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

Matthew Philip Foster

Abstract

Acknowledgements

Contents

List of Figures

List of Tables

List of Acronyms

AC active Contour

ANC adaptive normalised convolution

BSI biharmonic spline interpolation

C/A coarse acquisition

CCC cross correlation coefficient

CM curve matching

CME coronal mass ejection

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

MM mathematical morphology

MINDI Multi-Instrument Non-linear Data Inversion

NC normalised convolution

NDC normalised differential convolution

NN natural neighbour

NS navigational signals

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

SSE sum squared error

SSD sum of squared-differences

SVD singular value decomposition

TEC total electron content

TOA time of arrival

TOI tongue of ionisation

TSM thin sell model

TPS thin plate spline

PNSR peak signal-to-noise ratio

Chapter 1

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 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 global navigation satellite systems (GNSS) 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.

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, geomagnetic storms, the magnetosphere and global positioning system (GPS).

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 exam-

ines 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 using the **MIDAS!** (**MIDAS!**) tomographic imaging software [Mitchell and Spencer, 2003], and and discusses methods of motion estimation, including differential analysis and optical flow.

Chapter 6 is an initial study on the use of block matching and 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.

1.1 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 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 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.

1.2 Introduction to 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 be 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_0F_1 and f_0F_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.

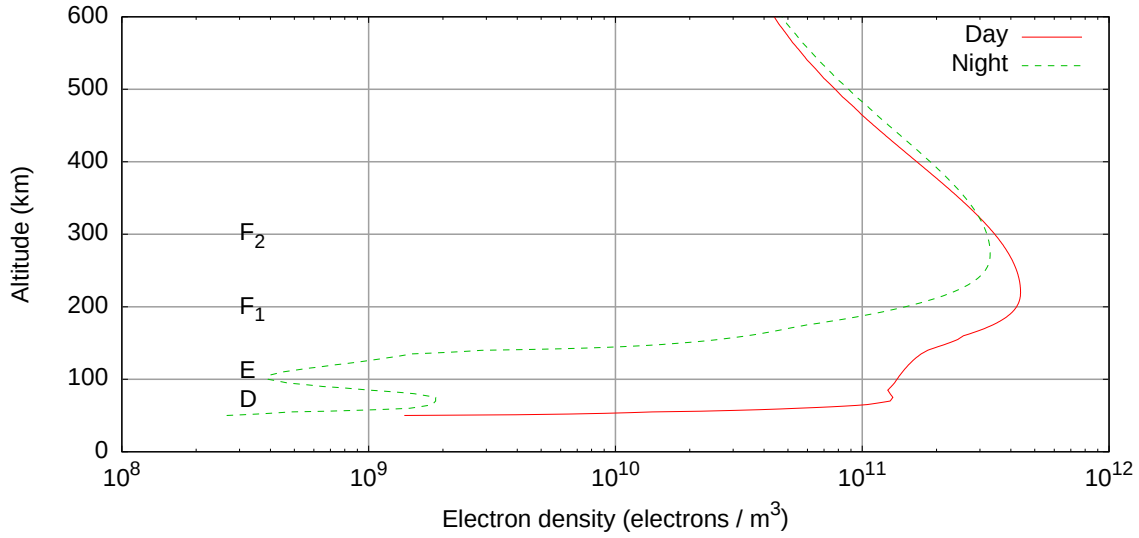


Figure 1.1: Day and night example electron density profiles, generated using IRI2001, for 30 June 2008 at noon and midnight.

1.2.1 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 goemagnetic 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 magnetoshpere, making

its shielding effects very important for all life on Earth.

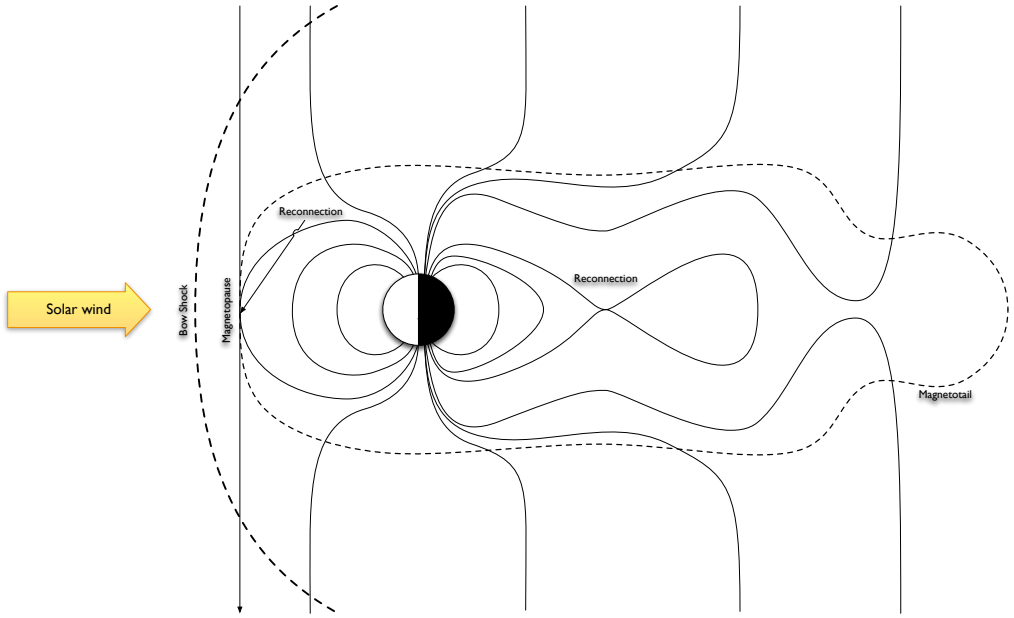


Figure 1.2: A schematic illustrating the Earth's magnetosphere. The solid lines represent magnetic field lines.

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 spiral[1]. 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.

There are various methods of remotely sensing the ionosphere's electron concentration, including the use of GPS Receivers and Satellites (see section 1.2.2); 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

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

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.

1.2.2 GPS

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.

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 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 ionosphere is the largest in magnitude. The ionosphere causes a frequency dependent delay in propagation of L-band signals. This delay varies according to electron 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.

TEC Mapping Using GPS

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. This makes TEC maps of the ionosphere difficult to produce in these areas.

1.2.3 Constructing TEC Maps from GPS Data

The two main methods of constructing maps of ionospheric TEC are interpolation [e.g. Foster and Evans, 2008, 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 to 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.

Chapter 2

Interpolation of Scattered Data

The chapter considered 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.

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 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:

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

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 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.

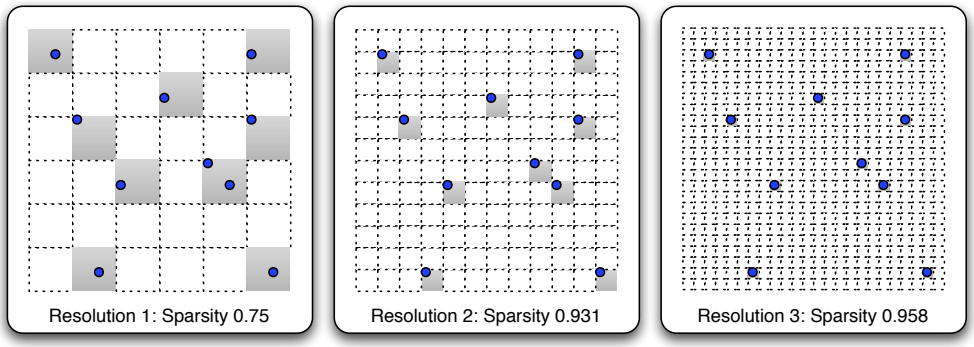


Figure 2.1: The effect that projecting scattered data into different resolution matrices has on sparsity. The small circles represent data points, and grey squares represent matrix elements which take the values of the data.

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:

$$f = \frac{f_i \otimes g}{c_i \otimes g} \tag{2.2}$$

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 than 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.

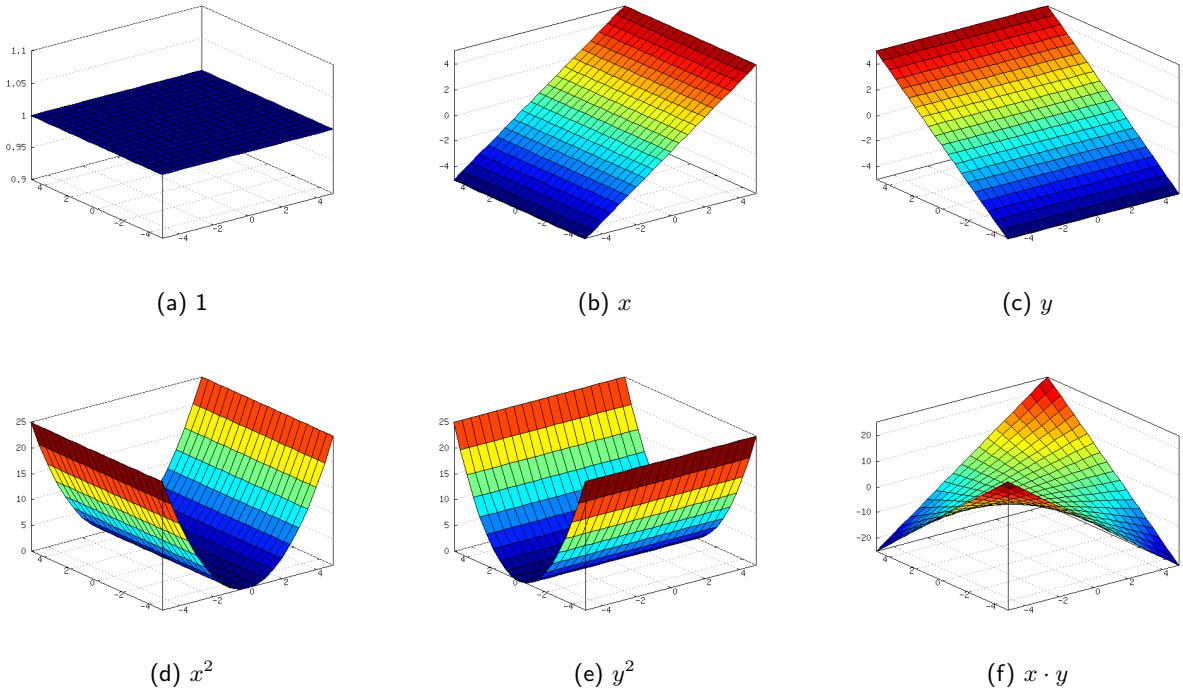
First order NC is the simplest improvement over zero-order NC, and uses bi-quadratic basis functions, where zero-order NC uses a constant. Nine-convolutions are required in all: six for the numerator, and three for the denominator. The following equation describes the process mathematically:

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

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.



(a) 1 (b) x (c) y

(d) x^2 (e) y^2 (f) $x \cdot y$

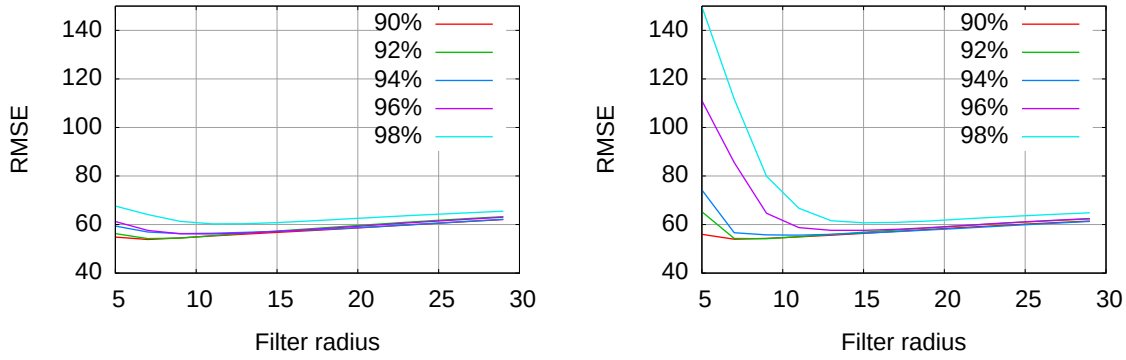
Figure 2.2: Polynomial basis functions as used by first-order NC

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 that 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.

(a) (b) (c)

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.



(a) Zero-order (b) First-order

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...). 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 proved 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:

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

where:

$$D_x(x,y) = x.g(x,y) \otimes f_i(x,y), \tag{2.5}$$

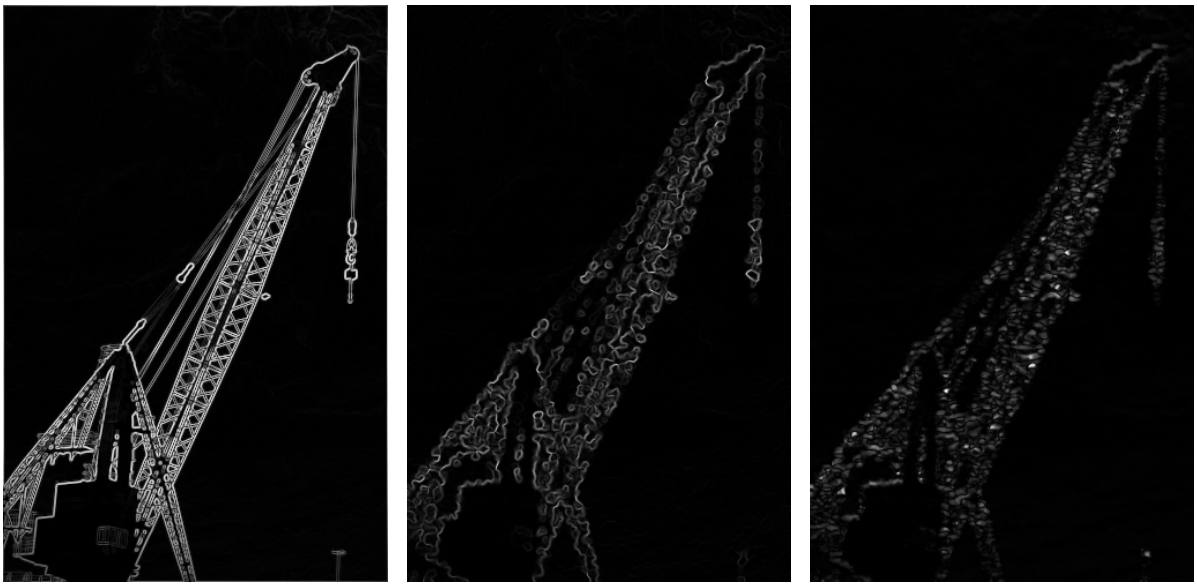
and:

$$N_x(x,y) = x.g(x,y) \otimes c_i(x,y). \tag{2.6}$$

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.



(a) (b) (c)

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.

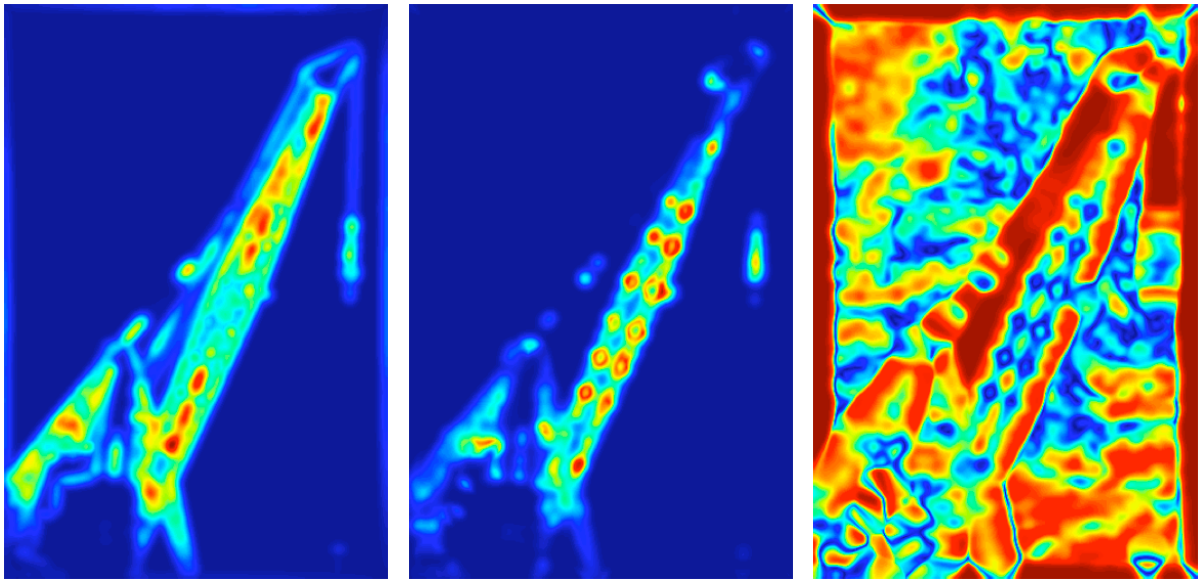
These properties can then be used to set the size and orientation of the NC filters in order to maintain features, and high frequency detail, whilst ensuring that no gaps appear in the output, a process which has been used in the past by Nitzberg and Shiota [1992], Almansa and Lindeberg [2000], and specifically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

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

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.

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:



(a) λ_1 (b) λ_2 (c) A

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.

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

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

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

$$\sigma_u = C(1-A)^{\alpha}\sigma_a \tag{2.9}$$

and

$$\sigma_v = C(1+A)^{\alpha}\sigma_a \tag{2.10}$$

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).

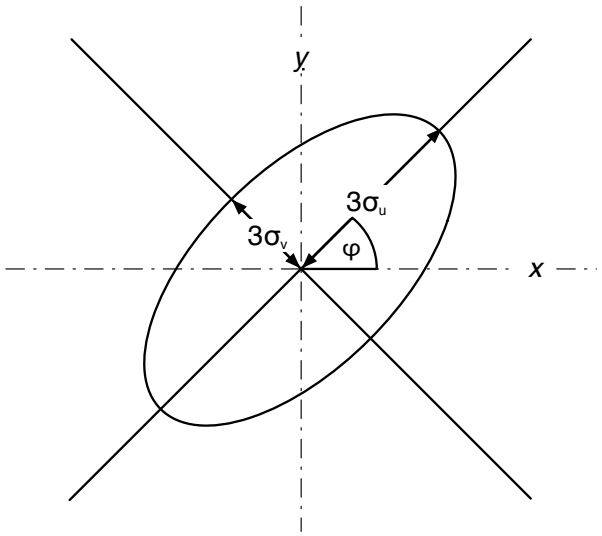


Figure 2.7: A schematic diagram showing shape, and salient parameters of a rotated 2-D Gaussian. The oval indicates 3σ confidence limits in each axis, which corresponds to a 99.7% confidence interval, and is a good place to truncate filters.

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].

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:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.11}$$

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

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

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 additionss 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:

   $$K = USV^* \tag{2.13}$$

   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 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:

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

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:

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

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

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

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:

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

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:

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

φ (and μ) are found using:

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

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.

