Data-driven Reconstruction of Sparse Remotely Sensed Data
PhD Transfer Report

M. P. Foster

November 27, 2006

Abstract

This report looks at the reconstruction of sparse remotely sensed data using a new technique know as normalised convolution. Traditional reconstruction techniques are examined and all are found to not perform well when faced with highly sparse data.

Normalised convolution and adaptive normalised convolution are introduced as alternatives to these traditional techniques, and found to perform well on highly sparse data, suggesting their suitability for the reconstruction of geophysical data sets.

An initial study using Global Positioning System derived measurements is presented, and normalised convolution based techniques are found to perform better than standard interpolation methods throughout the test. Optimum parameters are also derived, and relationships between input data and these parameters are found to be too sensitive in their current form.

Finally, conclusions and further work are presented.

Contents

List of Figures

List of Symbols

List of Symbols

$\#$	Count operator
a_i	Polynomial Coefficient
b_r	Receiver interfrequency bias
b_s	Satellite interfrequency bias
B_z	IMF z component
L_1	L_1 phase
L_2	L_2 phase
ℓ_p	Minkowski distance of order p
ρ	Autocorrelation
γ	Autocovariance
N	Vector Length (or number of pixels)
Q_{ij}	Input Sample
r_e	Radius of Earth
r	Radius of shell
θ	Elevation angle

1

Reconstruction of Geophysical Datasets

Remote sensing is the process of gathering information about an object or area from a distance [1]. This could include satellite imagery, such as the examples listed below, but the definition does not end there. There are a great many sources of remotely sensed data, encompassing many different fields, methods of measurement, spatial and temporal resolutions and distributions.

Definition 1 (Image)**.** For the purposes of this report, an image is a array with at least two dimensions, where *some* or all of the elements contain intensity values representing some kind of data. This includes traditional greyscale and colour images (which constitute two- and three-dimensional arrays respectively), images of spectral content which is outside of human vision, and data which does not represent electromagnetic radiation at all.

Types of remotely sensed imagery include [2]:

- Monospectral: images are composed of data from a singal spectral band, such as near infrared.
- Multispectral: images are composed of data from several spectral bands, which are sensed simultaneously. See figure 1.1, which shows a visible light image of Death Valley, US.
- Hyperspectral: images are composed of data from many vary narrow continuous spectral bands, from visible wavelengths, through to far infrared. Hyperspectral imagery attempts to give each data point a continuous spectrum of reflectance.
- Synthetic aperture radar: images are constructed using radar systems which make used of array antennas to synthesise very large apertures, with narrow, steerable beams. This increases the resolution far above that of systems using non-synthesised actual antennas. See figure 1.2 which shows a SAR image of Oregon, US.

Data could also have been gathered from spcific instruments, on-board satellites or probes, as with the following examples:

- The Doppler Wind Experiment on Cassini-Huygens sensed the wind velocity of Titan's atmosphere during the probe's descent, using instruments on both the Huygens probe and the Cassini Orbiter. This information was then relayed back to Earth, for analsyis.

- The Halogen Occultation Experiment on the Upper Atmosphere Research Satellite makes use of satellite solar occultation to measure solar attenuation. The attenuation profiles are then processed to extract data on the composition of the atmospheric limb which was measured.

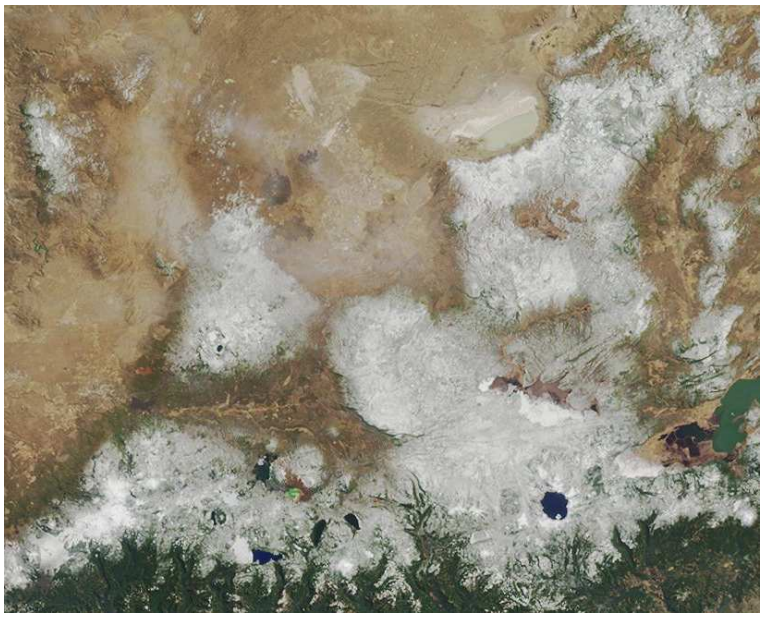


Figure 1.1: Colour satellite image of Oregon, US. (Public domain image courtesy of NASA)

Remotely sensed data sets are often sparse because of the way in which they are collected. For example, data from global positioning system (GPS) satellites are only available on specific paths between the satellites and visible ground stations (see figure 1.3, which shows the positions of 20 GPS ground stations in North America). Alternatively they might have come from ground based measurement stations which are spaced apparently randomly over a large area. Data could also have come from floating transducers, dotted all over the Earth's oceans, and drifting with the wind and currents, like the ARGO floating sensors ([?]). Figure 1.4, shows the positions of the sensors as of September 21st, 2006. In situations where it is impractical or prohibitively expensive to obtain blanket coverage, it is generally necessary to settle for a sampled look at things, and this often leads to sparse data. For this reason, it is useful to have as many techniques as possible ready to extract as much useful information as is available from the data.

Definition 2 (Degree of Sparsity)**.** According to Karvanen and Cichocki [3], a given data sets degree of sparsity can be quantified[1] using ℓ^0 norm, which is defined for a vector x, by:

[1] for the noiseless case, where samples present are not to be considered noise, as opposed to the case where

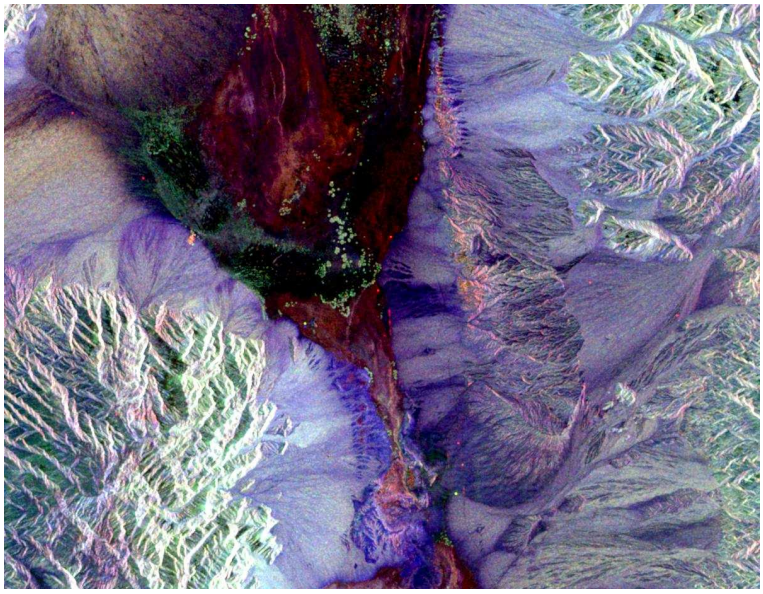


Figure 1.2: Synthetic aperture radar (SAR) image of Death Valley, US (Public domain image courtesy of JPL)

$$||\mathbf{x}||_0 = \frac{\#\{j, x_j \neq 0\}}{N} \tag{1.1}$$

Where # increments a counter whenever its argument evaluates to *true*, and N is the number of elements in the vector being measured. The degree of sparsity is therefore simply a fraction which describes the number of non-zero elements in a given vector. However, this conveys no information about the distribution of elements within the vector, and so in some cases, it may be preferable to define data sparsity using statistical moments, such as kurtosis.

The degree of sparsity of a given set of data can also be a consequence of the way it is projected into a form ready for reconstruction (discussed below). Quite often, data will consist of a set of scatter points, with each element being defined at a specific point in space. Each point must be projected into a matrix before any further processing can be done, where each matrix element represents a tile of space, with vertices defined by a grid or spatial coordinates. The area of the matrix elements, (and therefore the resolution of the grid) will determine the sparsity of the resultant matrix, as demonstrated by figure 1.5. This contrived example shows a set of data (the black circles) being projected first onto one grid, and then onto a second grid with double the resolution. The grey areas show where each point would be placed in the output matrix.

1.1 Reconstruction Techniques

Often, in order to make use of sparse data, some form of processing must be carried out in order to convert it into a full set of points. This process is known as reconstruction. Reconstruction

samples may be noisy

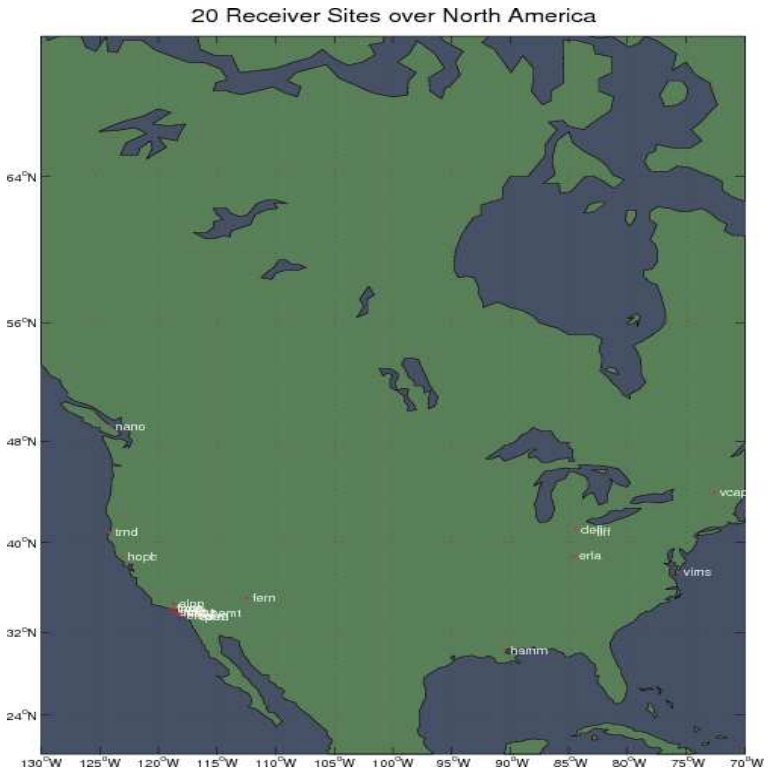


Figure 1.3: 20 GPS Receivers in North America

(or interpolation) is the process of recovering data points which lie between a set of samples. It can therefore be described as constructing a function which closely fits the data points, and should therefore be a good approximation of the missing points. This can be approached in two main ways: directly, and indirectly. Direct interpolation is discussed in the following section and involves specifically aiming to find coefficients for a given interpolating function. Indirect interpolation includes all techniques which are not direct, which can be broadly split into two further categories, *data-driven* and *model-driven*.

Model driven reconstruction uses available data to seed or bias models, which are then used to derive output values for the reconstruction. This kind of reconstruction has been used in many fields, for example palaeoecology (see [4], where a temperature model and pollen measurements were used to reconstruct treeline data), medical imaging (where intensities are modelled to help improve scan outputs), computer vision (see, for example Derou et al. [5]) and oceanography (see Guinehut et al. [6]) and other geophysical uses (for example GPS, see Mannucci et al. [7]).

The following section describes some common 'traditional' or direct interpolation techniques, before moving on to describe a class of techniques known as Kriging, which were designed to reconstruct geophysical data.

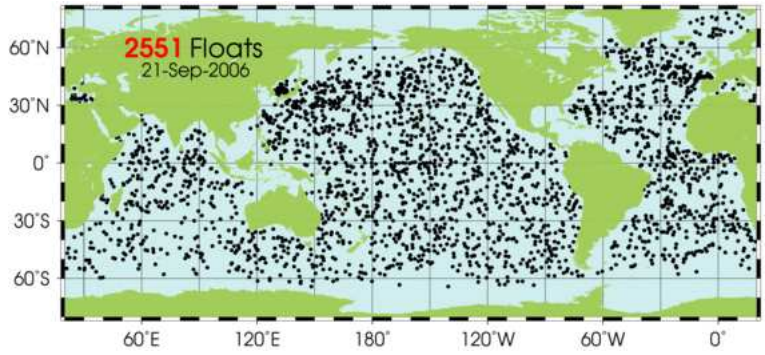
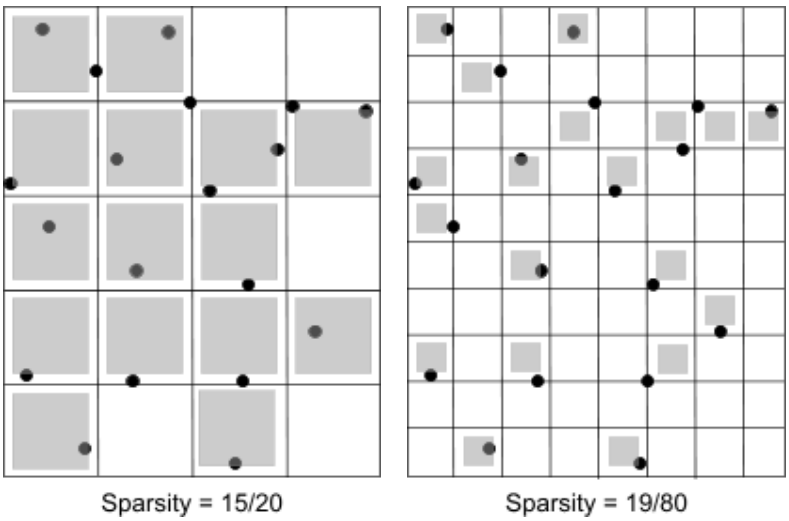


Figure 1.4: ARGO Buoy positions



(a) Low resolution grid $\Longrightarrow$ low sparsity

(b) Higher resolution grid $\Longrightarrow$ higher sparsity

Figure 1.5: Examples illustrating how grid resolution determines data sparsity

1.1.1 Direct Interpolation

Direct interpolation techniques work by attempting to find coefficients for a function, or set of functions, such that the functions pass through all available data points. Direct in this case simply means that the process attempts to find coefficients for the functions being used to interpolate the data set. Indirect techniques use other methods to find and parameterise the functions being fitted to the data. All of the techniques detailed below require data that sits on a regular grid.

Bilinear Interpolation

Bilinear interpolation is the extension of linear interpolation [2] into two dimensions. It works by performing linear interpolation in one dimension, and then repeating the process in the other.

[2] Linear interpolation joins (1D) sample points using straight lines.

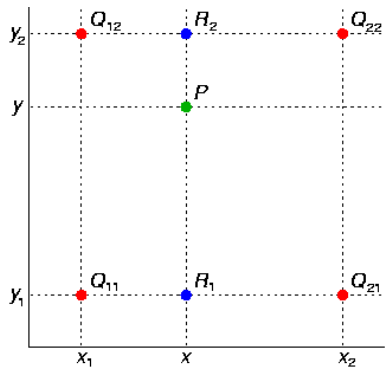


Figure 1.6: Bilinear Interpolation Layout
Bilinear Interpolation Layout. Point P is to be estimated, and Q_{ij} are available input samples. (Public domain image, courtesy of Jitse Niesen)

If the known samples sit on a square grid, with vertices labelled as shown in figure 1.6, then bilinear interpolation can be described by the following polynomial:

$$f(x,y) \approx a_1 + a_2 x + a_3 y + a_4 xy \tag{1.2}$$

Where the coefficients (a_i) are found using:

$$\begin{aligned} a_1 &= Q_{11} \\ a_2 &= Q_{21} - Q_{11} \\ a_3 &= Q_{12} - Q_{11} \\ a_4 &= Q_{11} - Q_{21} - Q_{12} - Q_{22} \end{aligned} \tag{1.3}$$

Linear interpolation is unsuitable for situations where a good representation of high rates of change is necessary, because there is no guarantee of continuity at the edges of the grid.

Bicubic Interpolation

The polynomial basis for bicubic interpolation has 16 terms and takes the following form:

$$\begin{aligned} f(x,y) \quad = \quad & a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + \\ & a_{11}xy + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + \\ & a_{22}x^2y^2 + a_{30}x^3 + a_{03}y^3 + a_{31}x^3y + \\ & a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^2y^3 + a_{33}x^3y^3 \end{aligned} \tag{1.4}$$

Or more simply:

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \tag{1.5}$$

The coefficients can be derived in a variety of ways, however the normal approach uses values at available grid vertices as well as their derivatives and cross derivatives.

When used on a grid, bicubic interpolation is the lowest order interpolation which maintains continuity of the function and its first derivatives across boundaries. When data are highly sparse, problems occur, because of the large number of coefficients which are needed to generate the derivatives.

Spline Interpolation

Spline interpolation is a scheme which uses low-order piecewise polynomials, defined between points known as knots or control points. The name spline comes from the drafting tool of the same name; a flexible strip designed to allow curves to be drawn which maintain good continuity with adjacent curves. Splines my be described as *uniform* if control points are equally spaced, or *nonuniform* if not. When using spines for interpolation, the data points are used as the control points, which means that many of the different types of spline cannot be used - splines which pass through their control splines are called *interpolating splines* and include linear, quadratic, cubic and natural cubic splines (a parametric representation of the physical spline curve mentioned above).

The main advantage of spline interpolation is that it gives results which are free from Runge's phenomenon. Runge's phenomenon is a problem which occurs when interpolating with high-order polynomials, and manifests itself as oscillations at the edges of the interpolation interval. A good introduction to splines can be found in [8, pp. 486–500].

