Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# Least-squares Optimal Interpolation for Direct Image Super-resolution

A thesis presented in partial fulfilment of the requirements for the degree of
Doctor of Philosophy
in
Engineering

At Massey University, Palmerston North,
New Zealand.

Andrew Gilman

2009

In Loving Memory

Of

Alexander Gilman

20 May 1955 – 5 February 2006

At Rest

Think of me as one at rest,
for me you should not weep
I have no pain no troubled thoughts
for I am just asleep
The living thinking me that was,
is now forever still
And life goes on without me now,
as time forever will.

If your heart is heavy now
because I've gone away
Dwell not long upon it friend
For none of us can stay
Those of you who liked me,
I sincerely thank you all
And those of you who loved me,
I thank you most of all.

And in my fleeting lifespan,
as time went rushing by
I found some time to hesitate,
to laugh, to love, to cry
Matters it now if time began
If time will ever cease?
I was here, I used it all,
and now I am at peace.

# Abstract

Image super-resolution aims to produce a higher resolution representation of a scene from an ensemble of low-resolution images that may be warped, aliased, blurred and degraded by noise. There are a variety of methods for performing super-resolution described in the literature, and in general they consist of three major steps: image registration, fusion and deblurring. This thesis proposes a novel method of performing the first two of these steps.

The ultimate aim of image super-resolution is to produce a higher-quality image that is visually clearer, sharper and contains more detail than the individual input images. Machine algorithms can not assess images qualitatively and typically use a quantitative error criterion, often least-squares. This thesis aims to optimise least-squares directly using a fast method, in particular one that can be implemented using linear filters; hence, a closed-form solution is required.

The concepts of optimal interpolation and resampling are derived and demonstrated in practice. Optimal filters optimised on one image are shown to perform near-optimally on other images, suggesting that common image features, such as step-edges, can be used to optimise a near-optimal filter without requiring the knowledge of the ground-truth output. This leads to the construction of a pulse model, which is used to derive filters for resampling non-uniformly sampled images that result from the fusion of registered input images. An experimental comparison shows that a $10^{th}$ order pulse model-based filter outperforms a number of methods common in the literature.

The use of optimal interpolation for image registration linearises an otherwise non-linear problem, resulting in a direct solution. Experimental analysis is used to show that optimal interpolation-based registration outperforms a number of existing methods, both iterative and direct, at a range of noise levels and for both heavily aliased images and images with a limited degree of aliasing. The proposed method offers flexibility in terms of the size of the region of support, offering a good trade-off in terms of computational complexity and accuracy of registration. Together, optimal interpolation-based registration and fusion are shown to perform fast, direct and effective super-resolution.

# Author's Publications

**A. Gilman**, D. G. Bailey, and S. Marsland, "Image Models for Image Super-resolution," in Proceedings of IEEE International Symposium on Electronic Design, Test and Applications, Hong Kong, 2008, pp. 55-60.

**A. Gilman**, and D. G. Bailey, "Noise Characteristics of Higher Order Predictive Interpolation for Sub-Pixel Registration," in Proceedings of IEEE Symposium on Signal Processing and Information Technology, Cairo, Egypt, 2007, pp. 269-274.

D. G. Bailey, and **A. Gilman**, "Bias of Higher Order Predictive Interpolation for Sub-pixel Registration," in International Conference on Information, Communications and Signal Processing, Singapore, 2007.

**A. Gilman**, and D. G. Bailey, "Near optimal non-uniform interpolation for image super-resolution from multiple images," in Proceedings of the Image and Vision Computing New Zealand, Great Barrier Island, New Zealand, 2006.

D. G. Bailey, **A. Gilman**, and R. Browne, "Bias characteristics of bilinear interpolation based registration," in Proceedings of IEEE TENCON, Melbourne, Australia, 2005.

# Acknowledgements

First and foremost, I would like to thank my supervisor Associate Professor Donald Bailey who has been in it from the start to the end. Thank you for letting me randomly pop up out of the blue and distract you with my nonsense for a good five years; thank you for all your support and guidance and correcting pages and pages of my drafts.

Secondly, I would like to thank my co-supervisor Dr Stephen Marsland for his help and friendship. Without his mathematical know-how and the ability and patience to transfer some of that knowledge onto me I would not be able to finish.

Next, I would like to thank my family – my Mom, Dad, Nastya, Grandparents and of course Larissa for all the love and support you have given me. I would not be here without you. A special thanks to Mom for taking over Dad's job of constantly phoning me up and pestering about how the work is going. I would also like to thank my friends Max and Chris for constant encouragement, support and advice.

Last, but by far not least I would like to thank my Love. Jess, thank you for being there for me twenty four-seven. I could not wish for a better partner in life.

# Table of Contents

# List of Figures

# Chapter 1

## Introduction

The quest for higher-resolution images seems to be unceasing. As the continuous development of new digital cameras clearly demonstrates, spatial resolution is a major factor in the quality of a digital image. High-resolution images contain finer detail and are usually more aesthetically pleasant for viewing, albeit at the cost of larger storage requirements. The extra detail in higher resolution images provides additional information that can generally be used by image processing applications to achieve better results; this is particularly important in applications where imaging provides data for further analysis, such as medical imaging, remote sensing, security systems, military applications and machine vision research.

The pixels of an image provide a set of samples of the scene. One of the factors that limits achievable resolution is the sampling density (spacing between the pixels) Therefore, an obvious approach to acquiring higher resolution images is to use image capturing devices with smaller sensor elements; this is an important area of commercial and academic research. However, there are physical limitations on the process of miniaturisation, principally that reductions in the size of pixel capture regions lead to higher levels of photonic shot noise[1] in each pixel. Therefore, there exists a limit in pixel size reduction at which the shot noise begins to cause serious degradation in image quality[2]. Another physical way of increasing spatial resolution is to pack more pixels on a chip by increasing its size. However, with the increased size, it becomes difficult to attain a good yield during chip manufacture, making large chips very expensive. Other physical difficulties, such as increased capacitance [1], make it difficult to speed up the charge transfer rate, reducing the practical effectiveness of this approach [3]. Consequently, a new approach for increasing spatial resolution that avoids these physical limitations is required.

---

[1] This type of noise is associated with the random arrival of photons at any individual sensor element, which is a homogenous Poisson process. As a result, the signal-to-noise ratio (SNR) due to shot noise is proportional to the square root of the expected number of photons arriving during the exposure. As the pixel size decreases, so does the number of photons collected by the pixel during the exposure time, and the signal-to-noise ratio decreases.

[2] It is estimated in [1] that this limit for the pixel surface area is about 40 $m^2$ for a 0.35 m CMOS process. Current consumer technology has already surpassed this limit [2].

Rather than modifying the sensor hardware, an alternative method of increasing the resolution is to capture multiple images of a scene and fuse them in some way. If there is some small movement of the camera over the sequence of frames then the combined sample density of this sequence is higher than that of any individual image. Combining the images in the sequence to produce a single higher resolution image is known as image super-resolution[3], and a fast, direct method of performing this procedure is the topic of this thesis. Current technologies for image capture can be used to produce the image set, and a higher resolution image produced from this set.

## *1.1. Background Information*

### 1.1.1. Imaging model

To understand the super-resolution problem it is essential to have a model of the image capture system. The image capture process is shown in figure 1.1.

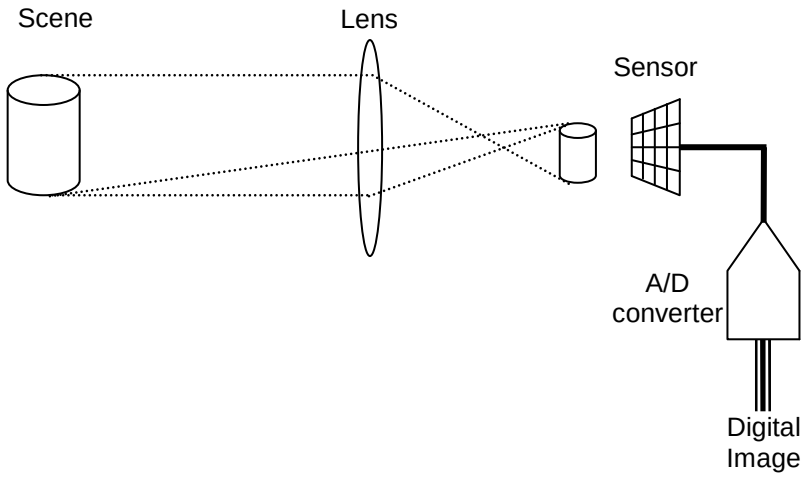


Figure 1.1 Image capturing process.

Light is reflected from the scene and focused by a lens onto a two dimensional sensor, typically a charge-coupled device (CCD) or CMOS sensor. The charge that accumulates on the sensor is integrated over some finite period of time and then converted into a voltage signal that, in turn, is digitised by the analogue-to-digital converter (ADC). As a result, a digital image is produced. The complete process can be represented in an observation model, such as that depicted in figure 1.2.

[3] The term super-resolution is also used in the context of recovering spatial frequencies in an image beyond the diffraction limit of the optical system [4]; however, these are two different fields of research and we focus entirely on the one defined above.

The warp in figure 1.2 represents the geometric transformation of the scene coordinates to image plane coordinates. It therefore embodies the characteristics of the lens (including any geometric distortions) as well as the position of the camera relative to the scene.

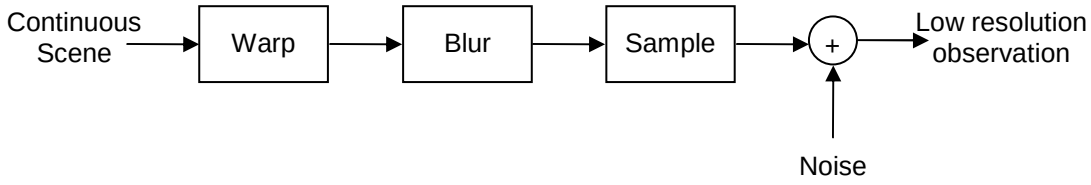


Figure 1.2 Observation model.

The imaging process will introduce blur from at least two sources. The first is from the optical transfer function of the lens. The second results from area integration within the sensor. Each sensor element has a certain physical size, which is maximised to give good sensitivity and reduce noise. Photo-electrons resulting from light incident on one part of the element cannot be distinguished from those on another part; all are integrated. Area sensing, therefore, is equivalent to convolving the continuous image with a rectangular function (assuming the pixels have the same dimensions and are rectangular) prior to sampling. This rectangular point spread function (PSF) corresponds to a two-dimensional sinc frequency response (see figure 1.3).

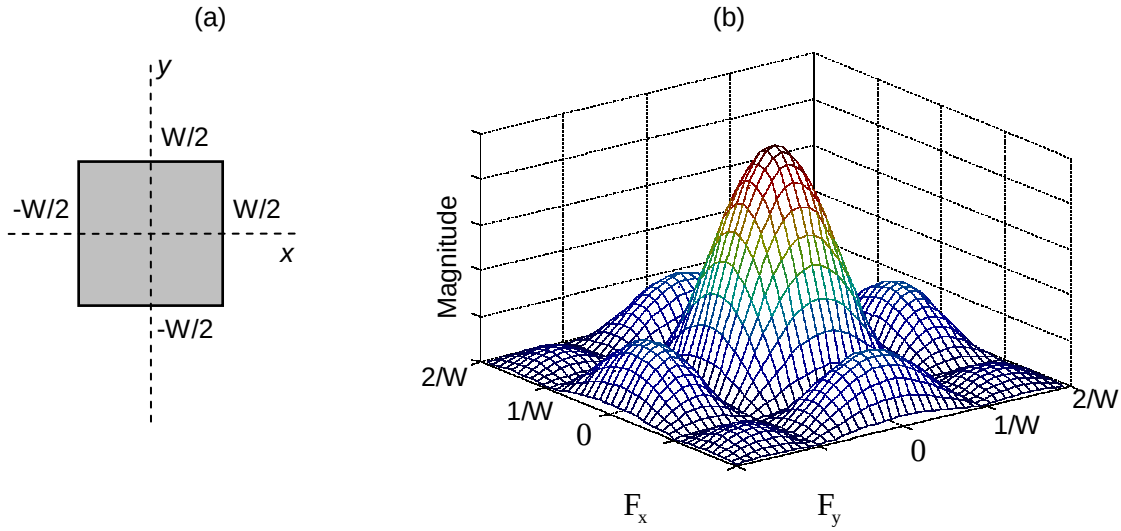


Figure 1.3 (a) Pixel dimensions and (b) magnitude of the resulting sinc PSF.

The final step within the image capture process is sampling. At this stage, the image is assumed to be contaminated with noise. All imaging systems introduce noise from a range of sources – imaging (shot) noise, thermal noise, quantisation noise, and so on.

It is reasonable to assume that the combination of all of these sources can be modelled by uncorrelated additive white Gaussian noise with zero mean and standard deviation $^2$, constant for all images [5, 6].

### 1.1.2. Super-resolution model

For super-resolution to be possible there must be multiple observations of the scene that each contain slightly different information. The multiple observations are generally created by capturing a set of images from slightly different positions. For a set of observations to contain more information than any one of its constituent images, a single image must not contain all of the information. If each low-resolution image was sampled at the Nyquist rate (or higher) then each image contains sufficient information to be able to recover the original scene exactly by using sinc interpolation [7]. This means that any single image would be able to provide all of the information required for reconstruction at any desired resolution and having multiple independent images would not provide an improvement (other than reducing any noise that may be present in the captured images). Hence, for multiple images to provide additional information, the sample frequency for each individual image must be below the Nyquist rate, with the images subject to aliasing.

Image super-resolution is therefore the reconstruction of a high-resolution representation of the scene based on a number of degraded (warped, blurred, aliased and contaminated by noise) input low-resolution images.

It is important that the process of creating multiple observations of the scene results in low-resolution images that are shifted relative to each other by a fraction of a pixel. If the relative translations are integer, there is no new information (about the overlap area) present in additional observations. However, if additional observations are shifted by different sub-pixel amounts the additional observations provide additional information, as the sampling (aliasing) affects these images in different ways. The process of constructing a higher resolution image increases the effective sample rate, untangling the aliased information so that the output image contains more information than that available from any of the individual input images.

The multiple observations are assumed to be taken effectively simultaneously so that there is no local motion within the image. With minimal rigid movement of the camera between captured images, the warp can be represented by a global geometric

transformation. In this thesis it will be assumed that the warp is global translational, as in many other works, for example [8-15]. This form of motion may be produced by a planar scene, with movement of either the scene or the camera parallel to the scene plane, or alternatively a 3D scene with movement of the sensor in the image plane. If the camera is moved relative to a 3D scene, it is assumed that the movement is sufficiently small for parallax to be negligible and that no perspective distortion is introduced.

Image blur also has implications for super-resolution. The relatively slow drop-off of response with frequency of area-sampling means that there may be significant aliased high-frequency content that can be recovered by super-resolution. The nulls in the frequency response also imply that some frequencies are lost completely and cannot be recovered without using additional knowledge of the scene; the presence of these nulls also makes super-resolution ill-conditioned. The roll-off of the lens optical transfer function is usually much sharper than that of area-sampling. The lens will therefore provide a practical limit on the extent to which the higher frequencies may be recovered. This thesis assumes that the dominant source of blur is from area-sampling, and that the lens blur is negligible in comparison. However, if necessary, the two point spread functions may be combined as a single blur function (under the assumption that the lens point spread function is spatially invariant).

The imaging model in figure 1.2 relates captured images to a continuous scene. The aim of super-resolution, however, is to create an image of the scene at a higher resolution (i.e. sampled at a higher rate). It is common to modify this model to a discrete version that relates low-resolution observations to a single high-resolution image, as shown in figure 1.4.

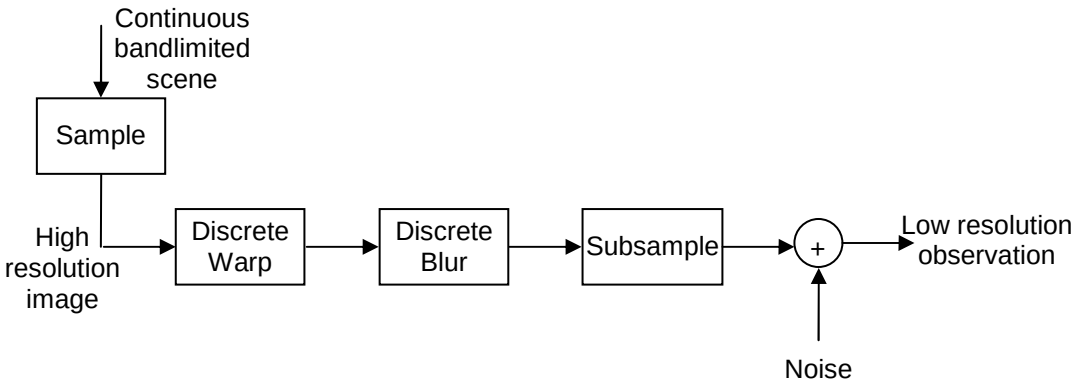


Figure 1.4 Observation model.

This observation model can be expressed mathematically as:

$$g_n(m,n) = s\downarrow\left(h(u,v)\otimes f\left(\mathcal{T}_n(m,n)\right)\right) + \quad (m,n), \tag{1.1}$$

where $\otimes$ represents the convolution operator and

$f$ - high-resolution image

$g_n$ - $n^{\text{th}}$ observed low-resolution image

$h$ - point spread function

$\mathcal{T}_n$ - geometric transformation of $n^{\text{th}}$ image

$s\downarrow$ - downsampling operator by a factor s

- observation noise.

There is a linear relationship between the low-resolution observations and the high-resolution image, in other words any low-resolution pixel can be expresses as a weighted sum of pixels in the high-resolution image. This relationship is often represented in vector-matrix notation, by arranging the high and low-resolution images into column vectors using lexicographical ordering:

$$\mathbf{g}_n = \mathbf{W}_n\mathbf{f} + \xi_n, \tag{1.2}$$

where $\mathbf{f}$ is the high-resolution image; $\mathbf{g}_n$ is the $n^{\text{th}}$ low-resolution observation; $\mathbf{W}_n$ is a matrix containing the linear weights that represent the combination of warp, blur and subsampling that relates the $n^{\text{th}}$ low-resolution image to $\mathbf{f}$; and $\xi_n$ is a vector containing observation noise.

## 1.2. Super-resolution Techniques

Super-resolution image reconstruction dates as back as the early eighties, when Tsai and Huang developed a very simple frequency domain approach to fuse a number of low-resolution images [14]. This has given the rise to the transform domain super-resolution methods [16-20], which are based on the aliasing relationship between the continuous Fourier transform of the original scene and the discrete Fourier transforms of the observed low-resolution images.

These were followed by approaches that are more in keeping with that followed in this thesis – fusing low-resolution data in the pixel domain. The most intuitive approach is to fuse a number of registered low-resolution images to create a

compound image with non-uniform samples. Then, a non-uniform interpolation technique can be utilized to interpolate and resample this compound image at the high-resolution uniform sampling intervals [9, 21-27].

Another common and effective technique is iterative back-projection [9, 28-31], a technique similar to that used in computer-aided tomography. The algorithm starts with an initial guess for the high-resolution image (an upsampled single low-resolution image, for example) and the model of the imaging process is applied to generate an ensemble of simulated low-resolution images. These are compared to the given observations and the error is measured. This error is then back-projected to update the initial guess. The procedure is iterated until either the error falls below an acceptable threshold, or a predefined number of iterations is reached. This method is relatively simple; however, regularisation and bounding of the solution space are rather difficult.

In a similar vein to iterative back-projection, but more mathematically sound, is the model inverse approach, which treats super-resolution as an inverse problem. The forward model (as described in section 1.1.2) is assumed to be responsible for the creation of the low-resolution images from an image of higher resolution, which is typically assumed to be band-limited. Although the observation model described by equation (1.2) is linear, direct computation of its inverse is infeasible, as it is ill-conditioned (and also large, so that even approximation methods run into computational problems of memory and time). The typical approach in the literature is to solve for the inverse iteratively and apply regularisation to make the model well-posed. Regularisers can be split into deterministic and stochastic approaches. The most common deterministic approach is constrained least-squares [10, 32-36], where as there are myriad stochastic approaches, such as [37-44].

A set theoretic approach [11, 37, 45, 46], using the projections onto convex sets (POCS) is an alternative framework for regularisation of super-resolution reconstruction using *a priori* knowledge. Each of the constraints is represented as a closed convex set in the solution space. The high-resolution image estimate is defined as a point in the intersection of the constraint sets and can be found by projecting an arbitrary initial estimate onto the constraint sets. This approach is also iterative.

## 1.3. Motivation for this Thesis

The majority of the existing image super-resolution methods are both iterative and computationally expensive. Although there have been attempts to speed up iterative methods (principally [10, 47]), they are still computationally costly. Thus, inverse methods are deemed unsuitable for fast super-resolution and a direct approach must be taken.

The main motivation behind this work is to investigate fast methods of performing super-resolution in the pixel domain for situations where aliasing is the principal degradation (rather than blur or noise for example). It was decided to concentrate the investigation on interpolation methods, as interpolation is the key to accurate sub-pixel registration, and is also necessary for image fusion. Interpolation methods also have the potential for real-time implementation, in particular, using linear filters.

A particular method of interest is least-squares optimal interpolation. This concept has arisen from previous work by Bailey and Lill [48] and Bailey [49] and is different to conventional interpolation methods where the interpolation kernel is defined by some predetermined function, such as a piecewise cubic polynomial for example. The least-squares optimal interpolator, or simply optimal interpolator, is derived from the data itself by minimising the sum of square errors. Minimising the square error is equivalent to minimising the $L^2$ norm, which is in-line with many existing super-resolution methods. The filter weights can be found using a closed-form solution; hence, the method may be direct and possibly suitable for fast super-resolution.

If the quality of the output of such a fast method is adequate, it can be employed directly in applications with constraints on available computational power. If, however, quality of the output is less than desired, a computationally efficient method can serve as a way to produce a good initial approximation, to be refined using iterative methods. It has been shown that the quality of the initial guess improves the outcome after a predefined number of iterations [50].

## 1.4. Thesis Aims

- To formulate the concepts of optimal interpolation and optimal resampling filters.
- To devise a method of performing optimal interpolation without the need to know the output.

- To demonstrate that optimal interpolation can be used to achieve fast, direct and effective image super-resolution.

These will be tested by:

- analytically investigating the performance of optimal filters for sub-pixel image registration using image models;
- experimentally comparing the registration performance with other registration methods in the literature;
- investigating the performance of optimal filters for fusion of non-uniformly sampled image data;
- experimentally comparing the fusion performance with other non-uniform interpolation methods described in the literature;
- demonstrating that image registration and fusion may be cascaded to give effective super-resolution;
- discussing the implementation of these techniques using small (fast) window filters.

## *1.5. Proposed Approach*

Unless the movement of the camera can be controlled with very high precision (as in microscanning [51] for example), the relationship between the input images is unknown and must be estimated from the images themselves. It is assumed that the lighting is constant, that there is no local motion within the scene, and that the images are related by pure translation. Image registration is employed to find the relative displacements between the input images. As a result of registration, the position of every pixel in the ensemble relative to the reference coordinates is known. A compound image can be formed by mapping all of these onto the reference grid. Although the compound image contains information from all of the input images, it is non-uniformly sampled. The compound image must be transformed into a more useful form by resampling onto a uniform grid. The resampled image may appear blurred because the reconstructed higher frequencies are attenuated by the capturing system's point spread function. A deblurring procedure, or image restoration, can be performed as a third and final step.

Thus, image super-resolution can be broken down into three consecutive steps: image registration, fusion/resampling and image restoration with each step being optimised separately of the other steps. This thesis is primarily concerned with optimising the first two steps – registration and resampling. As image restoration is a relatively mature field, it is not the focus of this work. The deblurring can be performed after the first two steps using any number of existing methods, for example [52-57]. In the case where the result of this direct method is to be used as a first approximation to speed-up iterative reconstruction, deblurring may not be desired in the first instance.

Both image registration and resampling of the non-uniform compound image require implicit reconstruction of the underlying continuous image from discrete samples, using interpolation. Chapter 2 reviews the topic of image interpolation, with particular emphasis on interpolation methods that may be implemented using linear filters. The problem of interpolation is under-determined, as there is an infinite number of continuous functions that pass through the defined sample points. Some prior knowledge about the shape of the underlying function is required in order to arrive at a unique and sensible solution. Different interpolation methods make use of different models in order to limit the number of possible solutions (preferably to a unique solution). However, these models are often not representative of real image data. In chapter 2, a new concept of optimal filtering is introduced.

An optimal filter is one that selects the filter coefficients in such a way as to globally minimise the error for a given filter size. Such filters are based on the image itself, so will provide the best possible model. Optimality is achieved in a least-squares sense, meaning these interpolation filters are guaranteed to achieve the lowest score for any error measure based on square error, such as mean square error (MSE) or peak signal-to-noise ratio (PSNR). Limited prior information on the image leads to near-optimal filters, and a novel filter based on step-edges (a common image feature) is developed. It is shown that this new filter performs better than commonly used filters based on arbitrary models.

Image registration is concerned with finding the best linear transformation between a reference image and a target image. This can be seen as a special case of resampling introduced in chapter 2, where the input and the output images are known in advance and the optimal resampling filter can be computed. Chapter 3 discusses how the

offsets between the images can be derived from the optimal filter's coefficients. This new method of image registration is shown to be superior to conventional methods of similar computational complexity.

In chapter 4 it is shown how the concept of near-optimal interpolation can be utilised in the fusion stage to perform resampling onto a uniform grid. For this purpose, the method is extended to interpolate non-uniformly sampled data. The proposed method is experimentally compared to popular non-uniform interpolation methods that are used with image super-resolution and shown to outperform these.

Chapter 5 combines image registration with fusion and resampling to analyse how image registration errors propagate into the output of resampling and to demonstrate that super-resolution is achieved. The conclusions and discussion of future work directions are given in chapter 6.

# Chapter 2

# Optimal Image Interpolation and Resampling

## *2.1. Image Interpolation*

The concept of image interpolation has been around since the beginning of the digital image era, and the more general concept of data interpolation (predicting unknown points between some known data points) has been around since ancient times. An article by Erik Meijering gives a good overview of how this problem evolved throughout the ages and developed into what is known today as data interpolation [58]. The abundance of research on this subject is clearly illustrated by the incredible number of references – over three hundred and fifty – in [58].

In the digital age, interpolation has become increasingly important and, with the introduction of digital imaging, the subject of image interpolation has become increasingly popular [59, 60]. Today, digital images and video can be found almost anywhere – a few prominent examples are medical imaging, astronomy, microscopy, industrial applications, communication and consumer electronics.

An abundance of existing one dimensional interpolation methods have been applied to the problem of image interpolation in a separable fashion (i.e., treating each spatial dimension separately) [59, 61]. Recently, attempts have been made at producing non-separable kernels for image interpolation [62-64]. However, these are designed specifically for uniformly sampled signals. Apart from methods based on splines [65-67], extension of such methods to non-uniform sampling is rare. Instead, resampling of non-uniform images is based on inherently two dimensional methods, such as the methods widely used in geostatistics [68] and other fields where scattered[4] data is common. These non-uniform interpolation methods will be reviewed in chapter 4.

Image interpolation and resampling have been approached from different angles. There are the traditional approaches of synthesis using basis functions, such as various piecewise polynomials [59] or the sinc function [69] for example. These methods can be considered as linear and space-invariant. For linear functions, (re)sampling an interpolated (continuous) image is equivalent to filtering the discrete image with a

[4] Scattered data is used here synonymously with non-uniform or irregular sampled data.

sampled interpolation kernel, and since the interpolation function is space-invariant, this process can be implemented as a convolutional filter. The correct term for this procedure is image resampling. However, it is often referred to simply as interpolation.