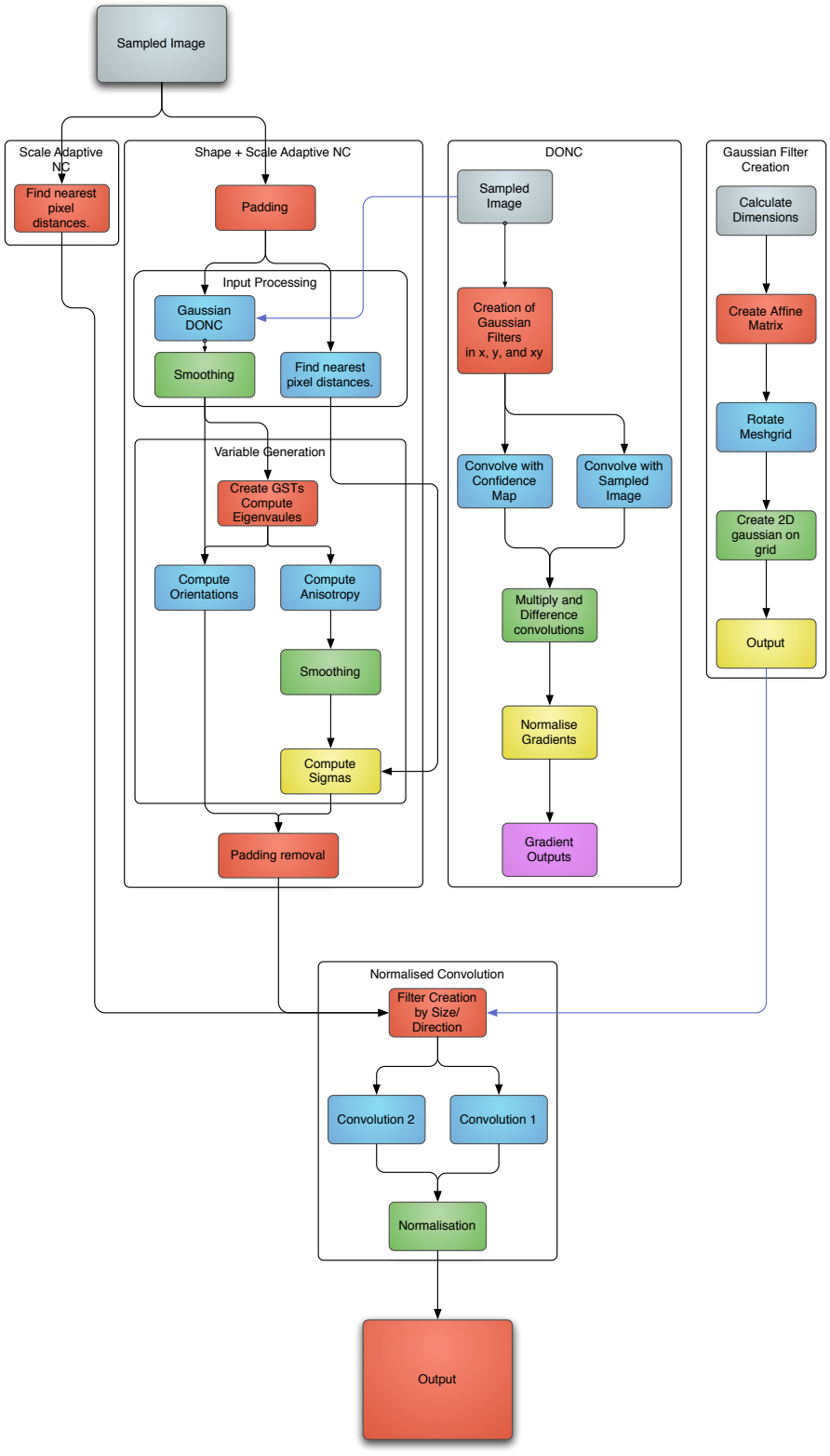


Figure 2.8: A Flow diagram showing the overall ANC process.

(a) Input Image (b) Sampled to $\approx 98\%$ (c) ANC Output

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 triangulations 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

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

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,

$$f(x, y) = c_1 x + c_2 y + c_3. \tag{2.21}$$

The coefficients $\mathbf{c} = (c_1, c_2, c_3)$ can then be found by solving $\mathbf{Ac} = \mathbf{f}$ 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.

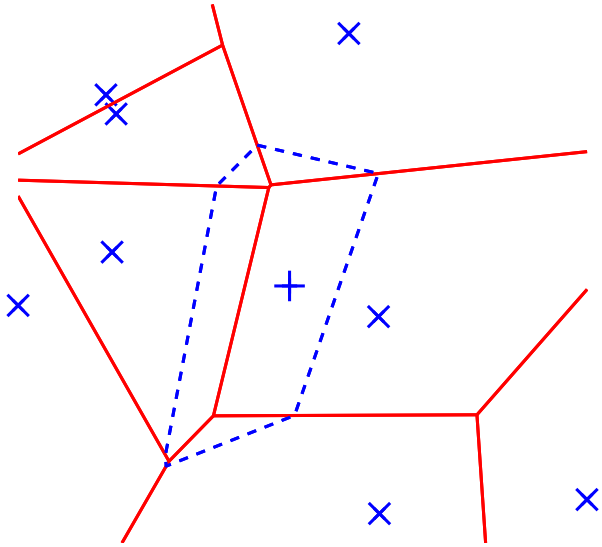


Figure 2.10: Voronoi diagram showing new cell (dotted line) overlapping cells from the original tesselation.

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 though $n+1$ 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

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

where p_m is a low-order polynomial surface with coefficients $c_0, c_1, \cdots, c_n$ 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	$\phi(r) = \lvert r\rvert^2(\log \lvert r\rvert - 1)$

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

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

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),

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

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

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

$$A\lambda = f. \tag{2.25}$$

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

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

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

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

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.

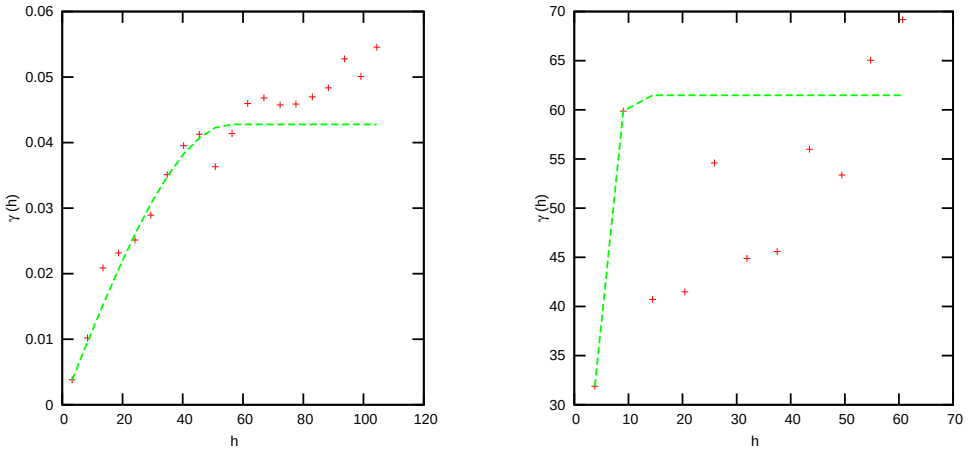


Figure 2.11: Semivariograms (+) and automatically fitted spherical model of (2.27) (dashed lines). (a) Simulated correlated data demonstrating a good fit and (b) degenerate TEC data showing a poor fit.

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.

Chapter 3

Evaluating Interpolation Performance

This chapter is based on material from Foster and Evans [2008], 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 volumetric 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, Gianinetto 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 multivariate 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 20^{th} 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 0.1 corresponds to a 10% error.

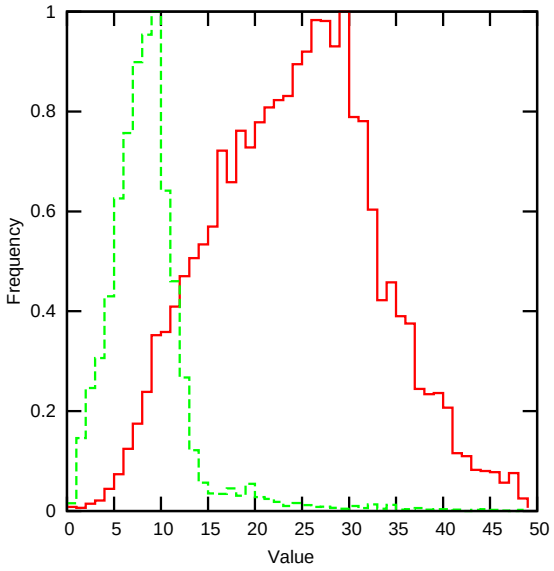


Figure 3.1: Normalised histograms of the two types of simulated correlated data described in section 3.3.1. Multivariate (solid) and univariate (dashed).

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% o 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

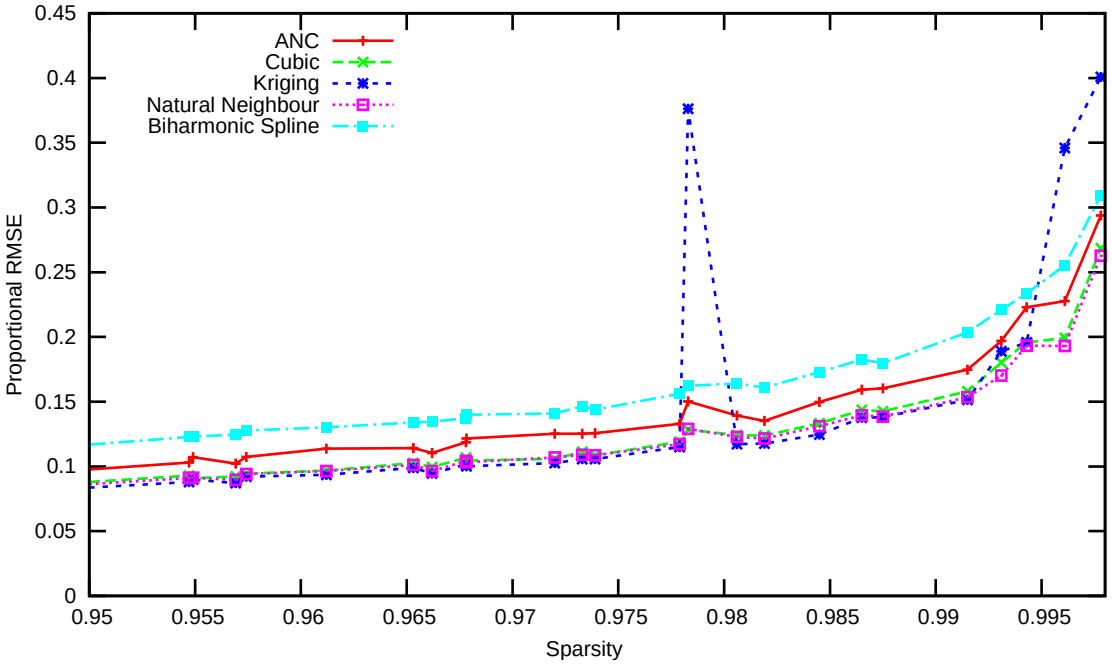


Figure 3.2: Proportional RMSE as a function of sparsity for simulated multivariate correlated data reconstruction.

and multivariate cases natural neighbour interpolation performed best with respect to changing radius of correlation. This is because 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, Astronomiches Instutut Universität Bern [2005], the aim here was to provide representative TEC data for the

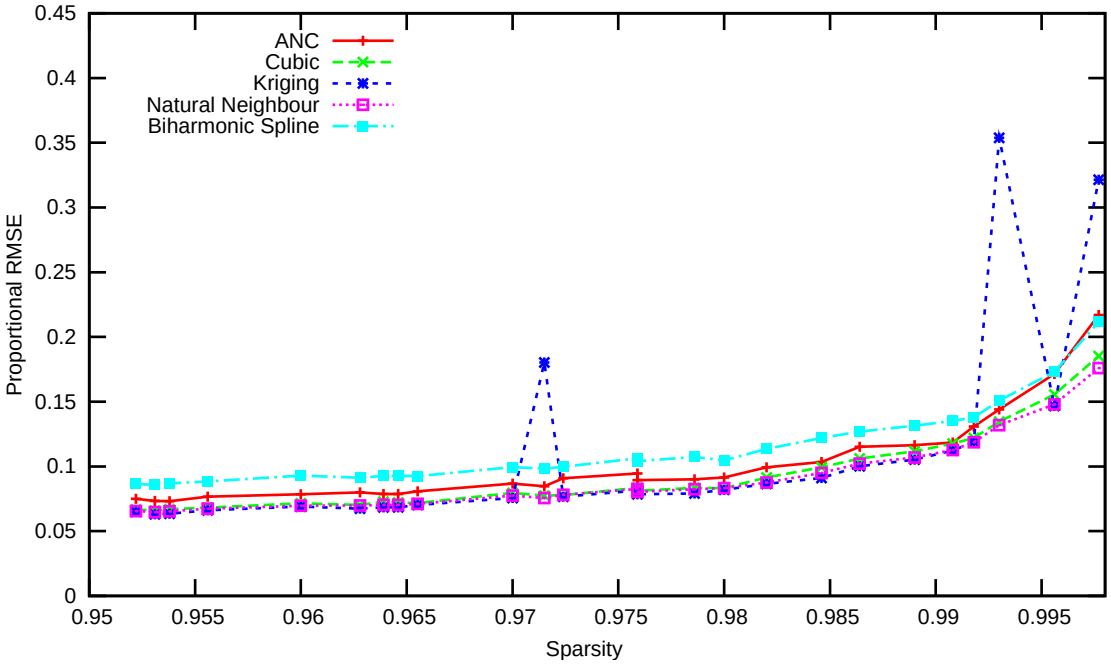


Figure 3.3: Proportional RMSE as a function of sparsity for simulated univariate correlated data reconstruction.

evaluation of the interpolation techniques.

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

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

where z is the satellite elevation angle (in degrees), H is the height of the shell (400 km in

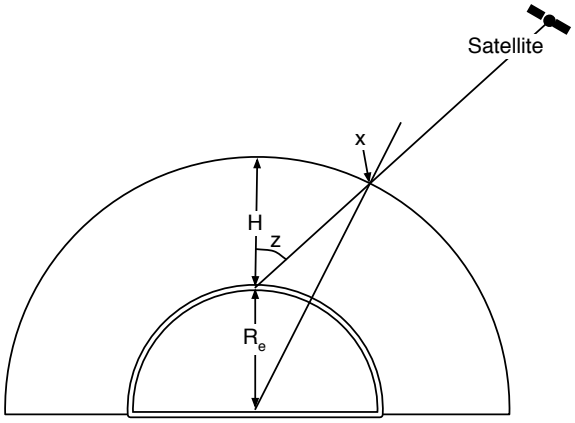


Figure 3.4: Thin Shell Ionosphere Model. R_e is the radius of the Earth, z is the elevation angle from the ground station to the satellite, x is the point at which the path between the satellite and ground station intersects the shell, and H is the height of the shell.

this case) and R_e is the radius of the Earth. Paths with elevation angles $<20°$ 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 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 results[1]). 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

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

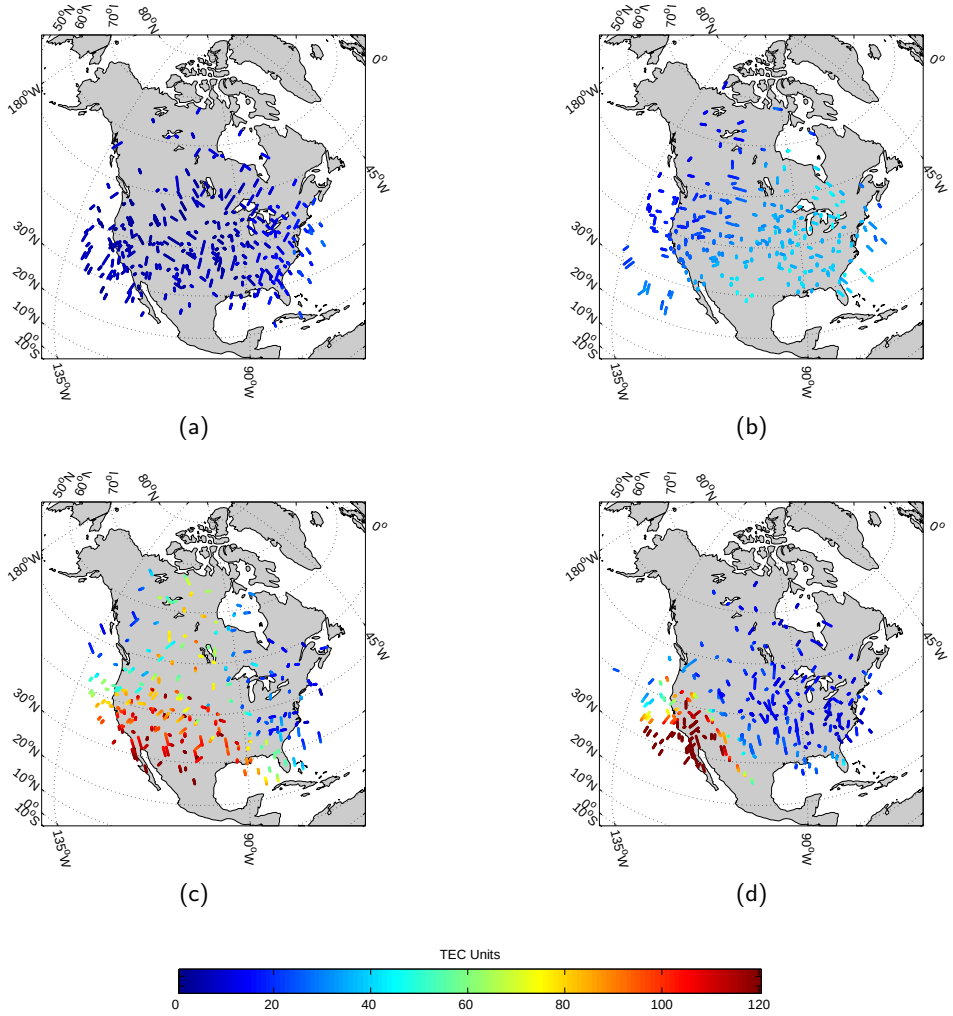


Figure 3.5: Example TEC input data for the cross-validation test in Section 3.3.2. All data are from October 30th, 2003 for the time periods (a) 1200-1215 UT, (b) 1600-1615 UT, (c) 2000-2015 UT and (d) 0000-0015 UT on the 31st.