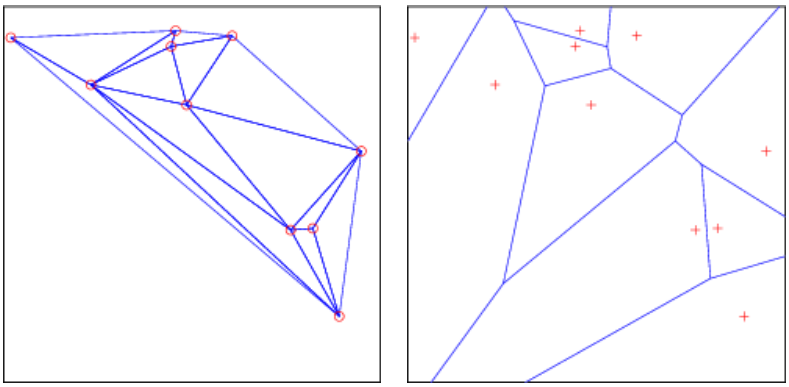
1.1.2 Gridding

Most interpolation schemes require data points to be defined on a regular grid. For this reason it is necessary to adapt and extend traditional interpolation techniques for use when this is not the case The main technique used for interpolating non-uniformly spaced vectors is known as *triangular* or *Barycentric* interpolation. As the name suggests, triangular interpolation involves triangulating the data using a *Delaunay triangulation*, and then performing the reconstruction using the triangle's vertices. Before explaining how Barycentric interpolation works, some definitions are necessary:

Definition 3 (Delaunay Triangulation)**.** The Delaunay triangulation of a set of 2D points is a set of tessellated triangles, with points at the corners, such that no point is inside the circumference of any triangle in the output. The circumference of a triangle refers to the circumference of the circle which is defined by the triangle's vertices - known as the *circumcircle*.

The geometric dual of the Delaunay Triangulation is known as the *Veronoi Tessellation*.

The Veronoi tessellation of a set of points divides the space containing the points into tessellating convex hulls, each containing one point. The space within each hull is closer to the point contained within it, than any other, except for at the hull edges, where the distance to two points could be equal.



(a) Delaunay Triangulation (b) Veronoi Tessellation

Figure 1.7: The Delaunay Triangulation and Veronoi Tessellation for a random set of points.

Definition 4 (Simplex and Convex Hull)**.** An n-dimensional *simplex* is the convex hull of a set of $(n+1)$ points. The convex hull of a set of points is the smallest convex polygon for which each of the points lies either inside the polygon or on its boundary. For the two-dimensional case (a 2-simplex) the hull has three points, and is therefore a triangle.

The name Barycentric interpolation comes from *Barycentric coordinates* – homogeneous coordinates which are defined by the vertices of an enclosing simplex. Most software uses the free Qhull library (`http://www.qhull.org/`) for the above operations.

Once the data have been triangulated, the following procedure is carried out for each point

which requires an estimate:

1. Extend a line from the point to the edges of the surrounding simplex. This can be done in any direction, as long as it is consistent throughout the entire process.

2. Calculate values at the points where the line crosses the edges of the simplex. This is done by interpolating using the vertices of the simplex.

3. Interpolate using the points calculated in step 2, to obtain a value at the desired point.

The interpolation used could be linear, cubic or some other direct interpolation technique. However, this technique only works when the Delaunay triangulation is successful, which means that at high sparsities it tends to fail – regardless of the complexity of the interpolation.

1.1.3 Kriging

Kriging is a technique which was developed in the 1960s by D. G. Krige, a South African mining engineer, for use in estimating ore reserves in mines and quarries from sparse measurements. It assumes that the parameter being estimated has a certain degree of spatial correlation, which is dependent only on relative proximity.

Kriging techniques have been sucessfully applied to a number of scenarios (for example, mining, mathematics and classification by Boucher et al. [9]), as discussed by Cressie [10]. Kriging was first applied to ionospheric electron density mapping by Blanch et al. [11], [12]. Later, Wielgosz et al. [13] compared Kriging with a multi-quadratic spline modelling method, and found that they were very similar in performance.

Kriging is a form of Bayesian inference, which starts by assuming a Gaussian *a priori* distribution for the input data. This is then combined with a Gaussian likelihood function for each of the samples. This combination leads to an *a posteriori* distribution whose mean and variance can be calculated using Bayes' theorem, and which should have a smaller variance than the prior distribution [14, pp. 300–333].

One very important concept in Kriging is the semi-variogram, which is defined as follows, for a one-dimensional series (see Cressie [15, page 58]):

Definition 5 (Semi-Variogram)**.**

$$V(k) = \frac{1}{2} Var\left[x_{t+k} - x_t\right] \tag{1.6}$$

Where x is the data series, and k is the *lag*. The semi-variogram can be constructed for as many lags as the data allows. For stationary processes, this is related to the auto-correlation and variance by:

$$V(k) = \gamma(0)\left[1 - \rho(k)\right]] \tag{1.7}$$

Where $\gamma(0)$ is the variance of the process. The semi-variogram is therefore a tool for assessing the spatial autocorrelation of a given function [16, pp. 273].

Creating two dimensional semi-variograms is more complicated, as the lag becomes a vector. This means it is necessary to group similar lag vectors together, to avoid too many data points.

There are several different Kriging methods, the most prevalent being *ordinary Kriging*, which is described below.

Ordinary Kriging

1. Construct a *semi-variogram* from input data set.
   This is a plot of Euclidean distance between each point, against the variance of each sample relative to the others.

2. Construct a *model semi-variogram*.
   This should model the trend in the input semi-variogram – and is constructed from prior knowledge.

3. The model semi-variogram is then used to compute *Kriging Weights*.
   These are then used in a weighted sum to compute output values:

$$F(x, y) = \sum_{i=1}^{N} w_i Q_i \tag{1.8}$$

   Where N is the number of input samples in the set, Q_{ij} is the ith scatter point, and w_i is the ith weight. Weights are calculated using a set of simultaneous equations, which are designed to minimise the least square error between the model and data.

Kriging is considered a model-based reconstruction technique, because it requires a model semi-variogram for the reconstruction.

1.2 Conclusions

This chapter has discussed the origins and meaning of sparse remotely sensed data, and has introduced the problem of reconstruction. It then described several direct reconstruction techniques. Unfortunately, when the data to be reconstructed are very sparse (where over approximately 95% of data points are missing), these techniques are unable to provide realistic outputs. This is for two main reasons:

- Not enough data are available to create a useful triangulation. This means that all Barycentric techniques, regardless of their interpolation method, will fail.
- Not enough data are available to form enough derivatives of the data to ensure continuity between reconstruction cells.

For this reason, it is necessary to look at alternative reconstruction techniques which can cope with highly sparse data. Kriging is one technique which might be appropriate, however, its reliance on a model semi-variogram means that it is unable to cope with data sets with unknown spatial autocovariance.

Normalised convolution requires no such prior information, and therefore has great potential for the reconstruction of un-modelled data. Chapter 2 discussed normalised convolution, starting with the first principles, and moving on to describe ways in which it can be improved to make use of information locked within the data set. This chapter will show that normalised convolution, and derived techniques are suitable for reconstructing sparse remotely sensed data.

Chapter 3 is concerned with the application of normalised convolution to GPS electron content mapping. First, GPS and the ionosphere are briefly described, giving the motivation for the reconstruction process. The process of converting GPS data into a form suitable for reconstruction is then discussed, and some example outputs are given.

Chapter 4 attempts to evaluate the performance of normalised convolution. This chapter discusses the generation of test data, before examining normalised convolution with a view to characterising how it reacts to changes in input conditions. Next, a scenario comparison is carried out – this attempts to find the optimum parameters for a given input. From these results, fits are derived, which reveal that more work on characterising the input distribution is needed.

Chapter 5 gathers the conclusions from the previous chapters, and explains further work that could be carried out, starting with implementation based tasks, such improving understanding of input characteristics and their effect on normalised convolution and further developments to adaptive schemes. The second section details new applications for normalised convolution based reconstruction, with specific reference to highly sparse GPS data, and temperature and salinity data taken by the ARGO float network.

2

Normalised Convolution

Normalised convolution (NC) techniques are interpolation algorithms which make use of available data and confidence (or certainty) meta-data – data which describes where samples are available, and where they are absent. This distinction allows algorithms to distinguish between absent data, and zero valued data, which helps improve algorithm output.

NC techniques were originally proposed in 1993, by Knutsson and Westin [17]. It has been steadily increasing in popularity, and has been applied to medical imaging, see, for example Estepar et al. [18], regularisation of tensor fields [19] and motion compensation [20]. NC techniques are particularly interesting with regards to reconstructing geophysical data because they have not yet been applied to this area.

The algorithm behind normalised convolution is very simple, involving just two convolutions and an element-wise division.

The first convolution is defined by:

$$D(x, y) = f(x, y) * g(x, y) \tag{2.1}$$

Where $f(x, y)$ is the sampled input data. $g(x, y)$ is known as the *applicability function*, (also known as mask or filter), and defines the locaisation of the convolution by constraining the area over which it works. Generally, Gaussian funtions are used as masks, although Knutsson and Westin [17] used a modified raised cosine.

The second convolution is defined by:

$$N(x, y) = c(x, y) * g(x, y) \tag{2.2}$$

Where $c(x, y)$ is the *certainty map* associated with the data $f(x, y)$. Equation 2.2 outputs a set of certainties associated with the first convolution.

In order to normalise the first convolution, it is simply divided by the second:

$$\tilde{f}(x, y) = \frac{D(x, y)}{N(x, y)} \tag{2.3}$$

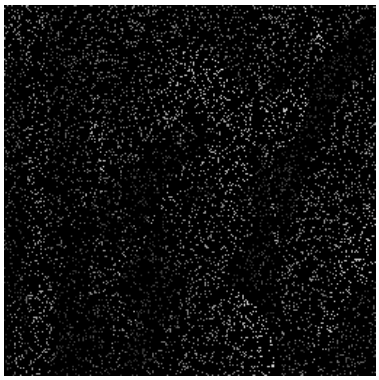
The output is therefore the first convolution, weighted by the confidence of the results generated.

Normalised convolution has the advantage that it works at very high sparsities, provided the filter used is large enough. It is also both computationally, and intuitively very simple, and requires no triangulation or calculation of derivatives.

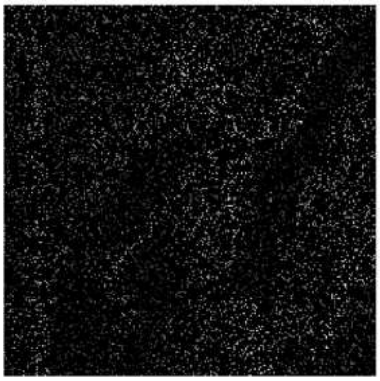
2.1 Sample Output

Figure 2.1(d) shows standard normalised convolution on the irregularly sampled test image Lenna.

(a) Lenna



(b) Lenna: 90% of samples removed



(c) Lenna: After first convolution (sampled image with filter)

(d) Lenna: Reconstructed using NC, after having 90% of samples removed. The NC filter was a Gaussian of size 11.

Figure 2.1: Images of Lenna, before, sampled and after NC

2.2 Filter Considerations

In general, the filter size should be kept as low as possible, in order to avoid over-smoothing the output data. Using a larger filter than necessary will causes the output to be to include values from a larger part of the surrounding area, which will lead to lower sensitivity to closer points.

However, if the image to be reconstructed has gaps which are larger than the size of the filter being used in the NC, the output will have gaps. These gaps reduce the quality of the output image, and can produce edge artifacts.

To mitigate the problems that this causes, the filter size at each point can be adapted, in relation to the distance to the nearest sample. This leads to the concept of adaptive normalised convolution which is discussed in section 2.3.

2.3 Adaptive Normalised Convolution (ANC)

Adaptive normalised convolution aims to increase output quality by adapting to the input data, by choosing the smallest possible filter which encompasses at least one data point. This will ensure that the output image has no gaps, and should increase the quality of the output. This can be done by using a *Euclidean Distance Transform* to calculate the distance to the nearest input sample in the confidence map.

Definition 6 (Euclidean Distance)**.** The Euclidean distance [21] between two objects is the distance one would obtain through standard measurement.

$$\ell_2(x, y) = \left(\sum_{i=1}^{N} |x_i - y_i|^2 \right)^{\frac{1}{2}} \tag{2.4}$$

Equation 2.4 is also known as the ℓ_2 norm, or Minkowski distance of order 2. Other difference transforms are formed by raising the components to different powers, before rooting them by the same power:

$$\ell_p(x, y) = \left(\sum_{i=1}^{N} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{2.5}$$

Commonly used distance metrics are the ℓ_1 norm, known as Manhatten distance, and ℓ_∞, the chessboard (or Chebyshev) distance, which is defined by:

$$\ell_\infty = \max\left(|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_N - y_N|\right) \tag{2.6}$$

The chessboard distance corresponds to the distance a chess king would have to travel to reach a given position on a chessboard.

However, things are not quite as straight forward as one would hope, and using only size adaptation has been offers very little improvement to the output image peak signal to noise ratio (PSNR) over simply choosing a large-enough filter, and using that over the entire input image. Tests using a ℓ_2 distance transform gave rise to a difference in PSNR of less than 1, when compared to standard normalised convolution using filters of a fixed size.

Figure 2.2: Lenna: Reconstructed using shape adaptive NC.

In order to give a higher level of improvement over standard NC, an adaptive NC system must make use of information on the structure of the actual samples as well as their spacing.

Adaptive NC (ANC) was first suggested by Pham and van Vliet [22], and improves on shape adaptive normalised convolution by using information derived from local gradients. This section of the report follows the development of an ANC filter after Pham and van Vliet [22]. First, techniques for estimating the gradients of the input image must first be introduced, for this reason, the next section introduces two gradient estimation techniques, and is followed by a return to the topic of ANC.

2.4 Gradient Calculation in Irregularly Sampled Images

2.4.1 Introduction

The two techniques introduced in this section provide ways of estimating the gradient of an image where some data are missing. They were first introduced by Knutsson and Westin [17], but are probably best summed up in [23], where various examples and comparisons with Sobel operators are given.

2.4.2 Derivative of Normalised Convolution (DoNC)

Applying differential operators to the normalised convolution (equation 2.2) is one way of obtaining an estimate of the image gradient. It also happens to be fairly computationally simple.

Applying the differential operator to only the x axis gives:

$$\Delta_x \left(\frac{D(x,y)}{N(x,y)} \right) \equiv \frac{D_x(x,y) \times N(x,y) - N_x(x,y) \times D(x,y)}{N^2(x,y)} \tag{2.7}$$

where:

$$D_x(x,y) = x.g(x,y) * f(x,y), \tag{2.8}$$

and:

$$N_x(x,y) = x.g(x,y) * c(x,y). \tag{2.9}$$

In the above equations, $x.g(x,y)$, is an edge enhancement filter which could be any arbitrary filter multiplied by a variable x. This effectively tilts the filter relative to the x axis. The example given by Piroddi and Petrou [23] is a raised cosine of the form (see figure 2.3):

$$g = cos^{\alpha} \left(\frac{\pi \sqrt{x^2 + y^2}}{8} \right), \tag{2.10}$$

Where α is the power by which is cosine is raised, and therefore controls the width of the central peak of the function.

This can then be extended to the y axis in a similar fashion, to give outputs consisting of the gradients in x and y, (Δ_x and Δ_y) in a similar fashion to the outputs of a Sobel edge detector.

2.4.3 Normalised Differential Convolution

Normalised differential convolution (NDC) is slightly more complex than DoNC, and works by constructing a set of filter matrices, (one for each point in the sampled image) which are then

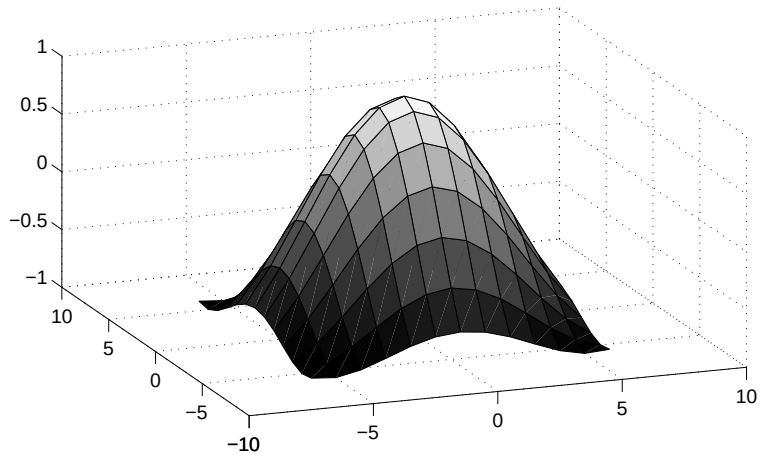


Figure 2.3: Raised cosine filter ($\alpha = 1$)

inverted, and used as a multiplier on the data. This gives good results, but is computationally more expensive than DoNC, because of the need to construct and invert a separate matrix for each point.

The matrix, N_Δ is given by:

$$N_\Delta \equiv \begin{bmatrix} N_{xx} & N_{xy} \\ N_{yx} & N_{yy} \end{bmatrix} \tag{2.11}$$

The terms in equation 2.11 are defined by the following equations (dependence on x and y is implicit), which define the data certainty in x, y, and the diagonals (xy and yx). N_Δ is therefore based entirely on the data confidence map.

$$N_{xx} \equiv N \times ((x^2.g) * c) - N_x^2 \tag{2.12}$$

$$N_{yy} \equiv N \times ((y^2.g) * c) - N_y^2 \tag{2.13}$$

$$N_{xy} \equiv N_{yx} \equiv ((x.y.g) * c) - N_x \times N_y \tag{2.14}$$

The other term in the NDC is a vector, formed using the input data and differentiated in x and y.

$$D_\Delta = \begin{bmatrix} D_x \times N - N_x \times D \\ D_y \times N - N_y \times D \end{bmatrix} \tag{2.15}$$

The output for each pixel is then defined as:

$$\begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = N_\Delta^{-1} D_\Delta \tag{2.16}$$

As with DoNC, the filters g could take any form, provided that when multiplied by combinations of the variables x and y, they form directionally sensitive filters. Smaller filters give more localised edges, but the minimum filter size is dependent on distance between adjacent samples.