There are also spatially adaptive variants of the above methods, for example [70-75]. These adaptive methods exploit geometric structure or regularity [76] of image features, such as step-edges, ridges and valleys. The orientation of such features is first extracted from the image, and then the interpolation kernel is locally adapted to perform interpolation along the features, rather than across them. Adapting the kernel to local data can improve the result; however, this is considerably more computationally expensive as the interpolation function is no longer spatially-invariant and resampling cannot be implemented as simple convolutional filters.

Alternatively, the problem of interpolation may be considered as an ill-posed inverse problem [77]. The image is assumed to have been produced from either a continuous scene or a higher resolution representation by a series of degradations (such as area-sampling and noise, for example). These degradations are modelled mathematically to form an observation model, which can be inverted to reverse the degradations. Schultz and Stevenson [78] approach this problem using a statistical framework with appropriate probabilistic models used for both the image and the noise. Various other methods exist, offering a least-squares inverse of the ill-conditioned matrix using various regularization terms [79-82]. However, most of these methods were designed specifically for up-sampling a uniform image by an integer factor, and are not easily extendable to non-uniform interpolation, as is required for super-resolution. Other methods that can or have been adopted, like [44] for example, do not have a direct solution for the same reasons as other inverse model super-resolution methods (such as large computational power and memory requirements and ill-conditioning).

Emphasis in this work has been put on methods that can be implemented as linear filters; hence, a direct solution must be feasible and the resulting kernel must be spatially invariant. Synthesis using basis functions fits well with these requirements and is analysed in more detail next.

## 2.2. Linear Methods

Let $f(x,y)$, $x,y \in \mathbb{R}$, be a continuous image and $f[m,n] = f(mT,nT)$ be the uniformly sampled version of $f(x,y)$, where $m$ and $n$ are integer indices and $T$ is the sampling period. The problem of interpolation is concerned with estimating the continuous function $f(x,y)$ from the discrete samples $f[m,n]$.

The generic approach to solving this problem is to represent the estimate of the continuous surface $\hat{f}(x,y)$ as a 2-D convolution of the discrete sequence $f[m,n]$ with a continuous basis function $(x,y)$, producing a continuous output:

$$\hat{f}(x,y) = \sum_{n,m\in\mathbb{Z}} f[m,n] \quad (x-mT, y-nT). \tag{2.1}$$

There are three requirements on the basis function $(x,y)$ for it to be an interpolation kernel:

1. It must satisfy the interpolation constraint [59], which requires that the interpolated surface pass through all of the discrete sample values. This is satisfied if:

$$(0,0) = 1, \text{ and } \quad (mT,nT) = 0, \text{ for all } m,n \neq 0. \tag{2.2}$$

2. It must satisfy the partition of unity constraint [7], also sometimes referred to as the flat-field response. When interpolating a set of discrete samples of a flat surface, a flat surface is desired. This is satisfied if:

$$\sum_{m,n\in\mathbb{Z}} \quad (x-m, y-n) = 1, \qquad \forall x,y \in \mathbb{R}. \tag{2.3}$$

3. The basis function must be symmetric. The frequency response of the 2-D convolution is given by the Fourier transform of the basis function:

$$\Phi(u,v) = \iint \quad (x,y) e^{j(ux+vy)} dxdy. \tag{2.4}$$

If the underlying continuous surface is not to be shifted, the frequency response $\Phi(u,v)$ must have zero phase, that is:

$$\Phi(u,v) = \Phi^*(u,v),$$

where * denotes complex conjugation. This requires that the basis function be symmetric:

$$(x, y) = (x, -y) = (-x, y) = (-x, -y). \tag{2.5}$$

### 2.2.1. Uniform image resampling

The main purpose of interpolation in image super-resolution is image resampling. Resampling is required as part of the image registration procedure and it is also required to produce a uniformly sampled output after image fusion.

A simple case is considered here, that of resampling an image with the same sampling rate but with a small offset, as is shown in figure 2.1. This case is of special interest because it directly relates to registration of translated images described in the next chapter.

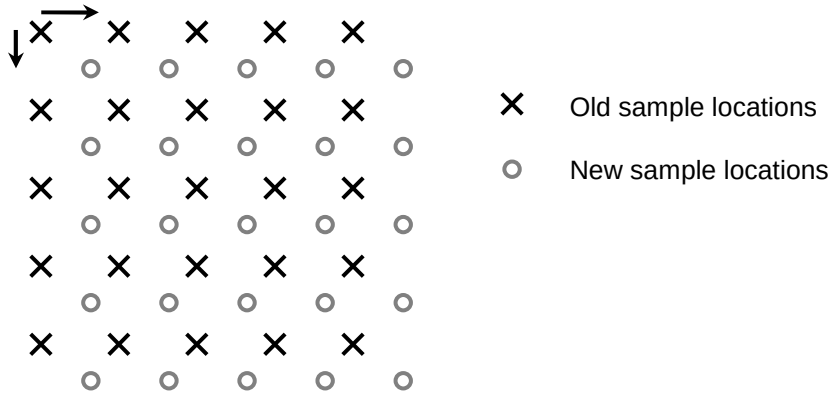


Figure 2.1 Two uniformly sampled images at the same resolution can differ by a sub-pixel offset.

Interpolation and resampling of a discrete image $f[m,n]$ at a location shifted by and in the $x$ and $y$ directions respectively can be accomplished by filtering the image with a sampled interpolation function:

$$\begin{aligned} \hat{f}_{,}[k,l] &= \hat{f}(kT + \ , lT + \ ) \\ &= \sum_{m,n \in \mathbb{Z}} f[m,n]\Phi[k-m, l-n], \end{aligned} \tag{2.6}$$

where the discrete resampling kernel $\Phi[k,l]$ is defined as:

$$\Phi[k,l] = (kT + \ , lT + \ ). \tag{2.7}$$

An alternative expression for equation (2.6) can be obtained by considering that convolution is commutative:

$$\hat{f}_{,}[k,l] = \sum_{m,n\in\mathbb{Z}} f[k-m,l-n]\Phi[m,n]. \tag{2.8}$$

Convolution describes the output of a system in terms of its impulse response. Alternatively, the output can be described as a weighted sum of inputs, where the weights form a digital filter [83]. Mathematically, these two ideas are equivalent; the one difference is that the convolution kernel (the impulse response) is time-reversed or rotated by 180 degrees in the case of a two dimensional filter.

In digital image processing it is common to use the term filtering, rather than convolution. Hence, in the following discussion the terms filtering and filter are used, rather than convolution and impulse response. Replacing the convolution kernel $\Phi[m,n]$ with a filter $h[-m,-n]$, equation (2.8) can be rewritten as:

$$\hat{f}_{,}[k,l] = \sum_{m,n\in\mathbb{Z}} f[k+m,l+n]h[m,n]. \tag{2.9}$$

### 2.2.2. Traditional basis functions

Numerous interpolation basis functions have been proposed in the literature. These can be subdivided into separable and non-separable functions. Most of the traditional interpolation functions were developed for one dimensional signals. When the need to interpolate multidimensional signals, such as images, arose, these one dimensional functions have been adopted to interpolate multi-dimensional data in a separable fashion.

A separable two dimensional kernel can be formed from a one dimensional basis function by considering each dimension separately:

$$_{sep}(x,y) = \quad (x) \quad (y) \qquad \forall x,y\in\mathbb{R}, \tag{2.10}$$

where $(x)$ is the one dimensional basis function. The main advantage of this is that the interpolation of images can be reduced to two one-dimensional convolutions, due to the associative property of convolution, usually with significant computational savings. The disadvantage is that many multidimensional signals (including images) are not necessarily spatially separable [62, 64]. Hence, the design of interpolation functions in a separable fashion is not necessarily optimal.

One common function that is traditionally used for interpolation of one dimensional signals is the sinc function [84], which is the optimal interpolator of a band-limited

signal, reconstructing it exactly. When sampling many electronic signals, it is possible to sample above the Nyquist rate, after applying appropriate anti-aliasing filters. However, almost all images contain some degree of aliasing, due to edges and occlusions. Interpolating such images with a band-limited function may produce unwanted artefacts, such as blurring and ringing.

Being band-limited in frequency, the sinc carries an additional problem of having infinite spatial extent with a relatively slow decay rate of $1/|x|$. This makes the realisation of an ideal sinc function impossible. A very large kernel may be used to approximate it, but this approach results in artefacts around the image boundary, as there is no valid data outside the image boundaries.

A number of approaches have been taken to make sinc interpolation realisable in practice. One such approach is to shorten the extent of the function by employing appropriate apodization. The Hann window [85] is commonly used for its continuously differentiable property. Wolberg [86] reviews a number of apodization functions to be used with sinc interpolation.

Another common approach in computer graphics is to window the sinc function with the mainlobe of another, a lower-frequency sinc. The resulting function is known as the Lanczos kernel [87]:

$$_{Lanczos}(x) = \begin{cases} \operatorname{sinc}(x)\operatorname{sinc}\left(\frac{x}{w}\right) & \text{if } |x| < w \\ 0 & \text{if } |x| \geq w \end{cases} \tag{2.11}$$

Lanczos kernels with regions of support of four ($w = 2$) and six ($w = 3$) pixels are common in image processing, and are referred to as Lanczos2 and Lanczos3 respectively.

Another approach is to approximate the sinc function with cardinal splines of various orders. It has been shown that as the spline order tends to infinity, the cardinal spline function approaches the sinc function [88, 89]. The actual interpolation procedure almost never explicitly uses cardinal splines, as these can be uniquely represented using a B-spline expansion [90, 91].

B-spline basis functions of orders zero and one:

$$\beta^0(x) = \begin{cases} 1 & \text{if } |x| < \tfrac{1}{2} \\ \tfrac{1}{2} & \text{if } |x| = \tfrac{1}{2} \\ 0 & \text{if } |x| > \tfrac{1}{2} \end{cases} \tag{2.12}$$

$$\beta^1(x) = \begin{cases} 1-|x| & \text{if } |x| \le 1 \\ 0 & \text{if } |x| > 1 \end{cases} \tag{2.13}$$

can be used directly in equation (2.1) as the interpolation function because they satisfy all three constraints (2.2), (2.3) and (2.4). The zero order spline is actually very similar to a nearest neighbour interpolator (although nearest neighbour has slightly different boundary conditions) and the first order spline is identical to the linear interpolator. Requiring only one-tap for nearest neighbour and two-tap for linear interpolation, these are the simplest kernels in use.

Higher order spline functions can be generated by convolving B-splines of degree zero with themselves a number of times:

$$\beta^n(x) = \underbrace{\beta^0 * \beta^0 * \ldots * \beta^0(x)}_{(n+1) \text{ times}}. \tag{2.14}$$

Unfortunately, B-splines of order three and above do not satisfy the interpolation constraint of equation (2.2), and if used directly in equation (2.1) will blur the result – this mistake has been made by researchers in the past [92]. For interpolation, cardinal splines (that satisfy the interpolation constraints) must first be built from B-splines [93]; however, there are efficient implementations for computing this, using digital filters [94].

Out of the family of polynomial splines, third order (the cubic) spline is probably most popular; Unser [95] suggests that a possible reason for this popularity is its minimum curvature property – out of all cubic piecewise polynomial functions, the cubic spline has the smallest second derivative.

Cubic spline interpolation gives a remarkable improvement in quality over nearest-neighbour and linear interpolation. However, it comes with an increased cost of computing the cardinal spline from B-splines. Even though the cubic B-spline has finite support, the data is pre-filtered with an infinite impulse response filter, effectively giving the cardinal spline infinite support. In contrast with the sinc function, however, cardinal splines decay at an exponential rate.

Other piecewise cubic polynomials have also been suggested. These functions are interpolating and, unlike the cubic cardinal spline, they have compact support, which makes them faster to compute. However, they do not posses the minimum curvature property mentioned before. Some popular examples of cubic polynomial kernels are the Key's cubic convolution family [96], the piecewise cubic Hermite interpolator [97] and Mitchell's piecewise cubic polynomials [98].

There have also been several attempts to improve interpolation by developing a two-dimensional non-separable kernel based on cubic polynomials [62, 64, 99]. Other non-separable kernels popular in scattered data interpolation are radially-symmetric basis functions (RBF) [100], such as thin-plate splines [101]. Although, these were applied to image interpolation in some special cases, generally their computational complexity is too high for normal use [102].

Many of these approaches are not specifically designed for image interpolation and do not take into consideration statistical properties common to most images, or the image capturing process. The non-linear approaches (such as [78] for example) that do account for these things may offer better interpolation, but at a much higher computational cost, as they require iteration and large matrix inverses. The main desire in this thesis is to develop a linear method that can be implemented as a linear filter, yet take into consideration properties of the images and the image capturing process. In order to do this we need to look at the methods by which the interpolant is fitted to the data.

## 2.3. Optimisation and Norms

The basis function $(x, y)$, from which the resampling filter $h[m,n]$ is derived, is generally chosen arbitrarily with little regard to the underlying data or image capture process. In the cases where the function is based on some underlying process, the assumptions are often severely violated – for example, the use of a sinc function to interpolate aliased images with sharp edges; or the use of linear interpolation on data that is not inherently piecewise linear.

In this thesis, the problem of choosing the 'best' (in some sense) resampling kernel is approached from an optimisation perspective. The error between the output and the desired output is minimised with respect to the resampling filter weights. However,

before the error can be optimised, it must be decided exactly how the error is measured. A typical approach is to use an error function based on some $L^p$ norm:

$$\left\|\mathbf{x}-\mathbf{y}\right\|_p = \left(\sum_{i=1}^{n}\left|x_i - y_i\right|^p\right)^{1/p},\ 1 \le p < \infty . \tag{2.15}$$

The two norms most frequently used in practice (in image processing) are the $L^2$ (the Euclidean norm) and the $L^1$ (the taxicab or Manhattan norm) norms, for example [33, 36, 66, 103]. The $L^2$ norm has the advantage of giving more weight to larger errors; hence, it favours many small errors over a few larger errors. However, this can be detrimental if the data contains outliers – in that case the $L^1$ norm, which weighs all errors equally, is a better choice (another way to see this is that the mean minimises the $L^2$ norm, while the median, which is a robust statistic, minimises the $L^1$ norm) [104].

In the case where there are no outliers, however, the $L^2$ norm has a number of advantages over the $L^1$ norm. The existence and uniqueness of the best solution in $L^2$ norm is guaranteed because it is strictly convex. The norm is smooth, continuous and most importantly differentiable. This means that in general the $L^2$ norm offers mathematical tractability and for a linear system, such as in the case of linear filtering, the best solution can be found using well known linear least-squares minimisation [105]. Another important advantage of the $L^2$ norm is that it is directly related to the variance; hence, by minimising the $L^2$ norm, the variance of the error is also minimized. This makes the $L^2$ norm the most suitable choice for smoothing data in the case of additive white Gaussian noise.

Better robust norms than $L^1$ also exist and some have found use in interpolation and super-resolution. Pham et al. [27], for example, apply a robust norm similar to that of bilateral filtering [106] to non-uniform image interpolation, while Patanavijit and Jitapunkul [107] apply the Tukey's Biweight norm [108] to image super-resolution. Later the same authors [109] also applied the Lorentzian robust error norm [108] with Tikhonov regularization. However, these robust norms lack mathematical tractability and result in a non-linear problem, increasing the computational cost of the solution by a significant margin.

The central characteristic of the $L^2$ norm that makes it favourable for fast interpolation is the ability to arrive at the 'best' linear filter through a direct, closed-form solution. An additional advantage results from the fact that the $L^2$ norm is directly related to many metrics that are used in the literature to measure the quality of interpolation, for example root mean square error [27], peak signal-to-noise ratio [66] and fidelity [110].

Leung et al. [81] have applied least-squares to image interpolation posed as an inverse problem. However, the approach is limited to image zooming; that is resampling a single image at an integer rate of two or more times the original sample rate, where the position of the new sampling grid is strictly limited. Their approach requires determining the inverse of a very large sparse matrix. The autocovariance matrix is also required at the output sample rate. The authors offer a number of autocovariance models and test these by comparing to an autocovariance function of a real image. It is shown that a kernel produced from a model based on a Markov random field resembles a kernel generated from actual image data the most.

Li and Orchard [71] have also proposed a least-squares-based approach. However, their approach is also restricted to image zooming, and cannot easily be adapted to resample an image with an arbitrary translation. They perform highly localized interpolation, using only the four nearest neighbours; hence only the central part of autocovariance is required. They exploit geometric similarity between low-resolution and high-resolution autocovariance functions to estimate the required high-resolution autocovariance from the low-resolution version obtained from the local neighbourhood. Even though the method uses a small kernel, the kernel depends on the local autocovariance and must be recomputed at every pixel, resulting in increased computational cost.

Chen et al. [111] improve on the work of the previous authors by slightly increasing the size of the interpolation kernel and hence, using a slightly larger portion of the autocovariance matrix. However, the low-resolution autocovariance is still used to approximate the high-resolution autocovariance function at every pixel.

The approach of Leung et al. makes explicit use of the image capturing model, whereas the approaches of Li and Orchard and Chen et al. make use of the capturing

model implicitly through the low-resolution autocovariance. Both of these approaches, however, have high computational complexity and very little flexibility.

In this chapter, a more elegant solution to the problem of interpolation and resampling using the least-squares framework is derived that allows an image to be resampled with an arbitrary translation. Chapter 3 will exploit this methodology for image registration and chapter 4 describes how interpolation of non-uniformly sampled data can be performed.

## 2.4. Least-squares Optimisation

The basis of our approach is the observation that the square $L^2$ norm of the error between the interpolated image and the desired output can be expressed as the sum of squares:

$$\left\|\mathbf{f} - \hat{\mathbf{f}}\right\|^2 = \sum_{k,l}\left( f[k,l] - \hat{f}[k,l]\right)^2 . \tag{2.16}$$

Substituting (2.9) into expression (2.16) and minimising it with respect to the filter weights results in a linear least-squares problem:

$$h = \arg\min_{h} \sum_{k,l}\left( f\left(kT + \quad, lT + \quad\right) - \sum_{m,n\in\mathbb{Z}} f[k+m, l+n]\, h[m,n]\right)^2 . \tag{2.17}$$

This minimisation must be performed subject to

$$\sum_{m,n\in\mathbb{Z}} h[m,n] = 1, \tag{2.18}$$

to satisfy the flat-field response constraint described by equation (2.3). This is referred to as ‘optimal’ interpolation in this thesis, because the resulting filter is optimal in the linear least-squares sense.

Note that the solution of (2.17) only requires the inversion of a relatively small matrix. The size of this matrix depends on the size of the region of support. Many common interpolation functions, such as linear, quadratic [112], Keys’ cubic convolution, and Lanczos kernels have very short spatial support. This indicates that small-kernel resampling filters can achieve adequate results.

### 2.4.1. Experiment I – Region of support

***Aim***

To investigate the extent of improvement that can be achieved by increasing the region of support of the optimal spatially-invariant resampling filter.

***Method***

The effect of the region of support size is investigated by evaluating a number of globally-optimal filters with increasing region of support and comparing the results. The interpolation method proposed in this thesis using 2D filters of sizes from 2×2 to 7×7 was used to resample the input images by half a pixel in each direction. This translation was chosen as it generally results in the worst error.


Figure 2.2 Four of the images used in this thesis, referred to as ‘bird’, ‘cat’, ‘face’, and ‘text’.

The experiment is performed using the four different images shown in figure 2.2. These images were chosen as to represent a wide cross-section of image statistics. Image ‘bird’ contains many edges with smooth areas in between. Image ‘cat’ contains a lot of highly textured areas and image ‘face’ contains a combination of the above. Image ‘text’ contains a lot of high frequencies, concentrated in its many sharp step-edges and is significantly different to the three other images of natural scenes.

The input and desired output images are created from a higher resolution version (2592×3888 pixels) of each image using an area-sampling imaging model. The high-resolution image is blurred using a four by four rectangular filter with flat response to simulate area-sampling and sub-sampled by a factor of four to create the input and the desired output images (the high-resolution image is shifted by two pixels in each direction before sub-sampling to create the output image) followed by quantisation to 256 grayscale levels. Optimal filter coefficients are then derived from the input and desired output pair and are used to resample the input image. The root mean square error (RMSE) is calculated between the resampled image and the desired output image and graphed against the filter window size.

### *Results*

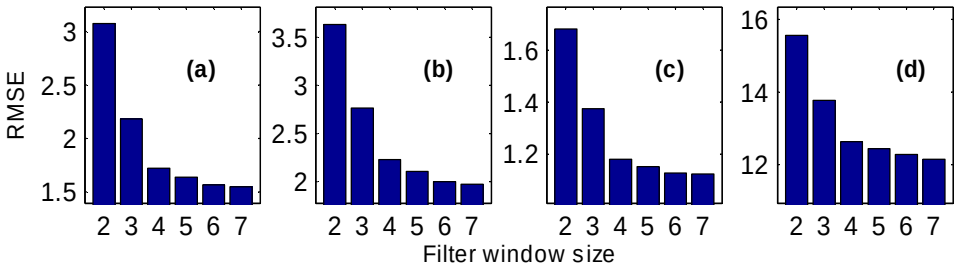


Figure 2.3 Plot of the RMSE against filter window size (2 represents 2×2 window and so on) for images (a) 'Bird', (b) 'Cat', (c) 'Face', (d) 'Text'.

The results are shown in figure 2.3, it can be clearly seen that the improvement decreases dramatically as the filter order increases. There is only a small improvement between a six-by-six and a seven-by-seven window for images 'bird' and 'text' and negligible improvement for the other two images. The small improvement does not compensate for the required increase in computational complexity, considering that a four by four filter has 16 coefficients, whereas the seven by seven filter has 49 coefficients that must be fitted.

### *Conclusions*

Increasing the region of support of the optimal filter beyond a certain point results in a diminishing improvement in interpolation quality at the expense of a growing increase in the computational complexity (computational complexity increases linearly with the number of filter coefficients). Good results may be obtained from a relatively small filter size, typically 4x4 here.

## 2.5. Near-optimal Interpolation

Optimal interpolation presents an efficient and flexible framework for image interpolation and resampling. However, there is an obvious problem with practical implementation – obtaining an estimate of the output requires knowing the desired output. In the remainder of this chapter, a number of ways of solving this problem are presented.

### 2.5.1. Experiment II – Optimal kernel similarity

Conventional linear interpolation methods have filter coefficients that are independent of the data and depend only on the offset. This is in contrast to optimal interpolation, where the filter coefficients depend explicitly on the image data. Any dependence on the offset is implicit through the dependence of the image data on the offset.

A useful question to answer is how different are the optimal kernels generated from different images? If optimal kernels from different images are similar, they should interpolate any image with reasonable results. This would create the possibility of generating a near-optimal kernel that could interpolate any image with near-optimal square error.

#### *Aim*

To investigate whether optimal kernels generalise well and can interpolate other images with near-optimal mean square error.

#### *Method*

The general set-up is similar to that of the experiment I – the input image is translated by half a pixel in each direction using a resampling filter. The four images shown in figure 2.2 are used again. Both input and output image are created from an image with four times the resolution using the area-sampling model.

First, each image is used to derive an optimal (for that image) filter kernel with a window size of four-by-four. Then the four images are resampled using each of these four kernels and root mean square error is computed. Each image is also resampled using Keys' cubic convolution [96] for comparison. This method was chosen as it is widely used and the resulting filter also has a four-by-four window.

***Results***

The resulting RMS errors are plotted in figure 2.4. It can be observed from the results that optimal filters derived from all four images can interpolate all of the images reasonably well. On visual inspection of the output images there are no differences in the resampled images between the different filters. It can also be noticed that, on average, filters optimised on other images actually outperform cubic convolution.

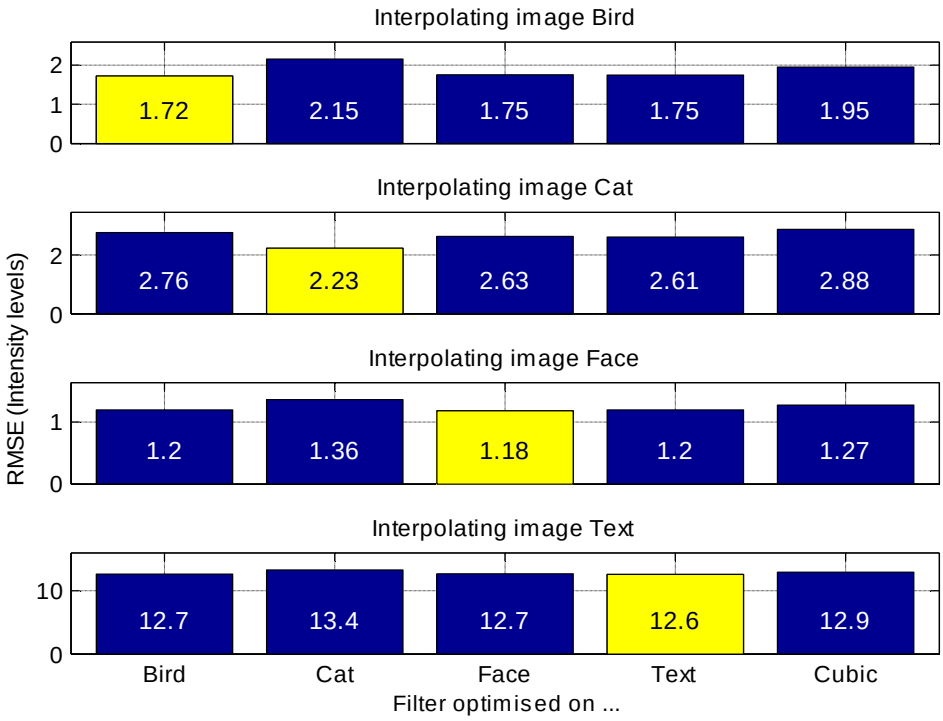


Figure 2.4 Results of near-optimal interpolation. Rows contain results from interpolating different images. Each column contains results of interpolating using a filter optimised on a particular image. The bar that represents the optimal filter is highlighted. The last column is the result of interpolating each image using Keys' cubic convolution [96].

***Conclusions***

The optimal kernel generalises fairly well – filters derived from one image are near-optimal on other images; hence, there must be some features common to all images that are dominant in the formation of the optimal kernel.

### 2.5.2. Experiment III – Step-edge dependence

Consider the process of optimising the resampling filter coefficients. These are chosen in a way that results in the minimum square error. This implies that the filter coefficients are most influenced by the pixels that result in large error, i.e., those that are hard to interpolate. Generally, the parts of an image that are hardest to interpolate

are the ones that contain a lot of high-frequency information, for example high-contrast edges. All four images contain many step-edges of different orientations. These features, therefore, should dominate the shape of the optimal kernel.

### *Aim*

To investigate the extent that step-edges are responsible for the shape of the optimal kernel.

### *Method*

The general set up is the same as experiments I and II. An optimal kernel is derived from an image containing only step-edges. The image consists of a set of concentric rings, pictured in figure 2.5, and contains many edges of different orientations and sub-pixel positions relative to the sampling grid, but very little else.

Figure 2.5 Image ‘rings’ - 630×630 pixels.

The resulting kernel is used to interpolate each of the other four test images. Performance of this near-optimal filter is compared with the best of the other near-optimal filters for each image, and also with Key’s cubic convolution.

### *Results*

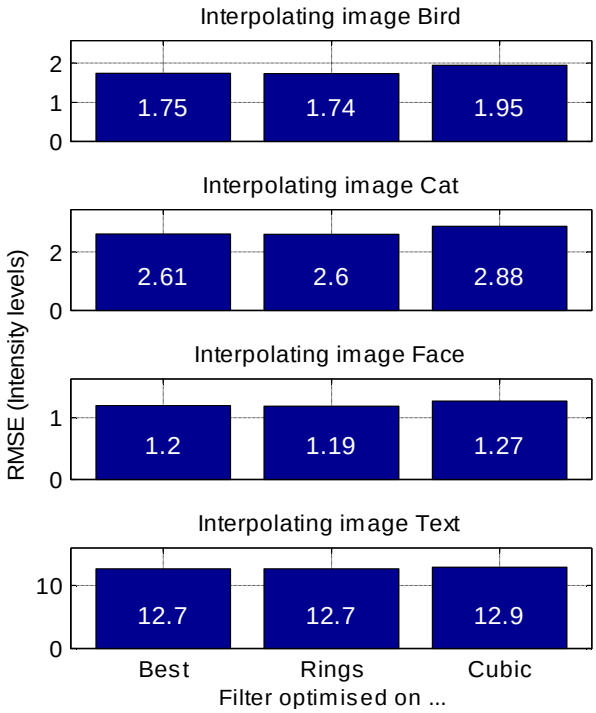


Figure 2.6 Results of near-optimal interpolation using the best-performing near-optimal filter from figure 2.4, a filter optimised on image ‘rings’ and Keys’ cubic convolution.