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

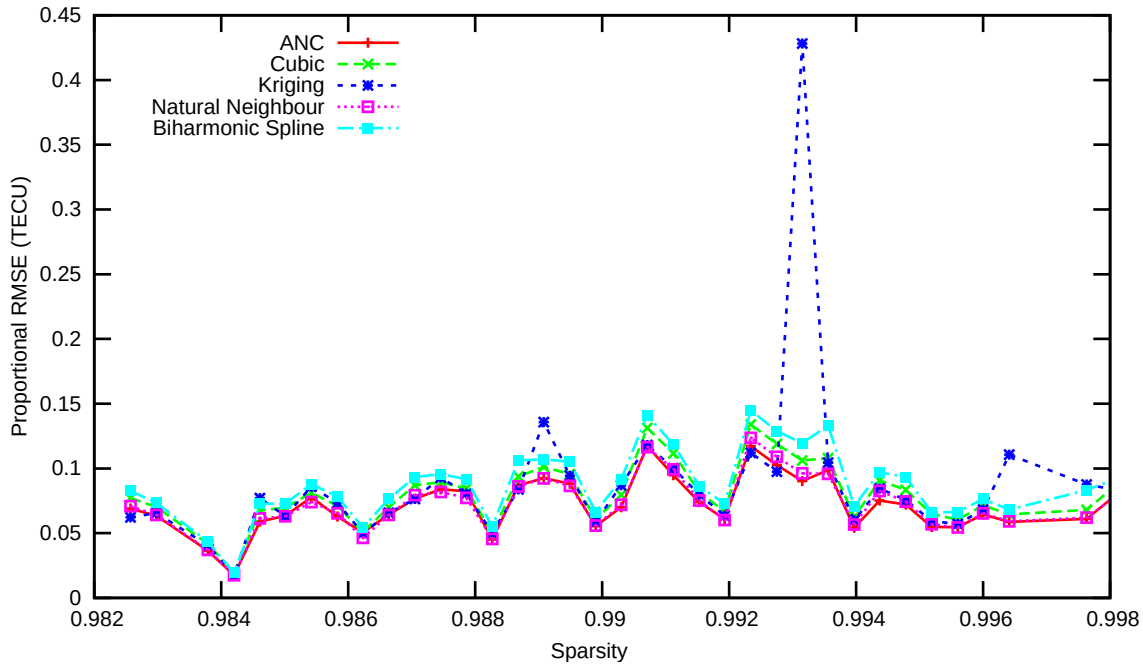


Figure 3.6: Proportional RMSE as a function of sparsity for TEC data from GPS path measurements. Errors were calculated using the cross-validation method described in the text (section 3.3.2).

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 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.

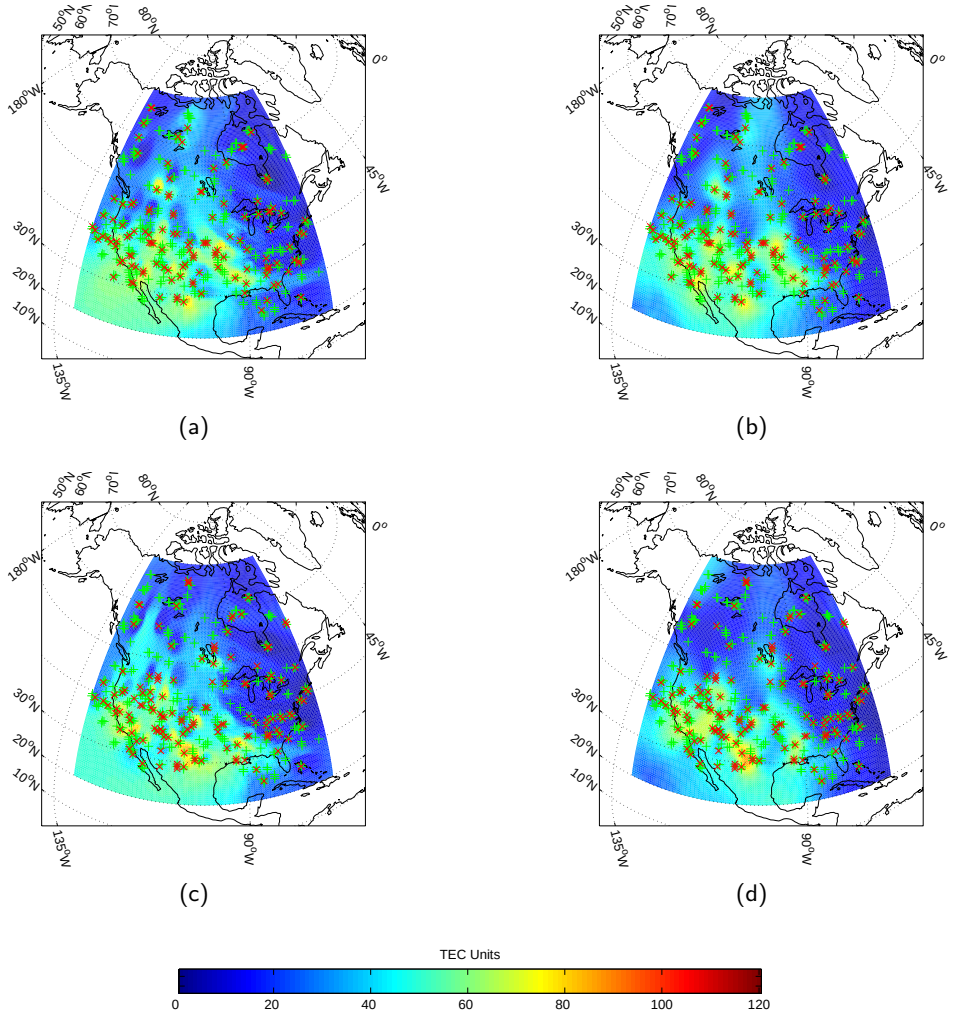


Figure 3.7: Example output images produced by ANC and kriging using two sets of the input data from Fig. 3.5c. Each set consisted of 25% of the available data ($\times$) and the remaining data ($+$) used to calculated the proportional RMSE. (a) and (b) Set 1 results (sparsity 98.94%) produced by ANC 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.

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 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. 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 semivarogram model being used fails to accurately fit the experimental semivarigram, 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 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.

Chapter 4

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.

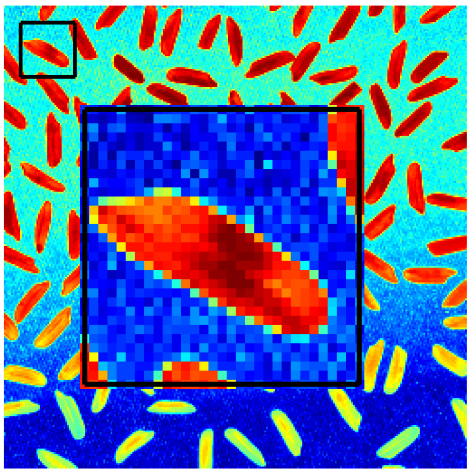


Figure 4.1: A greyscale image of rice grains, displayed using false colour, with the single grain used in Fig. 4.2.

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 together over a frame whose vertices are the data points, and with edges following those of the triangles.

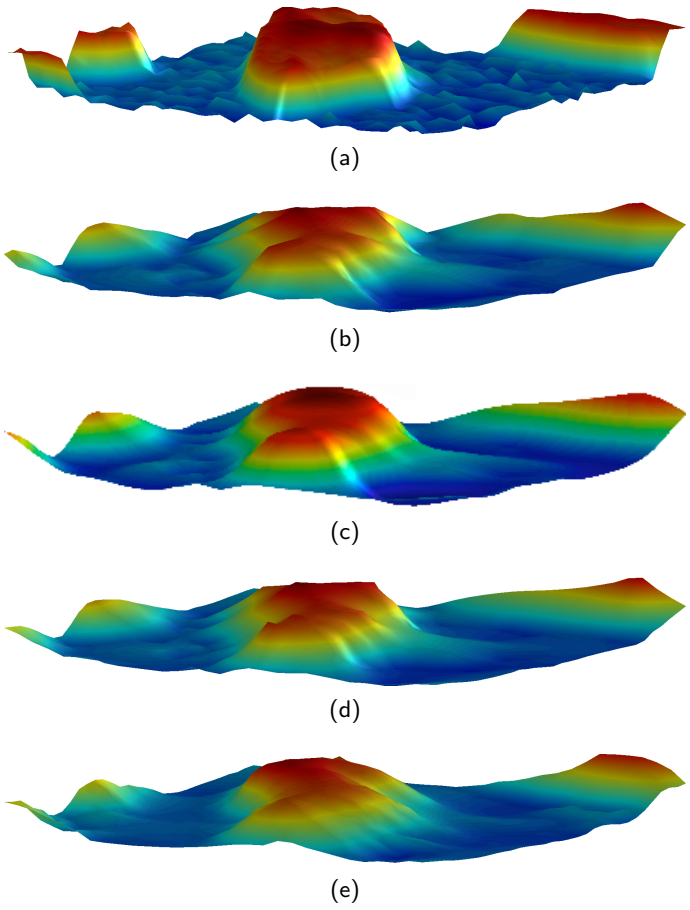


Figure 4.2: Examples of a section of an image of rice grains. The original image, displayed as a displacement map (a) was sampled to a sparsity of $\approx 99\%$, and is followed by linear (b), RBF TPS (c), NN (d) and ANC (e) interpolated versions of the sampled image.

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 is 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,

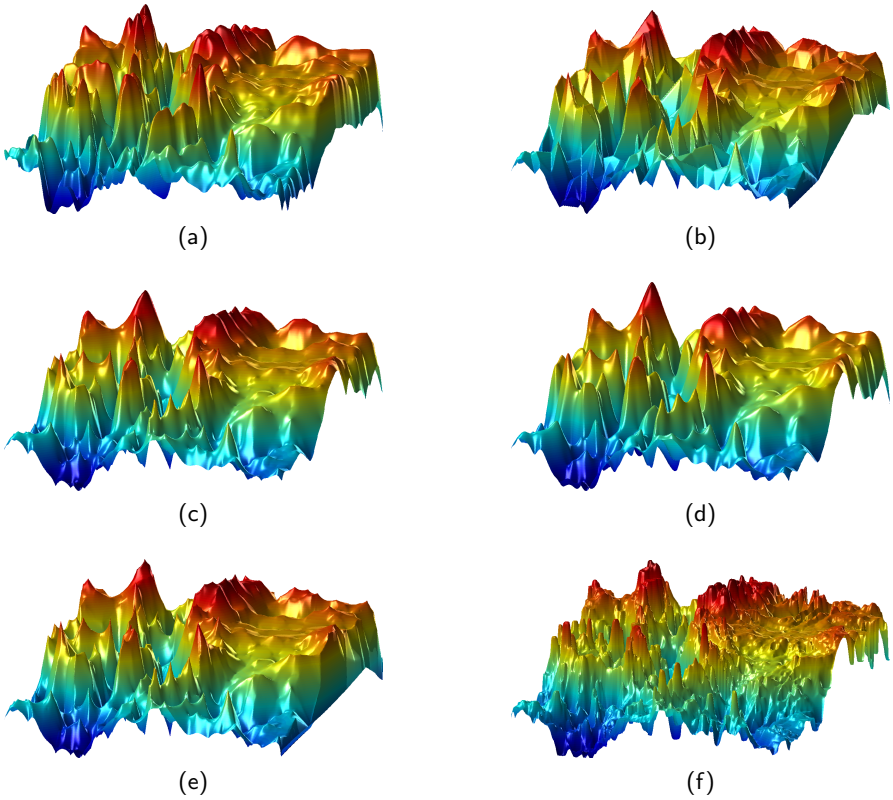


Figure 4.3: Example elevation data from the SRTM, shown as a false colour surface. (a) is the original input data, which was sampled to $\approx 99\%$ sparsity, and reconstructed by (b) was interpolated using linear triangulation based interpolation and (c) was interpolated using linear RBF interpolation.,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

this interpolation method is ideal.

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.

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 pints. 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.

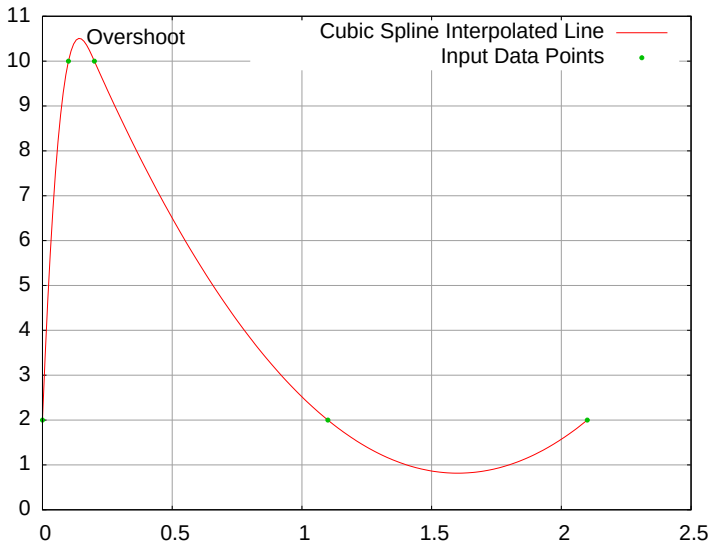


Figure 4.4: Examples of overshoot (and undershoot) when performing cubic interpolation.

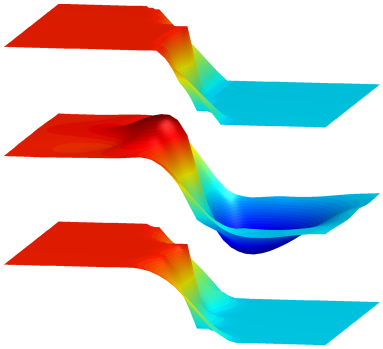


Figure 4.5: Examples of overshoot and undershoot in interpolated simulated edges. Images were reconstructed using (bottom) NN (middle) TPS RBF and (top) Cubic interpolation.

4.1.5 Adaptive Normalised Convolution

ANC [Foster and Evans, 2008], which uses convolution with rotated and scaled filters to perform interpolation. It produces outputs which includes anisotropy where necessary, and in this 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.

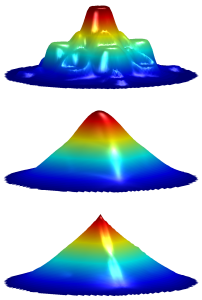


Figure 4.6: Example interpolated pyramids, demonstrating the pointy effect that NN causes (bottom), along with the same data interpolated with cubic interpolation based on triangulation (middle), and ANC with undersized filters (top).

4.1.6 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:

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \tag{4.1}$$

   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 vales. 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.

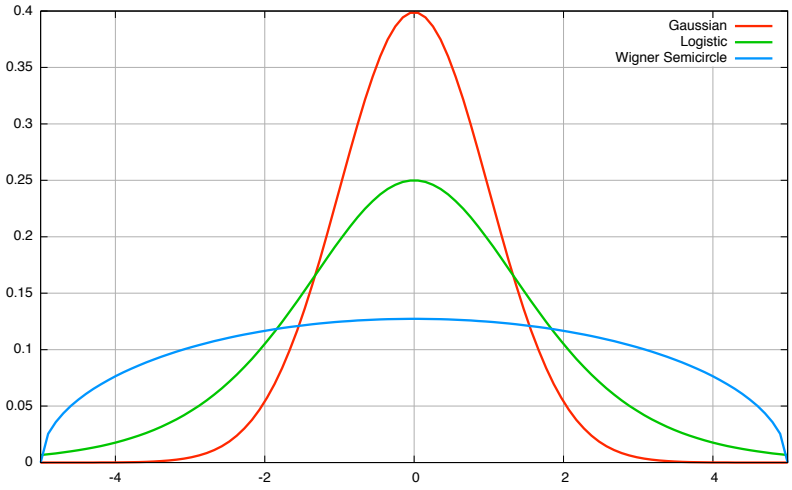


Figure 4.7: Various distributions with differing kurtosis values, including Gaussian (excess kurtosis 0), Logistic (excess kurtosis 1.2) and Wigner semicircle (excess kurtosis -1).

4.2.1 Error Skew

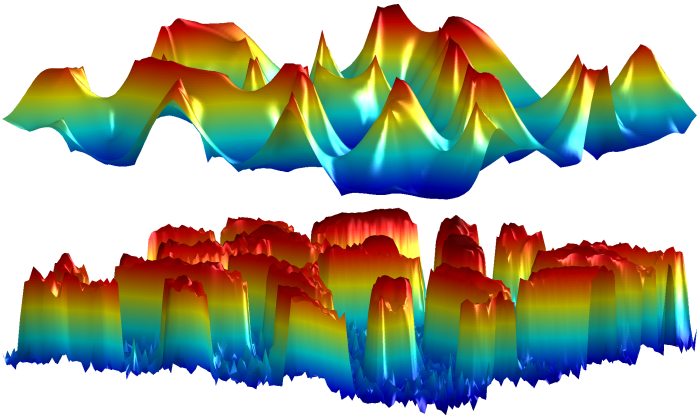
As a further example of the skew problem mentioned above in point 3, Fig. 4.8 shows a sample image and its reconstruction with kriging (a), the semivariogram (b), and the error histogram (c). This histogram shows that the technique is slightly skewed in this case, indicating that output values tend to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

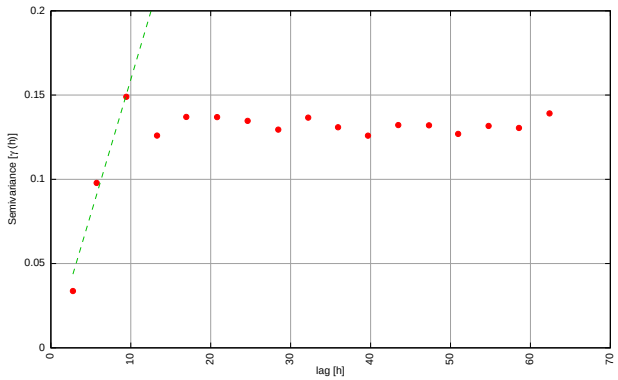
of rice), because each grain of rice has a different orientation, the level of autocorrelation in the image varies significantly across the image, which is also highly anisotropic. This means that the semivariogram in unable to fully capture the spatial variation of the image, which in turn leads to 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 evidences by the longer left hand tail in Fig. 4.8c. Problems with kriging are also discussed in section 3.4.

4.2.2 Confidence Limits

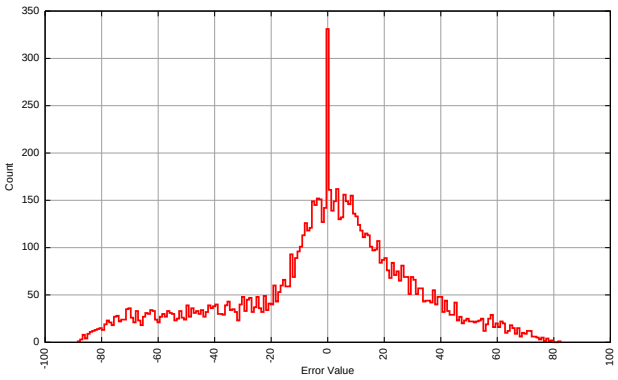
Error distributions can be used to calculate confidence limits on output errors. These can be calculated using the standard distribution of errors, or as absolute error bounds are probably more useful, the histogram of absolute errors could be used to calculate the limits.



(a)



(b)



(c)

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).

Chapter 5

Motion Estimation Techniques

This describes the most common motion estimation techniques, including examples of their application to total electron content (TEC) data where appropriate. The data is described below, and the discussed techniques include differential analysis, optical flow, block matching, and boundary tracking methods.