2.4.4 Sample Outputs

Figures 2.4 and 2.5 show DoNC and NDC edge magnitudes generated from an the Lenna image where 90% of data have been removed. Both functions use an identical 13×13 Gaussian mask. The output images are of a broadly similar quality, with the NDC having generally smoother output.

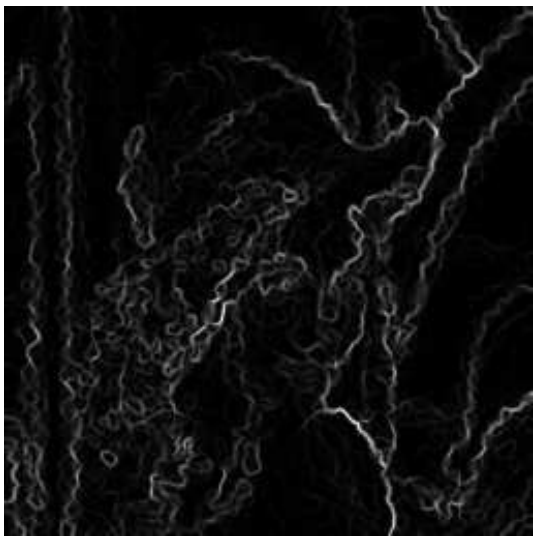


Figure 2.4: DoNC (90% data removal, Gaussian filter dimension 13)

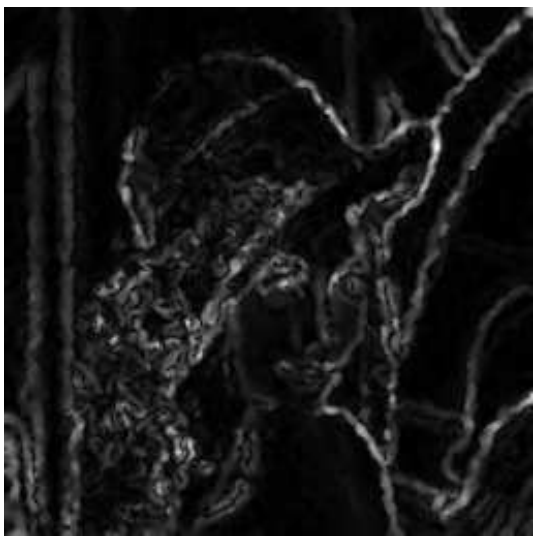


Figure 2.5: NDC (90% data removal, Gaussian filter dimension 13)

2.5 Adaptive Normalised Convolution Revisited

Adaptive NC works by making use of information on the underlying structure of the input image. This information, such as gradient and direction of local features allows a filter kernel to be sized and oriented to give the best possible output image quality. Having explained how it is possible to estimate the gradient of an irregularly sampled image, the process of ANC can be examined in detail.

First, the gradient of the input image should be estimated. This can be determined using either NDC or DoNC, with DoNC offering results of comparable quality at a lower computational cost (see tables 2.2).

Next, the gradients are multiplied together and smoothed using a Gaussian filter (where larger filter sizes are generally better) to give g_x^2, g_y^2 and g_{xy}.

All filters in ANC are usually (2D) Gaussian, which makes it easy to set the size in two dimensions by adjusting standard deviation values, and using the formula $d = 6\sigma + 1$.

After these new gradients have been computed, a *gradient scale tensor* (GST) is produced for each pixel in the image. (See [24] for more information on GSTs and their derivation). This is a two-by-two matrix, composed of pre-smoothed gradient products:

$$GST = \begin{pmatrix} g_x^2 & g_{xy} \\ g_{xy} & g_y^2 \end{pmatrix} \tag{2.17}$$

The GST's eigenvalues are then computed, and from these two values the following metrics can be calculated: (λ_1 and λ_2 are the largest and smallest eigenvalues respectively.)

- The local anisotropy:
$$A = 1 - \frac{\lambda_1}{\lambda_2} \tag{2.18}$$
- The local energy:
$$E = \lambda_1 + \lambda_2$$

Definition 7 (Eigenvalue)**.** The eigenvalues of a square matrix are the non-trivial roots of its characteristic equation. The characteristic equation is a representation of the matrix in one variable, normally λ. The characteristic equation of a matrix, $\mathbf{A}$ is

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \tag{2.19}$$

Therefore, the eigenvalues are found by solving equation 2.19.

The eigenvalues and gradients also allow computation of the local gradient direction and orientation. The values lie in the range $\pm\frac{\pi}{2}$, and are given with respect to the x-axis.

- The local gradient direction is the direction associated with the largest eigenvalue:

$$\varphi_1 = tan^{-1}\left(\frac{\lambda_1 - g_x^2}{g_{xy}}\right) \tag{2.20}$$

- The local orientation is the direction associated with the smallest eigenvalue (this value is used to set the filter direction, see 2.6):

$$\varphi_2 = tan^{-1}\left(\frac{g_{xy}}{\lambda_1 - g_y^2}\right) \tag{2.21}$$

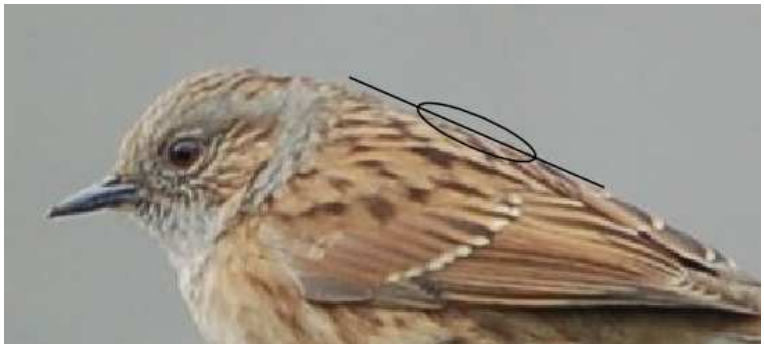


Figure 2.6: Steered Edge Enhancement Filter

As discussed in section 2.3 the other variable used in determining the filter size at each pixel is the Euclidean distance from each pixel to its nearest neighbour. This is known as σ_a.

The filter standard deviations are then given by:

$$\sigma_u = C(1 - A)^{\alpha}\sigma_a \tag{2.22}$$

$$\sigma_v = C(1 + A)^{\alpha}\sigma_a \tag{2.23}$$

The constants C and α allow the degree of dependence on the local image structure and anisotropy to be adjusted. Values of $C = 1$, and $\alpha = 1.1$ give good results.

After the filter size and orientation for each pixel has been calculated, the standard normalised convolution procedure can be followed, where instead of using a fixed Gaussian filter, the filter for each point is constructed using the calculated data. This means that standard convolution routines must be modified in order to use different filters at each point.

The entire process can be seen in Figure 2.7, which shows how many different features of the filter may be customised, including optimal smoothing filters at each stage - in practice however, smoothing anything but the anisotropy and gradient products actually reduces the output quality.

Experimentation on changing filter sizes suggests that for the gradient products, a larger filter size is better, where as initial results show that a more moderate filter size seems to be best for

smoothing the anisotropy.

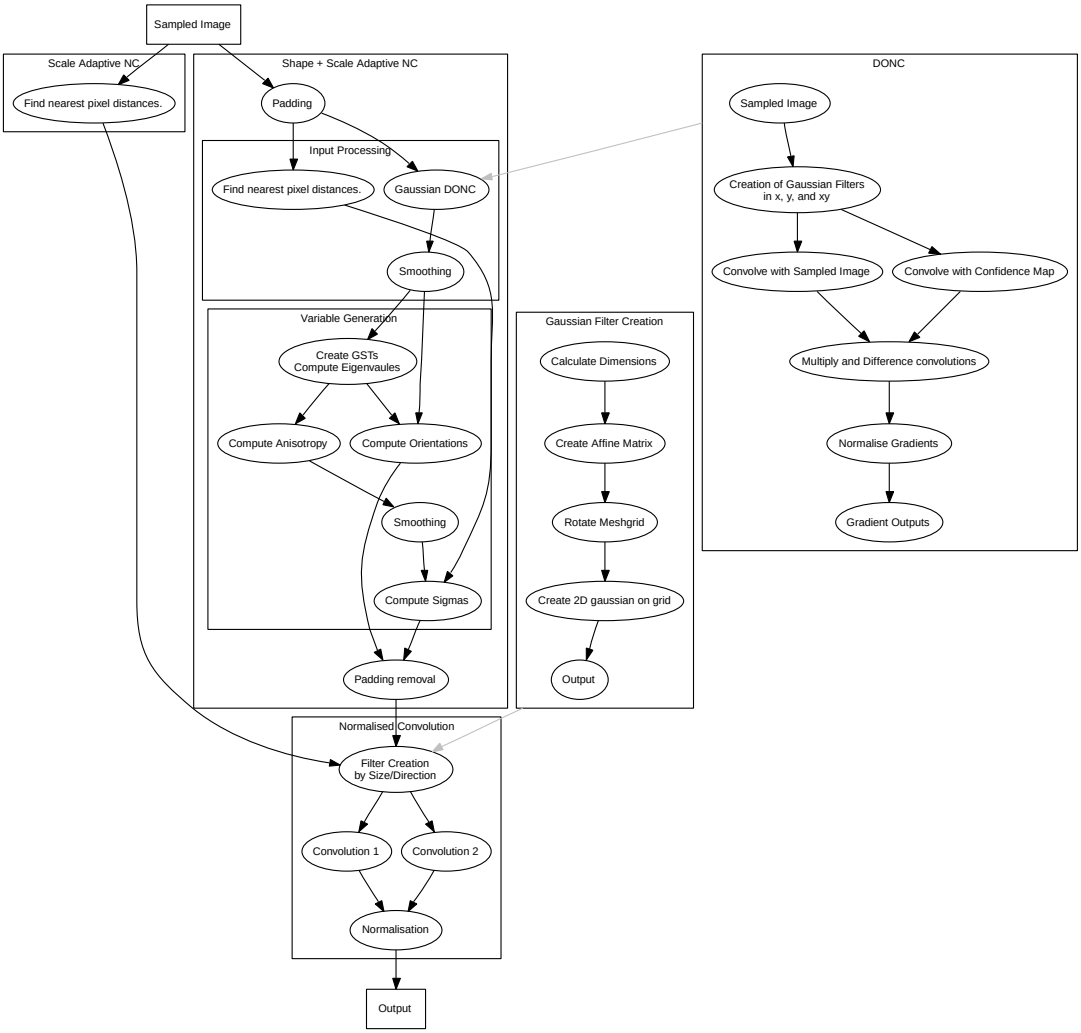


Figure 2.7: Flow chart showing NDC process.

2.6 Preliminary Performance Evaluation

This section shows outputs generated by normalised convolution and adaptive normalised convolution alongside results generated using MATLAB's *griddata* function, which facilitates interpolation of data using various direct methods. These results all use the test image 'Lenna' as their basis, and form a basic, preliminary technology evaluation.

This test was carried out by taking the test image, and randomly removing approximately 90%, 95% and 99%. Choosing the sample to remove followed the following proceedure, where *threshold* is the decimal value of the percentage of samples to remove:

```
For each pixel in the (intact) input image. {
    Draw a value from a uniform random distribution.
```

```
    if value < threshold
       set pixel to 0
}
```

This then allows various performance metrics to be calculated by comparing the output of and process with the original image. The two main performance metrics, (also called *fidelity criteria*) used in this report are peak signal to noise ratio (PSNR) and root mean square error (RMSE):

Definition 8. RMS Error and PSNR The RMS error between an original image $f(x, y)$, and a reconstructed estimate $\overline{f}(x, y)$ is given by [25]:

$$\text{RMSE} = \left[\frac{1}{MN} \sum_{x}^{M} \sum_{y}^{N} \left[\overline{f}(x, y) - f(x, y) \right]^2 \right]^{\frac{1}{2}} \tag{2.24}$$

Therefore a low RMS error implies that $\overline{f}(x, y)$ is a good likeness to $f(x, y)$.

The PSNR is defined as:

$$\text{PSNR} = 20 \log_{10} \frac{\text{Max allowed value}}{\text{RMSE}} \tag{2.25}$$

Generally, the maximum value used in the denominator of equation will be 255, since most images are represented by 8-bit pixels.

Figure 2.6 clearly shows that the ANC method has a slightly lower quality output at low sparsities than the interpolation methods, but that the relative quality increases as more data are taken out. In figure 2.9(d) the difference in quality between ANC and the interpolation schemes is particularly apparent.

In figures 2.9(a)-2.9(d) the 'adaptive' output gradients were generated using DoNC, and are very similar to the NDC outputs (not shown). The difference between DoNC and NDC is discussed in section 2.4.

RMSE	Linear	Cubic	Nearest	NC	ANC (DoNC)	ANC (NDC)
90%	15.42	15.41	18.41	17.51	15.59	15.47
95%	19.59	19.69	22.20	22.68	18.81	18.73
98%	25.43	25.79	28.14	49.40	23.49	23.80
99%	31.92	32.33	32.05	79.79	27.57	27.20

Table 2.1: RMSE values for 90-99% data removal.

PSNR	Linear	Cubic	Nearest	NC	ANC (DoNC)	ANC (NDC)
90%	24.370	24.375	22.830	23.265	24.275	24.340
95%	22.290	22.245	21.205	21.020	22.645	22.680
98%	20.025	19.900	19.145	14.255	20.715	20.600
99%	18.050	17.940	18.015	10.090	19.320	19.440

Table 2.2: PSNR values for 90-99% data removal.

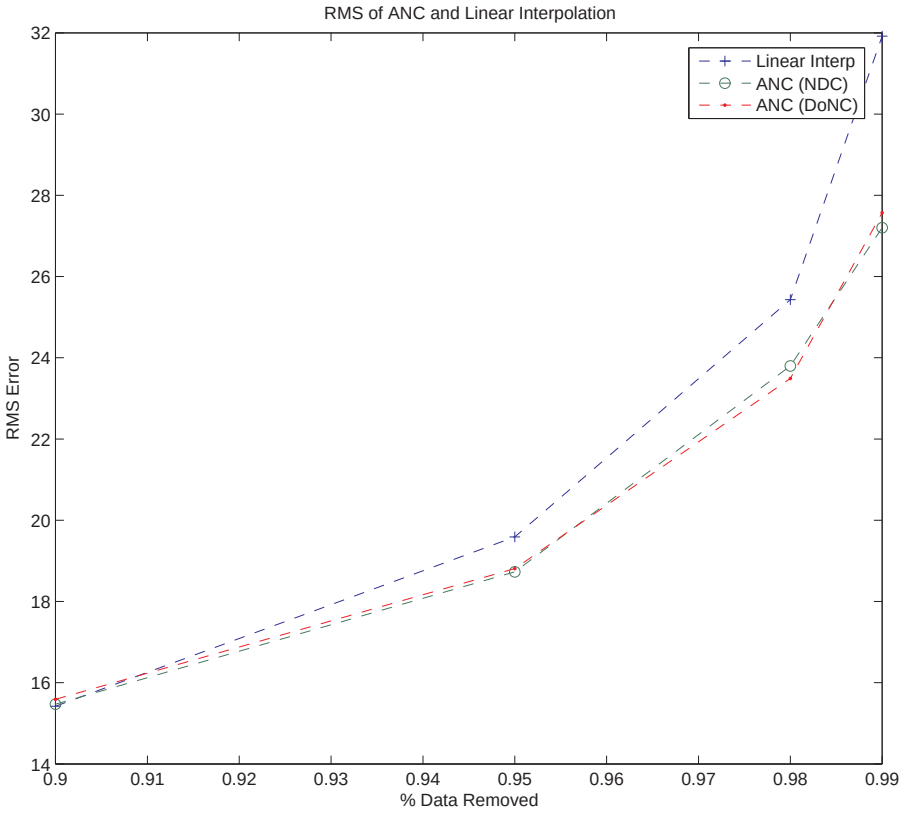


Figure 2.8: RMSE values for ANC and Linear interpolation with varying data removal.

2.6.1 Implementation Enhancements

Initially, a pure MATLAB based implementation was used. However, due to the high complexity of ANC, this was far too slow for regular running, with run time on small images (e.g. 320×240) approaching 10 minutes.