The result of applying a filter optimised on ‘rings’ to each of four test images is shown in figure 2.6. For each test image, the result obtained using the filter optimised on image ‘rings’ is very similar to the best near-optimal result, whereas the results from cubic convolution are poorer.

### *Conclusions*

Edges are the dominant features in interpolation filter optimisation for these four significantly different images. Optimising a filter on a set of edges of many orientations and sub-pixel locations relative to the sampling grid gives a near-optimal interpolation filter that should be suitable for most images.

This result is significant, because the ‘rings’ image is deterministic. In the experiments so far, the near-optimal filter is determined empirically, with the outer summation of equation (2.17) applied over all of the pixels of the generator image. While straight-forward, such a step is time consuming. With a deterministic image, a closed form solution for this summation should be possible for a given offset. This in turn implies a closed form solution for the filter coefficients.

## 2.6. Model-based Near-optimal Interpolation

A simple analytical image model that is representative of the key image characteristic (many edges) can be developed; then the interpolation filter coefficients can be optimised on such a model. If the optimisation problem has a closed-form solution, the filter coefficients can be expressed simply as a function of the desired resampling offset. However, unlike conventional interpolation functions, a filter derived from such a model can be justified theoretically. This section investigates the possibility of creating such a model and deriving a closed-form solution to the least-squares problem using the model.

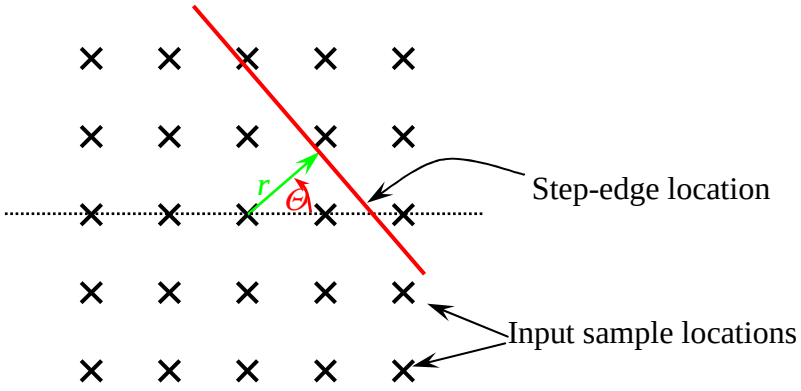


Figure 2.7 Location of a step-edge inside the resampling filter window.

Consider a model consisting of a single step-edge of height one and arbitrary orientation and position within the resampling filter's window. Such a model, denoted by $edge(x, y; r, \Theta)$, where $r$ is the perpendicular distance to the edge from the origin and $\Theta$ is the angle between the perpendicular and the horizontal axis is shown in figure 2.7.

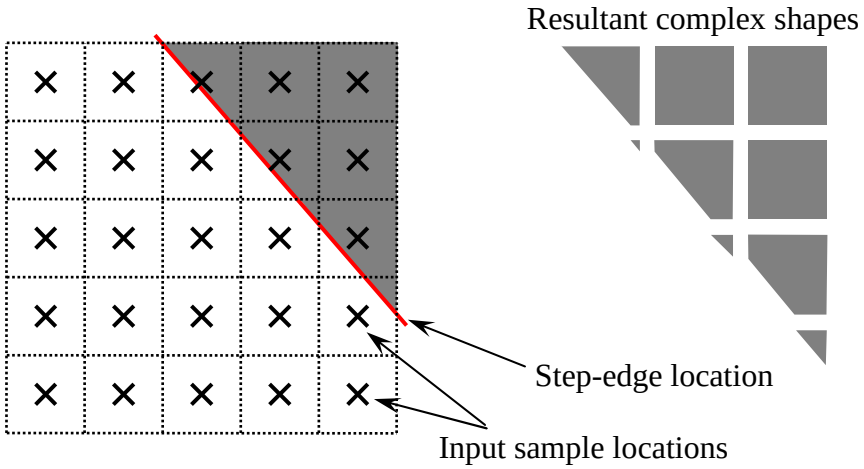


Figure 2.8 Area-sampling of a slanted step-edge.

Without any additional knowledge, $r$ and $\Theta$ can be considered to be random variables with uniform probability density functions. The filter weights can be optimised in a way that will minimise the square error on average, by considering the expected value of the input and output samples over all possible edge locations and orientations. This is equivalent to performing the minimisation on an image with a large number of edges of all possible orientations and positions relative to the sample grid.

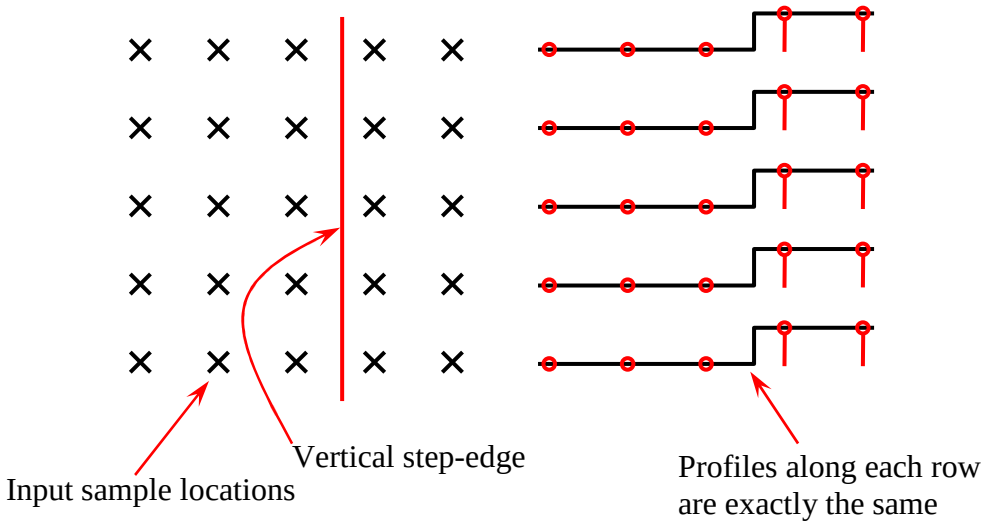


Figure 2.9 A vertical edge creates multiple linearly dependent constraints.

However, this minimisation is not straightforward. The image model must be area-sampled to produce the pixel values. Area-sampling of a slanted step-edge involves calculating the area of shapes similar to those shown in figure 2.8, when using rectangular pixels. Deriving a closed-form expression for the integration of such an area-sampled model over $r$ and $\Theta$ is non-trivial.

It has been shown in the past that vertical and horizontal edges have prevalence over other orientations in images of man-made objects, as well as some natural scenes [113, 114]. Therefore, one possible solution is to constrain the edge orientation to vertical and horizontal directions only. Using edges that are parallel to the rows and columns of the sampling grid significantly simplifies the derivation. However, this simplification results in a rank-deficient least-squares problem. A vertical step-edge runs perpendicular to the rows of the sampling grid creating multiple linearly-dependent constraints, as shown in figure 2.9. A similar situation occurs with a horizontal step-edge.

Another possible solution is to use a more complex shape, yet still based on horizontal and vertical step-edges. A 2D rectangular pulse model can be formed by multiplying a vertical and a horizontal 1D rectangular pulse functions of width and height one:

$$P_{1D}(x;p)=\begin{cases}1 & p-\tfrac{1}{2}<x\le p+\tfrac{1}{2}\\ 0 & \textit{otherwise}\end{cases} \tag{2.19}$$

$$P_{2D}(x,y;p_x,p_y)=P_{1D}(x;p_x)\,P_{1D}(y;p_y), \tag{2.20}$$

where $x=p_x$ and $y=p_y$ is the location of the centre of the pulse. The resulting model consists of two short horizontal and two vertical step-edges, with corners created at the intersections of these edges, as shown in figure 2.10. The reason for employing this particular shape is that a closed-form solution for the intersection of this pulse with a rectangular pixel is easy to determine.

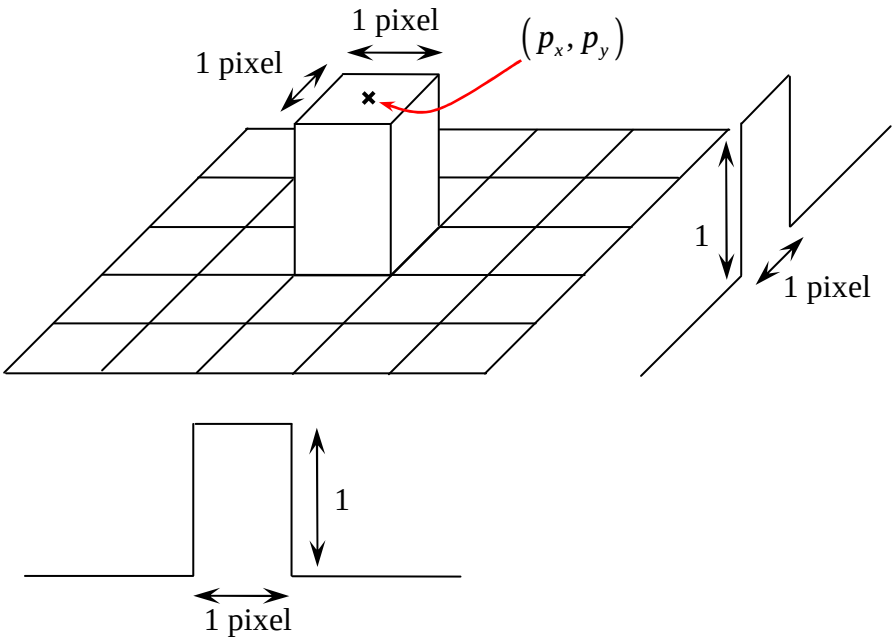


Figure 2.10 Rectangular pulse model.

The image model is area-sampled at the locations of input and output samples. Figure 2.11 illustrates the area-sampling of the rectangular pulse model (light shaded rectangle) centred on $(p_x, p_y)$ by a pixel centred on $(x, y)$. The value of this pixel is equal to the common area (dark shaded).

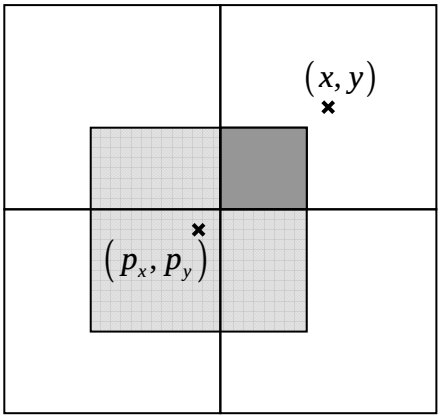


Figure 2.11 Area-sampling of the rectangular pulse model.

Area-sampling can be represented by a convolution of the model with a rectangular pulse of width and height equal to one (assuming square pixels with 100% fill factor) and point-sampling the result. This two dimensional convolution can be calculated in a separable fashion, as both the rectangular pulse model and the area-sampling function are separable in $x$ and $y$. The result of the convolution is the response of area-sampling the model centred at location $(p_x, p_y)$ by a pixel centred on $(x, y)$:

$$M(x, y; p_x, p_y) = M_{1D}(x; p_x) M_{1D}(y; p_y), \tag{2.21}$$

where

$$M_{1D}(x; p) = \begin{cases} 1 - p + x & \text{if } p - 1 < x \le p \\ 1 + p - x & \text{if } p < x \le p + 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.22}$$

Given a set of samples located at integer locations $(m, n)$ and spanning from $M^-$ to $M^+$ in the $x$ direction and from $N^-$ to $N^+$ in the $y$ direction and a set of filter coefficients $h[m, n]$ that weight these samples, it is desired to find the filter weights that minimise the square error:

$$^2 = \left( M(\ ,\ ; p_x, p_y) - \sum_{M^-}^{M^+} \sum_{N^-}^{N^+} M(m, n; p_x, p_y) h[m, n] \right)^2 \tag{2.23}$$

over all possible locations, $(p_x, p_y)$, of the rectangular pulse, where it has influence on any of the pixels in the filter window or the pixel at $(\ ,\ )$. The total error can be expressed by integrating $^2$ over all such values of $p_x$ and $p_y$:

$$E = \iint {}^{2} dp_x dp_y \,. \tag{2.24}$$

To shorten notation, the dependence of $M(\cdot)$ on $(p_x, p_y)$ will be made implicit, writing $M(x, y; p_x, p_y)$ as $M(x, y)$. The samples within the region of support of the filter are numbered from 0 to $R-1$ (where $R$ is the number of samples that fall within this region) in an arbitrary order, except that the sample at $(0,0)$ is given an index of zero. Then $M_i$ denotes $M(m, n; p_x, p_y)$ and $h_i$ denotes $h[m, n]$, where $i$ is the index that was allocated to the sample at $(m, n)$.

Rewriting expression (2.24) using this notation results in:

$$E = \iint \left( M(\ ,\ ) - \sum_{i=0}^{R-1} M_i h_i \right)^2 dp_x dp_y \,. \tag{2.25}$$

Expression (2.25) must be minimised with respect to the filter weights $h_i$ and subject to the flat-field response constraint of equation (2.18). It is possible to build this constraint directly into the minimisation by substituting

$$h_0 = 1 - \sum_{i=1}^{R-1} h_i \tag{2.26}$$

into (2.25). After some rearranging, this results in:

$$E = \iint \left( \bar{M}(\ ,\ ) - \sum_{i=1}^{R-1} \bar{M}_i h_i \right)^2 dp_x dp_y \,, \tag{2.27}$$

where $\bar{M}_i = M_i - M_0$ and $\bar{M}(\ ,\ ) = M(\ ,\ ) - M_0$. A direct closed-form solution to the minimisation of $E$ with respect to the filter weights given the resampling offsets and *is* complicated by the fact that the function $\bar{M}(x, y; p_x, p_y)$ is piecewise.

The filter weights may be determined by differentiating expression (2.27) with respect to each filter weight and equating the partial derivatives to zero. The order of differentiation and integration may be swapped, resulting in:

$$\frac{\partial E}{\partial h_j} = -2 \iint \left( \bar{M}(\ ,\ ) - \sum_{i=1}^{R-1} \bar{M}_i h_i \right) \bar{M}_j dp_x dp_y = 0 \,, \quad \forall j \in [1, R-1] \,. \tag{2.28}$$

Rearranging (2.28) results in the following system of equations:

$$\sum_{i=1}^{R-1} h_i \iint \bar{M}_i \bar{M}_j dp_x dp_y = \iint \bar{M}_j \bar{M}(\ ,\ ) dp_x dp_y, \quad \forall j \in [1, R-1]. \tag{2.29}$$

This system is best represented in matrix form:

$$\mathbf{Ah} = \mathbf{b}, \tag{2.30}$$

where $\mathbf{h} = [h_1 \quad h_i \quad h_{R-1}]^T$, $\mathbf{A}$ is an $(R-1)\times(R-1)$ matrix with element in column $i$ and row $j$ being of the form

$$A_{ji} = \iint \bar{M}_i \bar{M}_j dp_x dp_y, \tag{2.31}$$

and $\mathbf{b}$ is an $(R-1)\times 1$ vector with element in row $j$ being of the form

$$b_j = \iint \bar{M}_j \bar{M}(\ ,\ ) dp_x dp_y. \tag{2.32}$$

Each element of $\mathbf{A}$ and $\mathbf{b}$ is of similar form – a double integral of a product of $\bar{M}(x_1, y_1)$ and $\bar{M}(x_2, y_2)$. Let

$$\Psi(x_1, y_1, x_2, y_2) = \iint \bar{M}(x_1, y_1) \bar{M}(x_2, y_2) dp_x dp_y \tag{2.33}$$

Expanding $\bar{M}(x, y)$ in terms of $M(x, y)$ and $M(0,0)$:

$$\begin{aligned} \Psi(x_1, y_1, x_2, y_2) = & \iint M(x_1, y_1) M(x_2, y_2) dp_x dp_y \\ & - \iint M(x_1, y_1) M(0,0) dp_x dp_y \\ & - \iint M(x_2, y_2) M(0,0) dp_x dp_y \\ & + \iint M(0,0)^2 dp_x dp_y. \end{aligned} \tag{2.34}$$

Taking into account that $M(x, y)$ is a separable function, this expression can be simplified further:

$$\begin{aligned} \Psi(x_1, y_1, x_2, y_2) = & \Omega_x(x_1, x_2) \Omega_y(y_1, y_2) \\ & - \Omega_x(x_1, 0) \Omega_y(y_1, 0) \\ & - \Omega_x(0, x_2) \Omega_y(0, y_2) \\ & + \Omega_x(0,0) \Omega_y(0,0), \end{aligned} \tag{2.35}$$

where

$$\Omega(x_1, x_2) = \int M_{1D}(x_1; p) M_{1D}(x_2; p) dp, \tag{2.36}$$

and the subscript of $\Omega$ in (2.35) indicates whether the integration is over $p_x$ or $p_y$. Since the functions are piecewise, several cases need to be considered depending on the overlap between the functions $M_{1D}(x_1; p)$ and $M_{1D}(x_2; p)$. The overlap depends on the difference between $x_1$ and $x_2$, but because $M_{1D}(x; p)$ is symmetric, only three cases need to be considered. These three cases are demonstrated graphically in Figure 2.12.

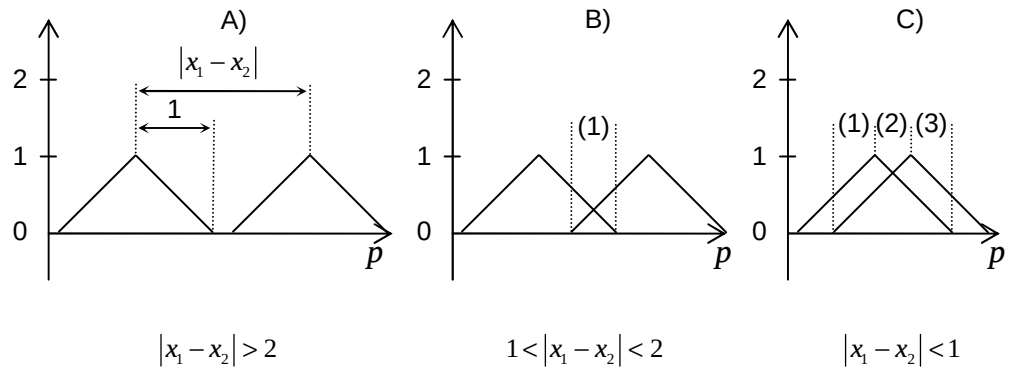


Figure 2.12 Three cases of equation (2.36). Refer to text for explanation.

In case A there is no overlap between the non-zero parts of $M_{1D}(x_1; p)$ and $M_{1D}(x_2; p)$; hence, the product is zero. In case B there is an overlap between the non-zero parts but only one piece (labelled (1)) of each function is overlapped; hence, the integral is calculated over this single piece. In case C the overlap is more complex (pieces labelled as (1), (2) and (3)) and the overall integral is a sum of the integrals of three pieces as outlined in figure 2.12. Calculating the integrals themselves is trivial and the resulting function is:

$$\Omega(x_1, x_2) = \begin{cases} 0 & \text{if } |x_1 - x_2| > 2 \\ \frac{1}{6}\left(2 - |x_1 - x_2|\right)^3 & \text{if } 1 < |x_1 - x_2| < 2 \\ \frac{1}{2}|x_1 - x_2|^3 - |x_1 - x_2|^2 + \frac{2}{3} & \text{if } |x_1 - x_2| < 1 \end{cases} \tag{2.37}$$

Now it is possible to calculate the elements of **A** and **b** by substituting the appropriate values for $x_1$, $x_2$, $y_1$ and $y_2$ into $\Psi(x_1, y_1, x_2, y_2)$. The solution to the linear system described by (2.30) can be computed by inverting the square matrix **A** and multiplying the result by **b**:

$$\mathbf{h} = \mathbf{A}^{-1}\mathbf{b}. \tag{2.38}$$

This results in a closed-form solution to the optimisation problem. Observe that from equation (2.31), the elements of $\mathbf{A}$ do not depend on the resampling offset. This allows multiple resampling offsets to be determined with the calculation of a single matrix inverse. One can find the filter coefficients simply by substituting the desired resampling offset into $\mathbf{b}$ and evaluating expression (2.38). This simple image model satisfies the desire to optimise filter coefficients on vertical and horizontal edges at all possible sub-pixel locations inside the filter window.

### 2.6.1. Experiment IV – Performance assessment of rectangular pulse model

***Aim***

To compare the performance of near-optimal interpolation using the proposed rectangular pulse model with a number of other interpolation methods popular in the literature.

***Method***

Model-based kernels were derived for 4x4 (Model4) and 6x6 (Model6) windows with an offset of half a pixel in each direction. The filters were applied to the four images, shown in figure 2.2, which were downsampled as described in the earlier experiments, using area-sampling model with a downsampling factor of four. The following popular interpolation methods are used for comparison: linear interpolation has a region of support of 2x2 pixels; Key's cubic convolution and the Lanczos2 methods both have region of support of 4x4 pixels; the Lanczos3 method has a region of support of six pixels; and finally the cubic spline has infinite support. The root mean square error was measured between the interpolated output and ground truth.

### *Results*

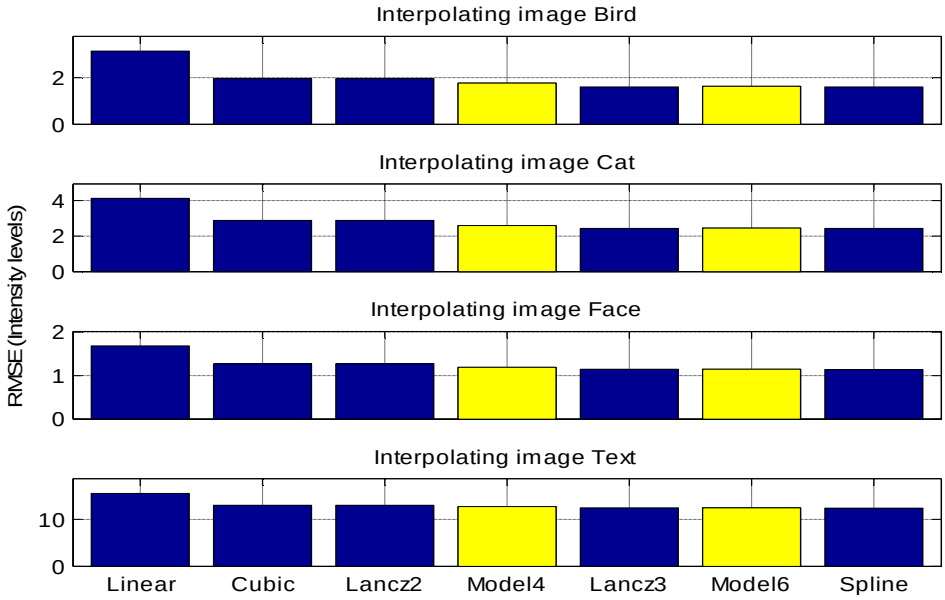


Figure 2.13 Comparison of near-optimal using rectangular pulse model (highlighted) to other popular methods.

The results are shown in figure 2.13 in a similar fashion to the previous experiments. Each column contains results from interpolating each image using the different methods. It can be clearly seen that model-based near-optimal interpolation outperforms both Keys' cubic convolution and the Lanczos2 method when used with a region of support of the same size as those methods. There is little difference in the performance between the six-by-six model-based filter, Lanczos3 filter and the cubic spline.

### *Conclusions*

An edge-based rectangular pulse model performs satisfactorily, as the results are comparable to, or better than, well-established methods.

## *2.7. Summary and Conclusions*

Conventional linear interpolation methods generate the interpolated surface through a convolution of a predefined basis function with the discrete data. This function, however, is usually chosen without image interpolation specifically in mind. A set of generic one-dimensional interpolation kernels has become popular for the quality of the result, although without any theoretical backing in terms of image interpolation.

A novel method for image interpolation and resampling has been presented in this chapter. It is based on the concept of a least-squares optimal interpolator that estimates the target image from the input with minimum square error using a globally-optimised space-invariant kernel. Unlike the traditional methods, the interpolation kernel is derived from the image data itself.

It is shown experimentally that the performance of optimal interpolation has asymptotic behaviour – beyond a certain filter window size, the improvement in square error becomes insignificant in comparison to the increase in required computational complexity. A small interpolation kernel can achieve almost as good a result as a global method.

The fact that computing the ‘optimal’ output requires the ideal output makes this method impractical for interpolation; however, the concept is valuable because it offers a flexible framework for deriving new kernels, based on various image models, making interpolation image specific.

For a model that is representative of a class of images, the corresponding interpolator becomes near-optimal for that class. One such model, based on a generic image feature, a step-edge, is introduced and shown experimentally to result in a kernel that performs comparable to or better than traditional methods on images with a wide range of statistics.

# Chapter 3

## Sub-pixel Image Registration

The first step of image super-resolution is to find the relative transformations between the input low-resolution images. If the low-resolution images are taken at very nearly the same time it can be assumed that there is no local motion within the scene. It can also be assumed that there is no difference in lighting conditions. Thus we are only trying to recover a translation between the images. The challenge for the registration is that it needs to be accurate to sub-pixel precision.

Registering images to the nearest pixel is not overly challenging and can be done using a large variety of methods [115, 116]. It will be assumed that the images are already registered to the nearest pixel and only sub-pixel registration is required. Only pairwise registration is considered here. One image is designated to be the reference and other images (targets) are each registered to it in a pairwise manner.

There are generally two approaches to performing sub-pixel registration. The first group of methods use features or landmarks, that is, points in the image that are unique within the images and easy to identify [116, 117]. Corresponding features in the target and reference images are detected automatically or matched by a human user. Then, the registration parameters are computed from these correspondences. Finding these pairs of matching features is sometimes known as the ‘correspondence problem’ [118].

Feature-based registration methods rely on some predetermined form or structure within the images. If the image consists predominantly of features smaller than a pixel, then feature-based methods cannot be used to determine the location of these features reliably. Since these methods are usually based on objects and features that must be able to be completely resolved in the low-resolution images they are less appropriate for super-resolution. For this reason, these approaches have not been investigated.

An alternative approach is to use so-called area- or intensity-based registration. These methods use pixel intensities from the whole image to perform registration. The procedure is set as an optimisation problem, where some metric between the

transformed reference image and the target image is minimised with respect to the registration parameters. The cost function is generally based on the $L^2$ norm for the reasons already discussed in section 2.3. However, even though the transformation function is a simple translation, the optimisation is non-linear, simply because images are generally non-linear functions of $x$ and $y$. This non-linear least-squares problem must be solved iteratively.

There are numerous optimisation procedures [119], such as Gauss-Newton or Levenberg-Marquardt, that may be employed to solve this problem. However, they are iterative and can be computationally expensive. The focus of this chapter is to investigate how optimal interpolation, as described in chapter 2, may be employed to perform image registration without the use of costly iterative optimisations.

## 3.1. Image Registration using Optimal Interpolation

Conventional sub-pixel image registration techniques create a continuous surface from the discrete samples of the reference image [120]. Then this surface is offset by some sub-pixel amount and resampled to be compared with the target image. The sub-pixel offset between the input image and the target is determined by searching for the offset that gives the best fit with the target. Hence, to gain any sub-pixel accuracy, some form of interpolation is required.

Bilinear interpolation can be employed here to achieve sub-pixel accuracy. The benefit of bilinear interpolation is that the resultant minimisation problem is linear in the registration parameters. A closed-form solution is possible in this case. However, this method may lack the required accuracy simply because linear interpolation is not very accurate, in general.