5.1 Data Sources

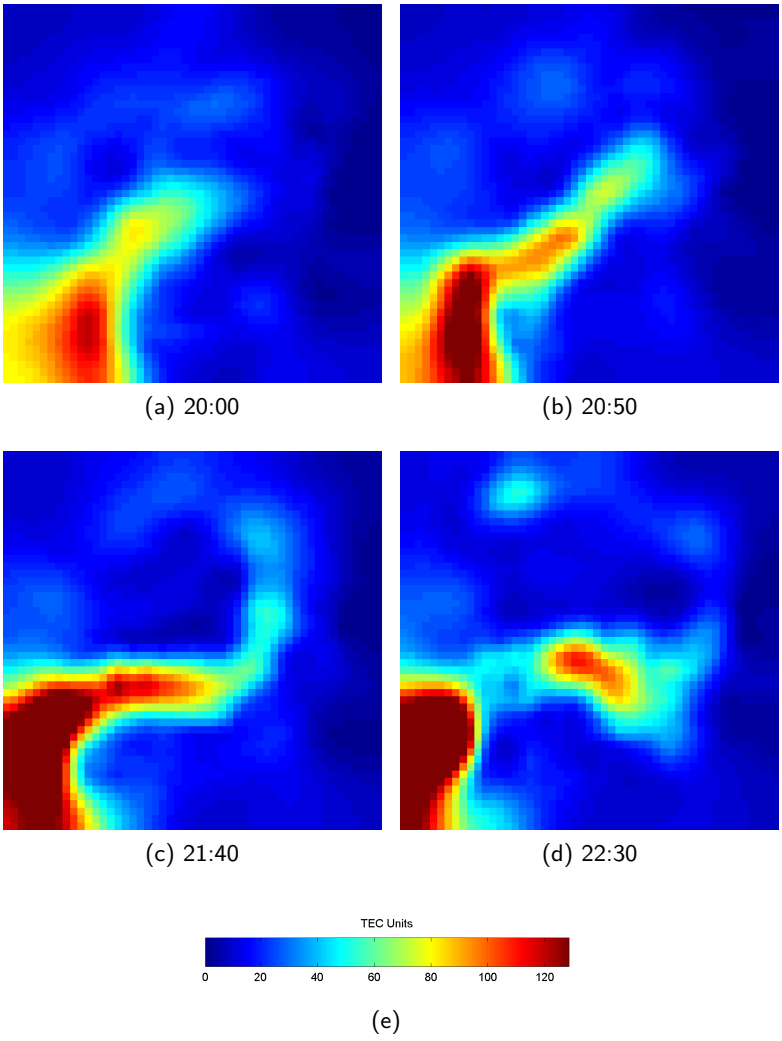
The chief source of data considered in the following chapters was gathered by global positioning system (GPS) receivers during the "Halloween Storm", which occurred during the 29—31 October 2003, with the peak activity between about 2000 and 2300 GMT, over North America. During this time, a large region of storm electron density (SED) appeared over mainland USA, and moved northward, over Canada and through the polar region to the night-side of the Earth.

The available data consist of images created using the Multi-Instrument Non-linear Data Inversion (MINDI) 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 pseudo-inversion.

The area 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 vertically from an altitude 100 to 1600 km, in 40 km increments. TEC maps were created by taking radial line integrals through the grid, giving a 2-D image with a

resolution of 25×25 pixels.

This low resolution is due to a lack of GPS receivers in the area of interest — it is also the chief source of difficulty in estimating the motion of SED blobs. The images contain data ranging in value from close to zero to ≈ 250, meaning that a compact representation using 8-bit unsigned integers is possible. This is advantageous because many image processing techniques are only able to operate on integer classes.



(a) 20:00 (b) 20:50

(c) 21:40 (d) 22:30

(e)

Figure 5.1: 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 in (e).

5.2 Differential Analysis

Subtracting images acquired at different times allows motion to be detected in a very simple, 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 $d(x, y)$, using:

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

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 provide measurements of motion direction. Cumulative difference images (essentially moving averages) can be used to get more information, but require a stationary 'background' to allow for its proper removal.

Fig. 5.2 shows an example of differential motion applied to four frames of TEC data. These have not been had a threshold applied, for the purposes of illustration. In all of these cases, whilst some motion *is* detected, this method would not accurately capture the motion of the tongue of ionisation (TOI), because of the level of change in noise between the images. Similarly, there is a change in average value through the sequence, meaning that a static threshold would probably not work.

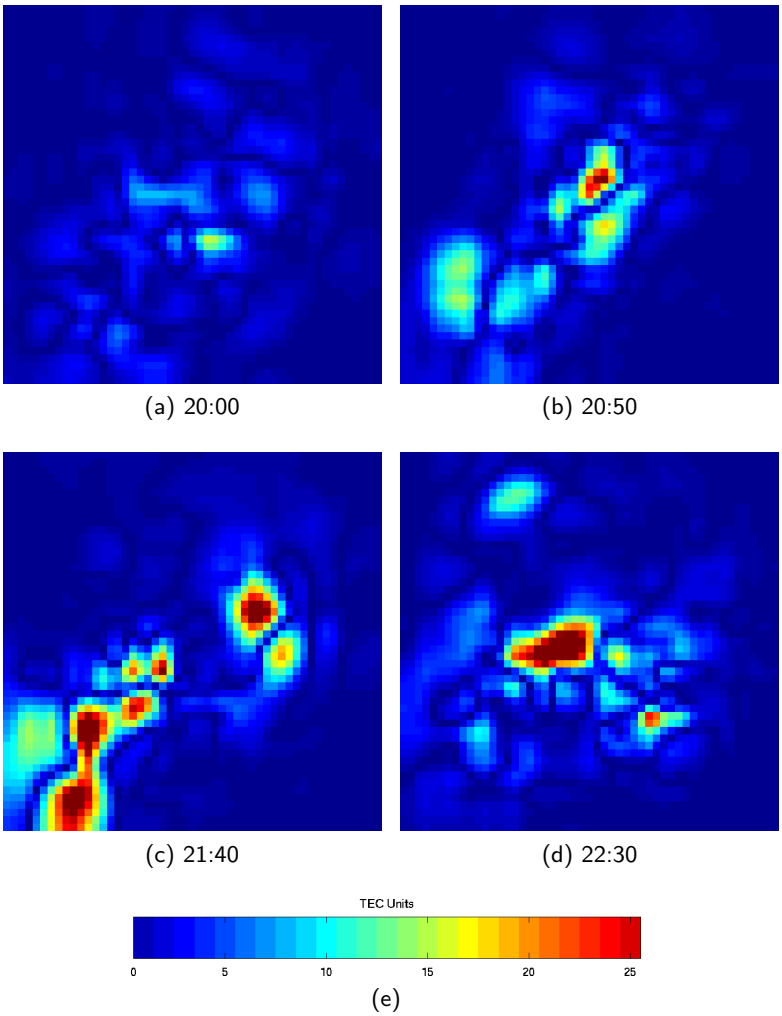
This technique is commonly improved by detecting moving edges, using a moving average filter. However, in this case edge detection operators are too large, relative to the scale of the image, to provide meaningful edge estimates.

5.3 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 applying smoothness constraints on the flow fields it produces.

The main two assumptions on which optical flow computation is based are:

1. An object point's brightness is constant over time



(a) 20:00 (b) 20:50

(c) 21:40 (d) 22:30

(e)

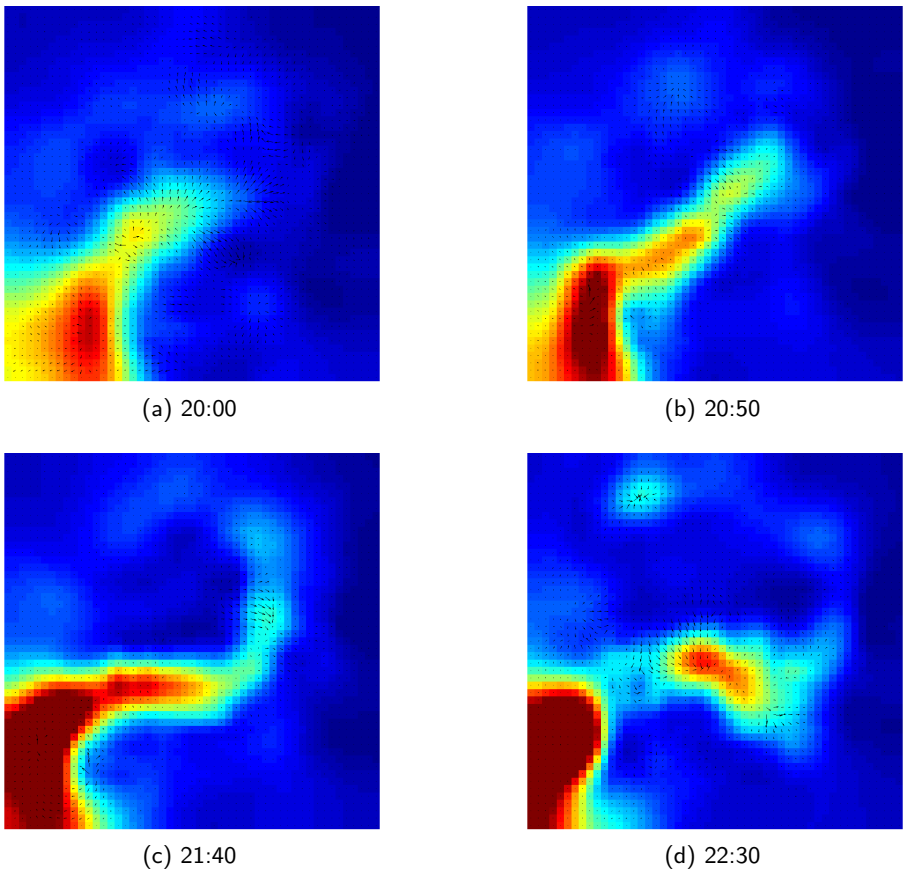
Figure 5.2: Absolute differences between frames at different times in the TEC data sequence. (e) show the colour scale used.

2. Nearby points in the image plane move in a similar manner.

Item 2 is known as the *velocity smoothness constraint*, and is common to most motion estimation systems. It is a very reasonable assumption (similar to the theory of regionalised variables, see § 2.6) which arises from the fact that images tend have regional autocorrelation (characterised by an autocorrelation distance). If this autocorrelation is not present, an image is essentially random noise.

The maximum intensity of the TEC images changes throughout the sequence, and large numbers of spurious features appear and disappear throughout. Also, optical flow requires objects to have moved very little between frames (which, as Fig. 5.2 demonstrates, is *not* the case).

Fig. 5.3 shows images produced using optical flow on the TEC images. 10 iterations were used, with a smoothing parameter of 0.5, the implementation used was that of Nixon and Aguado [2008], with some modifications. Whilst the motion fields produced by this method are very smooth, they are dominated by the background motion, and show a large number of small circular clusters of vectors, where background features appear and disappear across frames. They also fail to capture the main direction of motion in all but the early frames.



(a) 20:00 (b) 20:50

(c) 21:40 (d) 22:30

Figure 5.3: Vectors produces using optical flow processing on the TEC image sequence.

5.4 Template Matching

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].

This can by carried out by sliding the template image throughout the source, and using a

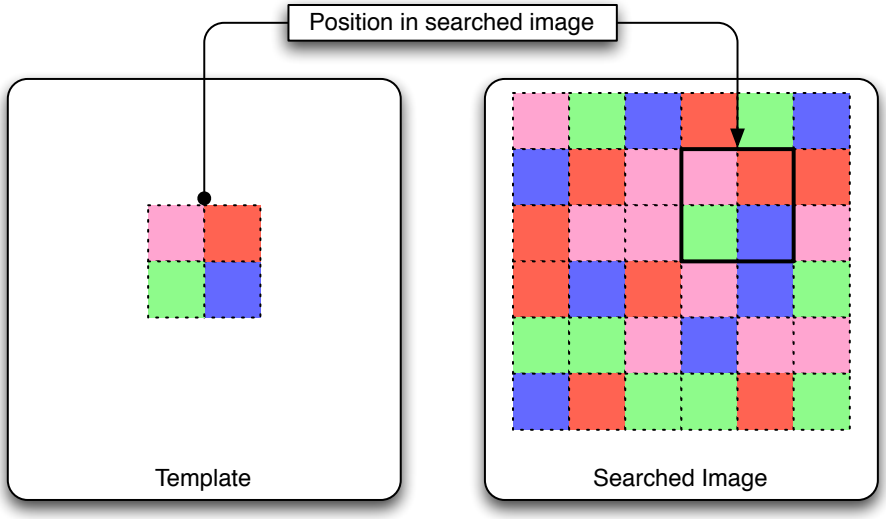


Figure 5.4: A schematic example of template matching

similarity or dissimilarity metric (such as cross-correlation, or sum of squared-differences (SSD) respectively) to measure compatibility between the two.

Motion estimation techniques based on this work by tiling entire source and destination images, and correlating all of the tiles. This gives rise to vectors associated with the strongest correlations. Motion estimation based on template matching is commonly used for video coding. Although they have also found use in many diverse problem of motion estimation, especially in cases where temporal resolution is too low for optical flow techniques to be effective. They have also been applied to a variety of remotely sensed imagery, such as photographs of glaciers [Evans, 2000a], visible and infrared satellite images of clouds [Evans, 2006, 1999] and radiometer images of the oceans [Dransfeld et al., 2006]. Such methods are particularly attractive because of their intuitive nature, and the possible shortcuts that can be taken to lower computational cost. Chapter 6 describes the process of block matching based motion estimation in more detail.

5.5 Boundary Tracking

5.5.1 Snakes

5.6 Other Techniques

Chapter 6

Motion Estimation Using Template Matching

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 regions under convection [see 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 is concerned with the development of data-driven approaches to estimating the motion of such regions, using image processing and computer vision techniques. As these are largely de-coupled from the underlying physics, they 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.

6.1 Template Matching

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,

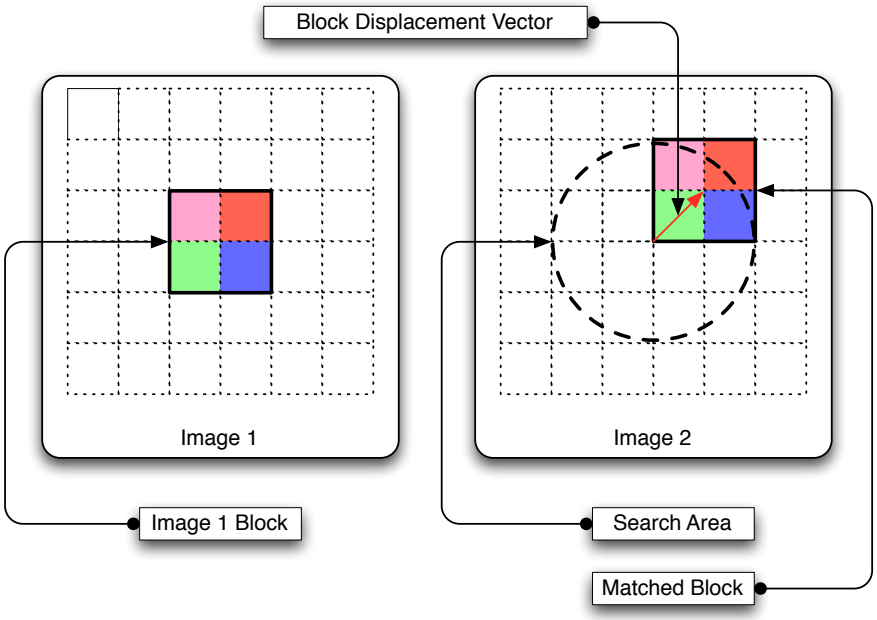


Figure 6.1: The template matching process. Image two is "searched" for values similar to those within the current template in image one. The result is a vector describing the displacement necessary to shift the template in image one to its position in image two.

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. 6.1 illustrates the basic process of template matching.

Template searching is done by measuring the similarity between the source template and various different destination templates in image two. The measurements can be done using several metrics, and various different structures, search techniques and similarity metrics. These are described below.

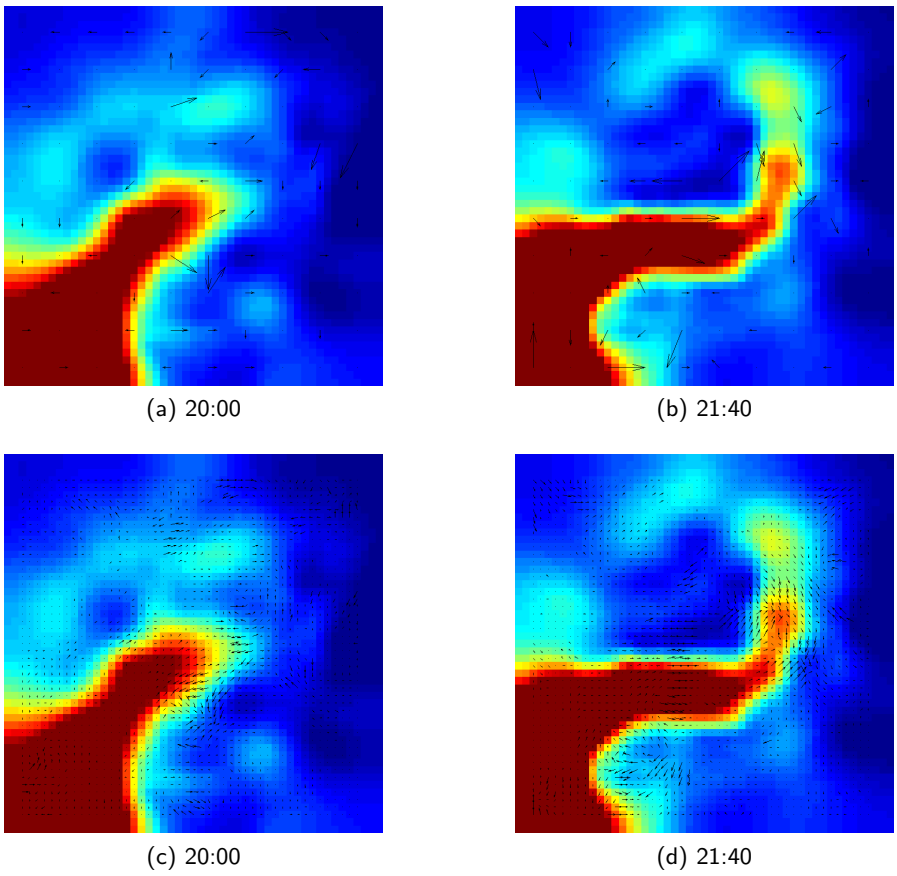
6.1.1 Template Structure

Template matching aims to find parts in an image which match a given template from the first 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 known as template matching. Figs. 6.2a and 6.2b show the result of block matching using 5×5 blocks, with a search radius of five pixels, on two frames of total electron content (TEC) data.

As described above, the block matching procedure aims to find matches between blocks in two images, referred to as the *source* and *destination*. The are several ways of laying out the blocks in the source image, including tiling, and overlapping blocks.

Performing motion estimation on two *tiled* frames, allows the second frame to be almost completely described in terms of the source frame, and vectors describing blocks motions. In this case, the source image is tiled entirely in non-overlapping blocks, and the block matching process returns one motion vector for each block.



(a) 20:00 (b) 21:40

(c) 20:00 (d) 21:40

Figure 6.2: Example outputs created using traditional block matching (a)–(b), and traditional block matching with overlapping blocks (c)–(d). Both sets of outputs use the TEC data sequence discussed in this chapter, up-sampled by two.