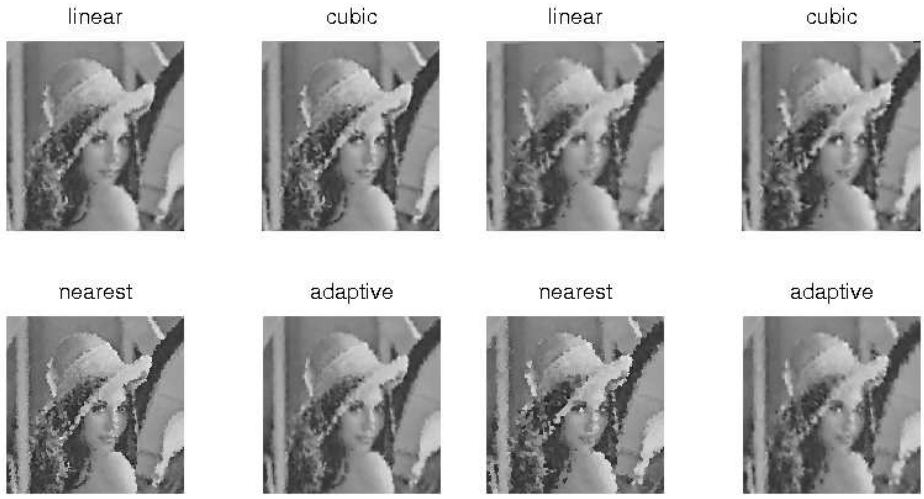
Profiling the code[1], revealed several areas which needed work to increase their efficiency, including:

- Generation of Gaussian filters: generating large Gaussian filters using a naïve implementation of the mathematics (that is, creating a mesh, and calculating a 2D Gaussian function over its entirety) is much slower than separating the filter into two 1D kernels, and convolving them:

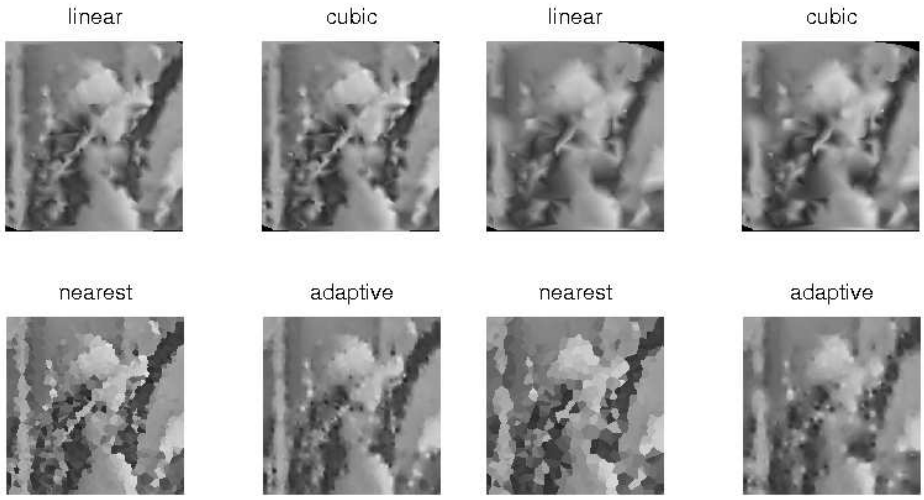
$$g(u, v; \sigma_u, \sigma_v) = \frac{1}{\sqrt{2\pi}\sigma_u} exp\left\{-\frac{1}{2}\frac{u^2}{\sigma_u^2}\right\} * \frac{1}{\sqrt{2\pi}\sigma_v} exp\left\{-\frac{1}{2}\frac{v^2}{\sigma_v^2}\right\} \tag{2.26}$$

- 2D Convolution using large isotropic filters: convolving using large 2D filters has a complexity of $O(n^2)$. If the filters are Gaussian, then they can be separated into two 1D filters once separated, the image can be filtered in two stages, by first using one of the 1D fil-

[1] A profiler is a tool for showing the time taken at each point in a program



(a) 90% data removal (b) 95% data removal



(c) 98% data removal (d) 99% data removal

Figure 2.9: Comparison of Matlab data gridding and ANC.

ters, and then filtering the resultant image with the second filter. This reducing filtering complexity to $O(2n)$ multiplications. For information on algorithmic complexity, and the O-notation, see Knuth [26, page 107].

Separability of filters is very useful, and it often possible to decompose 2D filter kernels into two 1D kernels using the following algorithm:

– Perform the singular value decomposition (SVD) on the filter kernel This will give three values U (output basis vectors), S (singular values) and V (input basis vectors), which can be recombined to form the original kernel using the equation $U.S.V^*$ where * represents the conjugate matrix transpose.

– Take the rank of the diagonal values of S. If the rank $= 1$, the filter kernel is separable.

  - If the kernel is separable, the two 1D vectors can be formed by taking the first columns of U and V and multiplying them by the square root of the single non-zero value from S.

- Anisotropic filtering: creating 2D filters and then rotating them using a reverse affine transform (see, for example Foley et al. [8, page 207]) is the most straight forward way of creating anisotropic filters. The affine transform works in reverse because values are needed at integer pixel coordinates. This is very slow when it needs to be repeated multiple times. Thankfully, the separability property of Gaussian filters can also be used to increase the speed of anisotropic filtering operations, as detailed in [27]. The procedure involves filtering first in one fixed direction, parallel to the x-axis, using:

$$g_x(x,y) = w_0 f(x,y) + \sum_{i=1}^{\lfloor N/2 \rfloor} w_i (f(x-i,y) + f(x+i,y)) \tag{2.27}$$

Where N is the size of the Gaussian filter, whose weights are given by $w_0 \dots w_N$. The filter used for the first filter pass has the standard deviation:

$$\sigma_x = \frac{\sigma_u \sigma_v}{\sqrt{\sigma_u^2 \cos^2\theta \sigma_v^2 \sin^2\theta}} \tag{2.28}$$

Where θ represents the angle of rotation of the filter, and σ_u and σ_v represent the original perpendicular standard deviations of the anisotropic filter.

The second pass of the filter operates on the output of the first, with the following standard deviation:

$$\sigma_\psi = \frac{1}{\sin\psi} \sqrt{\sigma_u^2 \cos^2\theta \sigma_v^2 \sin^2\theta} \tag{2.29}$$

The term ψ in this equation is found by using:

$$\mu = \tan\psi = \frac{\sigma_u^2 \cos^2\theta \sigma_v^2 \sin^2\theta}{(\sigma_u^2 - \sigma_v^2)\cos\theta \sin\theta} \tag{2.30}$$

This is left as a tangent, because μ is used at the intercept of the line along which the second filter operation works:

$$
\begin{aligned}
g_\theta(x,y) \quad = \quad & w_0 g_x(x,y) + \sum_{j=1}^{\lfloor M/2 \rfloor} \\
& w_j\{a(g_x(\lfloor x - j/\mu \rfloor, y-j) \\
& + g_x(\lfloor x + j/\mu \rfloor, y+j)) \\
& + (1-a)(g_x(\lfloor x - j/\mu \rfloor - 1, y-j) \\
& + g_x(\lfloor x + j/\mu \rfloor + 1, y+j))\}
\end{aligned}
\tag{2.31}
$$

The second filter operation therefore operates along a line oriented at ϕ radians to the x-axis. However, because $x \pm j/\mu$ will not always lie on a pixel coordinate, (2.31) makes use of linear interpolation between the pixels either side of the coordinate needed. The overall complexity depends on the filter size but is $O(2n)$ as for other separated filters.

A implementation in the C programming language, using the MATLAB external interface significantly reduced the time taken in anisotropic filtering.

Implementing these improvements increased the speed of ANC filter by approximately 20 times, reducing average sample run times from 497 seconds to 24 seconds.

2.7 Conclusions

This chapter introduced normalised convolution as an alternative to direct interpolation techniques. It then moved on to improvements that can be made in order to improve the quality of the NC output. To facilitate this, methods for finding edges in sparsely sampled image were introduced, and these methods were then used to explain how an adaptive normalised convolution algorithm could work.

This chapter has shown that NC and ANC are capable of reconstructing data at high sparsities with higher quality than direct techniques. Various ways of enhancing performance were discussed, including separating filters, and a fast implementation of anisotropic filtering, these allow ANC to run at very high speeds, only slightly slower than plain NC.

This speed enhancement, coupled with the good output quality at high sparsities suggest that NC and ANC would be good techniques to apply to highly sparse geophysical data sets, such as global positioning satellite (GPS) data for mapping electron content. The following chapter discusses how the ionosphere's electron content can be measured using sparse GPS receivers, and how these sparse measurements can be reconstructed using NC into electron content maps.

3

Application to Ionospheric Electron Content Mapping

Data derived from measurements taken using global positioning system (GPS) receivers can be used for far more than just navigation and positionging. One very important use of GPS data is the mapping and profiling of the ionosphere, a region of the atmosphere which is heavily influenced and altered by Solar activity. Ionospheric delays are the main source of ranging error in GPS, so understanding how these delays change under varying ionospheric conditions is important consideration in improving the GPS accuracy. The state of the ionosphere also has wide ranging consequences for long range radio communications and power distribution. For these reasons, and more, it is useful to be able to map the ionosphere using whatever data are available. This chapter introduces the ionosphere, GPS and then shows how GPS data can be converted into a form suitable for reconstructing with normalised convolution.

3.1 The Ionosphere

The ionosphere is a region of the atmosphere, spanning from about 90 to over 1000 km, which is formed when extreme ultra violet (EUV) light *photoionises* neutral atoms in the atmosphere. This process creates positive ions and electrons from neutral atoms and can only occur in sunlight.

Photoionisation is counteracted by two main processes:

- *Recombination*, involves electrons recombining with ions to form neutral atoms once again. There are two forms of recombination (*radiative* and *dissociative*).

  Radiative recombination is most common, and occurs when an electron and an ion recombine directly.

Dissociative recombination 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, and in the second stage, electrons combine with the positively charged molecule created previously.

The main parameter controlling the rate of recombination at any given altitude, is the number of available neutral atoms.

- *Attachment* occurs at lower altitudes where there are more neutral atoms, and involves electrons combining with neutral atoms to form negative ions.

Because of the way neutral atoms are distributed, the fact that their densities decrease with altitude, and differences in intensity of EUV wavelengths, the electron density of the ionosphere changes with altitude. The change of electron density with height is known as an electron density profile. The electron density 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, at night though, the recombination dominates, and the D, E and F_1 layers are almost entirely depleted, leaving only the F_2 layer, which survives over night.

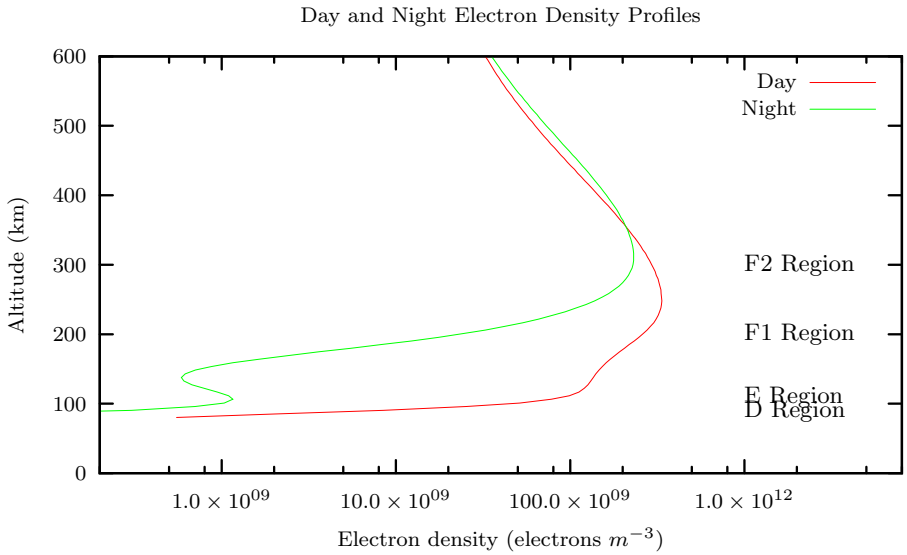


Figure 3.1: Day and night example electron density profiles, generated using IRI2001.

3.1.1 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, starting at 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 edge of the magnetosphere is known as the *magnetopause*,

outside of which is an area known as the *magnetosheath*, which is bounded by the *bow shock*, a region where the solar wind velocity drops suddenly see figure 3.2.

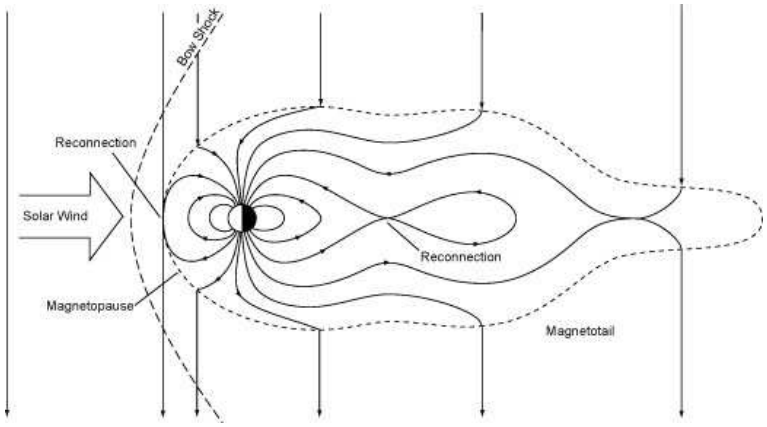


Figure 3.2: The Magnetosphere
A simplified schematic of the magnetosphere. Public domain image courtesy of USGS.

The interplanetary magnetic field is formed by the steady outflow of solar wind from the Sun, which carries the Sun's magnetic field far from its surface. The Sun's rotation causes the IMF to be spiral shaped, and intense variation in the Sun's surface magnetic field due to sunspots means that the orientation of the IMF when it reaches the magnetosphere varies with time. The 'vertical' component of the IMF is known as B_z, and its orientation determines whether or not solar wind plasma can enter the ionosphere. When B_z is pointing north or south, a large number of solar particles are injected into the ionosphere through regions of the geomagnetic field know as polar cusps, resulting in increased geomagnetic activity. 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. For this reason, when the orientation of the IMF is southward, it is able to link with the Earth's field, and this solar wind plasma is accelerated 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. Geomagnetic storms can cause large aurora and disrupt power distribution and communications.

The fact that the state of the ionosphere has such a large effect on communications and conditions on Earth means that it useful to study it in as much detail as possible, over as large a timescale as possible. There are various methods of remotely sensing the ionosphere's electron concentration, including:

- Ionosondes: which use swept high frequency pulses, and analysing the return signal.
- GPS Receivers and Satellites: see section 3.2
- Satellite Occultation: by mounting GPS receivers on GPS satellites, it is possible to recover electron densities along limbs between the receiver and other GPS satellites, as they are occluded by the Earth.

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

3.2 GPS

GPS (Global Positioning System) is a timing and positioning system run by the US Department of Defence. The GPS 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 configured such that there are four satellites in six orbital planes, each inclined at 55° to the equatorial plane.

Each of the 24 satellites transmits their own NS, containing information on the satellite, clock corrections, ephemeris and other information. The NS consists of 25, 1500 bit frames, delivered over 12.5 minutes, corresponding to a data rate of 50 bits/s. This signal is created, and then added (modulo two) to a 1.023MHz, pseudorandom-noise (PRN) code known as the coarse acquisition (C/A) code. The resulting code is modulated on to 1575.42 MHz carrier, known as L1, creating a spread-spectrum signal which can be used for ranging. A second spread spectrum signal is transmitted at 1227.60 Mhz, and is known as L2. These frequencies are generated by multiplying the fundamental GPS frequency (10.23 MHz) by 154 and 120 respectively.

Both L1 and L2 are modulated by a code known as the precision- (P-) code, which is encrypted by a further code called the Y-code. A cryptographic key is required to remove the Y-code, and allow use of the P-code. Many modern receivers make use of L2 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.

3.3 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.

3.3.1 Error Sources

Sources of ranging error (in approximate order of magnitude):

- Ionosphere – 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, 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 [28, pp. 294–307]. However this is still the main cause of ranging error in GPS. Because 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.

- Ephemeris data – this is the error in the transmitted position of the satellite. Ephemeris error is typically around 1 metre, although this error grows as the time from the last transmitted NS increases.

- Satellite clock – before being corrected by the control segment, clock errors can account for ranging errors of up to one metre.

- Troposphere – changes in refractivity cause errors of around 1 metre.

- Multipath – multipath interference accounts for around 0.5 metres of ranging error.

- Receiver errors – Errors in software and hardware can account for errors of various magnitudes. However generally these are rounding errors, which are negligible.

3.3.2 Fixed GPS Receivers

A great many fixed position GPS receivers are positioned around the world, collecting position data, which are then used for monitoring tectonic shift and crust strain, as well as for cartography and precision timing. If the correct data are saved, data recorded by these receivers can also be used to analyse delays caused by atmospheric regions - in particular the ionosphere. These delays can then be used to derive information on electron concentration. Thankfully many receivers output data in a format which can be assimilated fairly easily. GPS receivers are mainly situated in areas where there are geographical fault lines, they can also be very expensive, and so tend to be grouped in countries which are more affluent. These factors mean that the distribution of receivers is largely random, and because of the nature of the system, highly sparse.

3.4 Reconstructing GPS Data

As well as standard reconstruction methods, it is possible to use tomography to reconstruct 3D images of electron concentration, see for example [29]. These inversions can provide high resolution imagery, but require large amounts of data, and so do not very well work at very low sparsities. This means that normalised convolution is ideal for situations where tomography fails.

Before data can be used for 2D reconstructions, they must first be converted from L and P code phase measurements into spot values on a fixed height shell. This 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 [30, pp. 102]. The disadvantage of this approach is that information on the vertical structure is completely lost, the main advantage is that it is computationally simple.

Conversion from phase data to TSM data involves several steps, which are outlined below:

1. Input carrier phases must be converted into pseudoranges, and then differenced. For the L-band phases, this is done by multiplying them by the speed of light, and dividing by the signal frequency. The peudoranges are biased because of systematic errors, called interfrequency biases, which can be split into a receiver component b_r, and a satellite component b_s. Biases are generally calculated by assuming that the total bias is constant over the time between each particular satellite pass, and using a least squares fit to measurements to retreive the bias sum [7].

2. These pseudoranges are then differenced.

3. These difference values are then multiplied by 9.5196×10^5 to convert them into total electron content (TEC) units. This is known as slant TEC, because it corresponds to all of

the electrons encountered on the slanting path between the satellite and receiver[1].