Other forms of interpolation can be employed, for example cubic spline [95] or Keys' cubic convolution [96] or any of the other methods described in chapter 2. The use of these kernels results in improved interpolation and generally leads to improved registration accuracy. These methods, however, result in a non-linear objective function that requires an iterative solution.

Alternatively, optimal interpolation can be employed to perform the resampling step, since, both the input and the output are known. Instead of using a predetermined interpolation function, optimal interpolation uses the image data itself to determine the interpolation kernel. Pixel values in the reference image are used to predict pixel

values in the target image in such a way as to minimise the sum of square errors, effectively determining the 'best' (that is, optimal in a least-squares sense) interpolation kernel that relates the target to the reference.

Consider a discrete reference image $f[k,l] = f(kT, lT)$ and a target image $g[k,l] = f(kT + u, lT + v)$, shifted by $u$ and $v$ in $x$ and $y$ directions respectively. In a similar manner to equation (2.17), the optimisation problem that determines the resampling filter coefficients $h[m,n]$ may be defined as

$$h = \arg\min_{h} \sum_{k,l} \left( g[k,l] - \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} f[k+m, l+n] h[m,n] \right)^2, \tag{3.1}$$

where the filter window spans from $M^-$ to $M^+$ in $x$ direction and from $N^-$ to $N^+$ in $y$ direction. Equation (3.1) must be minimised subject to

$$\sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} h[m,n] = 1 \tag{3.2}$$

to satisfy the flat-field response constraint described by equation (2.3). Let us define $\Theta$ as the filter's region of support, equal to the number of degrees of freedom (number of coefficients minus one because of the above constraint). If $R$ is the number of coefficients within the filter window:

$$R = \left(M^+ - M^-\right)\left(N^+ - N^-\right), \tag{3.3}$$

then $\Theta = R - 1$.

To simplify the notation, each of $R$ coefficients of the filter is allocated an index $i$, such that $h_i$ denotes $h[m,n]$, where $i$ is the index that was allocated to the coefficient at $(m,n)$ and $f_i[k,l] = f[k+m, l+n]$. The coefficient at position $(0,0)$ is given an index of zero and the remaining coefficients are numbered from 1 to $\Theta$ in an arbitrary order. The least-squares objective function expressed by equation (3.1) may be rewritten using this shortened notation:

$$E = \sum_{k,l} \left( g[k,l] - \sum_{i=0}^{\Theta} f_i[k,l] h_i \right)^2. \tag{3.4}$$

Similar to the previous chapter, the flat-field response constraint can be built into the minimisation by substituting

$$h_0 = 1 - \sum_{i=1}^{\Theta} h_i \tag{3.5}$$

into (3.4). After some rearranging, this results in

$$E = \sum_{k,l} \left( \bar{g}[k,l] - \sum_{i=1}^{\Theta} \bar{f}_i[k,l] h_i \right)^2 , \tag{3.6}$$

where $\bar{f}_i[k,l] = f_i[k,l] - f_0[k,l]$ and $\bar{g}[k,l] = g[k,l] - f_0[k,l]$.

The filter weights may be determined by differentiating expression (3.6) with respect to each filter weight and equating the partial derivatives to zero. However, first, the objective function is rewritten in matrix form for simplicity of notation:

$$E = \left\| \bar{\mathbf{g}} - \mathbf{F}\mathbf{h} \right\|_2^2 , \tag{3.7}$$

where $\bar{\mathbf{g}}$ is the image $\bar{g}[k,l]$ arranged lexicographically into a column vector, $\mathbf{h}$ is a column vector containing the filter coefficients $h_1$ to $h_\Theta$ and $\mathbf{F}$ is a matrix made up of column vectors $\bar{\mathbf{f}}_1$ to $\bar{\mathbf{f}}_\Theta$:

$$\mathbf{F} = \left[ \bar{\mathbf{f}}_1 \quad \cdots \quad \bar{\mathbf{f}}_\Theta \right], \tag{3.8}$$

where $\bar{\mathbf{f}}_i$ is the image $\bar{f}_i[k,l]$ arranged lexicographically into a column vector.

The least-squares solution to minimisation of $E$ with respect to the filter coefficients is straightforward and may be written as

$$\mathbf{h} = \left( \mathbf{F}^T \mathbf{F} \right)^{-1} \mathbf{F}^T \bar{\mathbf{g}} . \tag{3.9}$$

Once the least-squares optimal filter coefficients for the resampling filter $h[m,n]$ have been determined, the relative offset of the two images must be “extracted” from these. Here, it is assumed that the offset in each direction is a linear combination of the filter coefficients:

$$\hat{u} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} w_u[m,n] h[m,n] , \tag{3.10}$$

$$\hat{v} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} w_v[m,n] h[m,n], \tag{3.11}$$

where $w_u[m,n]$ and $w_v[m,n]$ are linear weights that relate the filter coefficients to the horizontal and vertical offset estimates. To solve for these weights, consider a situation where the offset between the reference and the target images is equal to an integer number of pixels. For any such offset $(u,v)=(m,n)$, $m,n \in \mathbb{Z}$, $g[k,l]$ will be equal to $f[k+m,l+n]$. The optimal interpolation procedure will derive the filter coefficient corresponding to this integer shift to be one and all of the remaining coefficients to be zero, i.e.

$$\begin{cases} h[k,l]=1 & \text{if } k=m, l=n \\ h[k,l]=0 & \text{otherwise} \end{cases} \tag{3.12}$$

This must be true for every integer offset not larger than the filter's region of support (i.e. $M^- \le m \le M^+$, $N^- \le n \le N^+$). Substituting these constraints (one constraint for each integer offset inside the region of support) into equations (3.10) and (3.11) results in two linear systems of equations. Solving these systems, yields $w_u[m,n]=m$ and $w_v[m,n]=n$, resulting in:

$$\hat{u} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} m h[m,n], \tag{3.13}$$

$$\hat{v} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} n h[m,n]. \tag{3.14}$$

Thus, there is a direct closed-form solution to the pairwise registration problem using optimal interpolation. First, the optimal resampling filter coefficients are determined using linear least-squares (assuming $\mathbf{F}^T\mathbf{F}$ is invertible and it should be if there is no 'aperture problem' [121]). Then, expressions (3.13) and (3.14) are used to determine the relative offsets from the filter coefficients. The matrix $\mathbf{F}^T\mathbf{F}$ that must be inverted in (3.9) is only $\Theta$ by $\Theta$; hence, most of the computational complexity in this approach lies in two matrix multiplications $\mathbf{F}^T\mathbf{F}$ and $\mathbf{F}^T\mathbf{g}$. The advantage of this method for image super-resolution, where a number of target images must be registered to a single reference image, lies in the fact that $\left(\mathbf{F}^T\mathbf{F}\right)^{-1}$ depends only on

the reference image and must be computed only once for any number of target images.

The performance of this method is assessed in the next section by experimentally comparing it to common registration methods.

## 3.2. Performance Evaluation

Performance evaluation of sub-pixel image registration is challenging without the knowledge of the ground truth registration parameters. In the case where test images are obtained by simulation and ground truth is available, the error can be computed between the known offsets and estimated parameters. In the case of complex models, such as full nine-point homography [122], it is hard to derive a single error metric from the parameter vectors because these vectors have different units in different dimensions. However, in the case of translational motion, it makes sense to simply compute the Euclidean distance between an estimate ($\hat{\mathbf{u}}$) and the ground truth ($\mathbf{u}$) translation vectors as a measure of quality of registration. This statistic offers a single scalar measure of error in estimating a two-dimensional translation vector.

From an estimation theory point of view, an image registration algorithm may be considered an estimator. Information on the position of the images is not available directly; it must be extracted from the pixel values. A model must be assumed that relates the pixel values to the position of the images. The registration parameters are therefore the parameters of this model that must be estimated. The word estimate implies inherent uncertainty; this uncertainty is a result of the fact that the image pixel values are possibly contaminated by noise. The model itself may not exactly reflect the underlying relationship. Thus, the job of this estimator is to extract model parameters from noisy image data that may not necessarily conform to the model.

The average performance of such an estimator may be assessed by exploring the expected value of the square error $\left\langle e^2 \right\rangle$, where $\langle \cdot \rangle$ denotes the expectation operator. This is commonly expressed in terms of the bias and variance of the estimator:

$$\begin{aligned} \left\langle e^2 \right\rangle &= \left\langle \left\| \hat{\mathbf{u}} - \mathbf{u} + \langle \hat{\mathbf{u}} \rangle - \langle \hat{\mathbf{u}} \rangle \right\|_2^2 \right\rangle \\ &= \left\langle \left\| \hat{\mathbf{u}} - \langle \hat{\mathbf{u}} \rangle \right\|_2^2 \right\rangle + \left\| \langle \hat{\mathbf{u}} \rangle - \mathbf{u} \right\|_2^2 \\ &= \mathrm{var}\left(\hat{\mathbf{u}}\right) + \mathrm{bias}\left(\hat{\mathbf{u}}\right)^2 . \end{aligned} \tag{3.15}$$

This separation of the error into the random (variance) and systematic (bias) components allows for a clear distinction between any variation caused by randomness and variation cause by model mismatch.

An experimental comparison of the proposed registration method is made to some popular registration methods that appear in the literature. Bias and variance are estimated using sample mean and sample variance respectively.

### 3.2.1. Description of methods for comparison

This section reviews a selection of common registration methods that are used to benchmark the proposed method.

#### (i) Gradients-based methods

The idea behind gradient descent is to iteratively solve the minimisation, approximating the objective function at each iteration in some simpler way, for example by a linear (steepest descent method) or a quadratic (Newton's method) function [119]. The gradient of this approximation can be computed and used to point in the direction of a minimum. A step is taken in that direction and the procedure is repeated again.

Using gradient descent for image registration was first suggested by Lucas and Kanade [123]. The reference image is denoted by $f(\mathbf{x})$ and the template (target) image is denoted by $g(\mathbf{x})$. The goal of their method is to align the input image and the template by minimising the sum of squared differences (SSD) between the template image and a shifted version of the input image, with respect to the shift parameter $\mathbf{p}$:

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \left[ f(\mathbf{x}+\mathbf{p}) - g(\mathbf{x}) \right]^2 . \tag{3.16}$$

The sum is computed over all of the pixels in the template. This is a non-linear least-squares problem, as the images are generally non-linear in $\mathbf{x}$. The Lucas and Kanade algorithm splits the parameters into a current estimate, $\mathbf{p}$, which is assumed to be known, and an additive update $\Delta\mathbf{p}$ to be solved for in each iteration:

$$\sum_{\mathbf{x}} \left[ f(\mathbf{x}+\mathbf{p}+\Delta\mathbf{p}) - g(\mathbf{x}) \right]^2 . \tag{3.17}$$

This expression is solved with respect to the parameters' update $\Delta\mathbf{p}$ and then the parameters are updated:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} . \tag{3.18}$$

These two steps are iterated until the estimates of the parameters converge (that is, some norm of $\Delta\mathbf{p}$ drops below a predetermined threshold level).

The expression $\sum_{\mathbf{x}} \left[ f(\mathbf{x}+\mathbf{p}+\Delta\mathbf{p}) - g(\mathbf{x}) \right]^2$ that the Lucas and Kanade algorithm tries to minimise is still non-linear in $\Delta\mathbf{p}$. To solve it, the expression is linearised through a first order Taylor series expansion of the shifted input image $f(\mathbf{x}+\mathbf{p}+\Delta\mathbf{p})$ around the point of current estimate of the shift $\mathbf{x}+\mathbf{p}$:

$$\sum_{\mathbf{x}} \left[ f(\mathbf{x}+\mathbf{p}) + \nabla f \Delta\mathbf{p} - g(\mathbf{x}) \right]^2 . \tag{3.19}$$

In this expression, $\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$ is the gradient of the input image evaluated at $\mathbf{x}+\mathbf{p}$. In other words, it is the gradient of the input image at point $\mathbf{x}$, shifted to the point $\mathbf{x}+\mathbf{p}$ through interpolation.

This is a linear least-squares problem. A closed-form solution can be achieved by differentiating (3.19) with respect to $\Delta\mathbf{p}$, equating the partial derivatives to zero and rearranging:

$$\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} [\nabla f]^T \left[ g(\mathbf{x}) - f(\mathbf{x}+\mathbf{p}) \right], \tag{3.20}$$

where $H$ is an approximation to the 2×2 Hessian matrix:

$$H = [\nabla f]^T [\nabla f] . \tag{3.21}$$

The Lucas and Kanade algorithm consists of starting with some initial estimate for the shift and iteratively applying equations (3.21) and (3.18). Ideally, the algorithm will converge after a certain number of iterations.

From the description of the above algorithm, it can be seen that it is a Gauss-Newton type algorithm, often used in mathematics to solve non-linear least-squares problems. Other gradient descent non-linear optimisation algorithms, such as Newton,

steepest descent and Levenberg-Marquardt have also been applied to the image registration problem [120].

### (ii) Cross-correlation-based methods

There are several variations of correlation, but the basic method is to multiply one image with an offset second image on a pixel by pixel basis [124]. The product is then accumulated to give the correlation for the particular offset. In this manner, the correlation $c[i, j]$, $i, j \in \mathbb{Z}$ is given by:

$$c[i, j] = \sum_k \sum_l f[k+i, l+j]\, g[k, l] = \sum_k \sum_l f_{i,j}\, g_{0,0} \,. \tag{3.22}$$

The problem with simple correlation is that images are finite. Therefore, if the image is darker on one side than the other, this can introduce a bias that offsets the correlation peak. This may be overcome by normalising the correlation by the overlap area and average pixel value:

$$c_N(i, j) = \frac{\sum_k \sum_l f[k+i, l+j]\, g[k, l]}{\sqrt{\sum_k \sum_l f^2[k+i, l+j]} \sqrt{\sum_k \sum_l g^2[k, l]}} \,. \tag{3.23}$$

This is repeated for a range of different offsets $(i, j)$, with the offset giving the maximum correlation corresponding to the best match. As both $f$ and $g$ are sampled functions, this will only find the offset to the nearest pixel. To register with sub-pixel accuracy, an interpolation surface must be fitted to the correlation values, and the interpolated maximum found [48]. However, to fit a surface to the sampled data requires knowing the shape of the correlation peak. For an image that is piecewise constant, the expected shape of the peak is a pyramid, with a width of twice the size of the smallest features in the image. For large objects without a lot of fine detail, the pyramid can easily be detected and gives an accurate registration. However, since this application is to improve the resolution of the images, significant fine detail is to be expected, and the registration will be less accurate. This implies that only local information should be used to estimate the peak location, so only the maximum pixel, and the pixel on either side horizontally and vertically are used.

As the images are assumed to be pre-registered to the nearest pixel, the maximum of the correlation will occur at (0,0). The interpolation process is illustrated for the 1-

dimensional case in figure 3.1. Making use of symmetry, the two pixels on one side of the peak are sufficient to define a slope, which may be mirrored on the other side. Finding the intersection of these lines gives the sub-pixel location of the peak in one direction:

$$x_0 = \frac{c(1,0) - c(-1,0)}{2\big(c(0,0) - \min\big(c(1,0), c(-1,0)\big)\big)}. \tag{3.24}$$

A similar equation may be used to obtain $y_0$.

If the pyramid is orientation independent (conical), any *x* or *y* offset will mean that the samples are not directly through the cone axis, and so will be more rounded on the top. Also, the actual peak will be rounded on the top as a result of blur introduced by area-sampling, and by the lens. Therefore, an alternative approach is to fit a parabola to the three sample points in the horizontal and vertical directions, and find the maxima of the parabola. In the *x* direction, the parabola maxima is given by

$$x_0 = \frac{c(1,0) - c(-1,0)}{4c(0,0) - 2\big(c(1,0) + c(-1,0)\big)}, \tag{3.25}$$

and similarly for the *y* direction.

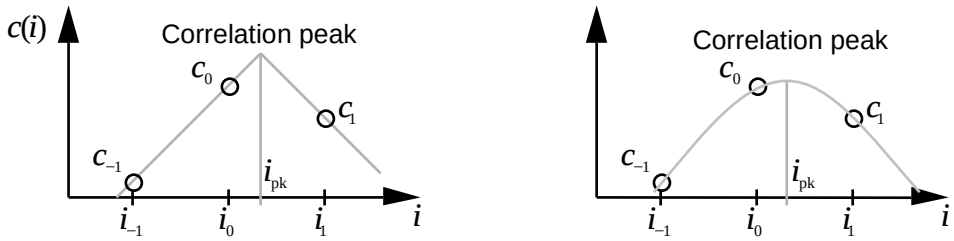


Figure 3.1 Locating the correlation peak in 1D. (a) Pyramid fit. (b) Parabola fit.

A sharper correlation peak may be obtained by subtracting the mean from the images before correlating. This gives the correlation coefficient [115]:

$$c_C(i,j) = \frac{C(f,g)}{\sigma_f \sigma_g} = \frac{\sum_k \sum_l \big(f[k+i,l+j] - \mu_f\big)\big(g[k,l] - \mu_g\big)}{\sqrt{\sum_k \sum_l \big(f[k+i,l+j] - \mu_f\big)^2}\sqrt{\sum_k \sum_l \big(g[k,l] - \mu_g\big)^2}}, \tag{3.26}$$

where $C(f,g)$ is the covariance, and $\mu$ and $\sigma$ are the mean and standard deviation of the pixel values in the respective images.

### (iii) Fourier phase

Phase methods of image registration work on the principle that an offset in the image domain gives a linear phase response in the frequency domain [125]. Let $F(\Omega,\Theta)$ and $G(\Omega,\Theta)$ be the discrete Fourier transforms of $f[k,l]$ and $g[k,l]$. Then, from the shift property of Fourier transform:

$$g[k,l] = f(kT-u, lT-v) \overset{DFT}{\Leftrightarrow} G(\Omega,\Theta) = e^{j(u\Omega+v\Theta)} F(\Omega,\Theta). \tag{3.27}$$

Therefore, by taking the Fourier transform of the two images, a plane may be fitted to the phase difference of the two images, enabling the sub-pixel offset to be calculated from the tilt:

$$F_{\phi}(\Omega,\Theta) - G_{\phi}(\Omega,\Theta) = u\,\Omega + v\,\Theta. \tag{3.28}$$

In practice, the images need to be windowed before Fourier transformation to prevent any discontinuity at the edge of the image from dominating the phase response (since the FFT assumes that the image is periodic) [85]. The phase difference also needs to be "unwrapped" by the addition of $\pm 2n$ at each point to make the phase as planar as possible. A least-squares fit of a plane to the unwrapped phase difference image gives the offset. Frequencies where the magnitude is low are more affected by noise, with the phase effectively random at some frequencies. Therefore, a better fit may be obtained if the individual phase points are weighted by the corresponding magnitude when performing the fit. When processing under-sampled images, there is also a concern with the aliased components distorting the phase shift. The effects of these may be reduced by fitting only the central part of the image, near the origin. This helps guard against aliasing in two ways. First, in most images, the low frequencies dominate, so the relatively large magnitudes compared to the aliased components means they will be less distorted. Second, the area-sampling blur has nulls at integer multiples of the sample frequency (for 100% fill factor). After aliasing, this corresponds to the DC component, so the low frequencies will be less affected by aliasing.

An alternative to equation (3.28) is to divide one image by the other in the frequency domain:

$$\frac{F(\Omega,\Theta)}{G(\Omega,\Theta)} = \frac{FG^*}{GG^*}. \tag{3.29}$$

The phase of this is the same as equation (3.28), however the magnitude should be equal to one. Points where the magnitude of equation (3.29) differs significantly from one are therefore corrupted either by noise or by aliasing. Identifying such points as outliers and eliminating them from the fit is another approach that has been proposed for using the phase difference in the presence of aliasing [126]. In a later paper [127], the same authors propose a different method for identifying and eliminating frequencies corrupted by noise and aliasing. They propose to eliminate the frequencies whose magnitudes are small relative to the rest of frequencies for both images. This is accomplished by setting a threshold and eliminating the frequencies whose magnitudes are smaller than the threshold from the least-squares.

### 3.2.2. Experiment V – Registration accuracy evaluation

#### *Aim*

To evaluate registration accuracy of optimal interpolation based registration in comparison to a selection of common methods from the literature, described in section 3.2.1.

#### *Method*

To assess the registration accuracy, it is necessary to work with a set of images where the offsets are known in advance. Capturing a sequence of images with precisely known offsets is difficult, if not impossible. Therefore, a high-resolution image was used as the image source, and a simple imaging model was used to simulate the capture of low-resolution sample images.

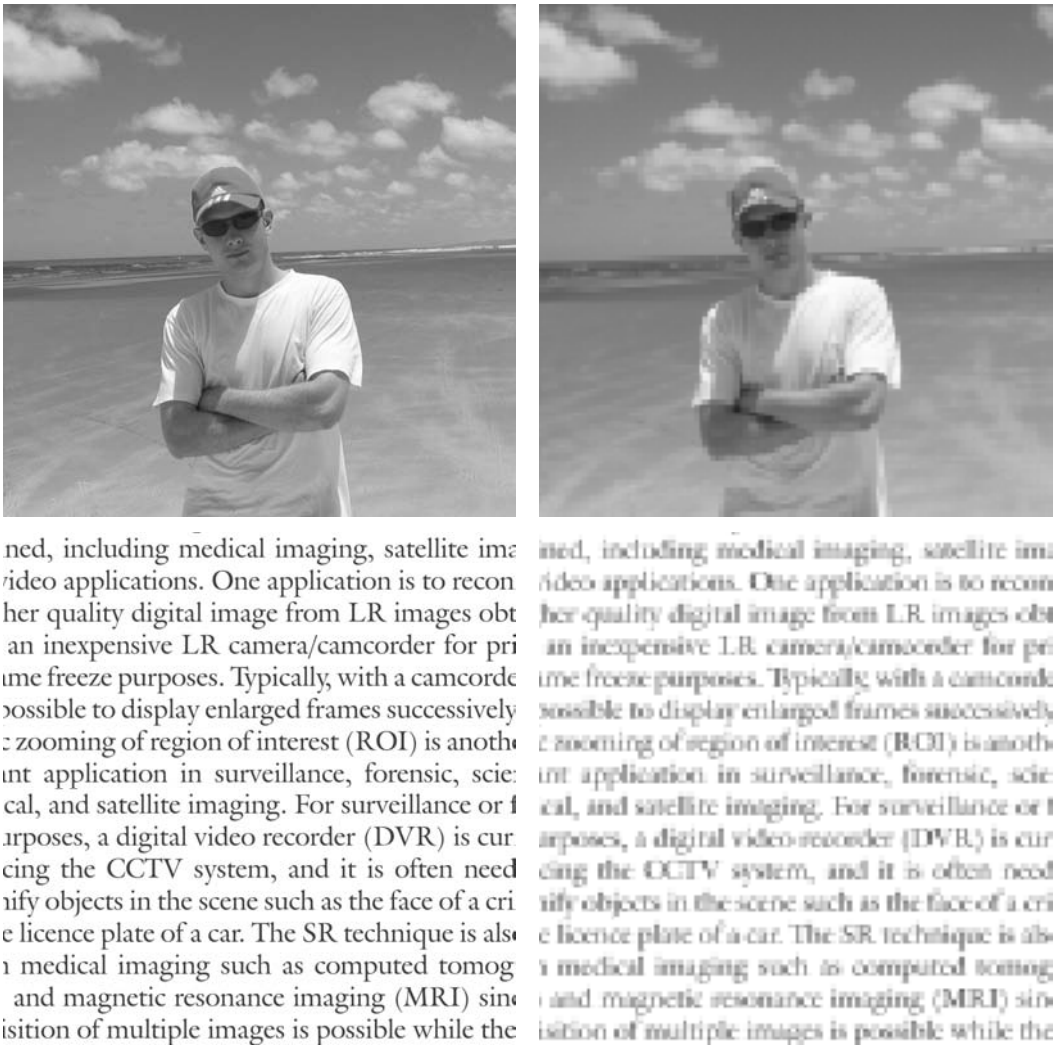

Figure 3.2 Images ‘beach’ and ‘text2’. Original high-resolution image on the left and a low-resolution image on the right.

Two test images were chosen for this experiment. These are shown in figure 3.2. The images were chosen for the following characteristics: the ‘beach’ scene is a typical image that is dominated by low frequencies. There are some sharp edges that will result in a limited degree of aliasing. The ‘text2’ image contains a lot of high frequency information, with a significant amount of aliasing. The source images were 1300x1300 pixels. These were filtered using a 10×10 box average to simulate area-sampling, and then downsampled by a factor of 10 in each dimension to give 128×128 pixels low-resolution images (low-resolution images were cropped to avoid edge effects from the filtering operation).

Downsampling by a factor of 10 allows image pairs to be generated with sub-pixel offsets of multiples of 0.1 of a pixel. Pairs of images with offsets between 0 and 0.9 pixels in each direction were created, resulting in 100 image pairs. To measure the

effect of noise level on accuracy, white Gaussian noise was added to the images prior to registration. Each pair of images was registered 100 times with different noise instances. The true offset was subtracted from the result and the bias and variance were estimated, using the sample mean and sample variance. This was repeated using different noise levels. Noise with standard deviations of 0, 0.64, 1.28, 1.92, 2.56, 3.2, 3.84, 5.12, 6.4, 7.68, 10.2, 12.8, 19.2 and 25.6 grayscale levels was employed in this experiment. The effect of noise contamination is demonstrated in figure 3.3.

$= 0$ $= 2.56$ $= 6.4$ $= 12.8$ $= 25.6$

Figure 3.3 Effect of noise contamination of images ‘beach’ and ‘text2’.

It is possible to generate more than one pair of images with the desired relative offset. For example, a pair of images with offsets (0,0) and (0.2,0.2) and a pair of images with offsets (0.5,0.5) and (0.7,0.7), both have a relative offset (0.2,0.2). The results of registering the first pair of images and the second pair of images may deviate from the true offset by different amounts even in the absence of noise. This is because the images carry inherent differences due to aliasing; hence, aliasing is another major contributor to registration error.

The second part of this experiment investigates the effect of aliasing on registration precision. One hundred image pairs with the same relative offset are created by shifting both the reference and the target by the same sub-pixel amount in the range between 0 and 0.9 in steps of 0.1 in both directions. These image pairs are registered, and the true offset is subtracted from the results. Bias and variance are estimated from the error distribution.

To gauge the average performance of sub-pixel registration, RMS bias and variance were computed over all 100 different sub-pixel offsets.

### *Results*

The results are subdivided into five sections. First, the variants in each class of methods (gradients, cross-correlation, Fourier phase and optimal interpolation) are compared within the class. Second, the best method from each class is compared with using optimal interpolation. RMS bias and standard deviation (standard deviation (SD) is used to keep the units of pixels) are plotted against noise level for each method.

#### (i) Gradients-based methods

The Lucas-Kanade (LK) method, as implemented by Baker and Matthews [120], was used in this experiment. Full implementation details are available in their paper and the source code is available from their project website [128].

The results of using this method on each test image are shown in figure 3.4. The registration accuracy is diminished as noise contamination becomes more severe. It can be observed that bias dominates over variance. Successive iterations significantly improve bias at higher noise levels on image ‘beach’. For low-noise, however, additional iterations after the third one seem to have little effect on results obtained from both images. The number of iterations before the bias stops reducing must depend on both the noise level and the image content.

Successive iterations have a contrary effect on variance – it grows, especially at higher levels of noise. This is particularly noticeable on image ‘beach’, where successive iterations reduce the bias and increase the variance at high noise levels to the point where variance becomes dominant source of error.

Hence, there must exist an optimal number of iterations with performance decreasing with further iterations. For low-noise, additional iterations after the third one seem to have little effect.