Instead of using tiled blocks, a denser output field can be estimated by using overlapping

blocks. Instead of tiling the blocks, they are shifted by a given amount, the limit being a one pixel shift. This gives more vectors, and a smooth output field, but has a correspondingly higher computational cost than using tiled blocks. This was the method used in this study. Figs. 6.2c and 6.2d show the result of block matching using 5×5 overlapping blocks, with a search radius of five pixels, on two frames of TEC data.

Densification of the output vector field can also be achieved by using sub-pixel search methods. These are described below.

6.1.2 Sub-pixel Block Matching

Motion between frames does not always occur in integer pixel increments. This is because the images being examined are usually sampled representations of some 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, and then performing block matching. To reduce interpolation artefacts, 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. 6.3 shows the process diagrammatically. The output of sub-pixel accuracy motion estimation system 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 radius of search. Fig. 6.4 shows the result of sub-pixel block matching on two example frames in the TEC image sequence, 5×5 blocks were used, along with a ten pixel search radius.

6.1.3 Search Methods

Research for video compression has produced a great many different search algorithms, most of which are concerned with obtaining a highly-compressible motion field, as quickly as possible. These do not guarantee accurate motion estimates, and instead trade accuracy for low entropy. Because of this, and due to the low resolution of the images available for use in this study, (which heavily constrain the maximum displacement, and keep the computational cost low), a very simple exhaustive search method was used. Block searches were carried out within a fixed radius of the source block. Considering the example given in Fig. 6.1, if the dashed circle is taken as the extent of the search, all blocks with their centres within a two-pixel radius of the centre of the circle would be searched.

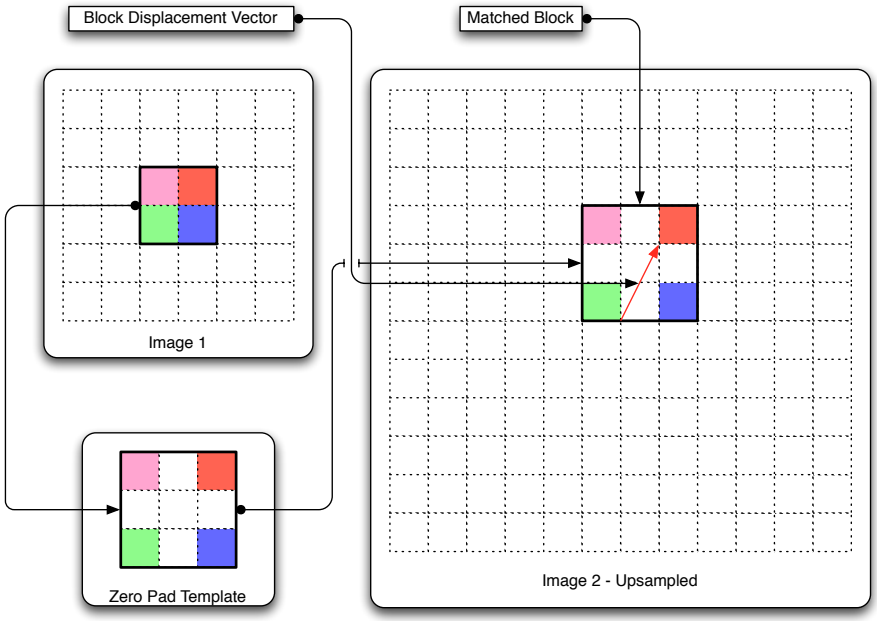


Figure 6.3: Block Matching with sub-pixel accuracy.

6.1.4 Similarity Metrics

Similarity metrics measure the correspondence (or correlation) between source and destination blocks. The choice of metric depends on computational cost, and performance, and, as with most problems, some trade-offs must be made. The most common (dis)similarity metrics are described below.

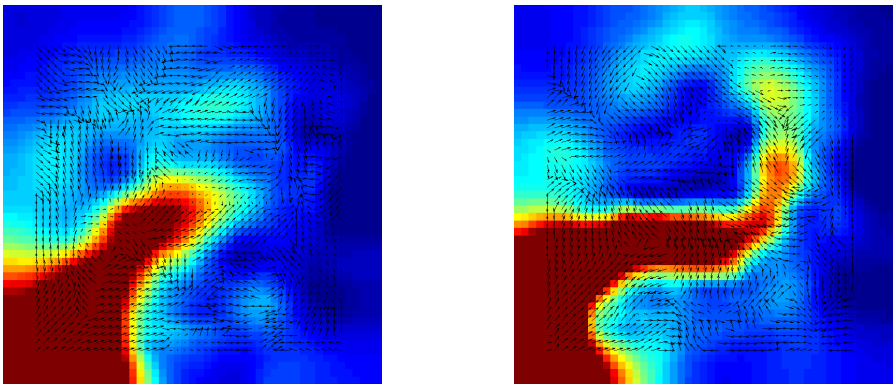
- The sum of absolute value of differences (SAVD) is a *dissimilarity* measure, defined as the sum of the absolute value of differences between corresponding pixels in a source block f and destination block g.

$$SAVD(f,g) = \sum_i \sum_j |f(i,j) - g(i,j)| \tag{6.1}$$

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

- The sum of squared-differences (SSD) is also a dissimilarity measure, and is very similar to the SAVD. It is defined as:

$$SSD(f,g) = \sum_i \sum_j (f(i,j) - g(i,j))^2 \tag{6.2}$$



(a) 20:00 (b) 21:40

Figure 6.4: Example outputs creating using sub-pixel block matching with overlapping blocks, on the up-sampled TEC data sequence, at two different times.

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

- The cross correlation coefficient (CCC) is widely regarded as the most effective metric. The CCC is a *similarity* metric, 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 fairly resistant to noise.

$$\rho(f,g) = \frac{\sum_i \sum_j \Big((f - \overline{f}) \times (g - \overline{g}) \Big)}{\sqrt{\Big(f - \overline{f}\Big)^2} \times \sqrt{(g - \overline{g})^2}} \tag{6.3}$$

  Where $\overline{f}$ and $\overline{g}$ 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.

A good comparison of these metrics can be found in Wu et al. [1997], which concludes that whilst SSD and SAVD are much faster than the CCC, the CCC produces better quality vectors. For this reason, and the fact that no real-time processing was necessary, the CCC was used in this study.

After the searches and correlations have been performed for a given block, the output will a set of displacement vectors, with a CCC attached to each vector. The most simple systems use the vector associated with maximum CCC as the estimate for a given block. This is 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 again the vector components as a 2-D surface. These *correlation surfaces* are often multi-modal, and have indistinct peaks[Anandan, 1989, Evans, 2000a], which can make choosing the best vectors difficult. This is the reason for using relaxation-labelling — a technique for smoothing motion fields.

6.1.5 Block Thresholds

The similarity metrics cannot provide meaningful values on regions which are flat. In the degenerate case of a completely uniform block, the denominator of the CCC is zero, and so the output is undefined. When blocks are almost uniform, noise is the main factor influencing matches. For this reason, it is a prudent to apply a threshold to one or more metrics in order to decide if which blocks should be examining for potential matches. Only blocks who's properties meet the set criteria are then used in the motion estimation (ME).

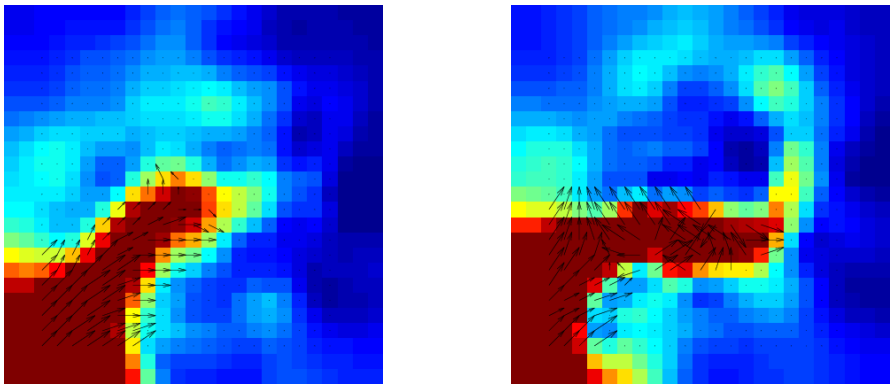
A large number of metrics exist which could be used to determine which blocks to ignore. However, in this case three simple methods were used. These were:

- The mean value of the block; experimentation showed a good threshold value is approximately $1.5\times$ the mean value of the frames.
- The standard deviation of the block; a good threshold value was found to be $1.5\times$ the average value obtained by using a standard-deviation filter with the same dimension as the blocks being used.
- The edge vector-magnitude. This is obtained by Sobel filtering the frames, in both vertical, and horizontal directions, and then taking the magnitude. The threshold used was $0.8\times$ the mean value.

All three of these methods behaved reasonably, although using an edge vector-magnitude threshold most closely mirrored the pattern of blocks containing interesting features. This is because the most trackable image motion occurs at the edges in the images. Fig. 6.5 presents example results on frames demonstrating good and bad results.

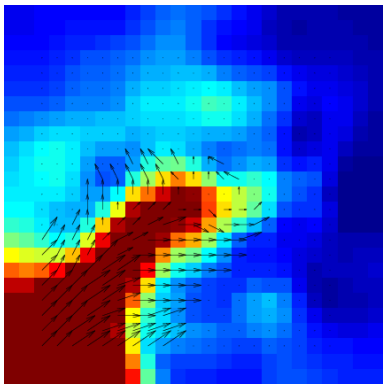
6.1.6 Relaxation Labelling

The MCC method described above has the tendency to produce noisy vector fields because correlation surfaces are often multi-modal. To mitigate this problem, additional smoothness

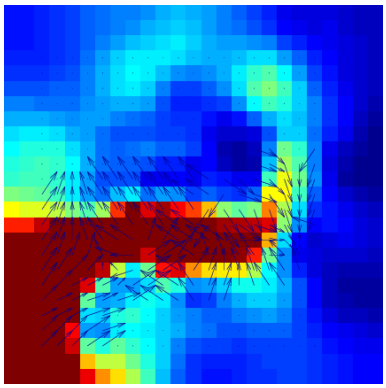


(a) Value threshold: Good Frame

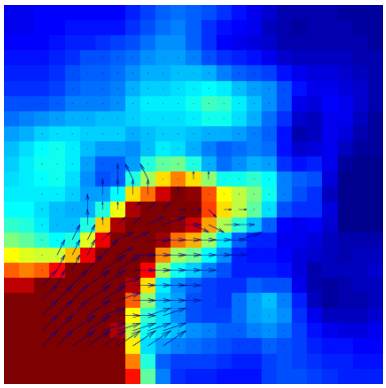
(b) Value threshold: Bad Frame



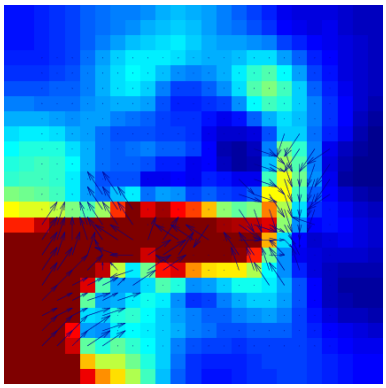
(c) σ threshold: Good Frame



(d) σ threshold: Bad Frame



(e) Sobel threshold: Good Frame



(f) Sobel threshold: Bad Frame

Figure 6.5: Example frames with good, and bad vectors overlaid. Blocks were thresholded using their mean values (a)-(b), standard deviations (c)-(d) and Sobel magnitudes (e)-(f).

contracts 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

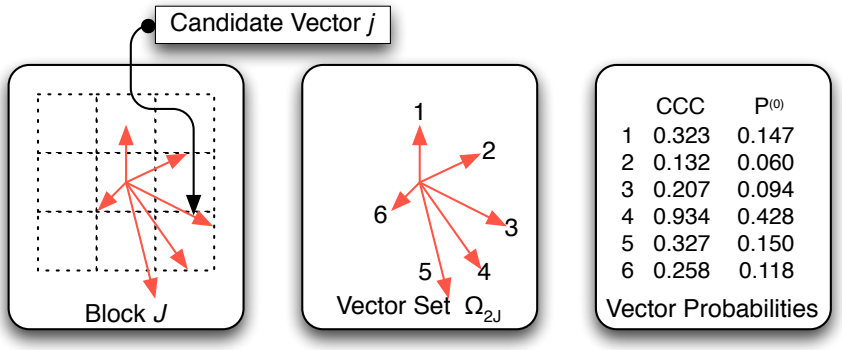
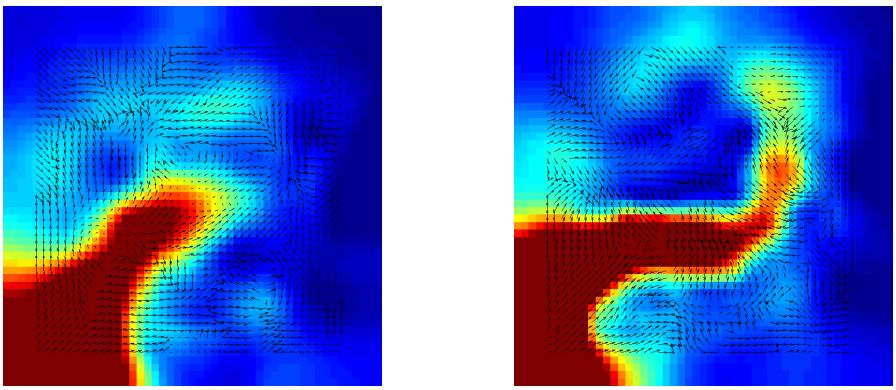


Figure 6.6: Terms used in the relaxation labelling equations (6.4)—(6.7).

fields, whilst ensuring spatial consistency. It is commonly used when examining remotely-sensed images of 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.



(a) 20:00 (b) 21:40

Figure 6.7: Example outputs created using sub-pixel block matching with overlapping blocks and relaxation labelling, carried out with the up-sampled TEC data sequence.

The relaxation process works as follows, terms are introduced through the process, and are also shown graphically in Fig 6.6:

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' CCCs 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:

$$P^{(0)}(J \rightarrow j) = \frac{\rho(J \rightarrow j)}{\sum\limits_{\lambda \in \Omega_{2J}} \rho(J \rightarrow \lambda)} \tag{6.4}$$

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

3. The probabilities are updated using the non-linear relaxation formula (6.5). 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.

$$P^{(n+1)}(J \rightarrow j) = \frac{P^{(n)}(J \rightarrow j)Q(J \rightarrow j)}{\sum\limits_{\lambda \in \Omega_{2J}} P^{(n)}(J \rightarrow \lambda)Q(J \rightarrow \lambda)} \tag{6.5}$$

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

$$Q(J \rightarrow j) = \prod_{I \in G_j} \sum_{i \in \Omega_{2J}} P^{(n)}(I \rightarrow I)R(I, J, i, j) \tag{6.6}$$

The output from this will depend on both the probabilities of the neighbouring block's vectors, and the output of the function $R(\ldots)$, which measures vector similarity, and has the following form:

$$R(I, J, i, j) = \exp\left[\frac{\Delta x_{I,i} - \Delta x_{J,j}}{\sigma}\right] \exp\left[\frac{\Delta y_{I,i} - \Delta y_{J,j}}{\sigma}\right] D(I, J) \tag{6.7}$$

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 functions which returns the relative distance between the blocks I and J, given by:

$$D(I, J) = \max\left(D_0 - D_{I,J}\right) \tag{6.8}$$

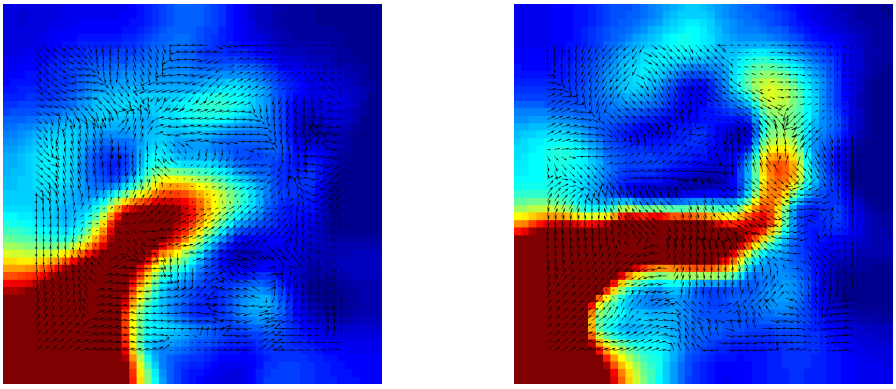
In (6.8), $D_{I,J}$ the sum of the horizontal and vertical distances between the blocks, normalised to the block dimension.

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

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. Blocks which were disabled because they did not exceed the imposed thresholds were tagged using a correlation of -2, which is outside of the range of acceptable values, and so simple to detect. Ten iterations were run, using a neighbourhood size of 3×3, and only considering vectors with cross correlations that were ≥ 0.5.

6.1.7 Post-filtering

Additional smoothing of output vectors can be provided by the application of a vector median [Astola et al., 1990] based post-filter. This process works by examining each vectors compatibility with its neighbours, and replacing it with the median vector for the neighbourhood if it differs by more than a given threshold. Fig. 6.8 shows some frames from the TEC data sequence which have overlaid vectors formed by post-filtering the vectors in Fig. 6.7, using a threshold of 0.6.



(a) 20:00 (b) 21:40

Figure 6.8: 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.

6.1.8 Discussion

This chapter has introduced example results illustrating each stage of the block matching process, as the stages were introduced. Whilst each stage of the process improves on the previous, the final outputs are still not good enough to accurately capture the motion throughout the whole sequence.

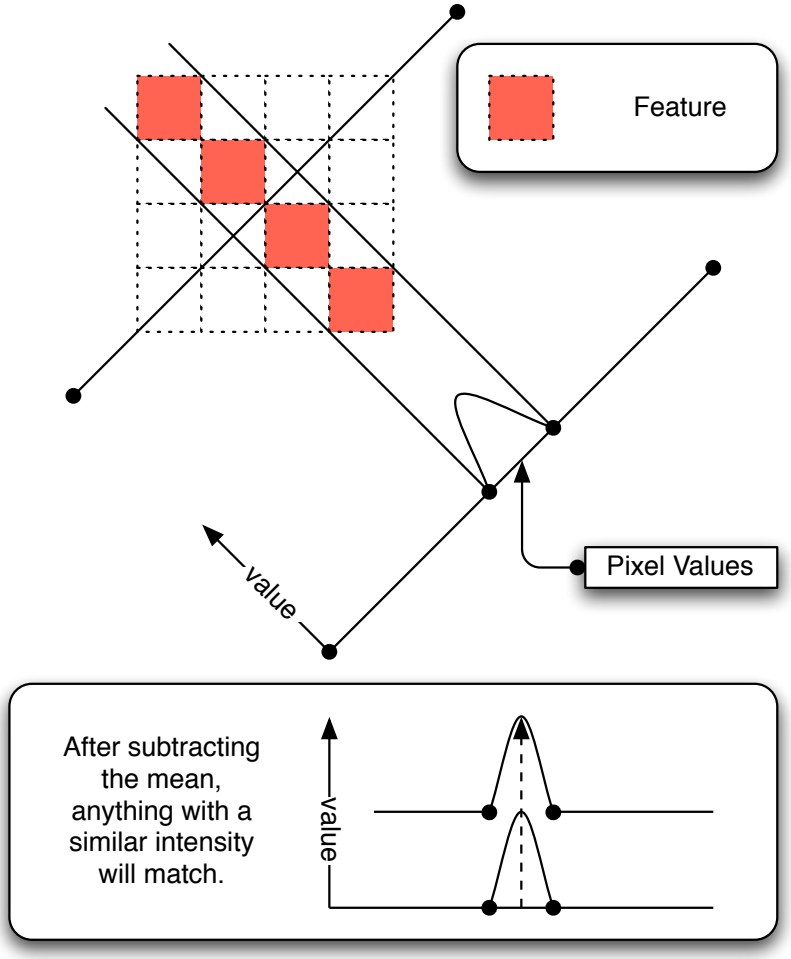


Figure 6.9: Matching mean invariance means that features only have to have the same shape to be matched with one-another.