4. Next, to project these value onto the shell, a correction must be applied to take the satellite elevation, and hence path length into consideration:

$$V' = \frac{V}{\sqrt{r^2 - r_e^2 \cos\theta}} \tag{3.1}$$

The output from this stage will be a set of scatter values, containing latitude, longitude and TEC information.

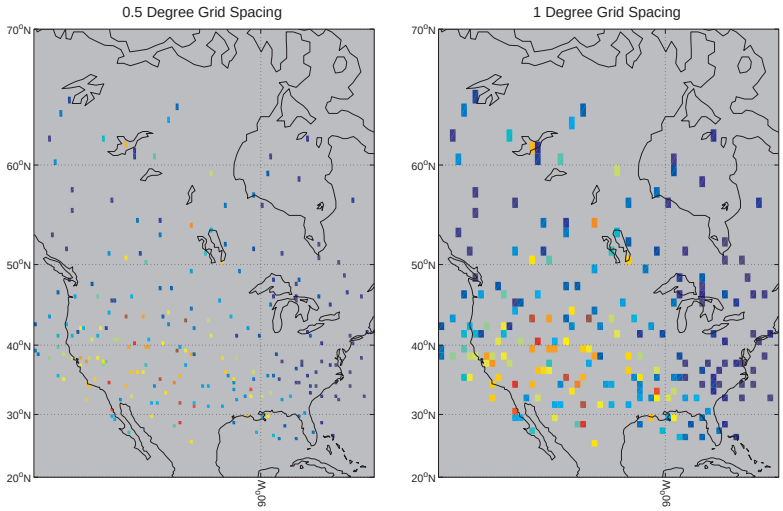
5. Finally, the scatter values must be projected onto a matrix. This is done using a grid of latitude and longitude values, which are used to set the values of the matrix grid. The resolution of this grid has an large bearing on the sparsity of the data to be reconstructed see chapter 1. Figure 3.4 shows the effect resolution has on sparsity using actual GPS input data.

Once these steps have been completed, the data are said to have been *gridded* and are ready to be reconstructed. For example input images, see figure 4.1. The dark rectangles represent the positions on input pixels. This input data is suitable for reconstruction using an suitable reconstruction method, for example cubic or linear Barycentric interpolation, however, when the grid resolution is high, or there are few receivers available, the corresponding input sparsity will be higher, and these interpolation schemes will fair badly.

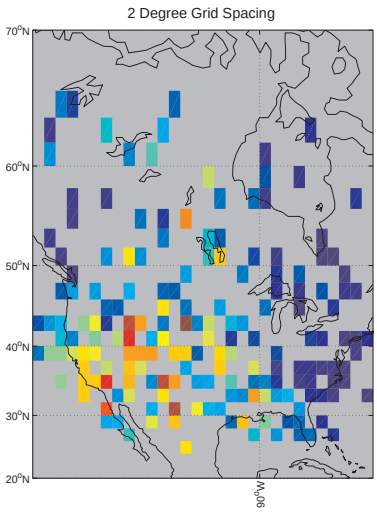
3.5 Outputs

This section show some example reconstructed data. Figure 3.4 and 3.5 were reconstructed using NC, with a filter size of 80×80, and ANC respectively and refer to 2040–2050 GMT on 30th October 2003. This corresponds to a period of intense geomagnetic activity, due to several large CMEs arriving at the magnetosphere in a short time.

[1] 1 TEC unit (TECu) is 10^{16} electrons per m^2.



(a) 0.5°grid (b) 1°grid



(c) 2°grid

Figure 3.3: GPS thin shell data, projected onto various grid resolutions.

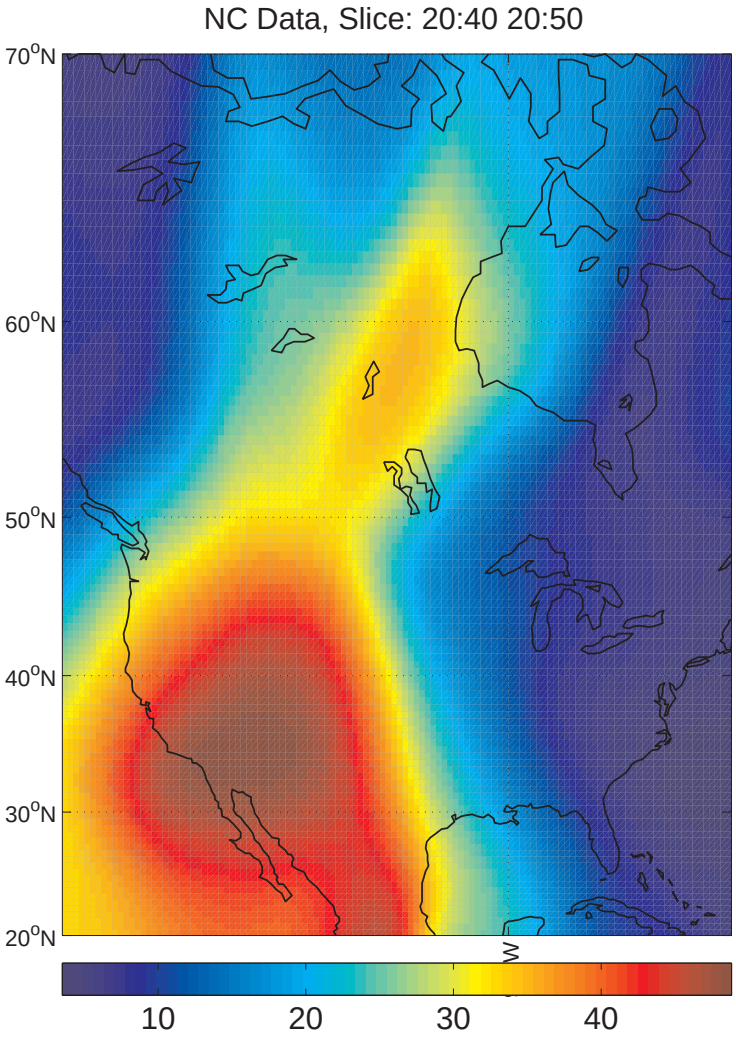


Figure 3.4: Example NC Reconstruction

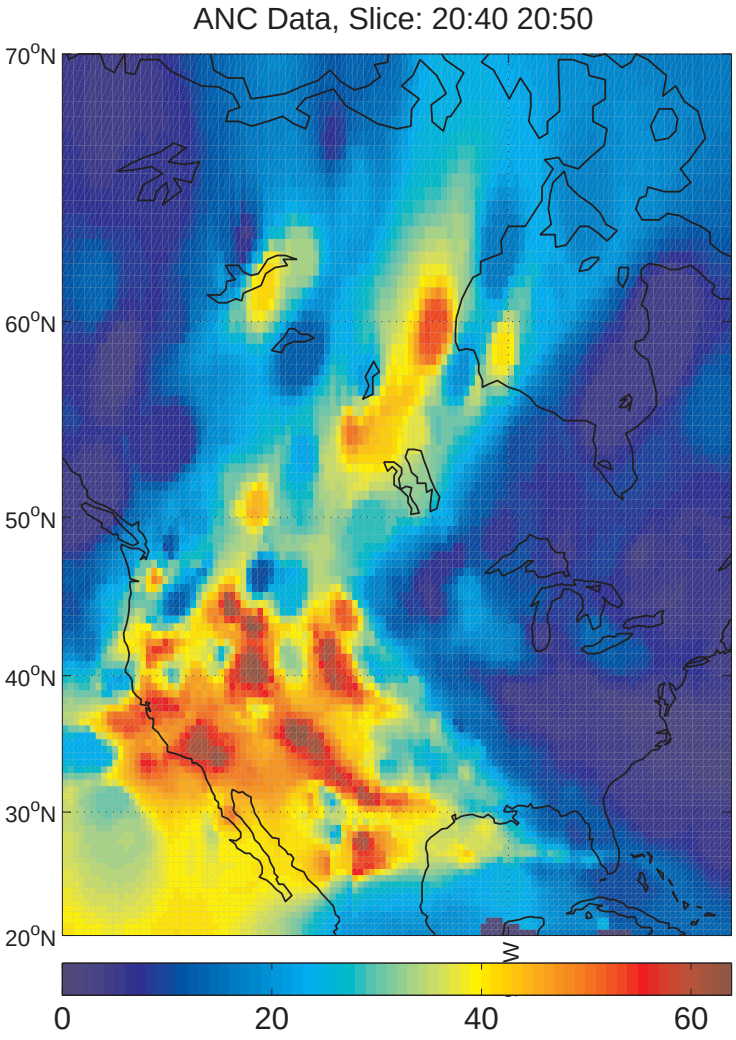


Figure 3.5: Example ANC Reconstruction

4

NC Performance Evaluation

In this chapter, the performance of NC will be evaluated, following some of the guidelines given in the Best Practice document [31, pp. 6]. This will consist of two stages.

1. The first section will be a *scenario evaluation* in which algorithm performance will be classified for specific scenarios, with a view to obtaining the best operating parameters.

2. The second stage will be a *technology evaluation*, which will attempt to characterise the algorithm in terms of how it behaves when input conditions are gradually changed, over a range of different parameter settings. The same data used to characterise the algorithm will then be used with other algorithms to allow comparisons to be made.

4.1 Experimental Procedure

The data used for reconstructions in this chapter will consist of GPS data which have been projected onto a thin shell, as described in section 3.4. Each sparse input frame consists of data drawn from a subset of available GPS ground stations, over a time period of 10 minutes. The data are then manipulated by removing the points associated with various subsets of the sites, which has the effect of changing the overall sparsity of the input. Figure 4.1 shows how changing the subset of sites changes the sparsity of the input points. For each subset of data (e.g. all, ½, ¼), the following procedure is carried out:

1. For each site in the input data subset:

2. Remove the site from the input data. This corresponds to several input points, because one ground station can generally see several satellites.

3. Reconstruct using the modified input data.

4. Compare the reconstructed data with the removed data. This should only be done at the specific points which were removed. This could be done using various metric, but the mean sum of absolute value of differences (SAVD) works well.

5. Loop to step 1.

6. Average the mean SAVDs to obtain site-invariant mean SAVD.

Definition 9. Mean SAVD

$$\text{Mean SAVD} = \frac{1}{MN}\sum_{x}^{M}\sum_{y}^{N}\| \left[\overline{f}(x,y) - f(x,y)\right] \| \tag{4.1}$$

Where, the symbols' meanings are as in (2.6). In particular, $\overline{f}(x,y)$ represents the reconstructed data at the point not used in the reconstruction, and $f(x,y)$ represents the values of the unused input data.

This process ensures that only unseen data are used for testing the reconstruction output, and that results will not be depend on which sites were removed.

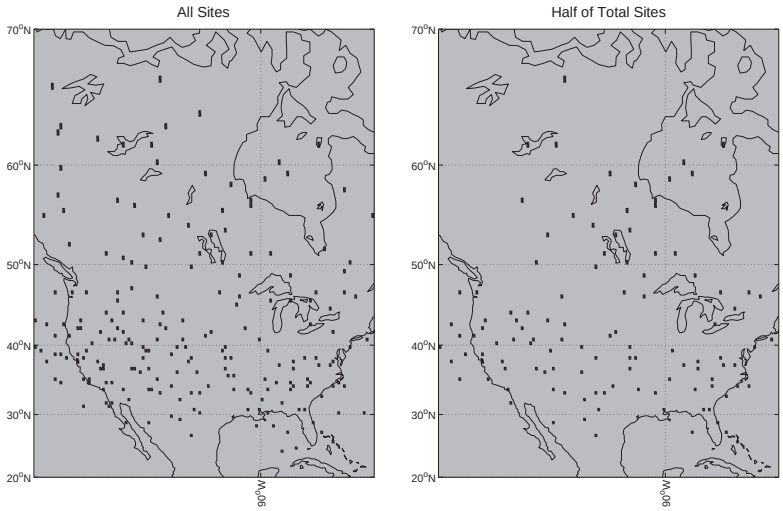
4.2 Scenario Comparison

In standard NC, the primary adjustable parameter is the size of the filter used for the reconstruction. The "optimum" mask size is the mask size for which the output error (when compared with unused, actual inputs), is the lowest.

To find the optimum mask size for a given set of input data, the same input data are reconstructed using a range of mask sizes. For this investigation, the masks used were isotropic Gaussians of dimension 50–200 increasing in steps of 10. After reconstructing the data with each mask size, the mean SAVD error can be calculated, as described above, and a graph constructed, showing mask size against the associated mean SAVD. This graph will generally have a similar quadratic form to figure 4.2, meaning that a clear minimum is present. The mask size that corresponds to this minimum is the optimum mask size.

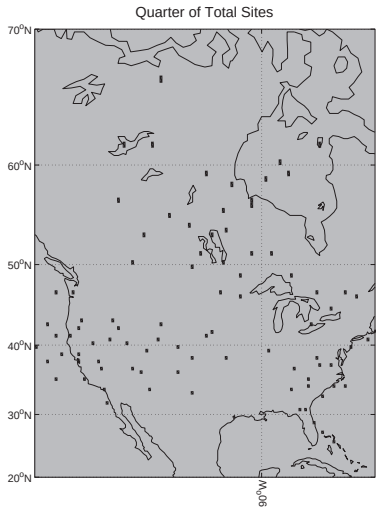
Another useful metric for comparisons is the mean distance to the nearest neighbour of any given pixel in a given matrix of input data (including point where no input data are present). This can be found using $\frac{\ell_2}{N}$, where N is the total number of pixels in the input, it therefore describes the mean *displacement* required to move from any given point in the input to the closest point with an associated value.

Figure 4.3 shows the optimum mask size and mean neighbour distances derived using the input data described in section 4.1. Unfortunately, not curve can be fitted because the data are so widely spread. Investigation of different characterisations of input distributions are necessary



(a) All available sites.

(b) Half of available sites.



(c) A quarter of available sites

Figure 4.1: Examples of sites used for reconstruction.

in order to derive more useful relationships between the input and optimum filter sizes (see section 4.1).

4.3 Technology Evaluation

Testing against strawman algorithms is a common way of evaluating an algorithm's behaviour. In this case, linear and cubic interpolation are obvious candidates because of their ubiquity, and availability in packages like MATLAB.

This section's tests used reconstructions which were carried out for 4.2, in addition to these reconstructions, the same input data were also reconstructed using linear and cubic Barycentric

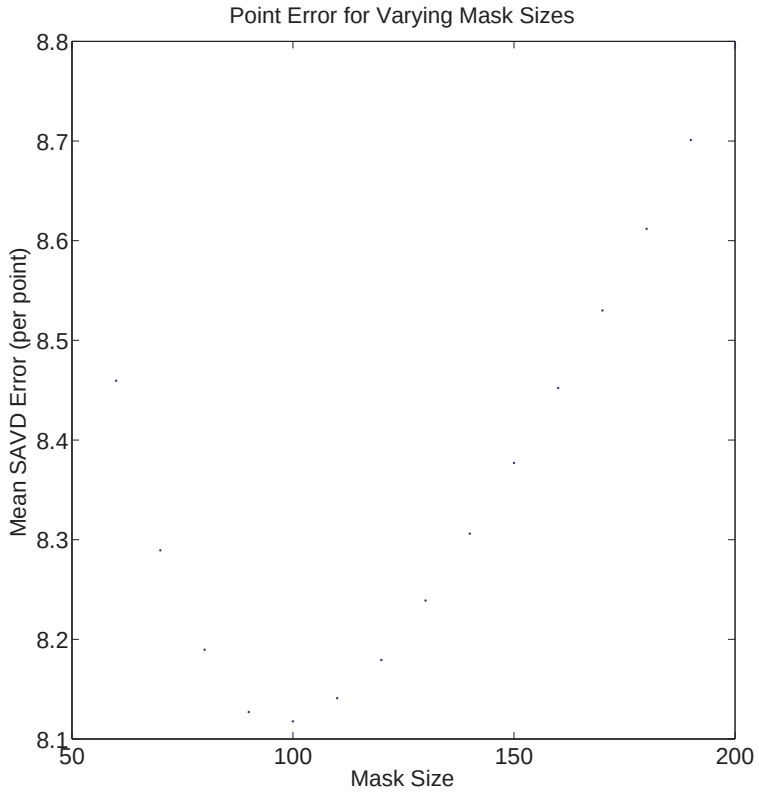


Figure 4.2: Mask Size and mean SAVD for input data where all sites are present. The optimum mask size is clearly 100.

interpolation. Mean SAVD values were calculated using unused data, as above, and the mean SAVD values used to represent the performance of NC were the values associated with the optimum mask size in each case.

These sets of values were then averaged across the input data sets which differed by only one site (not across the sets created by halving inputs etc.). Variances were also taken, and the mean distance to nearest neighbour metric was calculated, and averaged in the same way as the data above. This produced a data set describing the mean SAVD and distance to nearest neighbour for many different distances. Figures 4.5 and 4.6 show aggregated data points from 22 time slices, using full, halved and quartered sites. Figure 4.7 shows these two graphs overlaid, along with error bars showing SSD variances, and lines of best fit.

Only linear interpolation and NC results are shown, because cubic interpolation performed exactly the same as linear interpolation. Examination of the code used reveals that its gridding functions are based on a Barycentric interpolation, as described in section 1.1.2. Therefore it must be the Delaunay triangulation breaking down when the data are very sparse which causes the interpolation to perform poorly.

Figure 4.7 shows how the sum squared difference (SSD) changes as the average distance to the nearest neighbour increases. The cubic line of best fit shows the output error for NC being generally (slightly) lower than cubic.