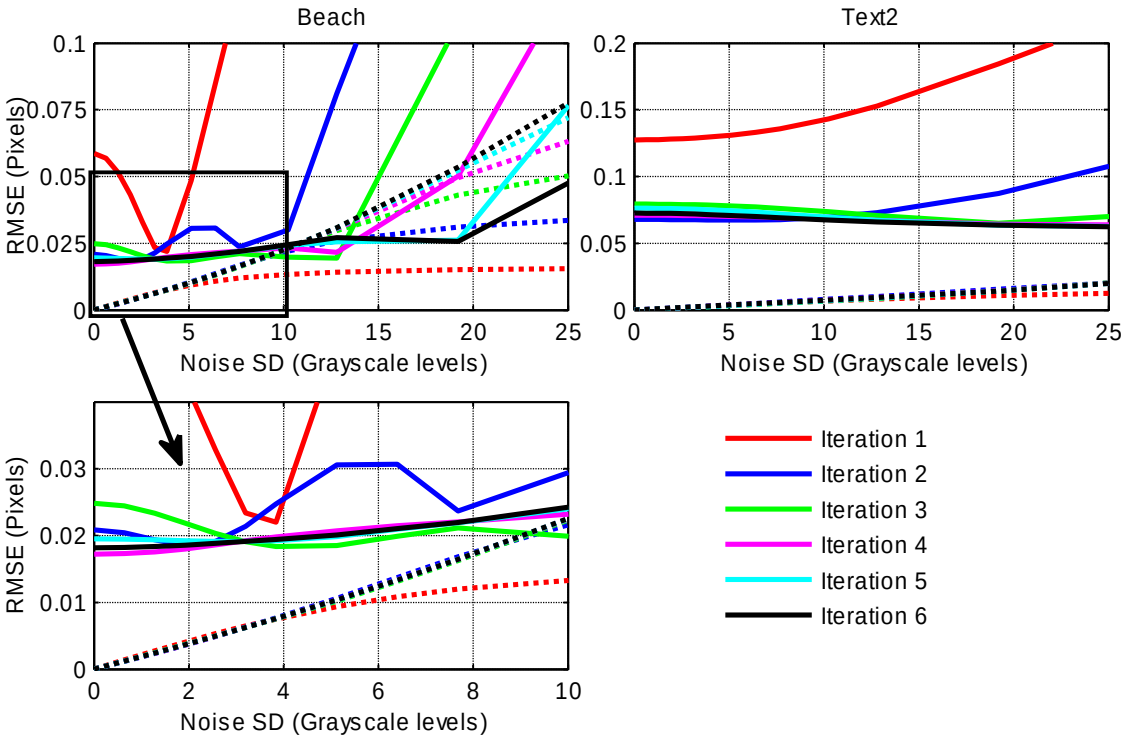


Figure 3.4 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying Lucas-Kanade method to images 'beach' (left plot) and 'text2' (right plot). Colours represent results after different iterations of the algorithm.

Figure 3.5 shows the effects of aliasing. Averaging the registration error over image pairs with high frequencies aliased in a different way has little effect on bias. However, the variance of the error increases. For image 'beach', variability due to aliasing is small in comparison to that due to noise and can only be noticed at very low noise levels. For image 'text2', however, variability due to aliasing dominates the variance and the level of variance is almost constant over all noise levels.

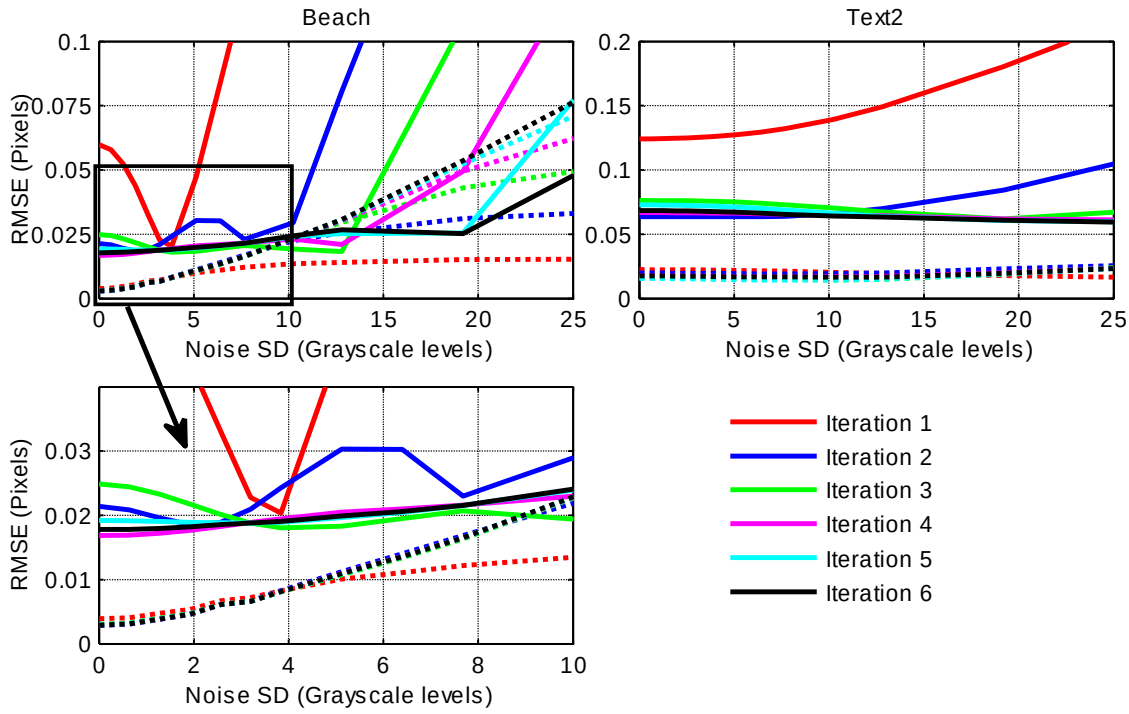


Figure 3.5 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying the Lucas-Kanade method to images 'beach' (left plot) and 'text2' (right plot). Colours represent results after different iterations of the algorithm. Results from 100 different image pairs with the same offset were averaged to show variability due to aliasing. See experimental method description for explanation of the difference to figure 3.4.

### (ii) Correlation-based methods

The results from cross-correlation methods are shown in figure 3.6. The basic correlation method, as described in section 3.2.1 is labelled 'Corr', the normalized correlation method is labelled 'NCorr' and the correlation coefficient method is labelled 'CorrCoef'. In all cases the pyramid fit gave significantly better results than the parabola fit. For this reason only the results that use the pyramid fit are shown.

The use of the basic correlation method on image 'beach' resulted in a much higher bias than from the two other methods. This was expected, as un-normalised correlation from equation (3.22) may result in large bias if there is a gradient over the image. For image 'text2' there was negligible bias from this source because the background is uniform.

Normalised correlation and correlation coefficient methods performed equally. It can be observed that bias dominates over variance for low noise situations in the case of image 'beach' and for any noise level in the case of image 'text2'. While variance increases with noise, bias stays constant, indicating that noise is not a large contributor to the bias – the slight increase at higher levels of noise is not statistically

significant given the significantly larger standard deviation. Most of this bias is probably caused by the mismatch between the employed pyramidal shape and the actual shape of the images' autocorrelation function. A better fitting model could possibly reduce this bias.

Figure 3.7 shows the effect of aliasing on variance for this method. The results are similar to that of the Lucas-Kanade method. For image 'beach' aliasing has only a small effect noticeable at low noise. For image 'text2' the effect of noise on variance is almost insignificant in comparison to the effect of aliasing. However, even with aliasing taken into account levels of bias are still significantly higher than the variance component.

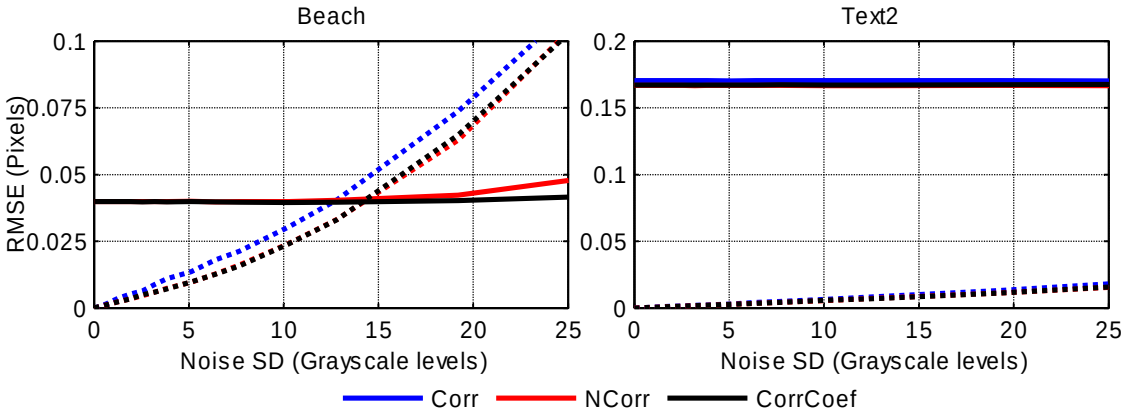


Figure 3.6 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying cross-correlation methods to images 'beach' (left plot) and 'text2' (right plot).

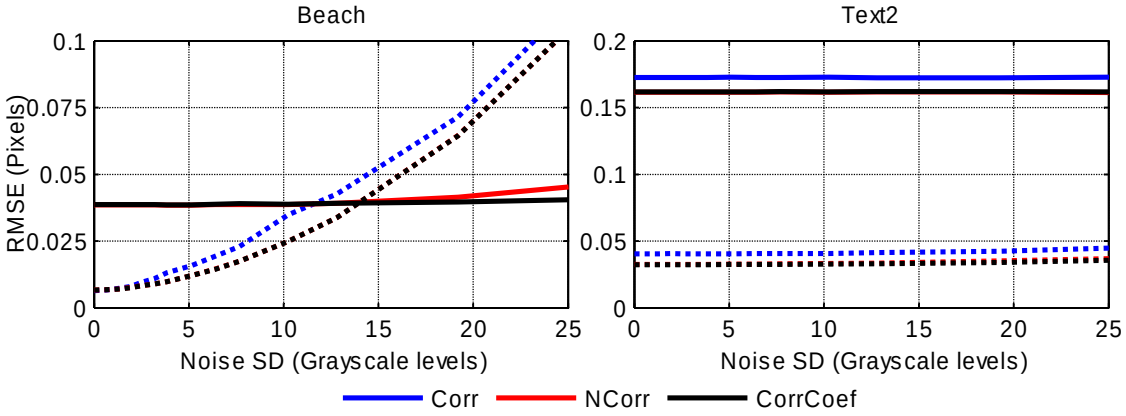


Figure 3.7 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying cross-correlation methods to images 'beach' (left plot) and 'text2' (right plot). Results from 100 different image pairs with the same offset were averaged to show variability due to aliasing.

### (iii) Fourier phase-based methods

The results from methods based on Fourier phase are shown in figure 3.8. The basic method that fits a plane to the central part (33 by 33 phase samples) of phase difference using magnitude-weighted least-squares is labelled ‘Raw’. To reduce any artefacts resulting from assuming that the images are periodic, images are first windowed using a Hann window before computing the Fourier transform. Results of this method are labelled as ‘Windowed’. The process of windowing can introduce bias if the image contains a significant DC level. As the image is shifted, the window does not shift, so the phase of the very low frequencies will be biased towards zero. To reduce this effect, average intensity level was subtracted from the images prior to windowing. The results of this method are labelled as ‘DC Removed + windowed’. Both methods by Stone et al. were also implemented, according to the algorithms provided in their papers [126, 127]. These methods are labelled ‘Stone `99’ and ‘Stone `01’.

As expected, the raw, unwindowed, method introduced a significant bias in ‘beach’ because of the periodicity assumption implicit in the discrete Fourier transform; windowing has significantly reduced this bias. For image ‘text2’, the raw method actually results in less bias than other methods. Windowed methods could possibly perform worse because windowing reduces the amount of information available to the algorithm to be used for offset estimation. Suppressing the DC component has a much smaller effect, although it does reduce the bias a little.

The ‘Stone `99’ algorithm performed adequately with no noise although it was very noise sensitive, and failed completely on ‘text2’ because there were often too few points remaining after trimming the outliers to obtain a reliable fit. The ‘Stone ‘01’ algorithm was better in this regard, and with the appropriate choice of parameters performed similarly to the simpler weighted fit. It took significantly longer (almost three times as long) to execute and the best parameters were strongly image dependent (the parameters recommended in their paper did not work for ‘text2’ in that the estimated $x$ offset was always very close to 0 pixels regardless of the actual offset although the $y$ offset was estimated correctly).

The addition of noise had relatively little effect on the bias. At any significant noise level, the accuracy is limited by the random component, rather than bias. The raw, unwindowed fit has the best noise characteristics because it is effectively averaging

over a larger number of pixels, reducing the effects of noise. The Stone et al. methods were more noise sensitive because they applied the fit to fewer samples.

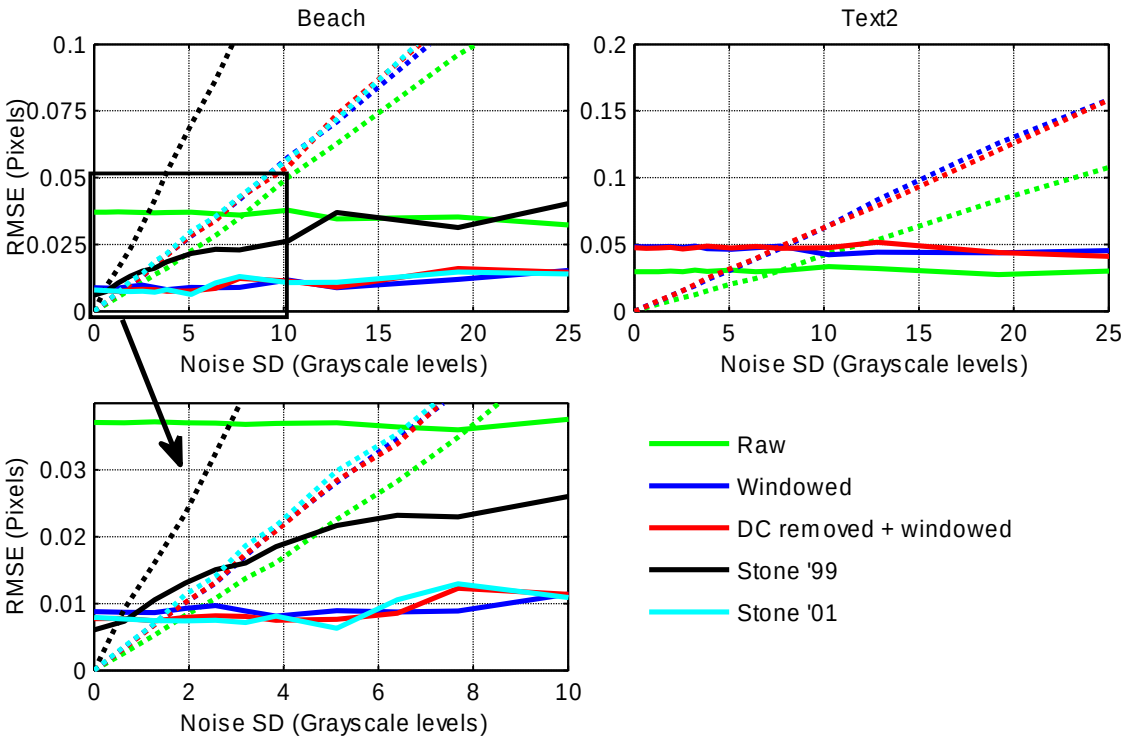


Figure 3.8 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying Fourier phase-based methods to images 'beach' (left plot) and 'text2' (right plot).

The effect of aliasing on registration error for this method is shown in figure 3.9. There is a noticeable increase in variance at low noise, especially for image 'text2'. This suggests that bias is affected by aliasing – pairs of input images with the same relative offset, but with higher frequencies aliased in a different way can result in significantly different registration error in the absence of noise. Even though these errors can be significant, as can be seen in figure 3.8, they average out to zero over all 100 image pairs with the same translations obtained by shifting the reference and target images over one pixel, as seen in figure 3.9. Consequently, much of the bias for 'text2' in figure 3.8 is a result of the aliasing present in the particular pair of images used. In figure 3.9, which averages over several instances, this error appears random and appears in the standard deviation rather than the bias. This shows that the Fourier phase methods are sensitive to the inherent differences between the images, such as aliasing and noise.

Overall, the DC suppressed, windowed weighted least-squares fit gave the best results for 'beach' and the raw method gave the best results for 'text2'. However, it

would be inaccurate to conclude from this that the 'Raw' method will perform better on heavily aliased images, as its superior noise sensitivity will be easily outweighed by a large bias for any image that does not fit well with the assumption of periodicity.

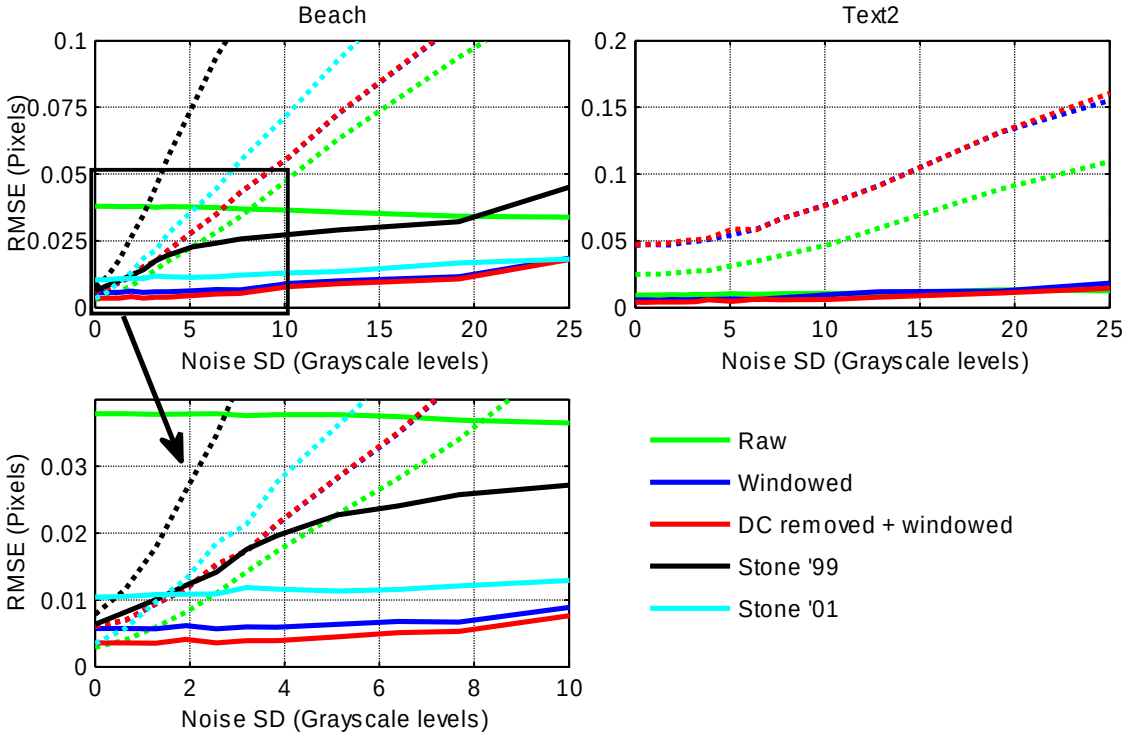


Figure 3.9 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying Fourier phase-based methods to images 'beach' (left plot) and 'text2' (right plot). Results from 100 different image pairs with the same offset were averaged to show variability due to aliasing.

### (iv) Optimal interpolation-based methods

The proposed method of image registration using optimal interpolation with filter window sizes of 2×2, 3×3 and 4×4 is evaluated. The results are shown in figure 3.10 with different colours representing filters of different order. It can be observed from the results (especially image 'beach') that the bias is strongly noise dependent. Adding some noise actually reduces bias for the 2×2 and 3×3 filters. This suggests that there is an interaction in the bias mechanism between signal and noise.

The 4×4 filter performs significantly better than 2×2 and 3×3 filters for both images. It results in very little bias with low-noise. However, increasing the filter order increases the sensitivity to noise in the form of larger variance, especially at higher noise levels.

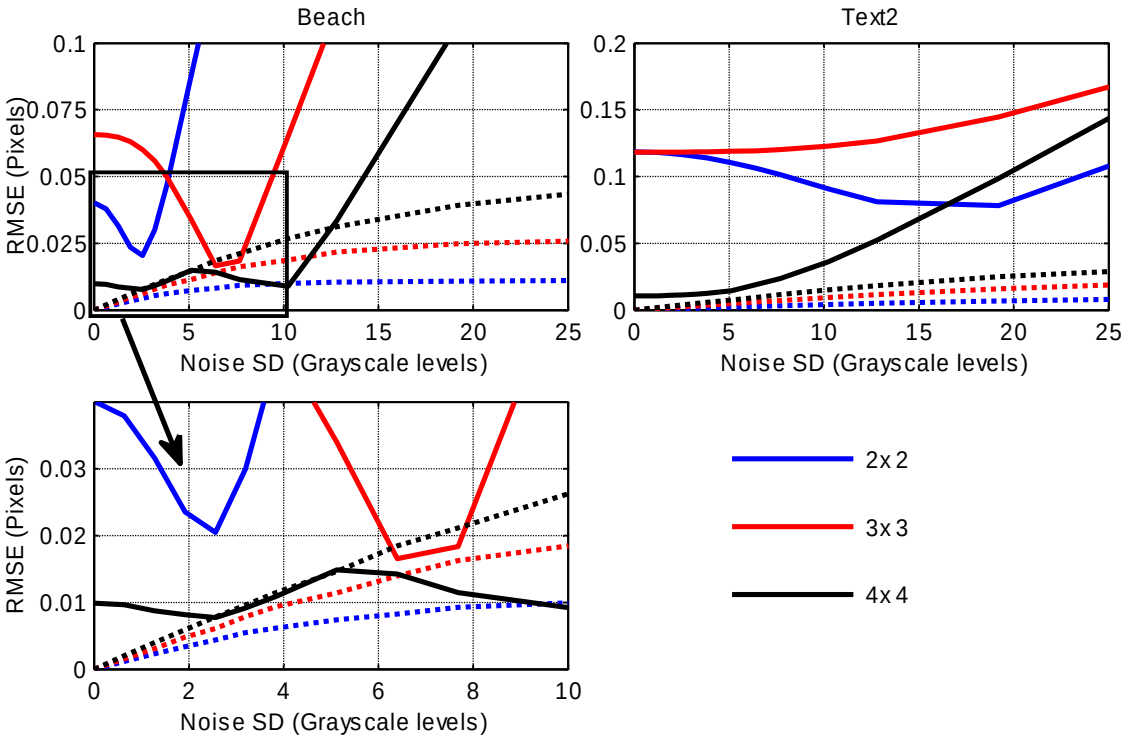


Figure 3.10 Plots of bias (solid) and SD (dotted) vs noise SD resulted from applying optimal interpolation-based method to images 'beach' (left plot) and 'text2' (right plot). Colours represent results obtained using filters of different order.

Figure 3.11 shows the effect of aliasing on registration error. There is a significant increase in variance at low noise, especially for heavier aliased image 'text2'. The effect of aliasing for this image is more subtle with higher order filters. For both images, 2x2 and 3x3 filters are still heavily dominated by bias. However, in the case of the 4x4 filter standard deviation increases to a level comparable in magnitude to bias for low noise.

For image 'text2' it can be seen that standard deviation is almost constant with noise (it actually decreases slightly for first and second orders); hence, variance resulting from aliasing dominates over the variance resulting from noise.

Overall, for low to moderate levels of noise, the 4x4 order filter performed better than the 2x2 and 3x3 for both images.

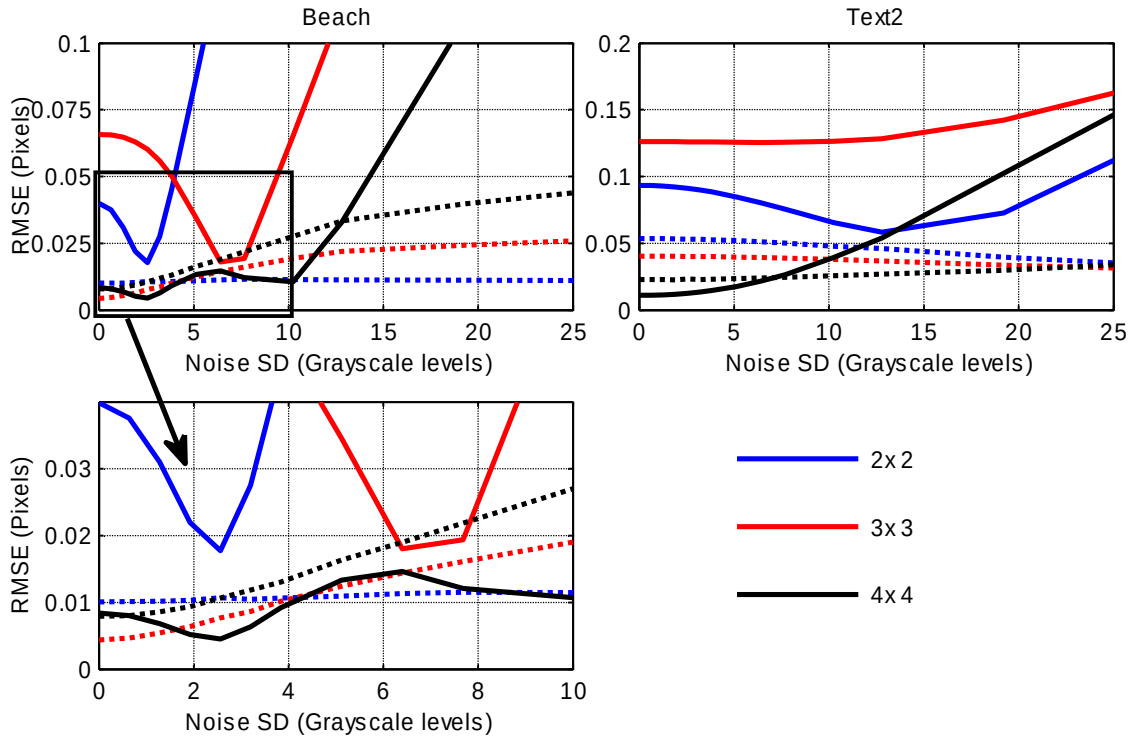


Figure 3.11 Plots of bias (solid) and SD (dotted) vs noise SD resulted from optimal interpolation-based method to images 'beach' (left plot) and 'text2' (right plot). Colours represent results obtained using filters of different order. Results from 100 different image pairs with the same offset were averaged to show variability due to aliasing.

### (v) Comparison

Figure 3.12 shows a comparison between the results obtained from the 4x4 order optimal filter, DC removed and windowed Fourier phase method and the fourth iteration of Lucas-Kanade method. The correlation based filters are not shown, as the results were only mediocre in comparison. Bias and variance were combined into the total RMS error and this was plotted against noise standard deviation.

It can be observed that all methods are capable of registering images with limited aliasing, such as 'beach', with good sub-pixel precision. Heavily aliased images such as 'text2' are slightly more challenging for other methods; however the proposed method performs equally well on both, heavily and limited aliased images.