Problems occur because of temporal incoherence in features being matched. For example, a lot of the background features in the images are not present in the preceding or following frames. This spontaneous appearance of features breaks a fundamental assumption in block matching: that the frames contain the same features.

Also, as the images are very small, the features are correspondingly small, and lack texture, and this often leads to multiple matches. To see why this is, consider the case of a small edge running through a block, as shown in Fig. 6.9.

This shows that once the mean has been subtracted, a large number of blocks will have similar values, making higher cross-correlations more likely than if the technique was not mean-invariant. Whilst this is normally a good thing, when using low resolution images the

number of possible different blocks is fairly low, and so more matches will occur. This increases the likelihood of choosing a false one.

6.1.9 Conclusions

On their own, the images used here are too low resolution, and contain too many oddities to allow accurate and realistic motion fields to be computed using block matching methods alone. For this reason, other approaches must be investigated. The following chapter address this by examining and developing techniques using segmentation and shape matching to estimate motion.

Chapter 7

Motion Estimation using Curve Matching

Section 6.1 introduced the problem of estimating motion present in low-resolution TEC imagery, and ended by concluding that block matching alone was unsuitable due to problems with matching small blocks.

This section follows the development of a motion estimation system which makes uses of shape boundary matching to find shape correspondences. These correspondences are then used to derive motion vectors. The main prerequisite for such a system is an accurate description of the shapes within each frame, and for this reason the system described in here has two main stages: segmentation and motion extraction. Section 7.1 describes the development of the first part, which segments the shape boundaries, and section 7.2 describes the second part, which make use of shape matching to perform the motion extract motion. The overall process is described pictorially in Fig. 7.1.

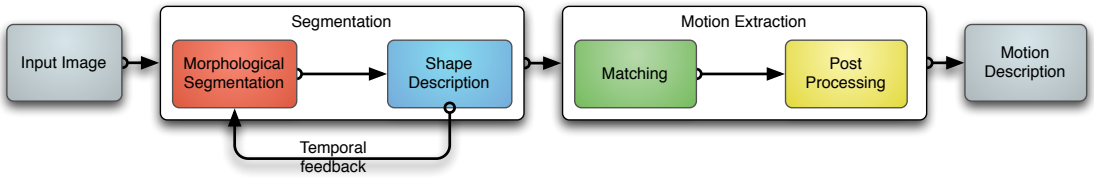
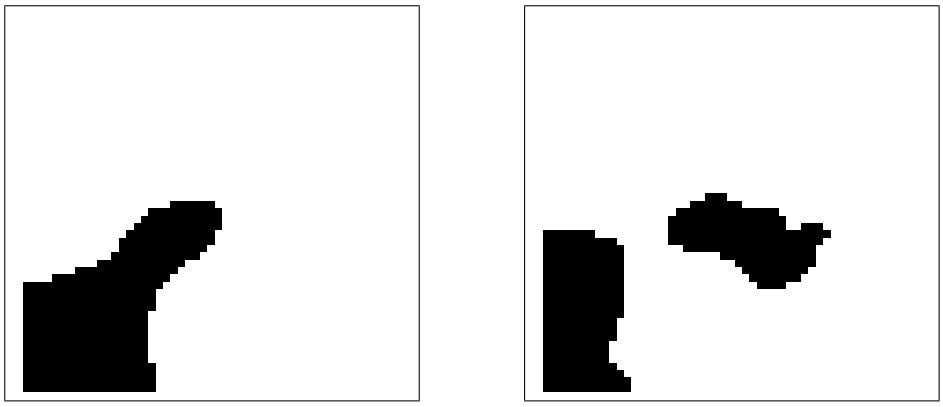


Figure 7.1: The processing pipeline for the curve matching process.



(a) Example Start Frame (b) Example Advanced Frame

Figure 7.2: Example segmented frames. Black indicates ROI, and white indicates background. The start frame was taken from early in the sequence, and the second from closer to the end. Both are indicative of typical segmentations obtained using the methodology described here.

7.1 Segmentation

This section describes the contents of the 'segmentation' box in Fig. 7.1. It begins by discussing segmentation using attribute morphology in (section 7.1.1), before moving on to the issue of how to compactly and efficiently represent and describe the segmented areas in (section 7.1.2).

The purpose of the segmentation stage is to partition features into two classes. Namely those which are interesting, and those which are not. In this case, the features (called regions of interest (ROI)) are blobs of electron density enhancement which are not part of the background, and the rest is, by definition, the background. The output from this segmentation stage could be a map of labelled areas, or simply a boolean mask, indicating the ROI.

7.1.1 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. In terms of greyscale morphology, and imagining the grey-level 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. 7.3 shows an example image cross-section, illustrating the concepts of level-sets and local maxima. For example, in Fig. 7.3, 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 ($\mathbf{E}(\ldots)$) between sets is written as $\mathbf{E}(C_1 \rightarrow C_2)$. The right hand panel of Fig. 7.3 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 metrics such as area or contrast, (see below), perimeter or moment of inertia.

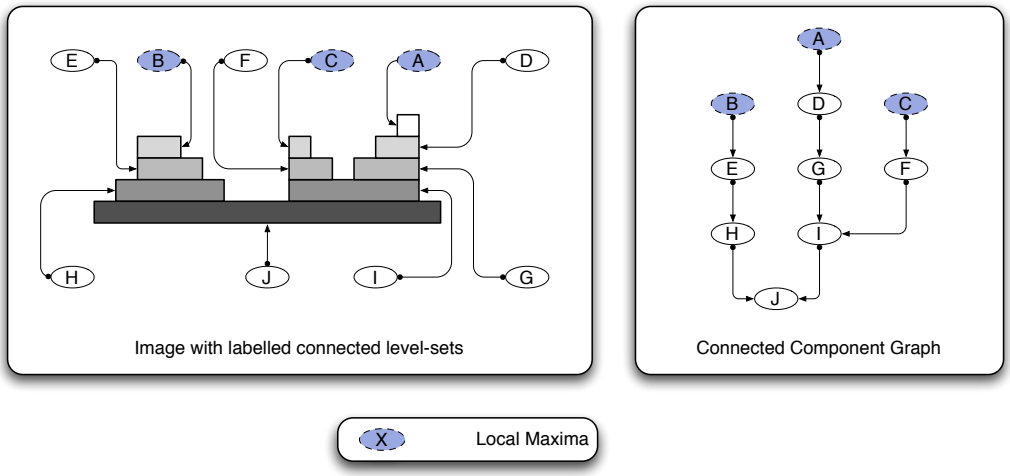


Figure 7.3: An example greyscale image cross-section with labelled connected level-sets, and connected component graph showing the relationship between the various level-sets.

One such situation 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, see, for example Acton and Mukherjee [2000], Mukherjee and Acton [2002].

Area opening removes bright or ascending objects which do not meet a specified minimum area, and area closing removes descending objects which do not meed the 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 starting at the bottom. Fig. 7.4 shows how an area opening using an area of three would alter an example image, and its connected component graph.

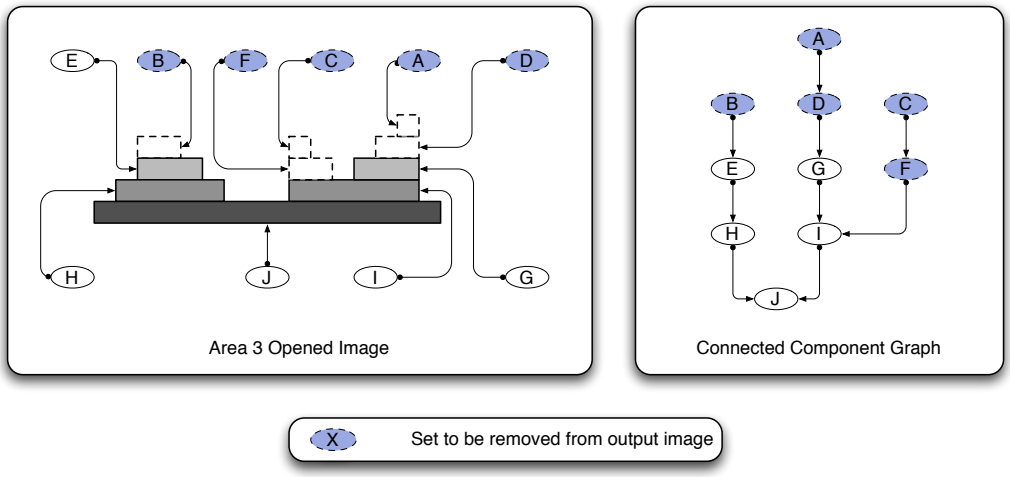


Figure 7.4: The example cross-sectional image from 7.3 showing the effect of an area closing with an area of three (assuming the image depth is only one pixel).

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 uses of the set's contrast. In the case of opening, the contrast is the difference between 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. 7.5 shows how a contrast opening would alter an example image, and its connected component graph.

Using contrast as the morphological attribute has advantages over areas based approaches in images where high gradients are present. As where the gradient is high, a small range of areas can correspond to a large vertical extent. In cases where this could be a problem, using contrast instead of area measures could be beneficial. This is described in more detail in Fig. 7.6.

There is no formal definition of the area or contrast of an electron density enhancement. For this reason, a semi-manual approach was used to decide on the contrast to be used for the segmentation. In this approach, the images were first segmented by hand, in order to establish a consistent set of ground truths. The contrast attribute producing images most closely resembling the hand segmented images was then chosen as the parameter to be used

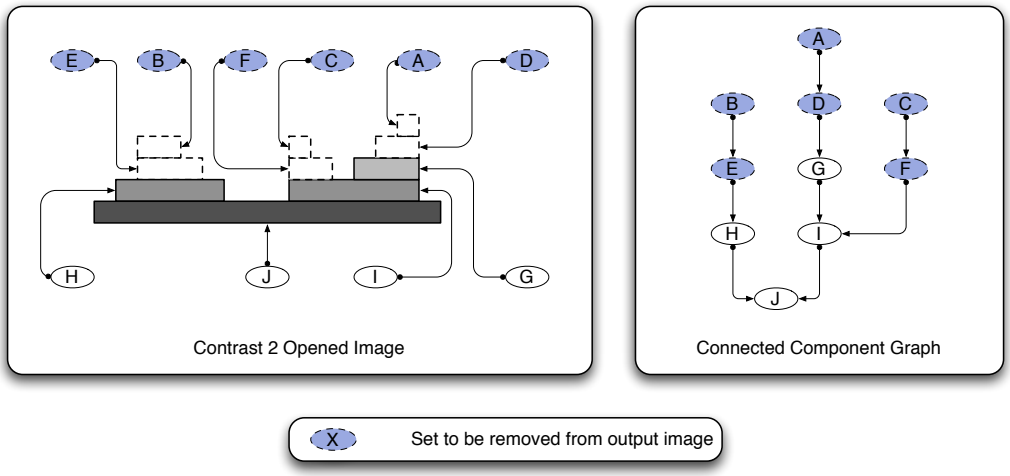


Figure 7.5: The example cross-sectional image from 7.3 showing the effect of an contrast closing with a contrast of two.

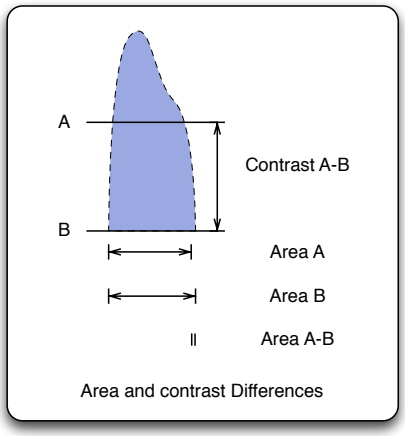
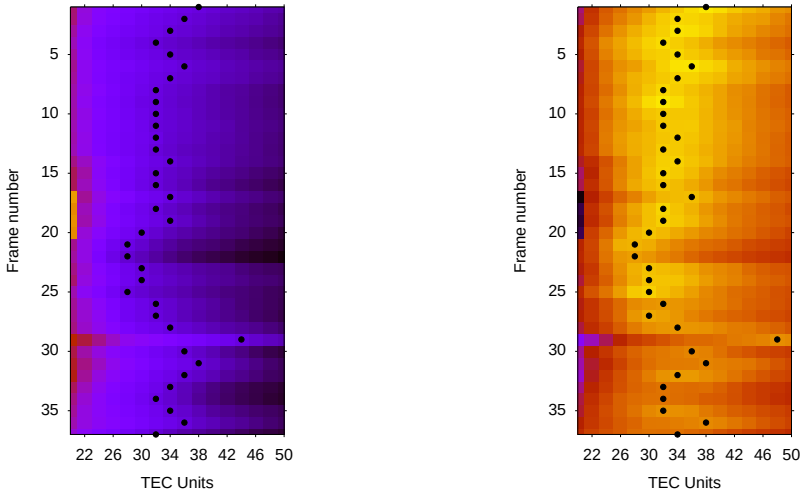


Figure 7.6: The difference between contrast and area parameters. The area can be very insensitive to height, leading to a small change in area consuming a large vertical section of image.

in the machine segmentation. The degree of resemblance was assessed using the CCC, and normalised area differences. Both of these techniques yielded very similar results. Fig. 7.7 shows how the area difference and correlations as images. Through this process, the best optimal[1] closing value for contrast was found to be 33.

Once the best staring value for contrast had been found, a system making use of feedback was designed. First, each frame is segmented. This consists of a contrast closing to remove

[1] in terms of producing the closest results to the ground truth segmentations



(a) Area Comparison (b) Correlation Comparison

Figure 7.7: 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.

the background, followed by an area opening to remove small features which are considered noise. For each frame following the first, the following procedure occurs: check if the area of foreground regions (ROI) differs from the previous frame by more than 10%. If it does, and is larger, then increase the contrast, and re-segment the frame. If it is smaller, then decrease the contrast, and re-segment the frame. By limiting the maximum number of iterations to 10, it can be ensured that the system cannot be overly skewed by anomalous frames. This process is illustrated in Fig. 7.8.

This process results in similar outputs to the ground truth images whilst maintaining area consistency between frames. However, in a few cases, the segmentation is too generous, meaning that some areas which are separate in the hand-segmented images are left joined by feedback process. This occurs when saddle-shaped structures join features.

To mitigate this problem, a further blob-separating 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 effectively joining two areas were severed. This is illustrated in Fig: 7.9.

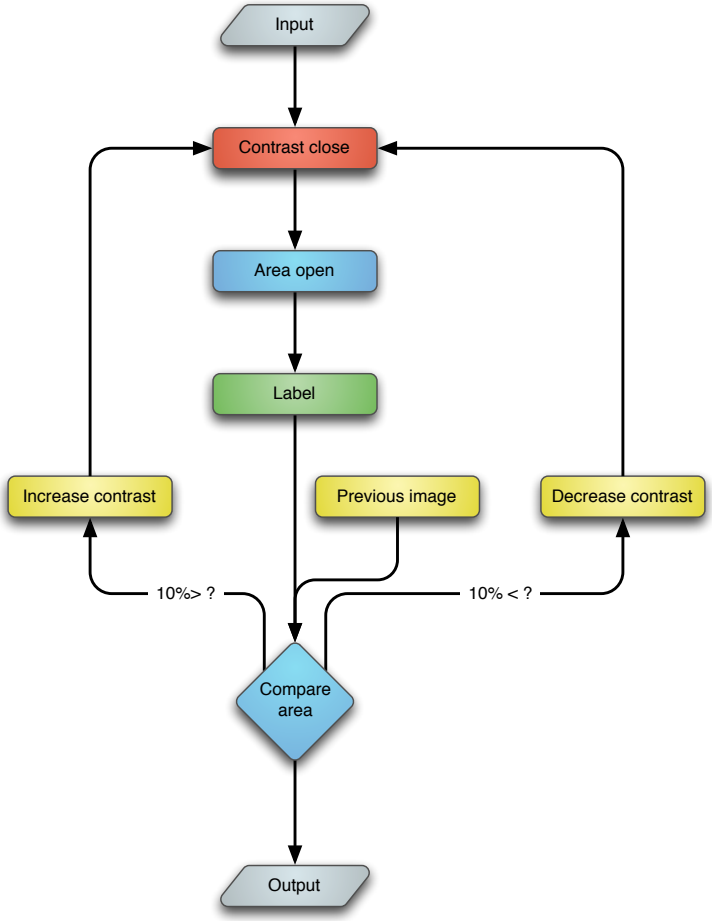


Figure 7.8: Segmentation process showing area based feedback.

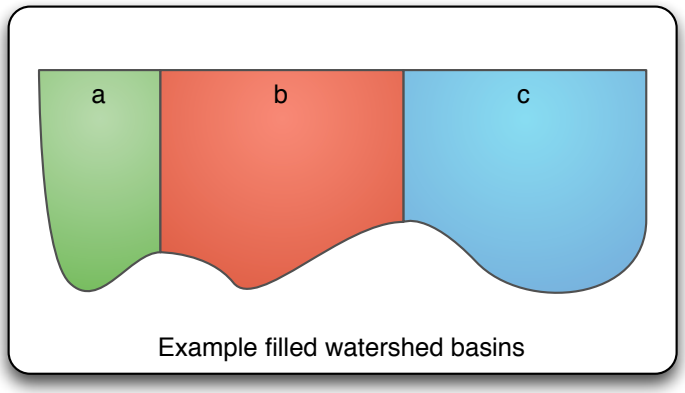
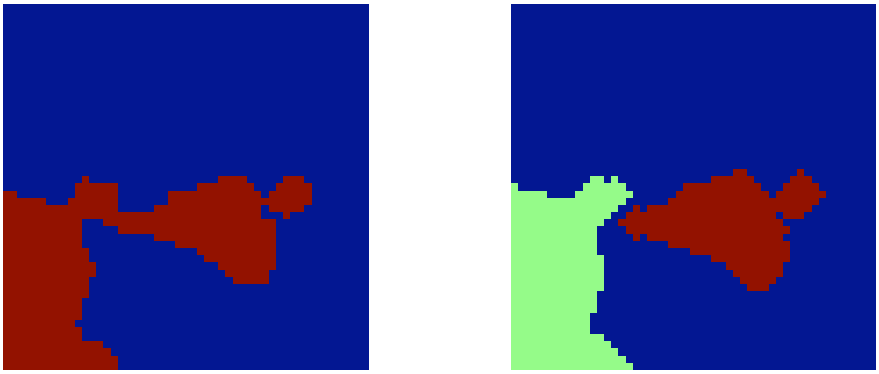


Figure 7.9: Cross-section showing how the watershed transform labels areas. Labels are propagated upwards from local minima, until they meed another adjacent label.