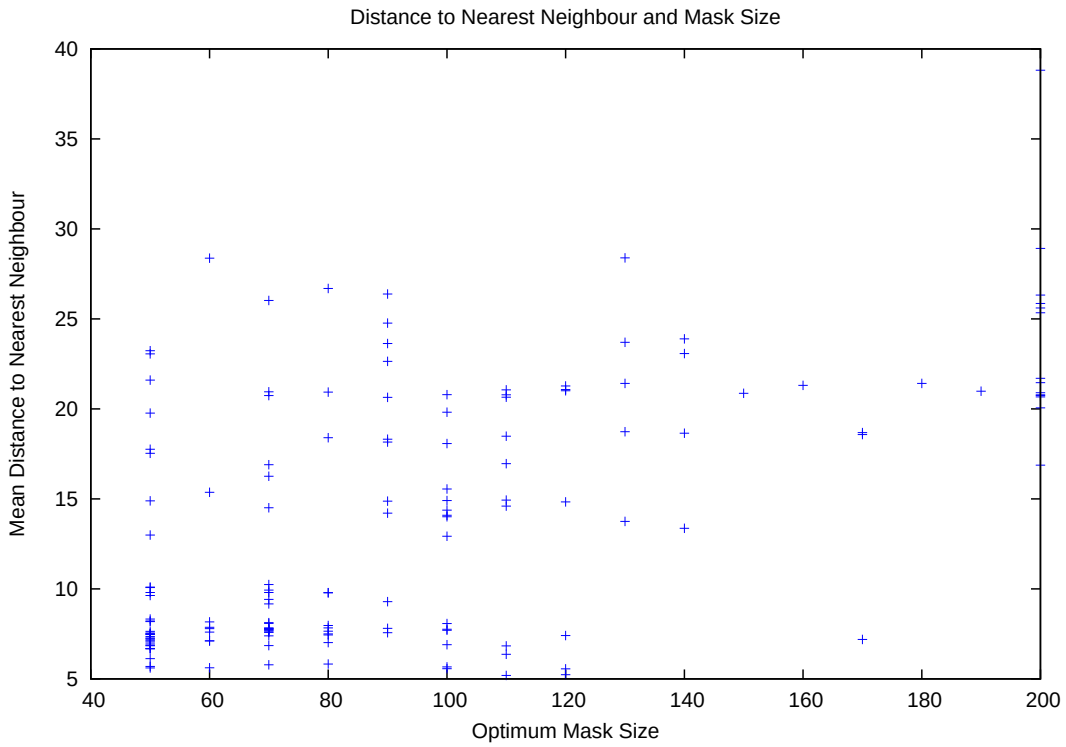


Figure 4.3: Optimum Mask Sizes and Mean Neighbour Distances

Reconstruction	Equation	RMS of fit residuals
NC	$f(x) = 11.290x^3 + 0.069x^2 - 1.526x - 0.001$	2.335
Cubic	$f(x) = -0.001x^3 + 0.090x^2 - 1.866x + 13.376$	3.68141

Table 4.1: Parameters for the fits in figure 4.7.

Where column 3 in table 4.1 is:

$$\text{RMS of residuals} = \sqrt{\frac{\text{sum squared residual (SSR)}}{\text{degrees of freedom (NDF)}}} \tag{4.2}$$

Another interesting result is the difference in average error variance between NC and linear interpolation. Across the entire input range, table 4.2 shows that the mean variance of NC was 12.5083, whereas the mean variance for cubic interpolation was 32.943, indicating that NC is much less sensitive to changes in the input distribution. Also, as expected, the variance of linear interpolation is slightly lower than that of cubic interpolation, suggesting that linear interpolation is slightly less sensitive to input sparsity than cubic.

Table 4.2 was generated by taking the mean of the mean SAVD values generated for each time period examined, and taking the mean value of the SAVD variances – they are therefore invariant of the sites used in the reconstruction. The figures in the NC (optimum) row correspond to the filter size which produced the lowest error output.

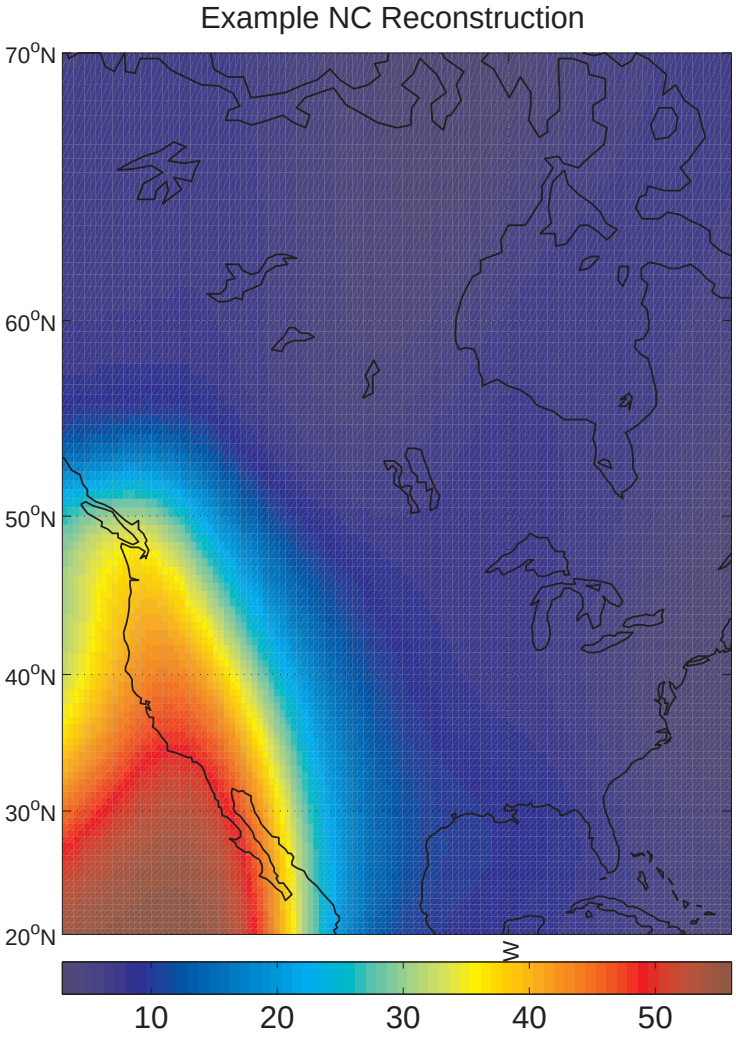


Figure 4.4: Example NC test output data.

Reconstruction	$\overline{\text{SAVD}}$	Variance
NC (optimum)	1.4763	12.5083
Linear	2.2413	30.802
Cubic	2.2929	32.943

Table 4.2: Mean SAVD and variance for various reconstruction types.

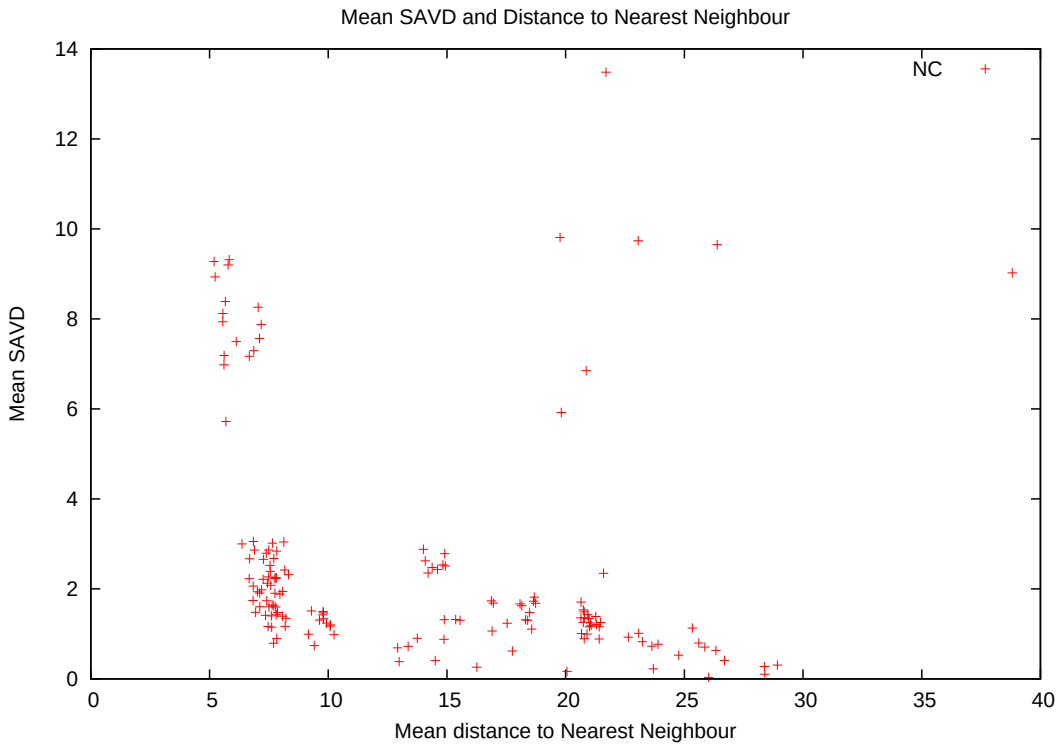


Figure 4.5: Mean SAVD and Mean Neighbour Distances for NC

4.4 Conclusions

This chapter has examined NC by looking at specific scenarios, and attempting to find optimum filter parameters, and by comparison with other popular reconstructions in order to assess its relative performance. This has highlighted three main results.

- Mean distance to nearest neighbour is not a very useful metric when searching for useful relationships between optimum mask size and the sparsity of input data, as figure 4.3 shows. Other metrics may provide more useful, or usable, curves than the relation. This is discussed in section 5.1.1.

- NC performs better than linear and cubic Barycentric interpolation schemes for all of the sparsities that were tested. This suggests that NC is better for reconstructing sparse data than triangulation based approaches.

- NC has a lower error variance than Barycentric interpolation. This suggests that NC's behaviour and output quality are less dependent to changes in the distribution of input points than Barycentric interpolation – a clear advantage in reconstructing who's sparsity and distribution are highly variable.

The next chapter aggregates the conclusions from earlier chapters, and this chapter, before describing various items of further work that these lead to, as well as other further work that has not been addressed directly in the body of the report so far.

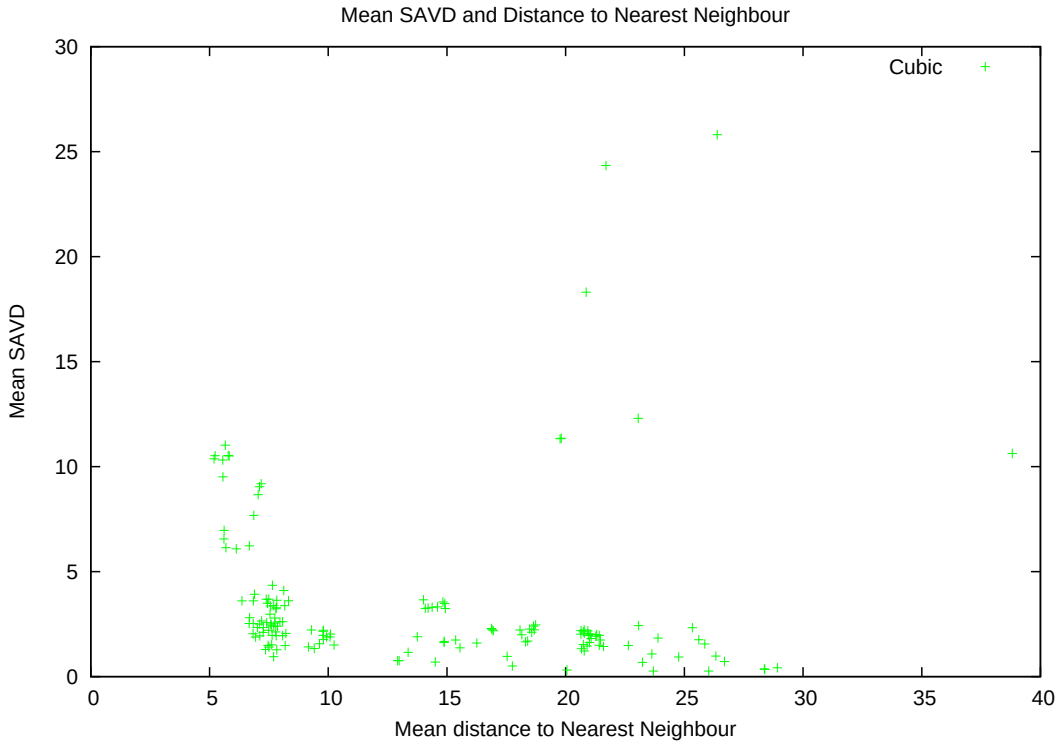


Figure 4.6: Mean SAVD and Mean Neighbour Distances for linear interpolation

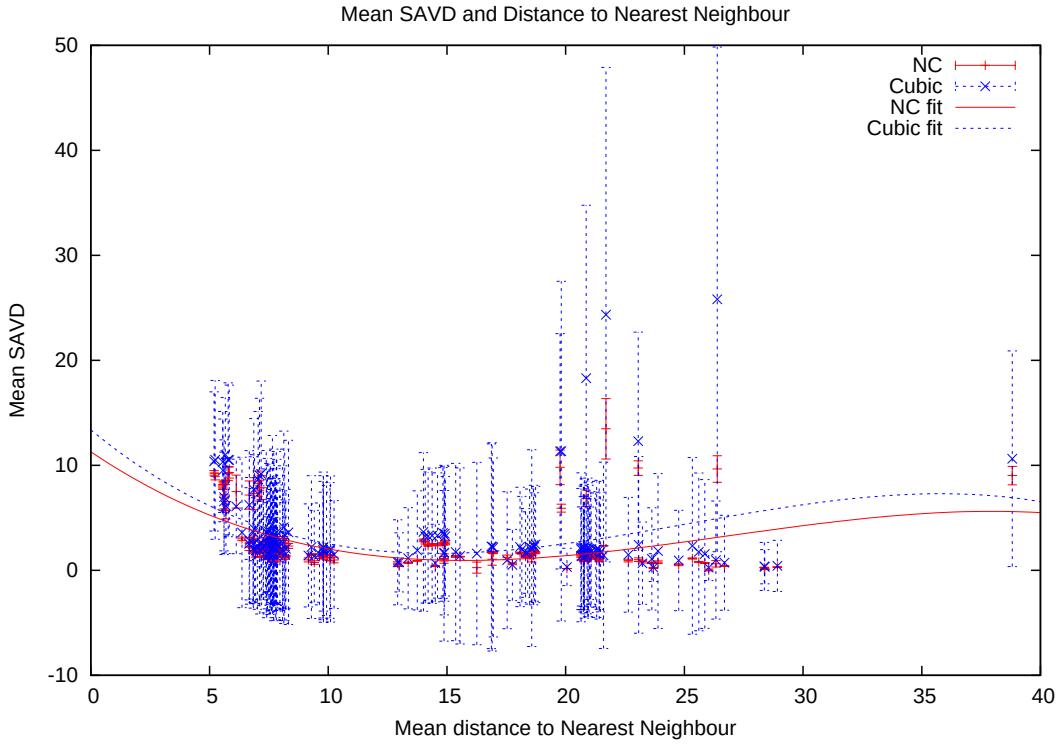


Figure 4.7: Errors and Neighbour Distances

5

Conclusions & Further Work

Normalised convolution techniques have been applied in several areas, such as medical imaging, and traditional image processing. So far though, they have not been applied to geophysical data.

This report has demonstrated that NC can be used to reconstruct data at sparsities where other techniques break down. It has also shown that NC provides consistently lower error that other interpolation techniques on real data – suggesting that it is a good choice for reconstructing such data.

NC has been sucessfully employed in the reconstruction of GPS TEC data, and shows lower error variances than linear and cubic interpolation across the range of input sparsities. This suggests better stability than triangulation based interpolation, which means that NC is a very good choice for datasets which are highly sparse.

Further work is needed in classifying input distributions in order to improve adaptation, because mean distance to nearest data point results in a distribution which is best characterised by an asymptote, implying extreme sensitivity to inputs.

An initial adaptive NC scheme has been implemented, and shown to perform well on the 'Lenna' test image, at high sparsities, however a more complete evaluation is needed in order to draw conclusions about its ability to reconstruct geophysical data.

Work on NC has hinted at the fact that deriving clear relationship between optimum filter sizes, and input characteristics would allow the development of better adaptation algorithms, which better adapt to the input data, and its spatial distribution. The current ANC implementation makes use of the ℓ_2 norm, as a simple product, where as figure 4.3 shows that this is clearly not the best approach. As mentioned above, further evaluation will reveal clearer relationships between input data and optimum filter parameters.

5.1 Further Work

There are three main categories of further work, *implementation*, covering improvements, changes and characterisation of the underlying algorithms in ANC, *application*, covering applying ANC and NC techniques to new data sets, and *other work* which contains work not directly related to the content of this report. The timescales involved in completing these tasks will be discussed in section 5.4.

5.1.1 Implementation

Input Characterisation

As well as sparsity, discussed in definition 2, a large number of other parameters can be used to characterise sparse data. This is because sparsity carries no information about how the data are distributed spatially.

The most releant characterisation of input distributions is likely to be the use of *image moments*.

Definition 10 (Image Moments)**.** Image moments allow an image to be characterised using various weighted sums of pixel values, or in this case, positions. Because image moments characterise two dimensional distributions, there are two parameters, p and q.

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{5.1}$$

If the image being examined is a binary image, then when both p and q are 0, the output will be the area (or the sum of non-zero pixels). Diving this by the total size of the image will give the sparsity of the image.

The *centroid* of the image (also known as centre of mass) can be found using the following:.

$$\overline{x} = \frac{M_{10}}{M_{00}}, \overline{y} = \frac{M_{01}}{M_{00}} \tag{5.2}$$

The can be used to derive various moments about the centre (or mean) of the image. These are known as central moments, and can be defined by the following equation:

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x, y) \tag{5.3}$$

Using second order central moments, information about orientation can be exctracted. This is

done by forming a covariance matrix, and solving for its eigenvalues and vectors. The angle of the eigenvector associated with the largest eigenvalue is the angle of the principle axis of the image.