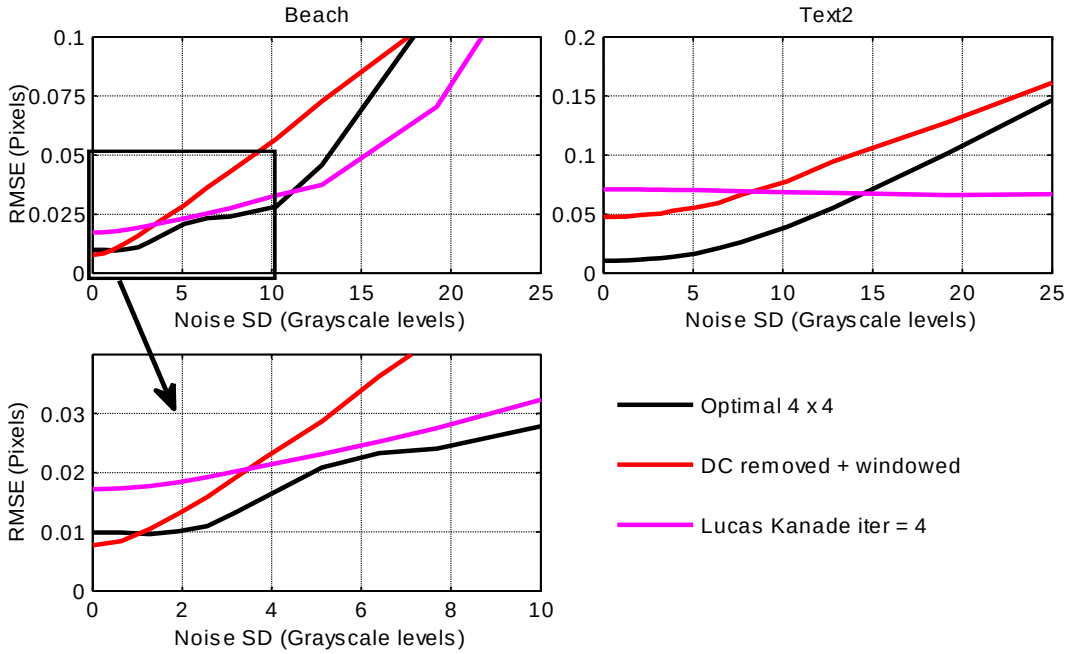


Figure 3.12 Plots of total RMS error vs noise SD for selected methods.

### *Conclusions*

The proposed method of registration using optimal interpolation compares well to existing methods, outperforming them when the noise is low. An accuracy of one hundredth of a pixel is possible for both heavily aliased images and images with limited degree of aliasing under these conditions.

The 2x2 and 3x3 optimal interpolation filters also perform relatively well at some noise levels. The interaction within the bias mechanism that results in low bias at some noise levels could be investigated and possibly exploited to achieve low bias levels over a wide range of noise levels. This would result in a lower order filter that could offer registration accuracy similar to that of a higher order filter.

## *3.3. Analysis of Bias in Registration using Optimal Interpolation*

In the previous section, it was shown that bias is the dominant source of error for optimal interpolation under high SNR conditions. Image super-resolution requires accurate sub-pixel registration and any strong bias is detrimental to good results. Analytical analysis of registration bias can be conducted to identify possible sources for the bias and to get insight into the interactions within the bias mechanisms.

To investigate this source of error, simple image models are employed to simulate the data. Two one-dimensional analytical models are employed, based on two types of

features: sinusoids that directly measure the frequency dependence of the bias and allow the effects of aliasing to be considered explicitly; and step-edges, which provide a more realistic model of typical images where a significant proportion of the high frequency content is contained within the edges between regions.

### 3.3.1. Sine wave image model

Consider estimating registration for a pair of sinusoidal signals in the absence of noise:

$$\begin{aligned} f[k] &= \sin \Omega k \\ g[k] &= \sin \Omega (k+u). \end{aligned} \tag{3.30}$$

Using one dimensional signals makes little difference to the derivation presented in section 3.1. Registration is estimated using optimal resampling filters of orders one and two. A sampled sinusoid is only a second order function, in that successive samples may be generated recursively using a second order filter:

$$s[k] = (2\cos\Omega)\, s[k-1] - s[k-2]. \tag{3.31}$$

As a result, there are only two linearly independent equations in the least-squares matrix in (3.9); therefore the optimal interpolation procedure is undefined for third and higher orders because the matrix $\mathbf{F}^T\mathbf{F}$ is singular.

The use of first order filter results in the registration estimate being

$$\hat{u}_1 = \left[ \sum_k \bar{f}_1[k]\, \bar{f}_1[k] \right]^{-1} \sum_k \bar{g}[k]\, \bar{f}_1[k]. \tag{3.32}$$

Substituting (3.30) into (3.32) results in

$$\hat{u}_1 = \frac{\sum_{k=-\infty}^{\infty} (\sin \Omega (k+1) - \sin \Omega k)(\sin \Omega (k+u) - \sin \Omega k)}{\sum_{n=-\infty}^{\infty} (\sin \Omega (k+1) - \sin \Omega k)^2}. \tag{3.33}$$

It is assumed that the signals are infinitely long: hence the infinite sums in the numerator and the denominator. These sinusoidal sampled signals ( $f[k]$ and $g[k]$) will contain all possible phase angles, unless the sinusoid frequency $\Omega$ is harmonically related to the sample frequency. Therefore the summations in (3.33) can be replaced with integrals over a single period:

$$\hat{u}_1 = \frac{\int_{-\pi/\Omega}^{\pi/\Omega} (\sin\Omega(k+1) - \sin\Omega k)(\sin\Omega(k+u) - \sin\Omega k)\,dk}{\int_{-\pi/\Omega}^{\pi/\Omega} (\sin\Omega(k+1) - \sin\Omega k)^2\,dk} = \frac{\cos\Omega(1-u) - \cos\Omega u + 1 - \cos\Omega}{2(1-\cos\Omega)} \tag{3.34}$$

The second order optimal filter has two independent coefficients, one positioned at $-1$ and one at 1. The use of this filter results in the registration estimate being:

$$\hat{u}_2 = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} \sum_k \bar{f}_1[k]\bar{f}_1[k] & \sum_k \bar{f}_2[k]\bar{f}_1[k] \\ \sum_k \bar{f}_1[k]\bar{f}_2[k] & \sum_k \bar{f}_2[k]\bar{f}_2[k] \end{bmatrix}^{-1} \begin{bmatrix} \sum_k \bar{g}[k]\bar{f}_1[k] \\ \sum_k \bar{g}[k]\bar{f}_2[k] \end{bmatrix}. \tag{3.35}$$

In a similar manner, substituting (3.30) into (3.35) and integrating over all possible phase angles will result in the registration estimate using the second order interpolator:

$$\hat{u}_2 = \frac{\sin\Omega u}{\sin\Omega}. \tag{3.36}$$

From equations (3.34) and (3.36) it can be seen that the bias ($=\hat{u}-u$) is a function of both sine-wave frequency and the offset between the images. It is clear that there is bias present even if the signals satisfy the Nyquist criterion.

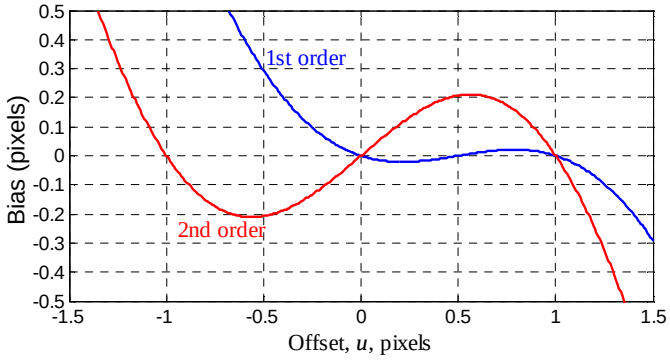


Figure 3.13 Bias in estimating the offset of sinusoids ($\Omega = \pi/2$) as a function of the offset.

Figure 3.13 shows the biases as a function of offset at a sampling frequency of half the Nyquist rate. Several observations may be drawn from this figure:

- In the absence of noise, there is no bias for offsets equal to an integer number of pixels (within the region of support of the interpolation kernel). This results from the use of the super-position principle in (3.13).

- The filter extrapolates poorly, as indicated by the growing bias outside the region of support.
- The bias from extrapolation is towards the region of support – outside the region of support the offset is underestimated, resulting in a positive bias for negative offsets and negative bias for positive offsets.
- The previous observation implies that there will be an odd number of offsets that give zero bias. It is this extra bias zero that gives the first order filter a significantly lower bias within the region of support.
- For the first order interpolator, the bias is away from the centre of the region of support. Similarly, for the second order filter, the bias is towards the end samples and away from the central sample.

The bias pattern remains much the same shape for different frequency signals (below the Nyquist frequency), although the amplitude of the error is larger at higher frequencies. As the offset is not known in advance (otherwise registration is unnecessary), the RMS error over all possible offsets in the range 0 to 1 provides an estimate of the average error for sub-pixel registration. Figure 3.14 compares the frequency response of the bias for the two filters. Both filters perform well if the sinusoid frequency is much lower than the sampling frequency. The first order filter is effective past the Nyquist frequency and gives meaningful estimates of the offset even for aliased signals. The performance of the second order filter deteriorates much earlier, with the estimated offset becoming meaningless even for unaliased sinusoids close to the Nyquist frequency. The reason for this instability is that the sample at $f(u+1)$ becomes the same as the one at $f(u-1)$ as the Nyquist frequency is approached. The matrix $\mathbf{F}^T\mathbf{F}$ becomes ill-conditioned for the second order filter.

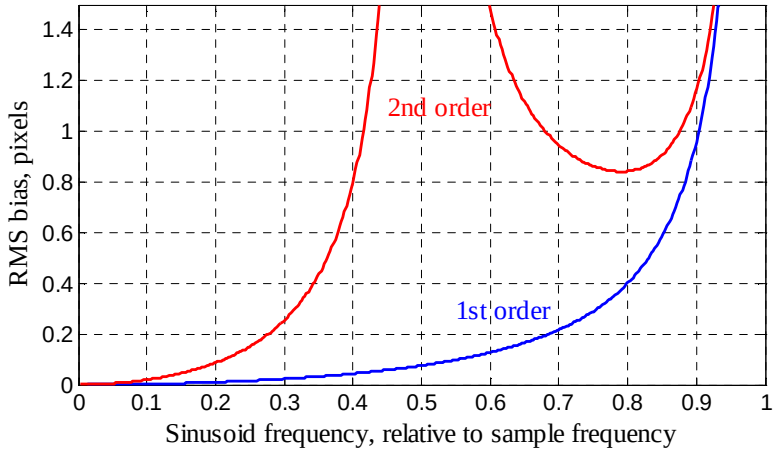


Figure 3.14 Frequency dependence of bias (sine-wave model).

Although these bias patterns would indicate that these methods are likely to perform poorly with aliased signals, in practise the situation is a lot more complex. Images seldom consist of a single sinusoid, and even when there are high frequency components (above the Nyquist rate) the low frequencies are still dominant. The bias mechanism is not linear, so it is not simply a case of adding the bias from each of the frequency components to get the overall bias. In an image with dominant low frequency components, the overall bias will be dominated by the low frequency component, although the higher frequencies would be expected to affect the bias.

The sinusoidal model gives some insight into the behaviour of predictive interpolation for signals of different frequencies. However, images include discontinuities, such as edges. The step-edge model can provide more insight into the bias mechanism of real images.

### 3.3.2. Step-edge image model

A more realistic model is that of a step-edge, since many images can be approximated by piecewise constant regions (with step-edges in between). In forming the image, a step-edge is blurred by area-sampling to a single pixel wide ramp. This means that the edge pixels take on an intermediate value depending on the exact position of the edge relative to the sampling grid.

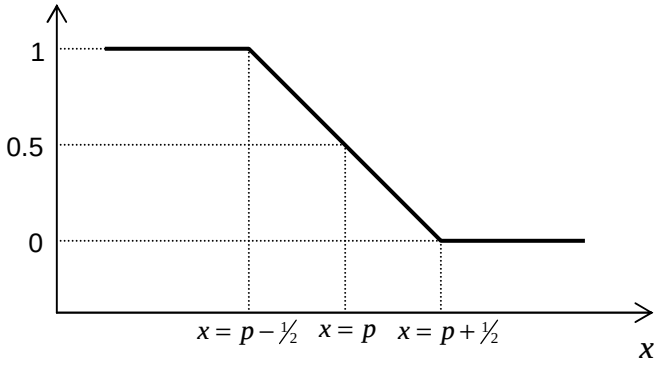


Figure 3.15 Step-edge image model.

Generally, there are many step-edges in an image and the location of these relative to the sampling grid is random, unless there is a strong periodic pattern present in the image, with a period harmonically related to the sampling rate. The filter weights are optimised to minimise the square error by evaluating the summations of the input and output samples over all possible locations of the edge.

Consider a model (pictured in figure 3.15) consisting of a single step-edge, blurred by area-sampling:

$$M_E(x;p)=\begin{cases}1 & \text{if } x-p<-0.5\\ p+0.5-x & \text{if } -0.5\le x-p<0.5\\ 0 & \text{if } x-p\ge 0.5\end{cases} \tag{3.37}$$

where $x=p$ is the location of the centre of the edge. A single output pixel (along with a set of input samples $f_i$ that are within the filter's region of support) is considered. The model function is sampled at the integer locations of the input samples $f_i$, located at $x=c_i$ and the location of the output sample $g$, located at $x=u$:

$$g(p)=M_E(u;p)=\begin{cases}1 & \text{if } u-p<-0.5\\ p+0.5-u & \text{if } -0.5\le u-p<0.5\\ 0 & \text{if } u-p\ge 0.5\end{cases} \tag{3.38}$$

$$f_i(p)=M_E(c_i;p)=\begin{cases}1 & \text{if } c_i-p<-0.5\\ p+0.5-c_i & \text{if } -0.5\le c_i-p<0.5\\ 0 & \text{if } c_i-p\ge 0.5\end{cases} \tag{3.39}$$

To compute the registration estimate for the first order, equations (3.38) and (3.39) are substituted into (3.32). The sum over all pixels in the image is replaced with an integral over all possible positions of the edge relative to the filter window:

$$\hat{u}_1 = \frac{\int_{-\infty}^{\infty} \left( \bar{M}_E(u;p)\bar{M}_E(1;p) \right) dp}{\int_{-\infty}^{\infty} \left( \bar{M}_E(1;p)\bar{M}_E(1;p) \right) dp}, \tag{3.40}$$

where $\bar{M}_E(x;p) = M_E(x;p) - M_E(0;p)$. Because of this subtraction, the edge only has effect when $p$ is in the interval between –0.5 and 1.5:

$$\begin{aligned} \hat{u}_1 &= \frac{\int_{-0.5}^{u-0.5} (-p-0.5)^2\, dp + \int_{u-0.5}^{0.5} -u(-p-0.5)\, dp + \int_{0.5}^{u+0.5} (p-1.5)(p-u-0.5)\, dp}{\int_{-0.5}^{0.5} (-p-0.5)^2\, dp + \int_{0.5}^{1.5} (p-1.5)^2\, dp} \\ &= \tfrac{3}{4}u + \tfrac{3}{4}u^2 - \tfrac{1}{2}u^3 \end{aligned} \tag{3.41}$$

The estimator bias can be computed by subtracting the true offset, $u$, from this expression:

$$\text{bias}\{\hat{u}_1\} = -\tfrac{1}{4}u + \tfrac{3}{4}u^2 - \tfrac{1}{2}u^3,\ 0 \le u \le 1. \tag{3.42}$$

A similar exercise can be completed to derive the bias for the second order filter:

$$\text{bias}\{\hat{u}_2\} = \frac{u - u^3}{5},\ -1 \le u \le 1. \tag{3.43}$$

Figure 3.16 shows the bias as a function of the true offset plotted for orders one to five. These functions consist of a series of piecewise cubic curves. It can be observed that for the first and second orders, the general shape is the same as that obtained using a sinusoidal model.

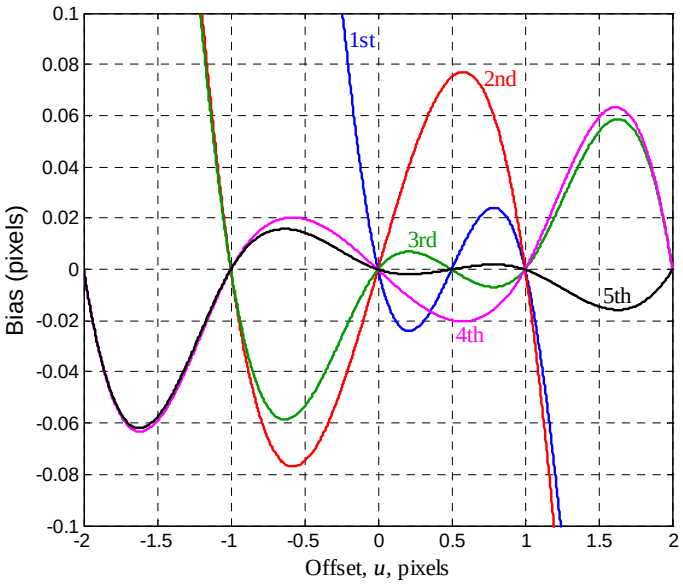


Figure 3.16 Bias characteristics of a blurred step-edge for different order interpolation filters (orders one to five).

The bias is symmetric around the centre of the region of support: 0.5 for the odd orders and 0.0 for the even orders. The sub-pixel performance of these estimators can be assessed by considering the bias in the range $-0.5 \le u \le 0.5$ for even order filters and $0 \le u \le 1$ for odd order filters. Using odd order filters results in a significantly smaller bias in comparison to filters of similar even order. As outlined previously, this phenomenon is most likely due to an extra zero bias at $u = 0.5$. In this sense, odd order filters appear to be more useful than even order filters.

For odd orders, increasing the order by two not only significantly improves the bias, but also extends the useful region of relatively low bias by expanding the region of support. Outside the region of support, the bias grows consistently in a similar manner regardless of the filter order.

### 3.3.3. Effect of noise on bias

The effect of additive white Gaussian noise can be analysed by considering equation (3.9) that computes the optimal filter coefficients. Recall that this equation consists of a square matrix inverse multiplied by a column vector. The matrix $\mathbf{F}^T\mathbf{F}$ is a $\Theta$ by $\Theta$ matrix with element in column $i$ and row $j$ of the form

$$\begin{aligned}\left[\mathbf{F}^T\mathbf{F}\right]_{j,i} &= \sum_{k,l} \bar{f}_i[k,l]\,\bar{f}_j[k,l] \\ &= \sum_{k,l}\left(f_i[k,l]-f_0[k,l]\right)\left(f_j[k,l]-f_0[k,l]\right)\end{aligned} \tag{3.44}$$

and the $\Theta$ by one column vector $\mathbf{F}\bar{\mathbf{g}}$ consists of elements

$$\begin{aligned}\left[\mathbf{F}\bar{\mathbf{g}}\right]_j &= \sum_{k,l} \bar{g}_i[k,l]\,\bar{f}_j[k,l] \\ &= \sum_{k,l}\left(g[k,l]-f_0[k,l]\right)\left(f_j[k,l]-f_0[k,l]\right)\end{aligned} \tag{3.45}$$

where the sums are over all of the pixels in the image $g[k,l]$. The images can then be contaminated by noise

$$\begin{aligned}\tilde{f}[k,l] &= f[k,l] + \quad [k,l] \\ \tilde{g}[k,l] &= g[k,l] + \quad [k,l]\end{aligned} \tag{3.46}$$

where $[k,l]$ and $[k,l]$ are independent and identically distributed noise images with zero mean and variances $\quad_f^2$ and $\quad_g^2$.

Momentarily dropping the dependence on sample number $[k,l]$ to shorten the notation, equation (3.44) in the presence of noise becomes

$$\begin{aligned}\left[\mathbf{F}^T\mathbf{F}\right]_{j,i} &= \sum_{k,l}\left(\left(f_i + \quad_i\right)-\left(f_0 + \quad_0\right)\right)\left(\left(f_j + \quad_j\right)-\left(f_0 + \quad_0\right)\right) \\ &= \sum_{k,l}\begin{pmatrix}\left(f_i f_j - f_i f_0 - f_j f_0 + f_0^2\right) + \quad_0^2 + \quad_i \quad_j - \quad_0\left(\quad_i + \quad_j\right) \\ + \quad_0\left(2f_0 - f_i - f_j\right) + \quad_i\left(f_j - f_0\right) + \quad_j\left(f_i - f_0\right)\end{pmatrix}\end{aligned} \tag{3.47}$$

Assuming that the summation is over a large number of pixels, the sum of uncorrelated noise components will go to zero, whereas the sum of a squared noise component will be equal the noise variance, resulting in

$$\left[\mathbf{F}^T\mathbf{F}\right]_{j,i} = \begin{cases}\displaystyle\sum_{k,l}\bar{f}_i[k,l]\,\bar{f}_j[k,l] + 2\quad_f^2 & \text{if } i = j \\ \displaystyle\sum_{k,l}\bar{f}_i[k,l]\,\bar{f}_j[k,l] + \quad_f^2 & \text{if } i \neq j\end{cases} \tag{3.48}$$

Expanding equation (3.45) in a similar fashion results in

$$\left[\mathbf{F}\bar{\mathbf{g}}\right]_j = \sum_{k,l}\left(\bar{g}[k,l]\,\bar{f}_j[k,l]\right) + \quad_f^2 \tag{3.49}$$

Note that the noise term added to the diagonal elements of $\mathbf{F}^T\mathbf{F}$ is twice that added to all of the other elements of $\mathbf{F}^T\mathbf{F}$ and the elements of $\mathbf{F}\overline{\mathbf{g}}$. When the noise level is insignificantly low in comparison to the signal level, this noise term will have no effect. However, when the noise dominates over the signal, the diagonal terms in matrix $\mathbf{F}^T\mathbf{F}$ will tend to become twice as large as non-diagonal terms and the terms of vector $\mathbf{F}\overline{\mathbf{g}}$. This will result in all of the optimal filter coefficients tending to $1/R$, which makes intuitive sense: if there is no image content and only noise, the best estimate will be to minimise the variance by giving equal weight to each of the pixels within the window.

The effect of this on the registration estimate can be considered by examining equations (3.13) and (3.14). Substituting $1/R$ for $h[m,n]$, results in

$$\hat{u} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} m/R = \frac{\sum_{m=M^-}^{M^+} m}{M^+ - M^- + 1} = \frac{M^+ - M^-}{2}, \tag{3.50}$$

$$\hat{v} = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} n/R = \frac{\sum_{n=N^-}^{N^+} n}{N^+ - N^- + 1} = \frac{N^+ - N^-}{2}. \tag{3.51}$$

Since $m$ and $n$ are integers, the offset will be estimated to be equal to the position of the centre of the filter window. Assuming that the filter window is square, this will be either (0,0) or (0.5,0.5), depending on whether there is an odd or even number of coefficients in each direction. At intermediate levels of noise the effect of this is to bias the offset estimate towards the centre of the filter's region of support and the actual amount of bias will depend on the noise level.

### 3.3.4. Effects of noise on bias with the step-edge model

The effect of noise on bias using the step-edge model is now considered. Repeating the derivation of equation (3.41), with the addition of random noise results in

$$\hat{u}_1 = \frac{\frac{3}{4}u + \frac{3}{4}u^2 - \frac{1}{2}u^3 + 3\sigma_f^2}{1 + 6\sigma_f^2}, \; 0 \le u \le 1. \tag{3.52}$$

For the low or no noise situation, the bias is dominated by the signal. When $\sigma_f^2$ is large, however, the expected value of the offset estimate tends towards 0.5, as noted

previously. For the second order filter, the offset estimate in the presence of noise becomes:

$$\hat{u}_2 = \frac{u - u^3/6}{\sigma_f^2 + 5/6}, \; -1 \le u \le 1. \tag{3.53}$$

In this case, again, as the noise level is increased, the estimate tends towards the centre of filter window – zero in the case of an even order filter. This phenomenon can be observed in figure 3.17. As the noise level grows, the bias function slowly tends towards $\text{bias}\{\hat{u}_1\} = 0.5 - u$ for the first order and $\text{bias}\{\hat{u}_2\} = -u$ for the second order, creating partial cancellation of the bias along the way. This is the effect seen in figure 3.10 with the reduction in bias for intermediate levels of noise.

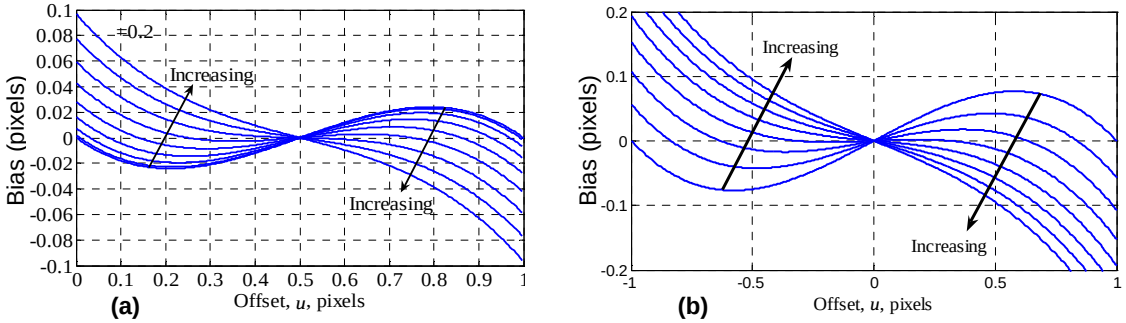


Figure 3.17 Change in step-edge bias under caused by increase in noise level: (a) filter order one, (b) filter order two.

Since the primary interest is sub-pixel registration, the bias is averaged over an offset interval of one pixel (to give indication of the performance on average) by calculating the RMS bias over that interval. Assuming the images are already pre-registered to within one pixel, for odd order filters offsets in the range of $0 \le u \le 1$ were considered. For even order filters, the symmetry about 0 suggests that it is more appropriate to consider offsets in the range $-0.5 \le u \le 0.5$. The results are shown in figure 3.18 for noise standard deviations up to the edge height (equal to one).

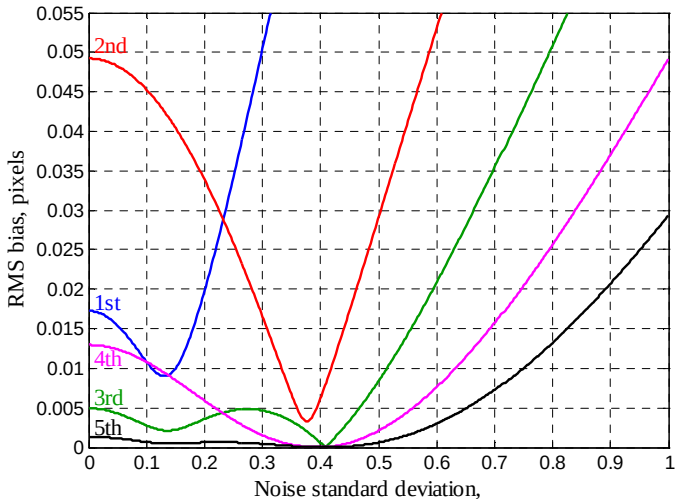


Figure 3.18 The effect of noise on bias for sub-pixel registration (step-edge model).

As observed previously, the addition of modest levels of noise can significantly reduce the bias because the noise induced bias is in the opposite direction to the bias from the signal (image) itself. This effect is even more marked with the higher order filters, with the bias being completely cancelled within the sub-pixel range for $\sigma_f^2 = 1/6$. The bias function for these higher orders is a fraction of two polynomials. For third and higher order filters, $\sigma_f^2 = 1/6$ is a root of the numerator, making the bias zero at this level of noise for any offset $u$.

Both the 3rd and 5th order interpolation filters give excellent (better than 1% of a pixel) registration performance over a wide range of noise levels. At higher noise levels, the higher the order of the filter, the smaller the bias.

### 3.3.5. Experiment VI – Model validation

In order to generalise the observations made from the analytic bias analysis using step-edge model, the results need to be validated. Real images contain many edges, and so the assumption that only a single step-edge is situated within the filter's window may not be valid. The usefulness of the infinite scale step-edge model is confirmed by comparing analytic results to experimental results using a real image.

***Aim***

To compare the results obtained from analytic bias analysis using a step-edge model with bias obtained from real image data.

### *Method*

A 1700×1700 pixels source image was filtered using a 1×20 horizontal box average filter to simulate 1D area integration. Shifting this blurred image by an integer number of pixels and sub-sampling horizontally by a factor of 20 (to 1700×85 pixels) produces a series of low-resolution images with known offsets in steps of 0.05 pixels. Image 'beach' was used as the high-resolution source image.