(a) No Post-processing (b) Watershed Transform Post-processing

Figure 7.10: The effect of using the watershed transform to split blobs which are joined by saddle points. Fig. 7.10a shows an image which has not been post-processed using them watershed transform and Fig. 7.10b shows an image which has.

7.1.2 Shape Description

Shape descriptors are methods of describing regions in a compact and efficient way. They also tend to allow various metrics to be easily computed. The shape descriptor used in this study was a simple coordinate list. From this list format, extracting other descriptors, such as histogram or chain-codes is straight-forward. All of the regions described were considered to have closed-contours. This means that they form contours, and that the list of coordinates can be considered as loops, with arbitrary start positions. It is therefore important to remember that the first and last elements are considered adjacent.

This study used simple coordinate lists to describe the contours, there were fitted with smoothing splines to decrease the effect of segmentation errors and noise, and to allow easy calculation of coordinates between boundary samples.

Now that segmentation has been described, the next step in the system is to perform shape boundary matching, and derive motion vectors. This is described in the following sections.

7.2 Introduction to Shape Context Matching

shape context (SC) matching is a method which 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. solving correspondences between points on their boundaries;
2. using 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*, these vectors will represent the relative motion between frames.

This section discusses using SC matching of shape boundaries to estimate the motion of ionospheric TEC enhancements.

7.2.1 Shape contexts

SC 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)

(a) Shape Boundary

(b) Shape context histogram

Figure 7.11: (a) a shape boundary. 7.11b the shape context histogram created from the boundary in (a), using the red-dotted point as the origin (bottom-left). The x-axis represents the angle θ and the y-axis represents $log(r)$. Darker colours indicate higher counts.

7.2.2 Shape Context Matching

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, $C_{ij} = C(p_i, p_j)$ 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:

$$C_ij \equiv \chi^2(p_i, p_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \tag{7.1}$$

As this only compares histogram bins with corresponding bins in the second histogram, other methods, such as 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:

$$K(h_i, h_j) = \sum_{l=0}^{L} |d_l(\mathbf{x})|, \tag{7.2}$$

where,

$$d_0(\mathbf{x}) = h_1(\mathbf{x}) - h_2(\mathbf{x}) \tag{7.3}$$

$$d_1(\mathbf{x}) = [d_{l-1} * \phi(\mathbf{x}, \sigma)] \downarrow_2 \quad l = 1, \ldots, L \tag{7.4}$$

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

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:

$$H(\pi) = \sum_{i} C(p_i, p_j), \tag{7.5}$$

subject to π being a permutation, that is, 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.

The result of minimising equation 7.5 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, however finding suitable metrics is problematic, as many such as curvature, can be very noisy, and introduce 'pits' into the cost matrix.

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: their relative motion.

7.2.3 Boundary Transformations

An additional step which can be performed 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 transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ which allows arbitrary points from one shape to the other. The

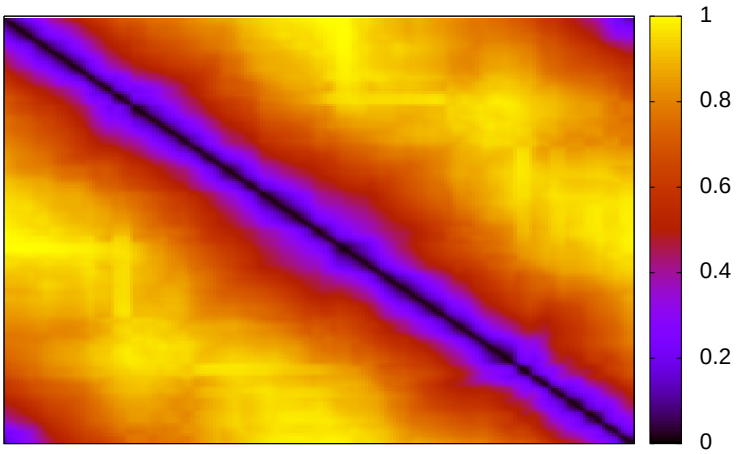


Figure 7.12: An example cost matrix created using the diffusion distance histogram metric, weighted by the distance between matching points.

two most commonly used techniques to choose T are least-squares fitting a standard affine model, and the estimation of warping planes using radial basis function (RBF) interpolation. This interpolation method is described in more detail in section 2.5. Regularisation can also be applied to smooth the warping planes as needed.

Affine Model

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

$$T(\mathbf{x}) = A\mathbf{x} + o, \tag{7.6}$$

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:

$$\hat{A} = (Q^{+}P)^{t}, \tag{7.7}$$

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 7.6. Fig. 7.13 show an example vector field generated

by and affine model.

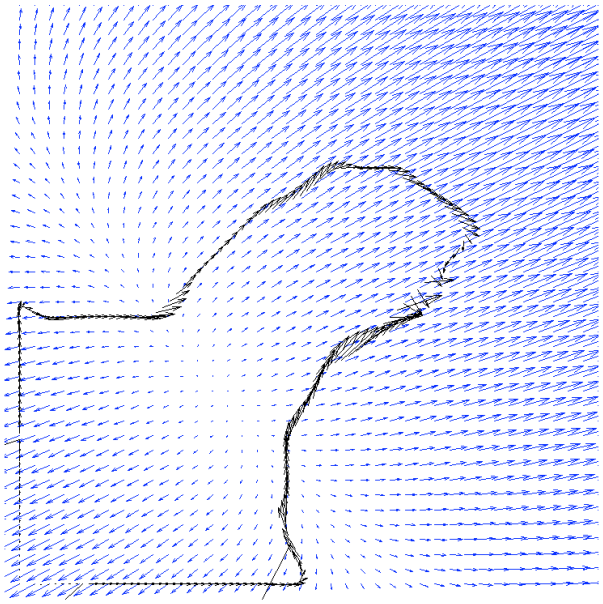


Figure 7.13: Shown in blue, an example vector field generated by least squares fitting an affine model of the form in 7.6. The black vectors show estimated warp vectors.

RBF Fitting and Regularisation

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

$$T(x, y) = (f_x(x, y), f_y(x, y)) \tag{7.8}$$

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:

$$A + \lambda I \tag{7.9}$$

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.

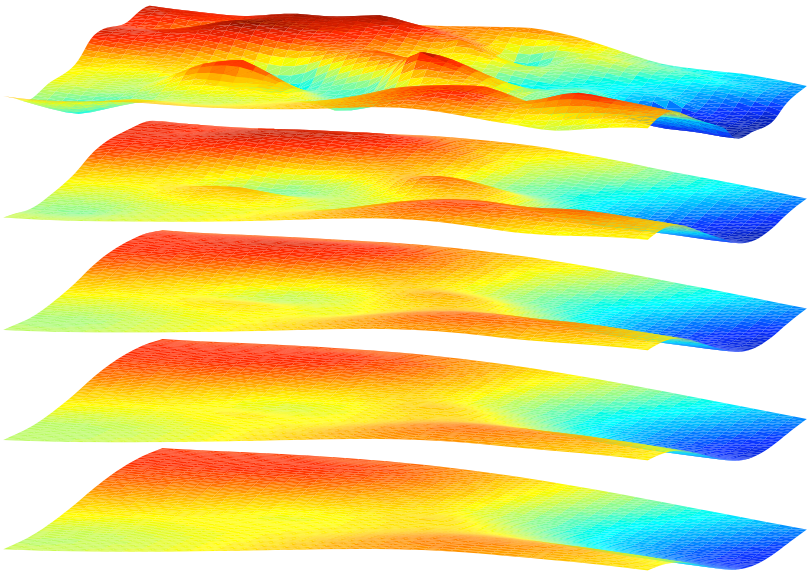


Figure 7.14: The effect of adjusting the regularisation parameter (λ) in steps of 0.25, starting at zero (top), and ending at one (bottom).

Fig. 7.15 show an example full-field of RBF interpolated vectors.

7.3 Implementation Issues

The following sections discuss some issues relating to the use of shape context matching for estimating the motion of TOI during ionospheric storms. Because of the novel use of shape matching techniques for ME, some problems and trade-offs can be expected.

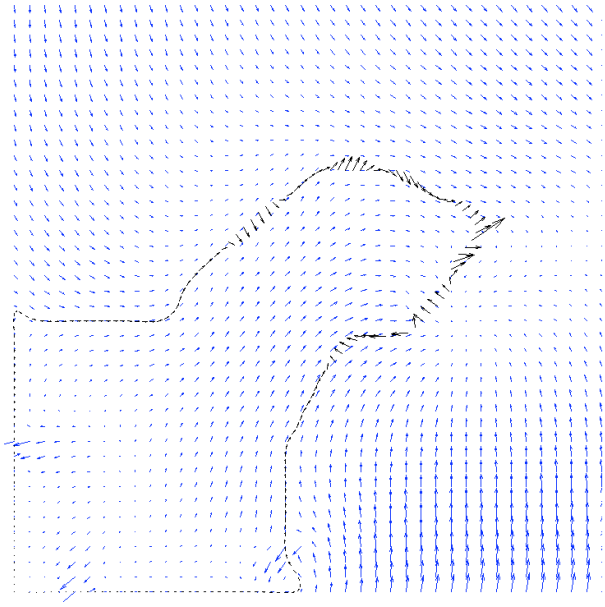


Figure 7.15: Shown in blue is an example vector field generated by RBF interpolation using a TPS function. The black vectors show estimated warp vectors from the calculated alignment.

7.3.1 Depletion Effects

As described in section 5.1, 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 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. 7.16.

This problem occurs towards the end of the image sequence under examination. Fig. 7.17 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.

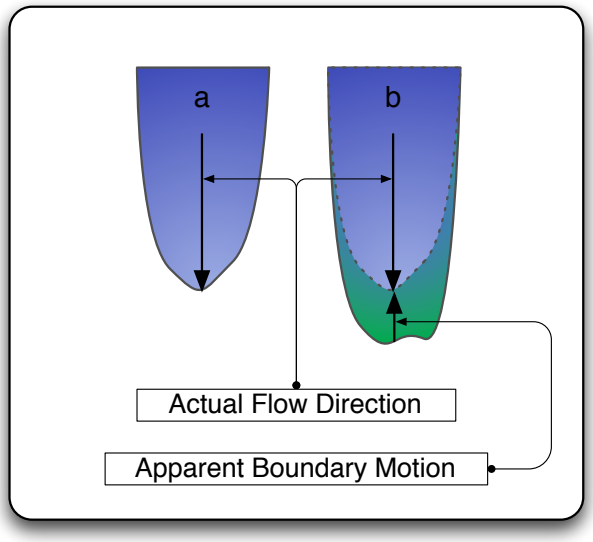
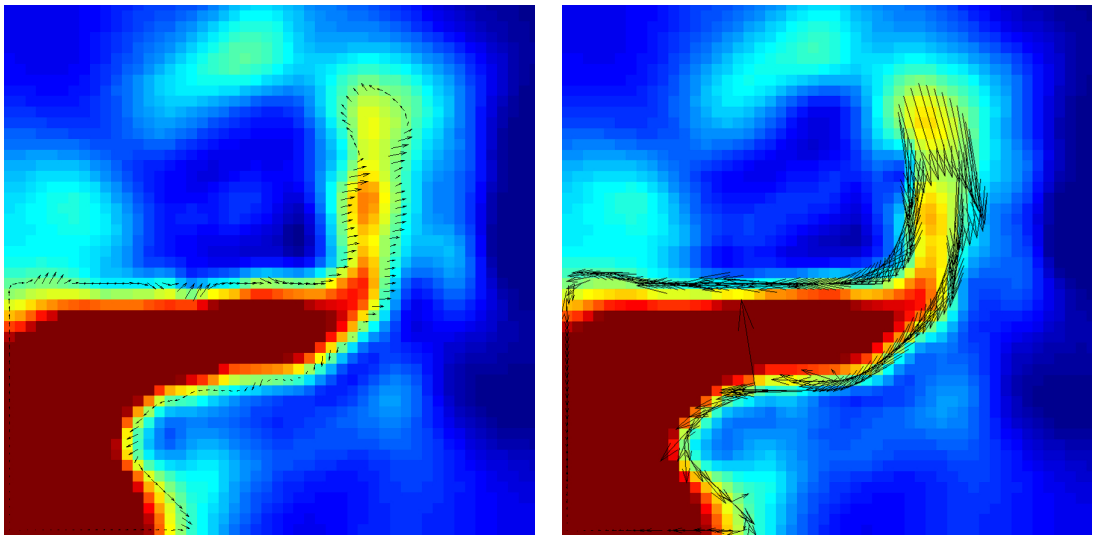


Figure 7.16: A diagram showing how a change in depletion rate can alter the apparent direction of flow.

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 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. Examples of two different threshold values can be seen in Fig. 7.18, (b) shows slightly less marked vectors than would be ideal, and (a) shows more than would be ideal. Fig. 7.18 shows histograms of the displacement vectors relative to the previous frame's displacement, and sheds some light on the problems with the marked vectors. In the red histogram, where there is no retreating problem, the range of different magnitudes present is small, where as in the green histogram, where there *is* a retreating boundary, there is a much larger range of values, with a fairly flat distribution. This is the reason for the high sensitivity to the chosen threshold level.
- Marking vectors in a manor similar to the above method, but making use of angle instead



(a) (b)

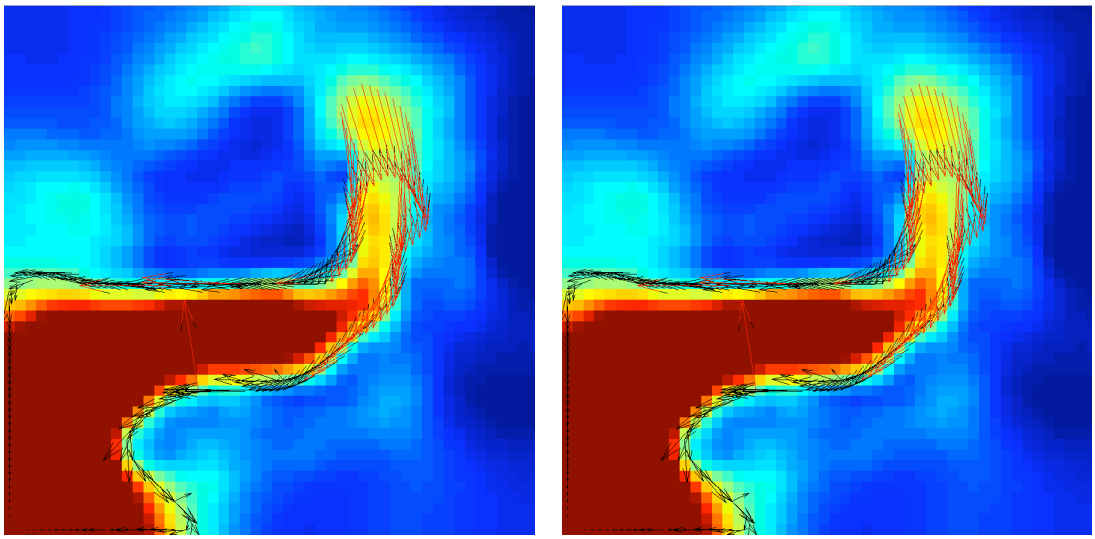
Figure 7.17: Example frames illustrating the retreating snout problem. In (a) the boundary reaches further up the frame than in (b), where due to a slight depletion in level, the boundary has retreated.

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. An example of this method is shown in Fig. 7.20a.
- 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 point a seemingly *wrong* direction. This method is illustrated in Fig. 7.20b.
- Replacing the detected vectors with new vectors generated by interpolating the previous



(a) (b)

Figure 7.18: In red, are examples of detected retreating vectors using thresholding the difference between the current frame and interpolated previous frame's vectors. (a) shows vectors marked when using a threshold value of 8, and (b) shows a those marked when using a value of 9.

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. The result of the interpolated case can be seen in Fig. 7.20c.

- 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.

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 Figs. 7.20a – 7.20c at the boundaries of regions where vectors are being replaced.

Fig. 7.21 shows example frames with boundary motion vectors plotted at the boundary points (where they were estimated). As an additional step, the vectors can be interpolated to cover the area within the boundaries. Regularisation can also be used to smooth the vectors as necessary. However, the reasonably low number of boundary vectors tends to ensure that the surface is fairly smooth. Fig. 7.22 shows some example frames with vectors interpolated to

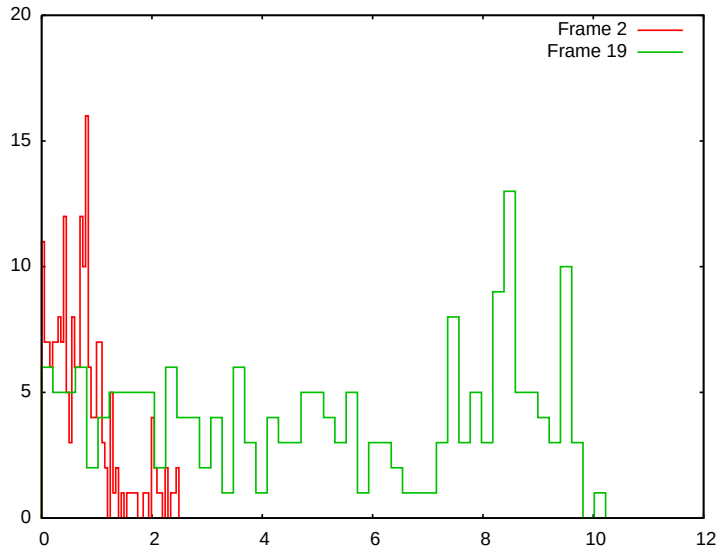


Figure 7.19: Histograms of displacement relative to previous frames interpolated vectors for a 'normal' frame (in red) and a frame showing the retreating snout problem (in green).

cover the entire shape area.

7.3.2 Other Issues

Shape context matching requires boundaries to have equal numbers of samples. This can cause problems when the boundary lengths vary across frames. The problems can manifest as an apparent rotation of the matched shape relative to the first, such that there is a tendency for all of the vectors to point in one direction or another relative to the boundary. This effect is illustrated in Fig. 7.23.

There is a possibility that this problem (referred to as the *phantom rotation* problem, from here) can be detected by examining the direction of vectors relative to vectors formed by subtracting boundary points. This is because the boundary vectors generally start in approximately the same position, and then continue in a clockwise rotation. Any bias relative to the vectors between points might indicate that the vectors tend to point one way along the boundary.

No.. this idea doesn't work! The rotation often spans several samples.

The estimated vectors can be assumed to consist of noise, phantom rotation and actual object

motion. If the phantom rotation can be detected and removed, and if regularisation is able to mitigate the noise. This should leave a reasonable motion estimate.

The main problem with phantom rotation is that it is difficult to detect. Fig. 7.24 shows some example boundary sections where it occurs, and (b) shows where it does not (a). In addition, Fig. 7.25 shows an example situation where it occurs on approximately half of the boundary positions.

