin and H. Knutsson. Tensor field regularization using normalized convolution. *Proceedings of the Ninth International Conference on Computer Aided Systems Theory (EUROCAST)*, 2809, February 2003.
- P. Wielgosz, D. A. Grejner-Brzezinska, and I. Kashani. Regional ionosphere mapping with kriging and multiquadric methods. *Journal of Global Positioning Systems*, 2(1): 48–55, 2003.
- G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press Los Alamitos, CA, USA, 1994.
- Q. X. Wu, S. J. McNeill, and D. Pairman. Correlation and relaxation labelling: an experimental investigation on fast algorithms. *International Journal of Remote Sensing*, 18(3):651–662, 1997.
- I. T. Young and L. J. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44:139–151, 1995.