The experiment is divided into two parts. First, the pattern of the bias as a function of the offset is validated. Pairs of images with relative sub-pixel offsets between –2 and 2 were created and registered using optimal filters of orders one to five. The ground truth offset was subtracted from the estimate to calculate the bias and this was plotted against the offset for each filter.

In the second part, the RMS bias over the sub-pixel region as a function of the noise is validated. Pairs of images with relative offsets between 0 and 1 for odd order filters and –0.5 and 0.5 for even order filters in steps of 0.05 were generated. Each one of these pairs were contaminated by additive white Gaussian noise with zero mean and variance $^2$. Registration was performed 200 times with different noise instances and the measurements were averaged to approximate the expected value. The ground truth offset was subtracted to produce the bias. Finally, the RMS average of the bias measurements over the sub-pixel range was obtained. This was repeated with a range of noise standard deviation levels between 0 and 0.06. The RMS bias was then plotted against the noise level.

### *Results*

Figure 3.19 shows the measured bias patterns obtained in the absence of noise. The pattern of the bias is very similar to those obtained analytically from the step-edge model (shown in figure 3.16). The bias from the test image is slightly higher than that estimated from the model. Some differences between the results obtained from the model and the results from a real image were expected, as the image does not consist of ideal step-edges alone.

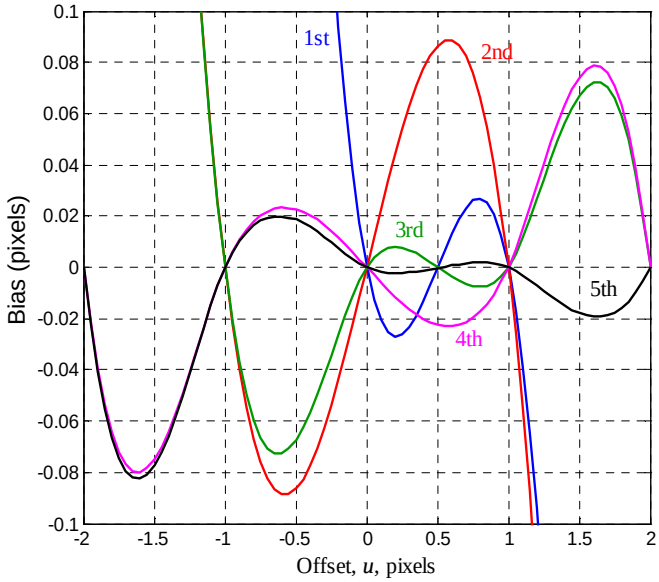


Figure 3.19 Bias characteristics measured from the "Beach" test image for different order interpolation filters (first to fifth order).

Figure 3.20 shows the root-mean-square bias in the sub-pixel range as a function of added noise. Notice that the scale of the x-axis is different to that of figure 3.18. In the previous experiment the signal consisted of a single step-edge of unit height. The test image in this experiment contains many step-edges of various heights. Because the effect of bias cancellation is caused by interplay of bias due to signal and bias due to noise, the noise strength is relative to signal strength. This explains the troughs in the bias functions shown in figure 3.20 occurring at a lower noise level than that in figure 3.18. Taking the scale differences into account, the pattern of the bias is very similar to that obtained analytically from the step-edge model (shown in figure 3.18). The cancellation is not complete because a typical image has a range of step heights, with each step requiring a different level of noise for complete cancellation. In spite of this, there is a distinct dip for low order filters, and the general pattern of the dependence of the bias on noise is similar.

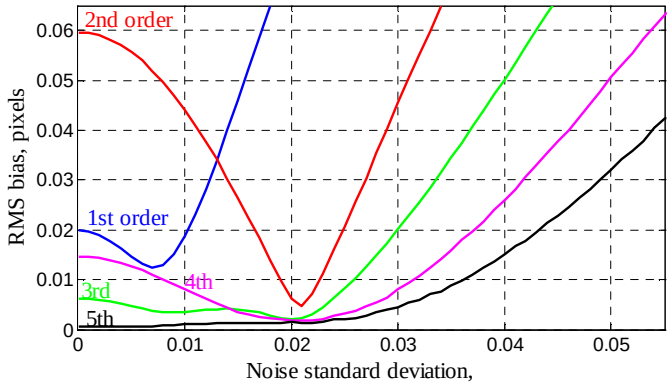


Figure 3.20 Noise characteristics of the bias measured from the "Beach" test image for different order interpolation filters (first to fifth order).

***Conclusions***

From these results it can be concluded that the piecewise constant model with area-sampling provides a reasonable representation of the dominant image characteristics in terms of registration using optimal interpolation.

## 3.4. Extension of Bias Analysis to Two Dimensions

In previous section it was shown that there exists an interaction in the bias mechanism between the bias from signal and the bias from noise. This interaction may cause a partial cancellation of the bias, as these two sources result in the estimate being biased in opposite directions. These results were derived using one dimensional model and even though they are validated using a real image, the conclusions may not extend to the two dimensional case. The way filter coefficients are combined to produce the offset estimate is separable, as seen from equations (3.13) and (3.14); however, the two dimensional optimal filter is not.

The bias of optimal interpolation-based registration is plotted here to show that the two dimensional registration exhibits similar trends. The bias results obtained in experiment V using image beach that are displayed in figure 3.10 are reused here. Bias is plotted in figure 3.21 as a function of the offset in $x$ and $y$ directions in the form of pincushion maps for various noise levels. Each pin originates from the true offset and points to the location of the estimated offset denoted with a cross.

For each filter, the noise levels were chosen so as to demonstrate that there is an interaction between the bias from signal and the bias from noise. For the 2×2 filter and 3×3 filters, the first row shows bias in the absence of noise – this bias points away from the centre of the filter window. The second row shows bias near the minimum.

The third row shows bias past the minimum, at an RMS level close to that of no noise situation. Last row shows bias at an RMS level double that of no noise situation. It can be clearly seen that bias has changed direction and is pointing towards the centre of the filter window, as predicted in the previous section. The strength of the bias increases as the noise level is raised.

For the 4×4 filter, the behaviour of bias with respect to noise level is more complex. However, similar to other filters, when noise level increases past a certain level, the bias is dominated by effects of noise and this bias points towards the centre of the filter window.

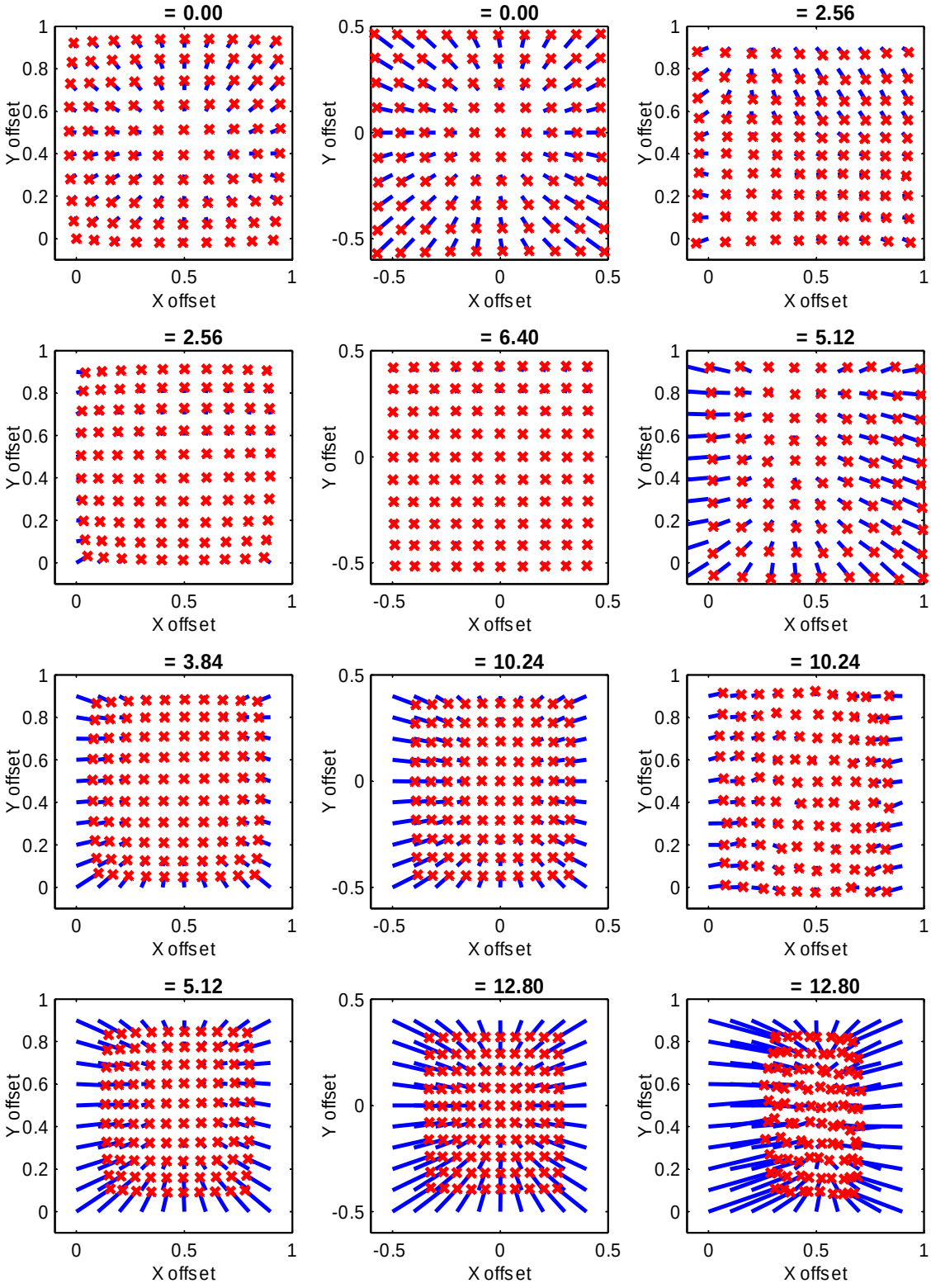


Figure 3.21 Pincushion maps of bias resulted from the use of 2×2 optimal filter (left column), 3×3 filter (centre) and 4×4 filter (right column) at various noise levels. Bias resulted from the 4×4 filter was scaled up by a factor of 5 for display purposes. Standard deviation is reported in grayscale levels. See text for further description.

## 3.5. Summary and Conclusions

Image registration is traditionally a non-linear optimisation problem that makes use of interpolation as a crucial link between the discrete and continuous domains. Ideally, improving interpolation should improve the accuracy of registration. This chapter introduces the idea of using optimal interpolation for image registration. Unlike traditional interpolation methods, optimal interpolation is derived from the data itself. No other linear method can yield a lower square error than this method.

The use of interpolation methods better than bilinear normally results in a non-linear problem that requires an iterative solution. An important advantage of using optimal interpolation over traditional methods is the way its use can linearise the problem. Optimal interpolation allows for a direct solution, by turning the problem around. Determining the optimal interpolation filter is a linear problem, and the sub-pixel offset is then computed linearly from the interpolation weights.

Sub-pixel accuracy of registration is crucial to image super-resolution. Experimental analysis shows that the registration error for the proposed method is smaller than that of a number of existing methods. Registration accuracy of one hundredth of a pixel is achievable in low-noise situations using a 4×4 optimal filter. This applies to both, heavily aliased images and images with only limited degree of aliasing. Please note, however, that this does not necessarily mean that offsets on order of one hundredth of a pixel can be successfully registered.

Analytic analysis of bias in estimating registration using optimal interpolation showed that the bias is a function of the noise level as well as the signal. Presence of noise in an image biases the estimate towards zero for even filters and 0.5 for odd filters. The interaction of bias from signal and bias from noise can result in the bias reducing for moderate noise levels, as has been observed in figure 3.10 and figure 3.20. At higher noise levels, however, the bias increases as the noise level grows.

Overall, optimal interpolation based registration offers reasonable results when used with aliased images – a property important for image super-resolution. The flexibility in terms of selecting the size of the region of support offers a good trade-off in terms of computational complexity and accuracy of registration.

# Chapter 4

## Image Fusion and Resampling

Image super-resolution reconstruction involves combining information from an ensemble of low-resolution images of the same scene to construct an image of higher resolution. The images in the ensemble are assumed to have pixels in slightly different locations. The previous chapter discussed the problem of obtaining the registration parameters from the images. It was established that the parameters can be obtained with a sub-pixel accuracy of about 0.01 pixel. This chapter investigates the problem of fusing the input images to form a higher resolution output. In chapter 5 the effects of combining these two stages, with their inherent errors, will be considered.

As a result of registration, the position of every pixel in the ensemble, relative to the reference coordinates, is known to some accuracy. A compound image can be formed by mapping all of the input pixels (from all input images) onto the reference grid, an example of which is shown in figure 4.1. This compound image has a higher sampling density than the original images and is therefore capable of representing higher detail.

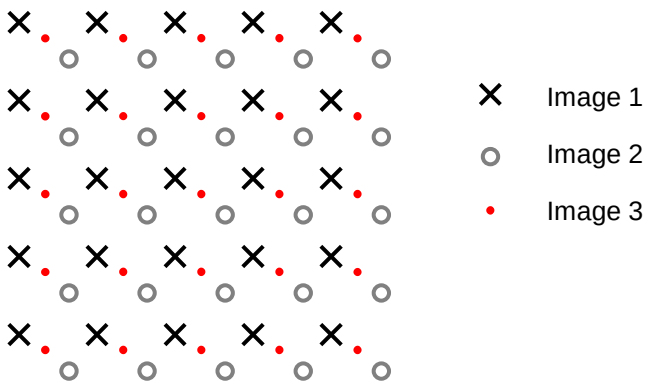


Figure 4.1 A non-uniformly sampled compound image formed by fusion of three images.

Although the compound image contains information from all of the input images, it has little use, as it is non-uniformly sampled. The compound image must be resampled onto a uniform grid to make it more useful. This resampling process involves interpolation of samples positioned at non-uniform locations. Unfortunately, the well established methods that dominate the image interpolation field, such as piecewise polynomial methods (bilinear, cubic convolution, cubic spline, etc.) or Lanczos filters, cannot be directly applied to non-uniformly sampled data.

This chapter extends the novel image interpolation method proposed in chapter 2 to interpolation of non-uniform data and investigates the performance of model-based near-optimal interpolation in comparison to standard methods of scattered data interpolation.

## 4.1. Non-uniform Interpolation and Resampling

Standard methods of resampling uniform images are specifically designed for a uniform sampling or based on separable (and also uniform) 1D interpolation. Neither case can be applied to resampling of non-uniform images. However, the idea of resampling using linear filters, where the output is a linear combination of input pixel values weighted by filter coefficients, is still valid and widely used. With non-uniform sampling, each output point has to be considered separately, as the number of input points in its vicinity and their relative positions will usually vary from output point to output point. However, in section 4.2.1 it will be shown how, in the special case of global translational motion, an efficient implementation can be achieved.

Reconstruction of the underlying surface by synthesis from basis functions (in a similar manner to uniform interpolation, as described in chapter 2) is possible; however, the basis functions must be two-dimensional. These are positioned at the locations of input samples and weighted by the sample value. The interpolated surface is constructed by simply summing these weighted basis functions and resampling at the output grid locations. A variety of methods that do this will be described in section 4.3.1.

## 4.2. Near-optimal Interpolation

Optimal interpolation uses the input and desired output pixel values to derive a resampling filter that is optimal in a least-squares sense. This procedure requires the output to be known beforehand. Near-optimal interpolation, however, relaxes this requirement by employing an image model to overcome the lack of knowledge of the desired output. This section shows how the same concept may be used for resampling of non-uniformly sampled images.

Consider interpolating a single output point located at $( \ , \ )$. The region of support around the output point may be defined in two ways. All input points that fall within a certain distance of the output point can be considered to influence that output

point. Alternatively, the $R$ closest input pixels can be considered to influence the output. The second approach, adopted here, is to retain explicit control over how many input samples are weighted to predict the output.

The value of the output point $g(\ ,\ )$ is computed using a weighted combination of the values of the $R$ closest input pixels:

$$g(\ ,\ ) = \sum_{i=0}^{R-1} f(x_i, y_i) h_i \,, \tag{4.1}$$

where $f(x_i, y_i)$ is the value of input pixel located at $(x_i, y_i)$ and $h_i$ is the filter coefficient that weights it. The filter order $\Theta$ is defined as the number of coefficients minus one (one degree of freedom is required for the flat-field response constraint). An index, $i$, is assigned to each input pixel within the region of support in the order of their distance away from the output, with $i = 0$ allocated to the closest input sample.

The closed-form solution derived in section 2.6 for uniform sampling can also be used here to compute the weights $h_i$ for the non-uniform sampling case. Recall equation (2.27), which expressed the error term to be minimised with respect to the filter coefficients. Even though the sample positions $(m, n)$ were integers with uniform sampling, there is no such requirement as far as the minimisation is concerned. Integer coordinates can be replaced with real coordinates $(x, y)$, resulting in:

$$E = \iint \left( \bar{M}(\ ,\ ; p_x, p_y) - \sum_{i=1}^{\Theta} \bar{M}(x_i, y_i; p_x, p_y) h_i \right)^2 dp_x dp_y \,, \tag{4.2}$$

where $\bar{M}(x_i, y_i; p_x, p_y) = M(x_i, y_i; p_x, p_y) - M(x_0, y_0; p_x, p_y)$ and $i$ is the index that was allocated to the sample at $(x_i, y_i)$. The filter coefficients may be found by minimising this expression in exactly the same way as equation (2.27) was minimised. The closed-form solution to this problem (assuming $\mathbf{A}$ is invertible) may be expressed in matrix form as:

$$\mathbf{h} = \mathbf{A}^{-1}\mathbf{b} \,, \tag{4.3}$$

where $\mathbf{A}$ is a $\Theta \times \Theta$ matrix of the form:

$$\mathbf{A}=\begin{bmatrix} \Psi(x_1,y_1,x_1,y_1) & \cdots & \Psi(x_\Theta,y_\Theta,x_1,y_1) \\ \vdots & \Psi(x_i,y_i,x_j,y_j) & \vdots \\ \Psi(x_1,y_1,x_\Theta,y_\Theta) & \cdots & \Psi(x_\Theta,y_\Theta,x_\Theta,y_\Theta) \end{bmatrix} \tag{4.4}$$

and $\mathbf{b}$ is an $\Theta\times 1$ vector:

$$\mathbf{b}=\begin{bmatrix} \Psi(\ ,\ ,x_1,y_1) \\ \vdots \\ \Psi(\ ,\ ,x_j,y_j) \\ \vdots \\ \Psi(\ ,\ ,x_\Theta,y_\Theta) \end{bmatrix}, \tag{4.5}$$

where $\Psi(x_1,y_1,x_2,y_2)$ is defined in expressions (2.35) and (2.37).

Consider the example pictured in figure 4.2, where an output pixel located at $(\ ,\ )$ is interpolated using a $5^{\text{th}}$ order filter ($\Theta=5$). The six closest points are indexed from zero to five in order of their proximity to the output, with $i=0$ allocated to the closest input sample. Elements of the five-by-five matrix $\mathbf{A}$ and five-by-one vector $\mathbf{b}$ are computed by substituting the corresponding values of $x$ and $y$ into $\Psi(x_1,y_1,x_2,y_2)$. Five of the filter coefficients are then computed using (4.3) with the remaining coefficient, $h_0$, computed using:

$$h_0=1-\sum_{i=1}^{5}h_i\,. \tag{4.6}$$

Finally, the filter coefficients are used to weight the values of the corresponding input pixels, using equation (4.1), to give the value of the output. The procedure is very simple and the search for the $R$ closest points can actually overpower the computation of the filter in terms of complexity in some situations, even when fast search techniques, such as the k-d tree [129] are employed.

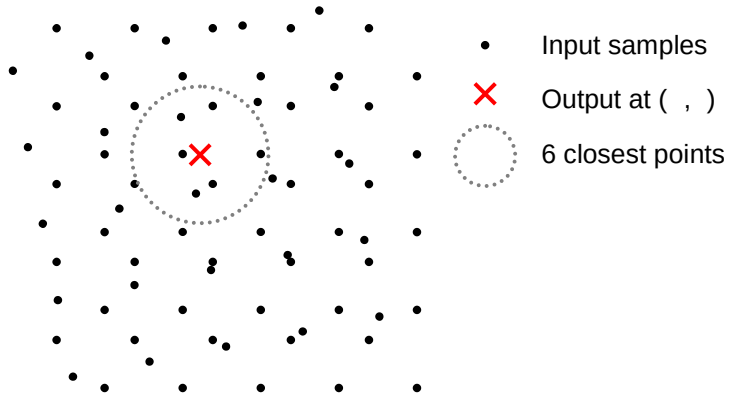


Figure 4.2 Resampling of a non-uniformly sampled image.

### 4.2.1. Special case – global translational motion

The assumption of global translational motion has two advantages in relation to image fusion. Firstly, it allows fast implementation using finite impulse response (FIR) image filters applied to input images. This removes the need for new data structures to represent the non-uniformly sampled compound image. Uniformly sampled input images are fast and easy to filter and the resampled compound image is simply a sum of interleaved filtered input images.

The second advantage lies in the fact that the above filters are global. With simulated data, where the desired output is known, the optimal global filters can be determined. This enables the performance of the proposed method to be benchmarked against these optimal filters.

#### (i) Efficient implementation

In the case of a global translational motion model, the resampling process can be implemented as a number of convolutional filters if the high-resolution sampling rate is an integer factor of the input sampling rate.

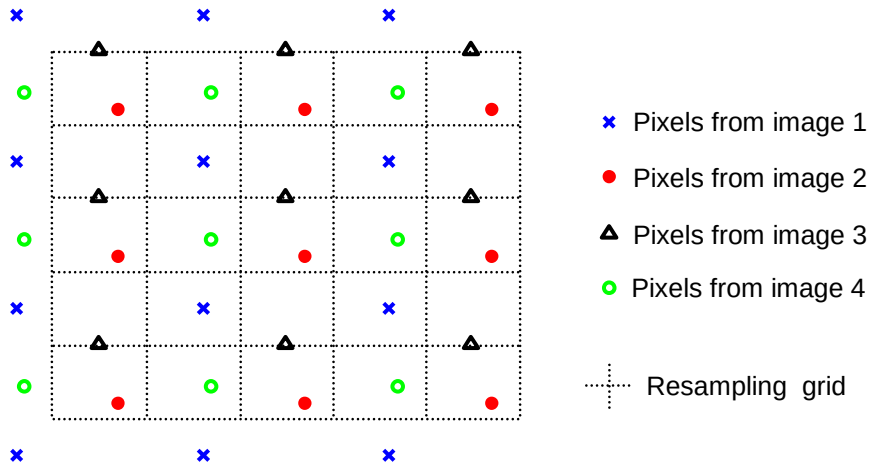


Figure 4.3 Fused image under global translational motion model. Output image pixels are located at intersections of dotted lines.

The distribution of the input pixels in the compound image exhibits a periodic pattern. Consider the example shown in figure 4.3. The pictured compound image was formed through fusion of four translated input images, each with sample rate *f*. It can be clearly seen from the figure that there is a pattern of input pixels that has the same periodicity as the input images. The output sampling grid is shown with dashed lines (the output pixels are located at the intersections of these lines). In this example, the resolution enhancement factor is two; hence the output grid has a sample rate of 2*f*.

The output sampling grid may be split into four sub-grids $A$, $B$, $C$ and $D$, such that each one has the same sample rate as the input images (see figure 4.4). For each sub-grid, the distribution of input pixels around each output point on the grid is the same (apart from image boundaries). Because the filter coefficients for the model-based filters depend on the relative distances between the input pixels and the output, the output pixels belonging to each sub-grid can be resampled with a single filter.

The four sub-grids require one resampling filter each, requiring equation (4.3) to be computed four times. Each one of these filters weight input pixels from the four input images. This means that each filter can be decomposed further into four regular filters applied to uniform input images, resulting in $4 \cdot 4 = 16$ filters. For the case of the 3rd order 'non-uniform' filter shown in figure 4.4, each regular filter has only a single coefficient (i.e. is just a weighted input image)! The output image is formed simply by interleaving the four sub-images.

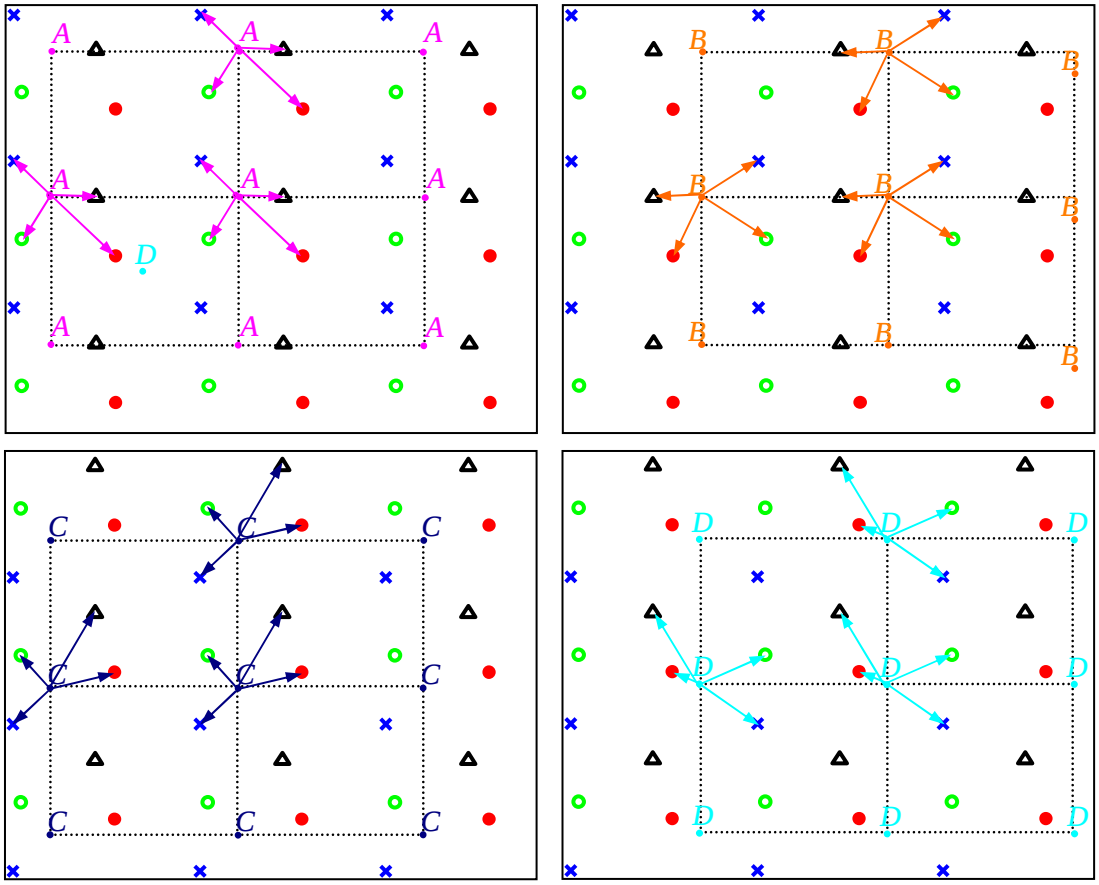


Figure 4.4 The output sampling grid can be broken down into four sub-grids, each with the same sample period as input images. Arrows indicate four nearest input pixels for each output point.

In the general case, given $L$ input images and integer resolution enhancement factor $N$, there will be $N^2$ filters, one for each sub-grid. Each one of these $N^2$ 'non-uniform' filters can be decomposed and implemented as the sum of results of $L$ regular filters applied to the input images. The whole procedure requires $N^2L$ regular filters applied to the input images. The average order of these filters is $R/L$ and since $R$ is small, these filters are even smaller.

### (ii) Optimal filter coefficients

In the case of simulated data, the desired output is known and can be used to compute the optimal resampling filters of a given order, in a similar manner to uniform optimal interpolation described in section 2.3. The output image is comprised of $N^2$ sub-images, each with the same sampling rate as the input images. Each sub-image is resampled using its own filter. Each one of these filters is optimised by minimising the square error between the resampled sub-image and the ground truth values with respect to the filter coefficients. These optimal filters yield the smallest

mean square error out of all linear methods that use the same configuration of input pixels.