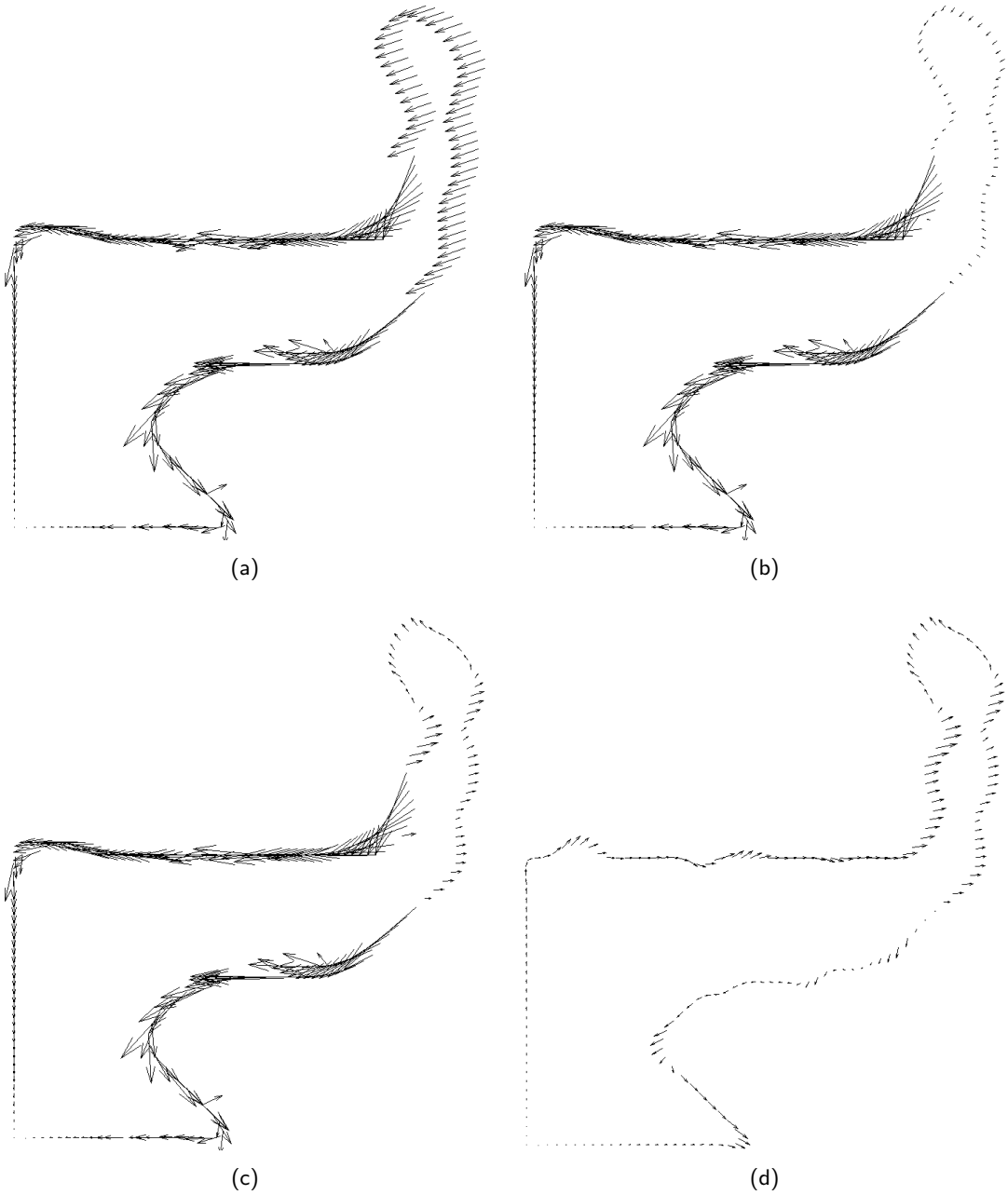
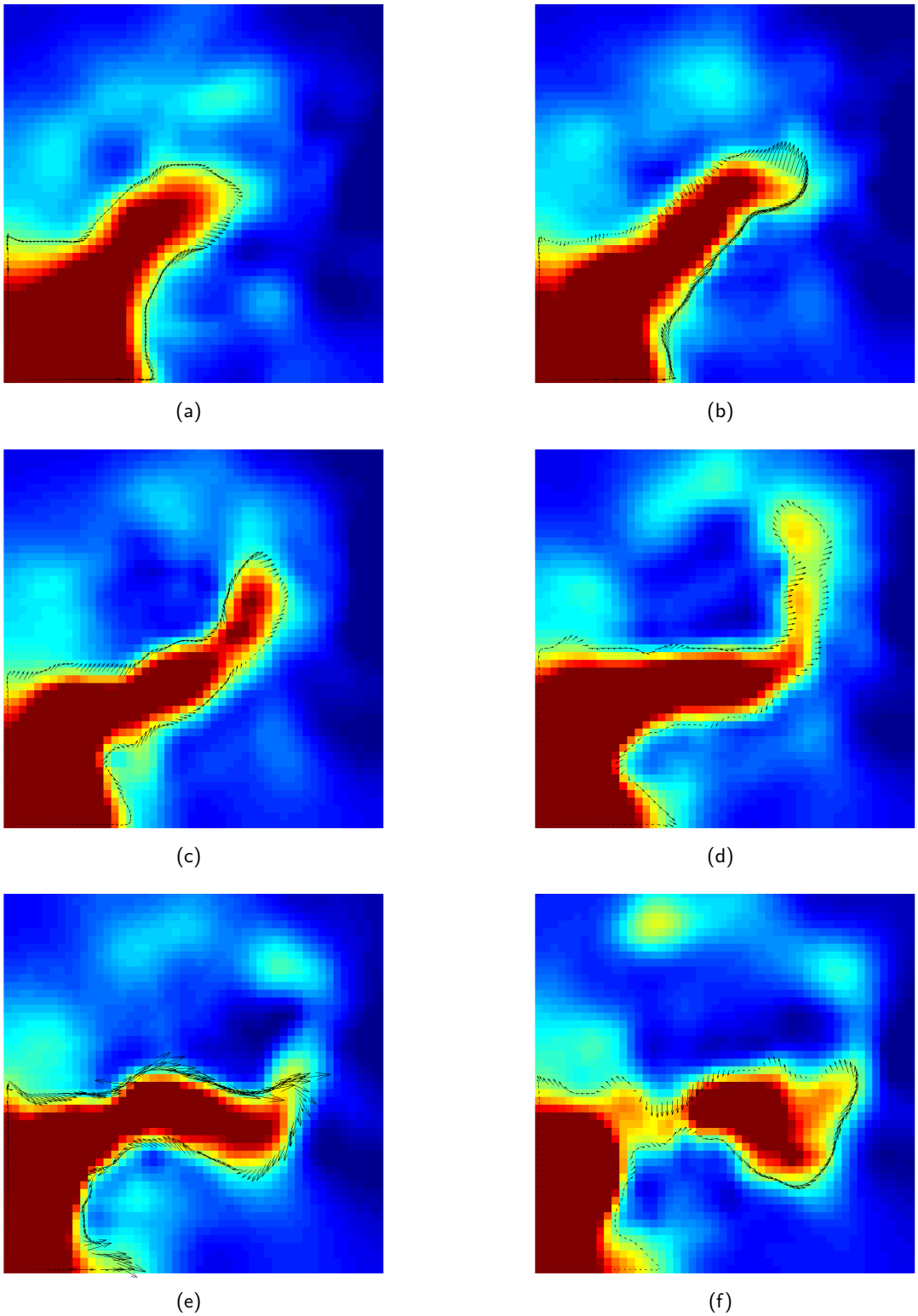
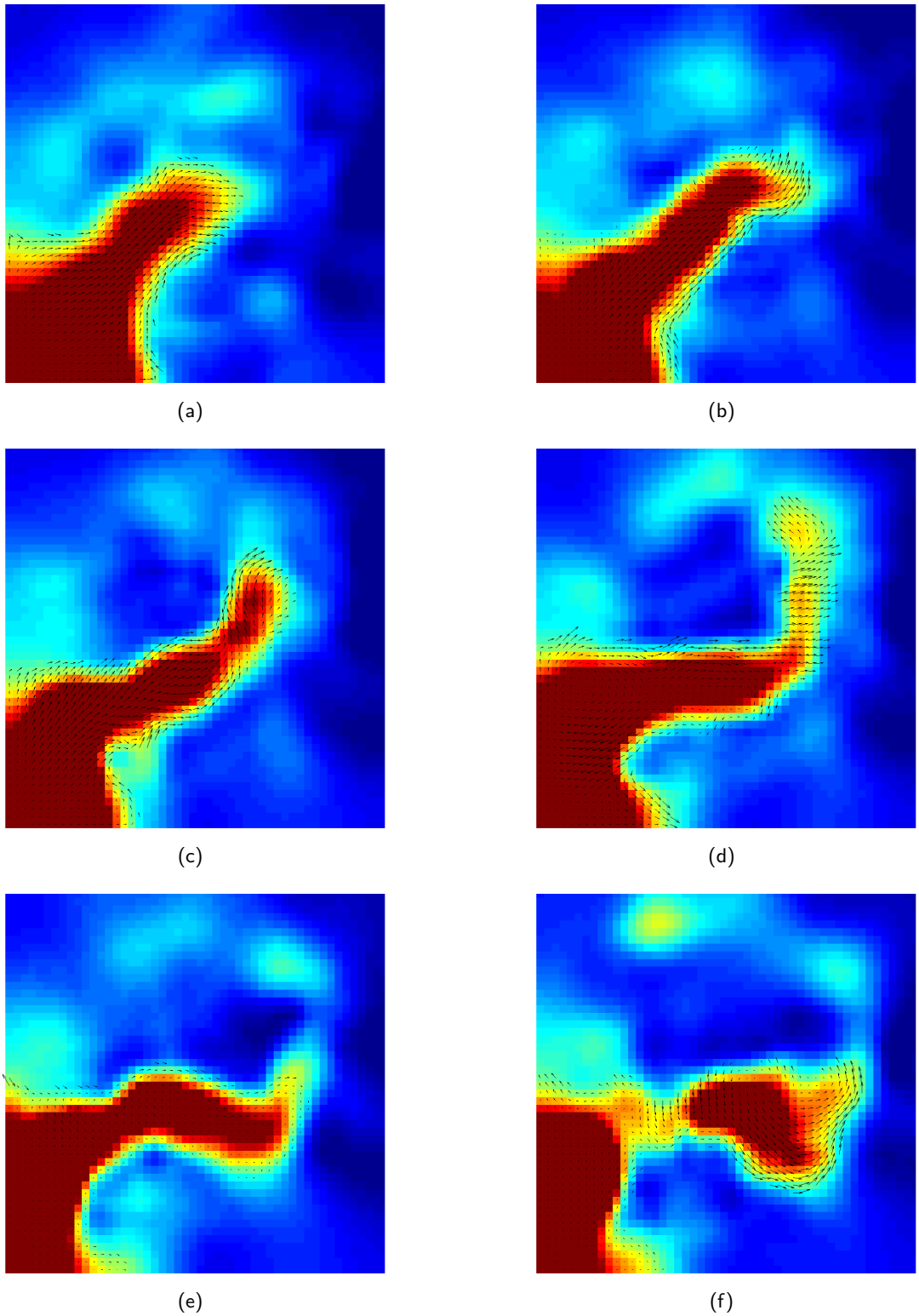


Figure 7.20: The effect of replacing all of the marked vectors in 7.18a with: (a) the median of the remaining vectors; (b) the fitted affine model of the previous frame's vectors; (c) the previous frame's vectors interpolated using RBF TPS interpolation at the positions of the current frame's vectors. (d) shows the effect of replacing all of the vectors in the frame with the previous frame's vectors interpolated onto the current frame's vectors positions.



(a) (b) (c) (d) (e) (f)

Figure 7.21: Example frames with overlaid vectors. Frames displayed are equally spaced throughout the data-set, from 1800-1200.



(a) (b) (c) (d) (e) (f)

Figure 7.22: Example frames with overlaid interpolated vectors. Frames displayed are equally spaced throughout the data-set, from 1800-1200.

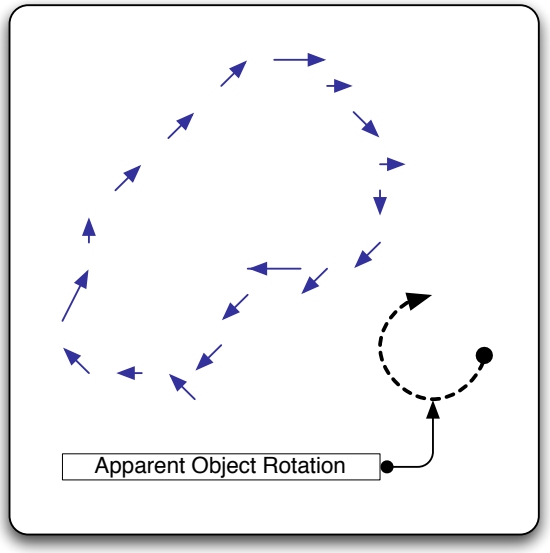
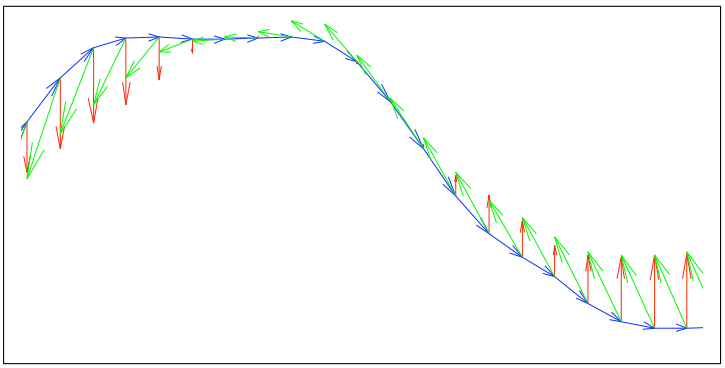
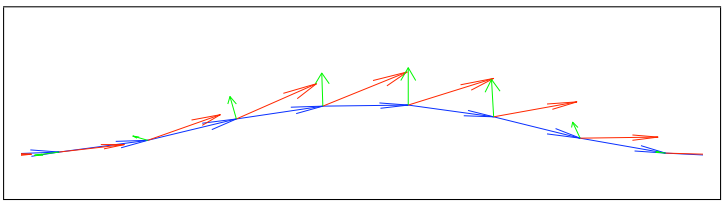


Figure 7.23: A diagram showing the problem of apparent object rotation.



(a)



(b)

Figure 7.24: Examples of vectors. In blue, is the difference between sample points, in red is the detected motion and in green is the difference between the two. (a) shows a situation where there is no phantom rotation, and (b) shows a situation where there is. In this case, the green vectors would probably be a more appropriate representation of the motion.

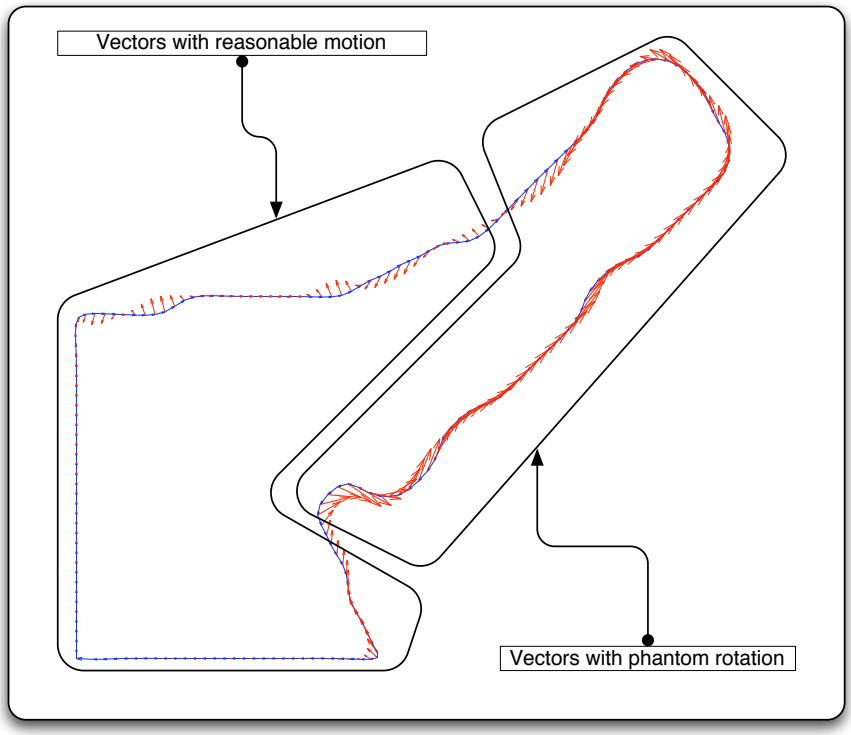


Figure 7.25: An object boundary where some of the vectors properly describe the boundary motion, and some of them show phantom rotation.

7.4 Conclusions

References

S. T. Acton. Fast algorithms for area morphology. *Digital Signal Processing*, 11(3):187–203, 2001. URL http://www.sciencedirect.com/science/article/B6WDJ-458W494-N/2/45ff4488eb45899f5480eb1cb21a076b.

S. T. Acton and D. P. Mukherjee. Scale Space Classification Using Area Morphology. *IEEE Trans. Image Process.*, 9(4):623, 2000.

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.

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.

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.

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 Transactions 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 Transactions on Geoscience and Remote Sensing*, 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, Astronomiches Instutut 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.

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.

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. Glacier surface motion computation from digital image sequences. *IEEE Transactions on Geoscience and Remote Sensing*, 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.

A.N. Evans. Cloud motion analysis using multichannel correlation-relaxation labeling. *IEEE Geoscience and Remote Sensing Letters*, 3(3):392–396, July 2006.

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. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(7): 2153–2164, July 2008. ISSN 0196-2892. doi: $10.1109/\mathrm{TGRS}.2008.916642$.

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.

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: $10.1109/\mathrm{TGRS}.2007.895414$.

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. EGNOS test bed ionospheric corrections under the October and November 2003 storms. *IEEE Transactions on Geoscience and Remote Sensing*, 43(10):2283–2293, 2005.

V. Hlavac, M. Sonka, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS, 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 Inernational 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.

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 Transactions on Geoscience and Remote Sensing*, 45(7):2325–2335, July 2007. ISSN 0196-2892. doi: 10.1109/TGRS.2007.896546.

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 Transactions on*, 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.

E. Maeland. On the comparison of interpolation methods. *Medical Imaging, IEEE Transactions on*, 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 modernsignal and image processing. *Proceedings of the IEEE*, 90(3):319–342, 2002.

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 Transactions 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 Transactions on Geoscience and Remote Sensing*, 43(2):321–326, 2005.

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.

BD Ripley. *Spatial Statistics*. Wiley-Interscience, 2004.

P. Sakov. Natural neighbour interpolation software, 2005. URL http://www.marine.csiro.au/~sakov".

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.

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.

I. Stanislawska, G. Juchnikowski, L. R. Cander, L. Ciraolo, P. A. Bradley, Z. Zbyszynski, and A. Swiatek. The kriging method of TEC instantaneous mapping. *Advances in Space Research*, 29(6):945–948, 2002.

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.

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.

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. URL http://www.informaworld.com/10.1080/014311697218980.

I. T. Young and L. J. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44:139–151, 1995.