Evaluation of ANC

A more complete technology evaluation will help assess how well ANC works on GPS data of various sparsities, compared to other algorithms. Currently, only a limited comparison using the 'Lenna' test image is available. Testing ANC on the same input data as were used to test NC, whilst changing parameters like aspect ratio, and smoothing will help to characterise ANC under varying input conditions.

A scenario comparison will help determine the optimum input parameters for running on GPS data or various sparsities. Currently, these parameters include various smoothing filters, as well as mask parameters, such as aspect ratio and size constraints. Changes to the adaptation methods could alter these. Scenarios to be examined should include a wide variety of input sparsities and distributions, in order to ensure the statistical significance of the results. If possible, a relationship between the optimum parameters and input distribution and sparsity will be derived.

Comparing the operation, and output of kriging with ANC will probably also prove instructive.

5.2 Applications

5.2.1 Reconstruction of Highly Sparse Historical GPS Data

Various data GPS data sets exist which are too sparse to realistically reconstruct using tomography. It should be possible to reconstruct them using NC or ANC in order to study the ionosphere's behaviour. The advantage of this approach over direct interpolation schemes is the lower error at high sparsities. The reconstructed data could then be analysed using motion estimation and relaxation labelling, allowing anomalies to be tracked as they convect across the polar cap. Vectors formed during the tracking could also be checked against modelled $E \times B$ drift vectors, allowing further validation of NC and ANC.

5.2.2 Reconstruction of Other Geophysical Data

ANC and NC could be used to reconstruct different geophysical data sets. For the reasons mentioned in the introduction, sparse data sets are very common in geophysics, and so there should be many areas in which NC techniques can be used.

One example is ARGO float data. ARGO buoys are neutrally buoyant devices designed to measure salinity and temperature as a functions of depth in the world's oceans. Normally they sit at a 'parking depth' of 2000 m, but every 10 days, they inflate an external bladder, and rise to the surface over about 6 hours. During this time, the device measures temperature and salinity, and when it reaches the surface, it transmits the data is has gathered to the Système Argos (`http://www.cls.fr/html/argos/welcome_en.html`), which also calculates the bouy's position to within 100 m. Système Argos is a doppler based positioning service, which operates using at least two satellites, and also allows data uplink and downlink.

Because ANC is data driven, meaning that specific models of the input data are not required, it could be used in situations where a result is required quickly for visualisation or sanity checking. It could also be validated against model-reconstructed data, in order to find the optimum reconstruction parameters, eventually allowing it to be used as a quick drop-in replacement for the models.

Another area of research could be the derivation of 'slices' or salinity or temperature – or fields at a given depth. This could be challenging, because at any given time only a few of the bouys will be at that depth. This means that some kind of inference, based on past measurements, or current measurements at different depths might be necessary. In this case, fusion of ANC with models would be very useful.

5.2.3 ANC Adaptation Improvements

As mentioned above, scenario comparisons of both NC and ANC could be fed back into ANC to improve the choice of filter parameters for given input scenarios. Similarly, in data sets where additional information is available, it could be beneficial to make use of this information in the reconstruction.

In cases where, for example, spatial covariance is known, fusing this information into the filter adaptation could improve the accuracy of the output. For this reason, further examination of kriging, and in particular semi-variograms will be conducted.

Also, examining Bayesian techniques could pave the way to providing accuracy estimates, as with Kriging and Kalman filters.

Other possible improvements could include using different distance transforms, including different order Minkowski distances, and Veronoi diagrams, which could improve the way in which a given points nearest input samples are found.

5.2.4 Extension into Extra Dimensions

There are a large number of cases where three-dimensional normalised convolution would be useful. For standard NC, this would be fairly trivial to implement by replacing the standard filters with 3D ones.

Extending AND into 3D would probably be more complicated. The main problem being extending the adaptation algorithms, whilst keeping them relatively fast.

Libraries like PeakStream (`http://www.peakstreaminc.com/`) could accelerate the process by harnessing the unused power of a GPU. For 3D convolutions and transformations, the speed increase could be vary large.

5.3 Other Work

Normalised convolution's ability to reconstruct sparse data suggests that it has potential to work as a compression scheme. Recent work has highlighted the similarity of a field known as *compressive sensing* (see `http://www.dsp.ece.rice.edu/cs/`), which attempts to use a sparse sensing in order to compress data as it's recorded. Normalised convolution based techniques will lead to simple and fast compressive sensing based systems.

5.4 Plan of Future Work

This section details goals for the next two years, the remaining duration of the project.

5.4.1 Goals for Year Two

- Complete work on characterising input data, and characterise current ANC algorithm.
- Complete case study using Halloween Storm data, publish results.
- Submit paper to BMVC/ICIP. Present posters or talks, subject to acceptance.
- Begin work on ARGO data.

5.4.2 Goals for Year Three

- Complete work on ARGO data. Publish results.
- Write thesis.

5.4.3 Thesis Chapters

This list show briefly how the project thesis will be structured:

- **Introduction**: including background on reconstruction and gridding, then normalised convolution.
- **Adaptive Normalised Convolution**:
  showing how normalised convolution can be adapted to improve output quality.
- **Application to GPS TEC Mapping**:
  starting with GPS background, and moving on to reconstructing GPS TEC data.
- **Application to ARGO Ocean Salinity and Temperature Mapping**:
  again, starting with an introduction to the ARGO system, and moving onto reconstructing data.

A

Variograms for Spatial Data Analysis

This chapter represents initial investigation into variograms, which was completed after the initial report was completed.

A.1 Introduction

Variograms, and by extension semi-variograms have found wide use in spatial data analysis since the first half of the 20th century, as methods for estimating the spatial autocorrelation of data sets. However, the term *variogram* was not coined until 1962, by Matheron [32] – a man widely regarded as the father of spatial statistics. Whilst originally conceived as a tool for estimating ore reserves in the mining industry, variograms are now considered one of the most important tools in spatial data analysis, an area which has become known as geostatistics.

The basic definition of the variogram is as follows:

$$\text{var}(Z(\mathbf{s}_1) - Z(\mathbf{s}_2)) = 2\gamma(\mathbf{s}_1 - \mathbf{s}_2), \text{for all } \mathbf{s}_1, \mathbf{s}_2 \in D \tag{A.1}$$

In the above equation, D represents the thing that has been sampled, and $\mathbf{s}_1$ and $\mathbf{s}_2$ represent sample indices. $Z(\mathbf{s}_i)$ refers to the value associated with sample $\mathbf{s}_i$). It is a function of the value of difference between $\mathbf{s}_1$ and $\mathbf{s}_2$, which corresponds to the distance between samples.

The variogram above is given by $2\gamma(\cdot)$, dividing by two gives the semi-variogram ($\gamma(\cdot)$) – the distinction is important because, as Cressie [33] puts it:

> ... there is too much to loose from missing 2s.

The classical variogram estimator was defined my Matheron [32] as:

$$2\hat{\gamma}(\mathbf{h}) = \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} = (Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2 \tag{A.2}$$

Where (using the notation of Cressie [33, pp. 69]):

$$N(\mathbf{h}) \equiv \{(\mathbf{s}_i), Z(\mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}; i, j = 1, \ldots, n\} \tag{A.3}$$

This means that $N(\mathbf{h})$ is the set of pairs of sample positions, and $|N(\mathbf{h})|$ is the number of distinct pairs in $N(\mathbf{h})$. In time series analysis, $\mathbf{h}$ is known as lag. For the one-dimensional case this will be equal to the binomial coefficient $\frac{N!}{K!(N-K)!)}$ where $k = 2$, and N is the number of available samples. The two dimensional case is more complicated, and will be discussed later.

As mentioned above, the variogram provides an analysis of spatial autocorrelation of a set of measurements. When data for a one-dimensional series with no gaps, the variogram can be approximated by [16, pp. 273]:

$$\gamma(\mathbf{h}) \simeq c\text{var}(Z)(1 - \text{a.c.f}(\mathbf{h}) \tag{A.4}$$

Where a.c.f$(\mathbf{h})$ is the autocorrelation function, and c is a constant.

When the data points are spaced irregularly, the condition imposed by (A.3) must be relaxed, since not all lags will be possible. In this case, the estimator is smoothed by defining a tolerance region around each lag, and then estimating the variogram coefficients as in (A.2). In this case though, a weighted average could be used in place of the arithmetic mean (see, for example Omre [34] who found that the robust estimator shown below (A.5) gave slightly worse results that (A.4) for the 'idea case' but better results otherwise). Tolerance regions should be chosen such that as many as possible contain a statistically significant number of samples (over 30). Generally the tolerance regions will be chosen such that each region is disjoint, although a moving window based estimator could be used, in which case, there will be overlap between regions.

A.2 Robust Estimators

The classical estimator, given in (A.2) is not particularly robust, meaning that the results will be significantly altered by contamination by outliers. For this reason, various people have looked at robust statistical methods for variogram estimation. Two possible robust estimators are:

$$2\bar{\gamma}(\mathbf{h}) = \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} = \frac{\left\{|Z(\mathbf{s}_i) - Z(\mathbf{s}_j)|^{\frac{1}{2}}\right\}^2}{\left(0.457 + \frac{0.494}{|N(\mathbf{h})|}\right)} \tag{A.5}$$

and:

$$2\tilde{\gamma}(\mathbf{h}) = \frac{\left[\text{med}\left\{|Z(\mathbf{s}_i) - Z(\mathbf{s}_j)|^{\frac{1}{2}} : \mathbf{s}_1, \mathbf{s}_2 \in N(h)\right\}\right]^4}{0.457} \tag{A.6}$$

Where the denominator terms correct for bias. The next section shows how these estimators behave relative to one another.

A.3 1D Example

The following MATLAB code generates a data set with autocorrelation, and then samples it using *random stratified sampling*, which is described in section A.5:

```
data = conv(ones(15,1)./15, randn(200,1));
samp = 4;
% random stratified sampling
xx = [1:samp:length(data)-1];
xx = xx + round(rand(size(xx)) * samp);
yy = data(xx);
```

The three estimators, and the variogram generated using (A.4) are shown in figure A.2.

The main features of note are the fact that the autocorrelation estimator is far too smooth, and the fact that the robust median estimator is very noisy compared to the other plots. In general, though, there is good correlation between all four curves. The fact that they are all similar to the autocorrelation function, a standard analytical tool, means that they can be trusted as good estimates of spatial relationships within the input data.

A.4 2D Variograms

When the data sets is 2D (or higher dimensional), the lag $\mathbf{h}$ becomes a vector. This leads to the question of how to partition the lags, since data could have spatial autocorrelation which is a function of direction as well as distance. Data which have variograms which are not dependent on direction are known as *isotropic*, similarly, when the variogram is a function of direction, the data are *anisotropic*.

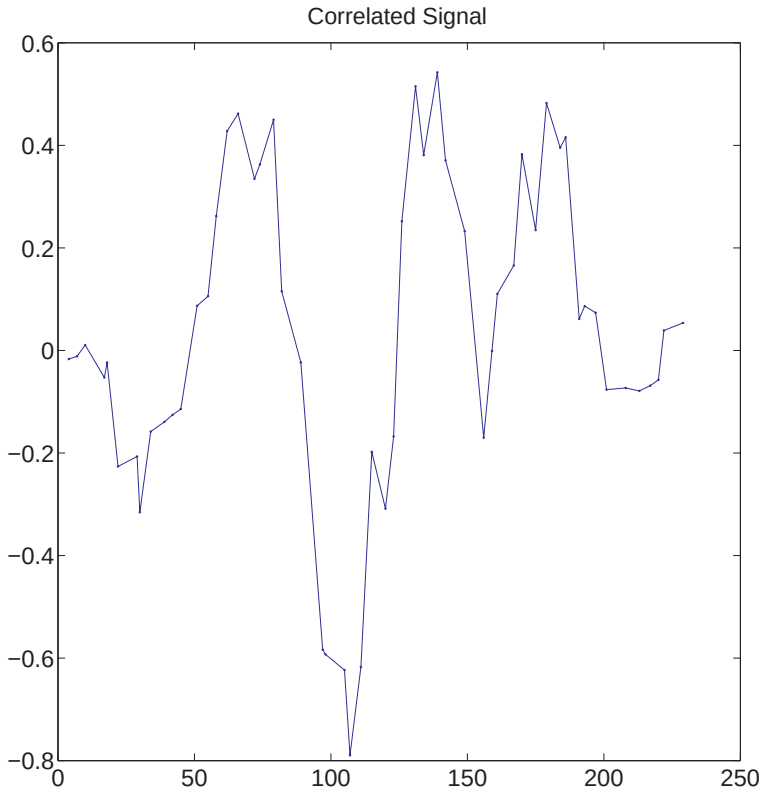


Figure A.1: Example autocorrelated signal.

Standard practice involves converting the Cartesian lag vectors into a polar form, and then grouping all lags within a certain range of angles *and* magnitude together. Converting angles to be modulo π increases smoothing further.

A.5 Example: Simulated Isotropic Data

Data in this section were generated according to the method given in Omre [34], that is:

- Create a 120×120 field of normally distributed data.
- Filter this with a circular filter of radius 15 – thereby adding autocorrelation, since the circular filter acts as a kind of running mean.
- Sample the data using *stratified random sampling* (see below). Use a block size of 8×8, and take one sample from each block.

The data used for this report can be seen in figure A.4.

Stratified random sampling [35, pp. 19] works by dividing the sample points into subareas (or *strata*), and then randomly choosing a set number of points from within each area. This approach has been shown to work well in situations where there is strong local positive corre-

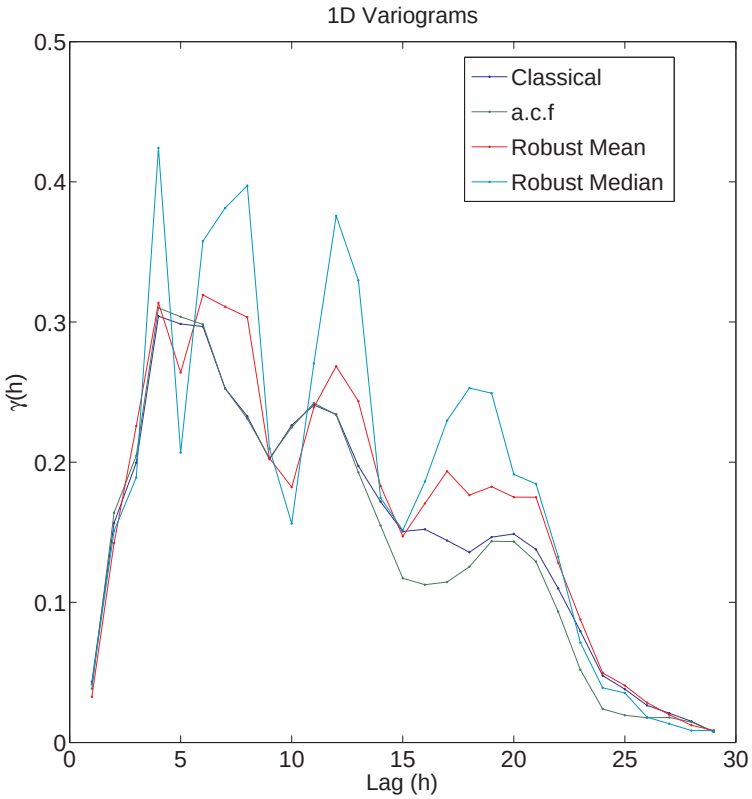


Figure A.2: Example Sample Variogram, using the classical estimator (A.2), the two robust estimators (A.5) and (A.6) and the a.c.f estimator (A.4).

lation – such as the imposed correlation above.

The variograms shown in figure A.5 were generated using a tolerance of ± 2.5, and encompass all lag angles. It is interesting to not the similarity of all three estimators in this case – in particular the robust median follows the classical estimator very closely.

A.6 Example: Wolfcamp Aquifer Data

In order to verify 2D variogram generation code, a test was run on some commonly available data: the Wolfcamp Aquifer piezometric-head data, taken in 1986. Piezometric head data readings give the elevation of the water table at a given point, measurements are taken by drilling a hole into the aquifer and allowing the water to rise until it reaches equilibrium. The height of the surface above sea-level forms the measurement. The Wolfcamp Aquifer data were taken in a region about 300 miles square (~ 200 km), around Amarillo, Texas, to assess the best position for a nuclear waste 'repository'. Figure A.6 shows the measurements as a 3D scatter plot. The data are clearly anisotropic, in that there is a clear different in how the values change with direction. For this reason, two variograms were taken, following the lead of Cressie [33, pp. 261]. Lags were partitioned into 8 km bins, and one variogram is constructed using lags with angles from the range $[0 \leq \theta < \frac{\pi}{2}]$ radians, the second variogram was constructed with

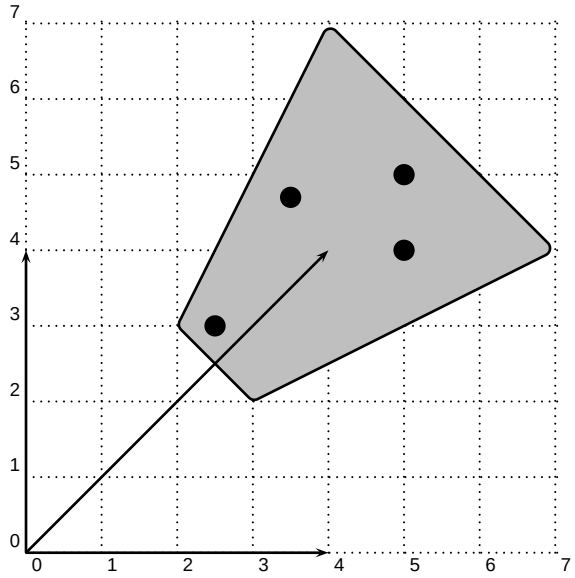


Figure A.3: A Lag vector, showing an example tolerance region. The black spots show lag vector end points which would be accepted.

angles from the range $[\frac{\pi}{2} \leq \theta < \pi]$, where θ is the angle found when converting from Cartesian to polar form.

A.7 Example: GPS Data

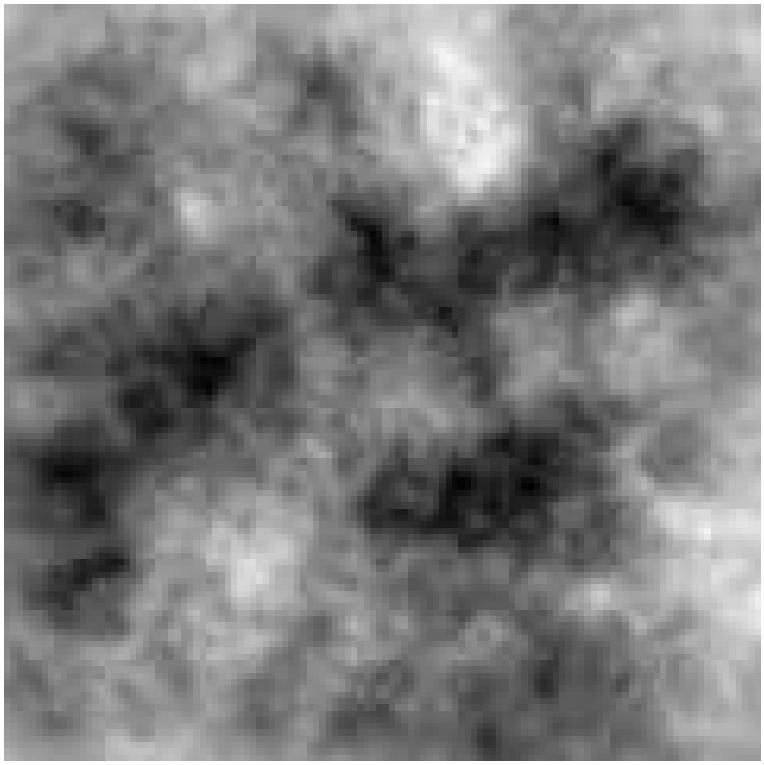


Figure A.4: 2D autocorrelated data.

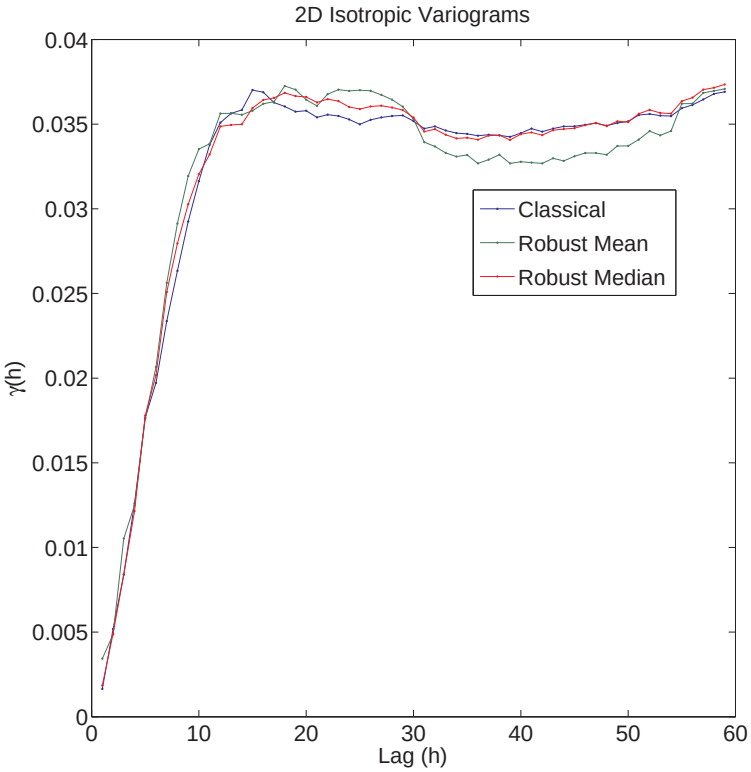


Figure A.5: Isotropic Variograms created from simulated 2D autocorrelated data, and then sampled using random stratified sampling.

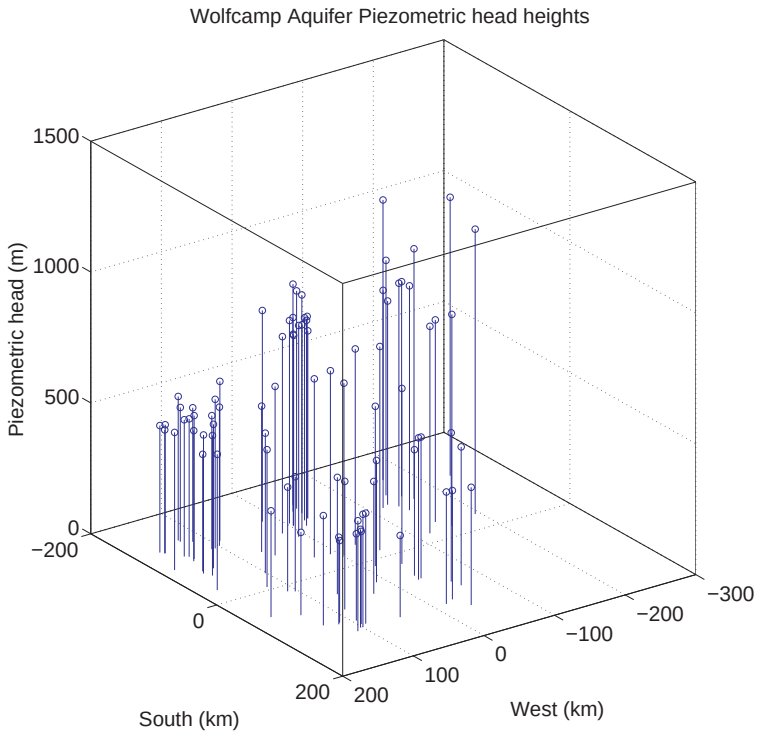


Figure A.6: Stem plot of Wolfcamp Aquifer data. Coordinates are from an arbitrary (unspecified) origin.

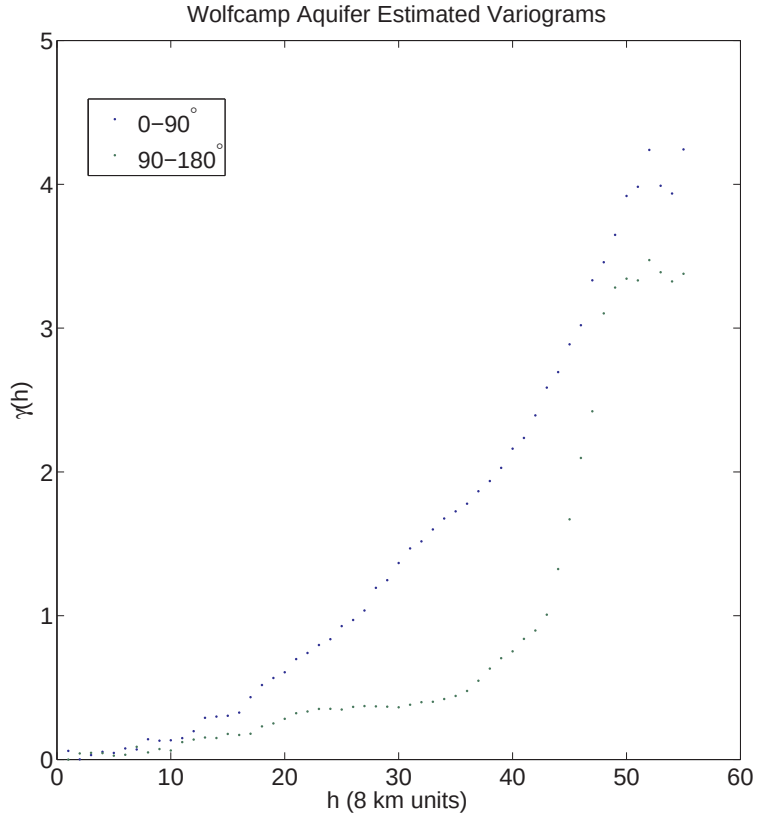


Figure A.7: Classical sample variogram of Wolfcamp aquifer data

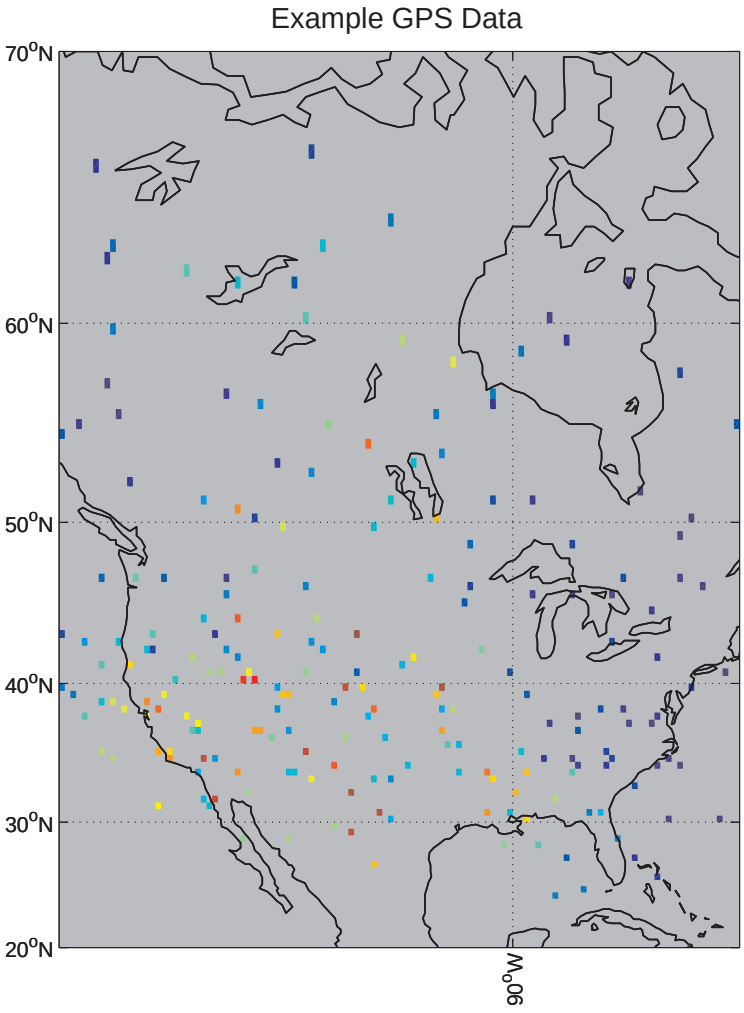


Figure A.8: Example GPS data

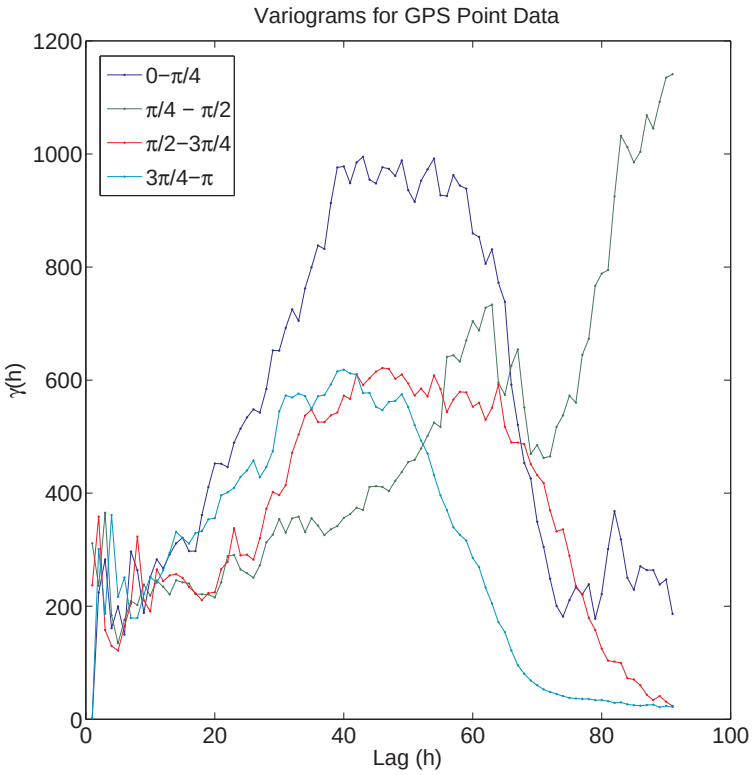


Figure A.9: Classical Sampling Variogram of GPS data

References

[1] F. F. Sabins, Jr. *Remote Sensing: Principles and Interpretation 2/E*. W. H. Freeman and Company, 1987. ISBN 0-7167-17930-X.

[2] T. M. Lillesand and R. W. Kiefer. *Remote Sensing and Image Interpretation*. John Wiley & Sons, Inc., 1994. ISBN 0-471-30575-8.

[3] J. Karvanen and A. Cichocki. Measuring sparseness of noisy signals. In *ICA2003*, 2003.

[4] C. Heiri, H. Buggmann, W. Tinner, O. Heiri, and H. Lischke. A model-based reconstruction of holocene treeline dynamics in the central swiss apls. *Journal of Ecology*, pages 206–216, 2006.

[5] D. Derou, J. Dinten, L Herault, and J. Niez. Physical-model based reconstruction of the global instantaneous velocity field from velocity measurement at a few points. In *Workshop on Physics-Based Modeling in Computer Vision*. IEEE, 1995. ISBN 0-8186-7021-5.

[6] S. Guinehut, P-Y. Le Traon, G. Larnicol, and S. Philipps. Ocean data assimilation. *Journal of Marine Systems*, 46:85–98, 2004.

[7] A. Mannucci, B. Iijima, U. Lindqwister, and B. Wilson X. Pi, L. Sparks. GPS and ionosphere. In *URSI Reviews of Radio Science*. JPL, March 1999.

[8] J. D Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice 2/E in C*. Addison Wesley, 1997. ISBN 0-321-21056-5.

[9] A. Boucher, K.C. Seto, and A.G. Journel. A novel method for mapping land cover changes: Incorporating time and space with geostatistics. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(11), 2006. ISSN 0196-2892.

[10] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252.

[11] J. Blanch, T. Walter, and P. Enge. Application of spatial statistics to ionosphere estimation for waas. In *Proceedings of ION NTM*, 2002.

[12] J. Blanch. *Using Kriging to Bound Satellite Ranging Errors Due to the Ionosphere*. PhD thesis, Dept. of Aeronautics and Astronautics, Stanford University, 2003.

[13] P. Wielgosz, D. A. Grejner-Brzezinska, and I. Kashani. Regional ionosphere mapping with kriging and multiquadric methods. *Journal of Global Positioning Systems*, 2003.

[14] V. Hlavac M. Sonka and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS, 1999. ISBN 0-534-95393-X.

[15] N. A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, Inc., 1991. ISBN 0-471-84336-9.

[16] C. Chatfield. *The Analysis of Time Series*. CRC Press LLC, 2004. ISBN 1-58488-317-0.

[17] H Knutsson and C.-F. Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Proceedings of Computer Vision and Pattern Recognition (Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition'93)*, pages 515–523, New York City, USA, June 16–19 1993.

[18] 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.

[19] C-F. Westin and H. Knutsson. Tensor field regularization using normalized convolution. In *Proceedings of the Ninth International Conference on Computer Aided Systems Theory (EUROCAST)*, volume 2809 of *Lecture Notes in Computer Science*, February 2003.

[20] G. Farneback. *Polynomial Expansion for Orientation and Motion Estimation*. PhD thesis, 2002. URL `citeseer.ist.psu.edu/ack02polynomial.html`.

[21] P.E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, February 1980.

[22] 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.

[23] R. Piroddi and M. Petrou. *Dealing with Irregular Samples*, volume 132, pages 109–165. Elsevier Inc, 2004.

[24] L. J. van Vliet and P. W. Verbeek. Estimators for orientation and anisotropy in digitized images. In *Proc. 11th Scandinavian Conf. Image Analysis*, 1999.

[25] R. C. Gonzalez and R. E. Woods. *Digital Image Processing 2/E*. Prentice Hall, 2001. ISBN 0-13-094650-8.

[26] D. E. Knuth. *The Art of Computer Programming: Volume 1 Fundamental Algorithms 3/E*. Addison Wesley, 1997. ISBN 0-201-89683-4.

[27] J. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, August 2002.

[28] A. Leick. *GPS Satellite Surveying 2/e*. John Wiley & Sons, Inc., 1995. ISBN 0-471-30626-6.

[29] 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.

[30] B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS Theory and Practice 5/e*. Springer Wein New York, 2001. ISBN 3-211-83534-2.

[31] N. A. Thacker, A. F. Clark, J. Barron, R. Beveridge, C. Clark, P. Courtney, W. R. Crum, and V. Ramesh. Performance characterisation in computer vision: A guide to best practices. URL `http://www.tina-vision.net/docs/memos/2005-009.pdf`. April 2005.

[32] G. Matheron. Trait de gostatistique applique. *Editions Technip, Paris*, Tome 1:334, 1962.

[33] N. A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, Inc., 1991. ISBN 0-471-84336-9.

[34] H Omre. The variogram and its estimation. *Geostatistics for Natural Resources Characterization, Part*, 1:107–125, 1984.

[35] B D Ripley. *Spatial Statistics*. Wiley-Interscience, 2004. ISBN 0-471-69116-X.