### 4.2.2. Experiment VII – Optimal and near-optimal filters

***Aim***

To examine the performance of the proposed method of model-based near-optimal interpolation for different filter orders and compare to the performance of optimal resampling filters experimentally.

***Method***

The quality of the resampling of the compound image depends on many factors: the method, the image content, the number of low-resolution images and their relative positions, the resolution enhancement factor, the position of the output grid, the sub-pixel registration accuracy and, of course, the level of noise.

The resolution enhancement factor and the motion model are generally determined by the application. Here, it is assumed that the high-resolution image is reconstructed from four translated low-resolution images and the resolution enhancement factor is two. In a similar manner to previous experiments, the low-resolution input images and the high-resolution ground truth image were generated using an area-sampling imaging model from a source image of much higher resolution.

A selection of test images with different characteristics is used in this experiment to make sure that the results can be generalised over a wide range of images. These images have already been used in chapter 2 and are shown in figure 2.2. The source images are 2592×3888 pixels and a downsampling factor of 20 is used to generate the low-resolution input images, resulting in 125×190 input images and a 250×380 ground truth high-resolution image. Portions of low and high-resolution versions of each image are shown in figure 4.5. The scale of image text is selected such that the text is unreadable in a low-resolution image, but can be read in a high-resolution image. The scale of the other images was chosen to have visible aliasing in the low-resolution images. To assess the accuracy of resampling alone, the noise variance is assumed to be negligible and the registration parameters known *a priori*.

Figure 4.5 Cropped parts of low- and high-resolution versions of images ‘bird’, ‘cat’, ‘face’ and ‘text’.

The positions of the input images relative to the output grid play an important role in the quality of reconstruction. It can, for example, happen that the input pixels are positioned exactly over the location of the output, as pictured in figure 4.6, making the resampling trivial. Alternatively, two or more images can happen to have exactly the same offset, or all four images can be offset only in one direction, making the resampling much harder. To measure the performance over a wide range of such

situations, it is proposed to reconstruct the high-resolution image from many random combinations of input low-resolution image offsets.

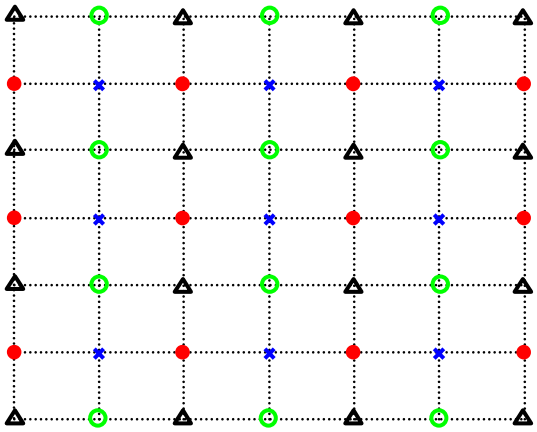

Figure 4.6 The fused image does not require any interpolation, as the input samples fall on the positions of the output samples.

The image formation model allows for 400 sub-pixel displacements (a downsampling factor of 20 in each direction). There are approximately ${}^{400}C_4 = 10^9$ possible combinations (without repetitions) of four input images' sub-pixel displacements. This is an unreasonably large number of simulations to perform. A Monte Carlo simulation is performed instead, with a thousand random samples being drawn from the population of all possible combinations of offsets. Four low-resolution images with selected offsets were generated and then fused together to form a compound image, which is in turn resampled on a uniform grid with a sample rate double that of the input images. The root mean square error was computed between the reconstructed high-resolution image and the ground truth. The same one thousand combinations of offsets were used for both methods. This general set up, apart from the resampling methods, will be employed in all further experiments in this chapter.

### *Results*

Inverse cumulative distribution density function (iCDF) plots [130] are used to interpret the results. These plots are also known as percent point function or quantile function plots. They make it easy to compare the statistical distributions of measured errors between different methods. The plots are formed simply by ranking the errors measured from the 1000 selected combinations of offsets in ascending order and plotting these against probability ranging from zero to one. The lower on the *y*-axis a line on the plot is, the better that method performs. The flatter a line, the smaller the

variance of error distribution for that method. The 50$^{th}$ percentile, or median, is used to judge the performance of a given method on average.

Figures 4.7 to 4.10 show the result of applying (a) optimal filters and (b) model-based near-optimal filters of orders two to 14. These graphs are a good demonstration of why non-uniform interpolation quality should not be judged based on a small number of results. The distribution of errors can be quite spread out, and it also changes significantly as the filter order is increased. For low orders, the variance of error distribution is much larger than for filters of order 7 and higher. The improvement in RMSE at the lower end of the scale is noticeably less than the improvement at the higher end of the scale for all order filters.

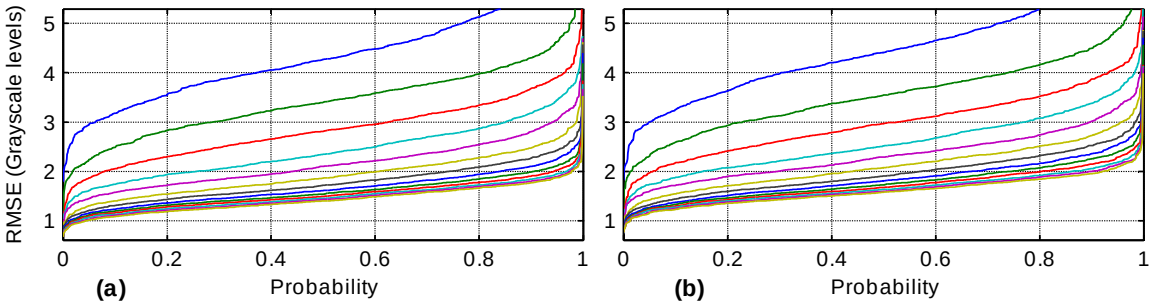


Figure 4.7 iCDF plots using image Bird and filter orders two to 14: (a) optimal (b) model-based. Top blue line is order two, green line below is order three, red line below that is order 4 and so on to order 14 right at the bottom.

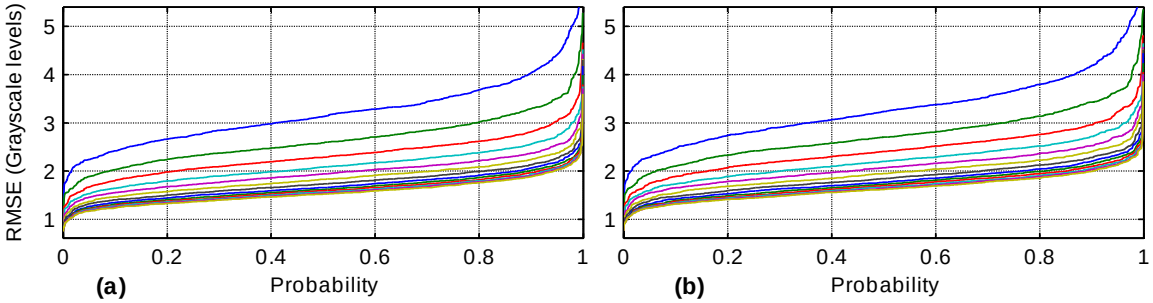


Figure 4.8 iCDF plots using image Cat and filter orders two to 14: (a) optimal (b) model-based.

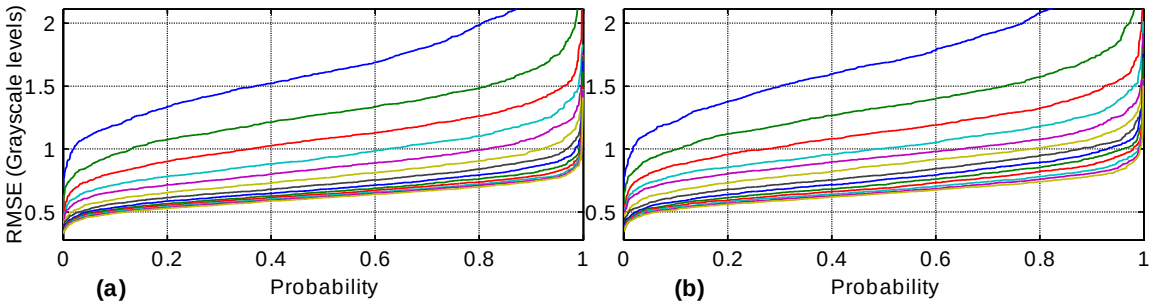


Figure 4.9 iCDF plots using image Face and filter orders two to 14: (a) optimal (b) model-based.

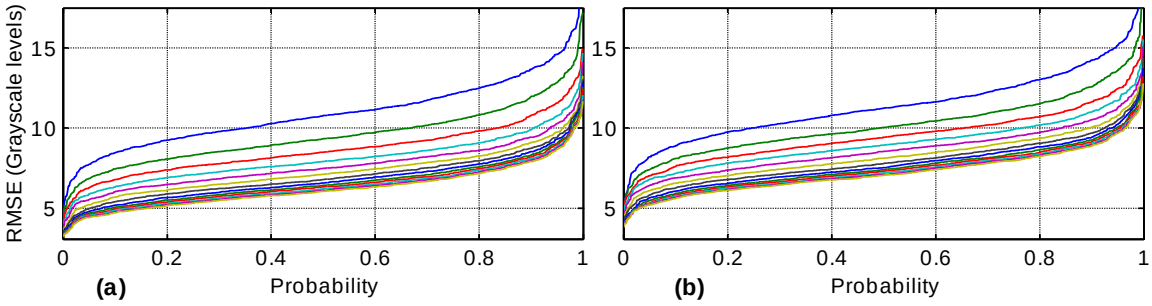


Figure 4.10 iCDF plots using image Text and filter orders two to 14: (a) optimal (b) model-based.

From these graphs it can be seen that optimal and near-optimal error distributions are very similar in terms of the shape and general variability. However, for a closer comparison of how the results compare figure 4.11 shows the median performance of optimal and near-optimal filters applied to the four images. This is equivalent to the RMSE reading from the previous graphs at $P = 0.5$.

It can be seen that the performance of model-based near-optimal interpolation follows closely the performance of optimal interpolation for all images except for 'text'. For image 'text' there seem to be a small mismatch between the model and the image. The improvement in performance from using a higher order resampling filter diminishes with increasing order. For all four images, the improvement after order 12 is relatively small.

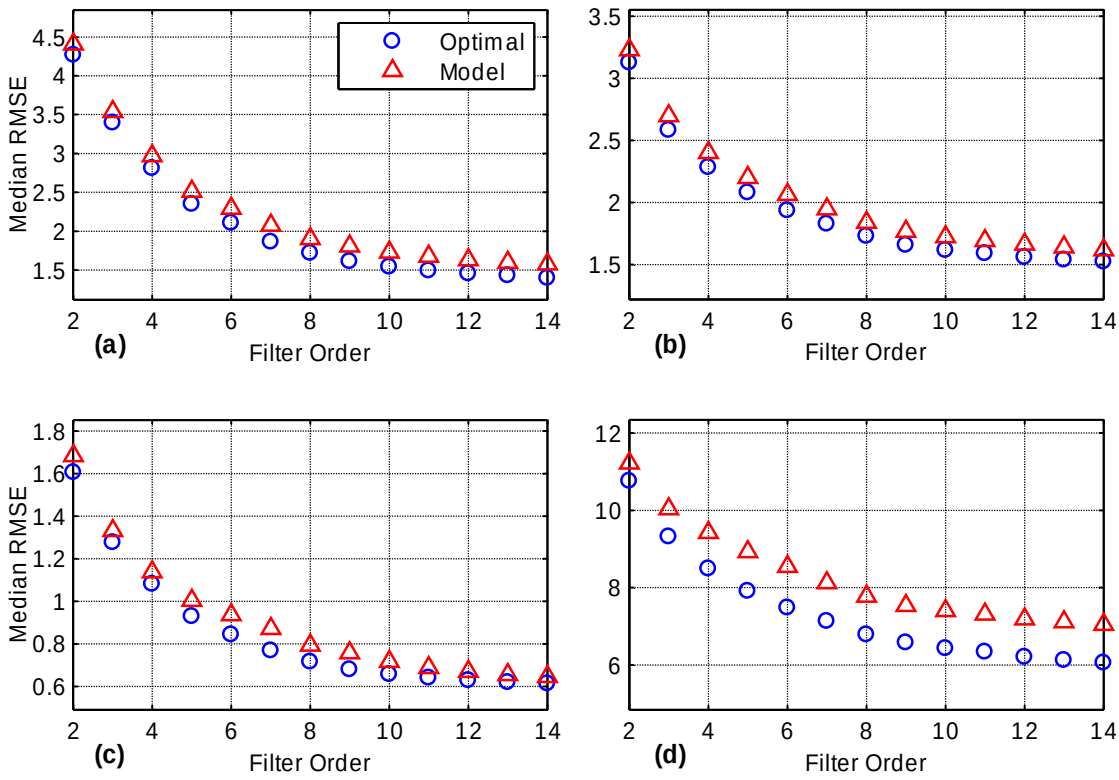


Figure 4.11 Median RMSE vs Filter order for images (a) 'Bird' (b) 'Cat' (c) 'Face' (d) 'Text'.

### *Conclusions*

Resampling filters based on the proposed rectangular pulse model produce results close to that of an optimal linear filter, except for image text. Increasing filter order yields improved results; however, this improvement diminishes as the order goes up.

The average extent of the region of support is estimated to relate these findings to the other methods. This is estimated based on the average distance from the output to the furthest input pixel used in the filter over all 1000 input image configurations. Figure 4.12 shows a plot of this average distance versus the filter order. A $10^{th}$ order filter, for example, has 11 closest input points in its window. The region of support for this filter is approximately two (high-resolution) pixels, because a filter with region of support radius equal to two would include 11 nearest input points on average for the 1000 offset combinations.

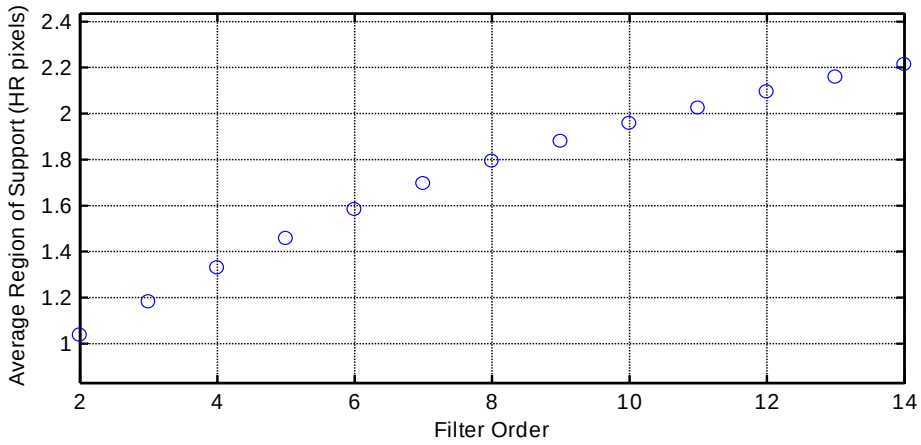


Figure 4.12 Average region of support for filter orders 2 to 14 in the case of test offsets.

## 4.3. Performance of Resampling Methods

In this section, the performance of the proposed method for resampling a non-uniform compound image is assessed experimentally against a selection of existing linear methods for interpolation of scattered image data with a focus on methods that are suitable for fast image super-resolution. Root-mean-square error is used as an objective measure of quality of interpolation. The performance is also benchmarked against optimal interpolation in the presence of noise.

### 4.3.1. Methods for comparison

A scattered distribution of pixels in a compound image is generally not separable in the *x*- and *y*-directions. Hence, it is impossible to adopt 1D interpolation kernels to this resampling problem. One exception to this is the nearest neighbour interpolator. For each output pixel, the nearest neighbour method uses the value of closest input pixel, resulting in very fast resampling. The quality of this resampling, however, is rather mediocre. A number of popular methods that are appropriate for fast implementation are briefly described in this section. All of the methods, except adaptive normalized convolution, are implemented in an efficient manner, as described in section 4.2.1(i), using regular filters, when applied to uniform input images.

#### (i) Inverse distance weighted, or Shepard methods

A linear combination of a number of input pixels must be used to interpolate a given output pixel with an acceptable result. The question is how to weight the values of these pixels. One simple solution is based on the idea that the interpolated value

must depend more on nearby pixels than those far away. The input pixels are weighted by an inverse function of their distance away from the output; hence, the name of the method – inverse distance-weighted interpolation. Sometimes it is also referred to as the Shepard method, after the first contributor [131].

Many different inverse distance functions have been proposed [68]; some of these are shown in figure 4.13, where $d$ is the Euclidean distance:

$$d(x, y) = \sqrt{x^2 + y^2} \,. \tag{4.7}$$

These examples are given to demonstrate the trade-off between data conformity and noise averaging. The simple inverse distance functions in the first row go to infinity at $d = 0$; this defeats any attempt to average out the noise. The functions in the second row go to one at $d = 0$ to combat this. Unfortunately, as the Shepard method is a global method, the points far away can collectively outweigh one or two important points close to the output, because the weighting function never goes to zero for large $d$. The functions in the third row also never go to zero, but drop off to insignificant levels fairly quickly. The size of the exponent in each case is selected based on the size of detail in the image and amount of averaging desired. If the function is too narrow, little noise averaging is performed. If the function is too broad, the result is overly blurry.

$$\begin{array}{ccc} \dfrac{1}{d_k} & \dfrac{1}{d_k^2} & \dfrac{1}{d_k^3} \\ \dfrac{1}{(d_k+1)} & \dfrac{1}{(d_k+1)^2} & \dfrac{1}{(d_k+1)^3} \\ e^{-d_k} & e^{-d_k^2} & e^{-d_k^3} \end{array}$$

Figure 4.13 Some examples of inverse distance weighting functions.

A modified version of the Shepard method only uses input points in close proximity to the point being interpolated, taking into account that many datasets have only limited spatial correlation. This is normally implemented using a piecewise weighting function, for example [132]:

$$w_k = \left( \frac{[R - d_k]_+}{R d_k} \right)^2, \tag{4.8}$$

where $w_k$ is the weight allocated to $k^{\text{th}}$ point, $R$ is the radius of the region of support, $d_k$ is the distance to $k^{\text{th}}$ point from the output and:

$$\left[R-d_k\right]_+ = \begin{cases} R-d_k & d_k < R \\ 0 & d_k \geq R \end{cases} \tag{4.9}$$

This modification reduces the computational complexity of the method and at the same time makes sure that a cluster of points far away cannot outweigh important points near the interpolated point. The modified method with $R = 2.5$ (high-resolution pixels) is implemented for comparison here. It is labelled as **'**IDW'.

In addition to the problems outlined above, the Shepard method is only sensitive to the distance of the input points from the output, not the actual location of the inputs. This enforced radial symmetry may result in poor interpolation of features like edges, ridges and planes.

### (ii) Triangulation methods

An alternative to using a linear combination of nearby points is to triangulate the data and fit some function to the triangular patch that the output point is in. Firstly, it must be decided how exactly the data is triangulated. Obviously there are many different ways to triangulate a given dataset. Delaunay [133] triangulation is widely used for its so called maxmin angle property [134]. Out of all possible triangulations of a given dataset, Delaunay triangulation will result in a set of triangles with maximum minimum angles. This is beneficial, as some researchers suggest that thin narrow triangles must be avoided for interpolation [26]. Once the data is triangulated, a surface must be fitted to the triangular patches. With three constraints (one from each vertex values), an obvious function to fit to a patch is a plane. A planar fit results in a surface that is $C^0$ continuous, but not $C^1$ continuous (smooth) – the surface is continuous across the boundaries of the triangles, but the derivatives are not.

Smoothness can be achieved by fitting a higher order surface. A popular method uses the Clough-Tocher [135] triangular interpolant to achieve a surface with $C^1$ continuity. This method is described in detail by Amidror [136]. A slightly different approach is implemented here, which was used in the context of super-resolution by Lertrattanapanich and Bose [26].

Instead of fitting a full bivariate cubic polynomial like the Clough-Tocher method, the authors leave the uniform twist monomial $xy$ out and fit the remaining terms:

$$z(x,y) = c_1 + c_2x + c_3y + c_4x^2 + c_5y^2 + c_6x^3 + c_7x^2y + c_8xy^2 + c_9y^3 . \tag{4.10}$$

The authors justify the omission of the $xy$ term (over other terms) by the fact that omission of this particular term maintains geometric isotropy. The remaining nine degrees of freedom require nine constraints. Three constraints from each vertex are used: the value and the $x$ and the $y$ components of the normal vector. Because there are no constraints along the connecting edges of the triangular patches, the resulting surface is continuous, but not smooth like the Clough-Tocher method. The Lertrattanapanich and Bose method is, however, less computationally expensive.

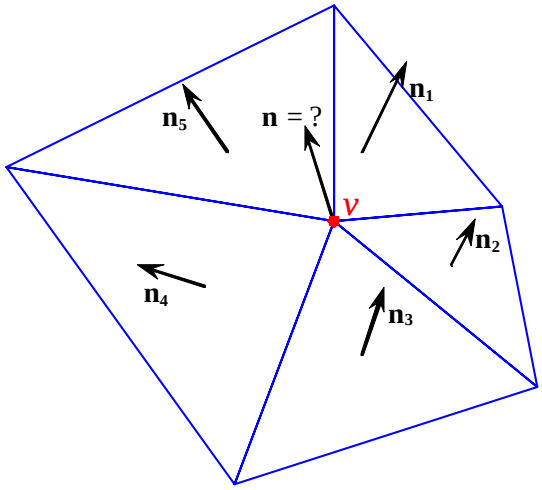


Figure 4.14 The normal vector at vertex $v$ is estimated as a weighted sum of normal vectors of patches around it, where the weight is determined by the triangle area [26].

The normal vectors at each vertex must be approximated from the pixel values. To take advantage of multiple low-resolution images available, these are computed from the compound image in the following way. To estimate the normal vector at a vertex $v$, each triangle around it is fitted with a planar patch and the normal vectors of these patches are computed, as shown in figure 4.14. These are weighted by the area of corresponding triangles, summed and normalised by the total area to give an approximate normal vector at vertex $v$:

$$\mathbf{n} = \frac{\sum_{k=1}^{K} A_k \mathbf{n}_k}{\sum_{k=1}^{K} A_k}, \tag{4.11}$$

where $A_k$ is the area of $k^{th}$ triangular patch and $K$ is the total number of neighbouring triangles.

Several algorithms for constructing triangulations exist. The method implemented here is based on triangulation using the Quickhull algorithm [137]. Both planar patches and the method by Lertrattanapanich and Bose are implemented for the purposes of comparison and are labelled 'Tlin' and 'Tcub' respectively.

Even though the Delaunay triangulation attempts to make the triangles as close as possible to equilateral, this is not always possible [138]. This is a big disadvantage for triangulation methods, as normal vectors estimated from narrow triangles may be sensitive even to small amounts of noise.

### (iii) Local polynomial approximation

Another popular alternative to using inverse distance weighting that does not depend on triangulation is to model the image around a certain point by performing a local polynomial expansion around that point. This was first proposed by Haralick [139] and dubbed the image facet model. Farnebäck [140] extended this idea by incorporating a Gaussian weighting function, essentially combining polynomial fitting with inverse distance weighting, integrating the idea that closer points are more highly correlated.

Van de Weijer and Van den Boomgaard [141] extended previous work by replacing the quadratic norm with a robust Gaussian norm and solving for coefficients iteratively. Pham et al. [27] were the first to apply this method to the reconstruction of an image from non-uniformly sampled data and incorporated local adaptivity by considering image gradients.

Two polynomials are used here for comparison, a quadratic ( $z_1$ ) and a cubic ( $z_2$ ):

$$\begin{aligned} z_1(x,y) &= c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 \\ z_2(x,y) &= c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3 \end{aligned} \tag{4.12}$$

For any output point, all input points within some predefined region of support are used to fit these bivariate polynomial functions using weighted least-squares. The contribution of each point in the least-squares fit is weighted using the following inverse distance function:

$$w_k = d_k^{-} \cos\left(\frac{d_k}{2R}\right), \tag{4.13}$$

where $d_k$ is the distance to $k^{\text{th}}$ point, $R$ is the radius of region of support, and $=2$ and $=2$. The method using the quadratic polynomial is labelled 'quadratic' and the method using the cubic polynomial is labelled 'cubic'. Both of these use $R$ equal to four high-resolution pixels.

Even though the primary interest of this work is fast linear methods, adaptive normalised convolution is implemented for comparison. The method employs the quadratic polynomial $z_1$; however, the region of support and the weighting function is adaptively varied to give more weighting to pixels situated along the edges. The details of this method can be found in [27]. This method is labelled 'adaptive'.

### 4.3.2. Experiment VIII – Comparison to existing methods

***Aim***

To assess the performance of near-optimal interpolation against a selection of existing linear methods for interpolation of scattered image data with a focus on methods that are suitable for fast image super-resolution.

***Method***

The general experimental set up is the same as in experiment VII. The performance of near-optimal interpolation is compared to that of the other methods, described in section 4.3.1. A $10^{\text{th}}$ order near-optimal filter (Model 10) was selected for comparison, as it seems to offer a good trade-off between the quality of interpolation and computational complexity.

***Results***

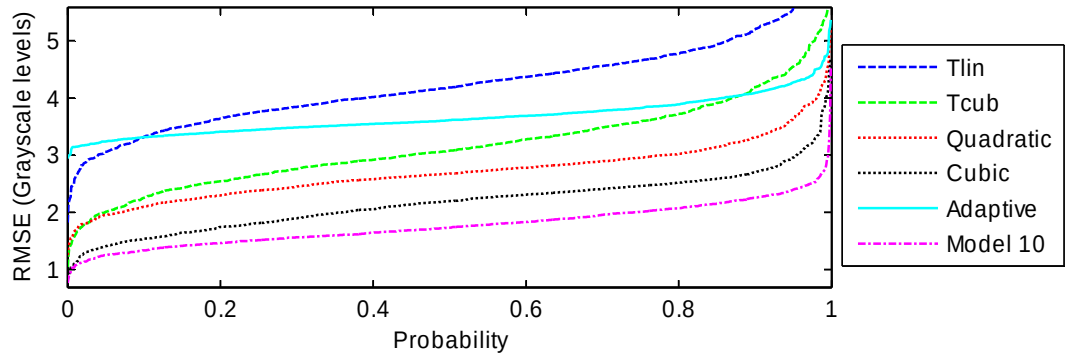


Figure 4.15 iCDF curves for image Bird.

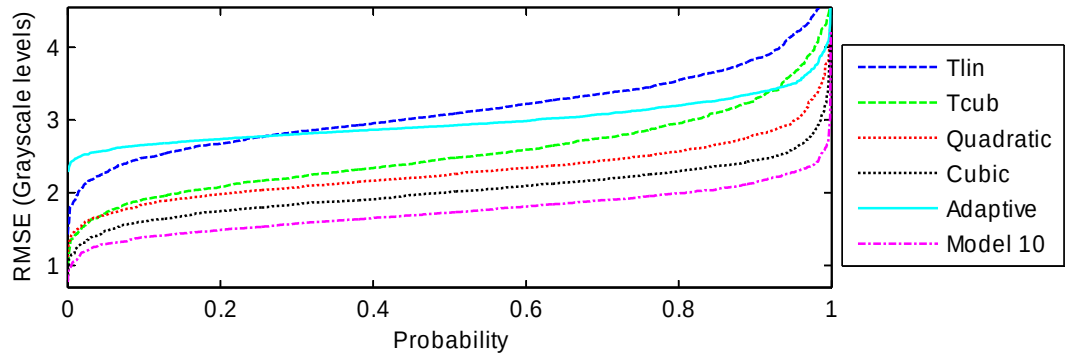


Figure 4.16 iCDF curves for image Cat.

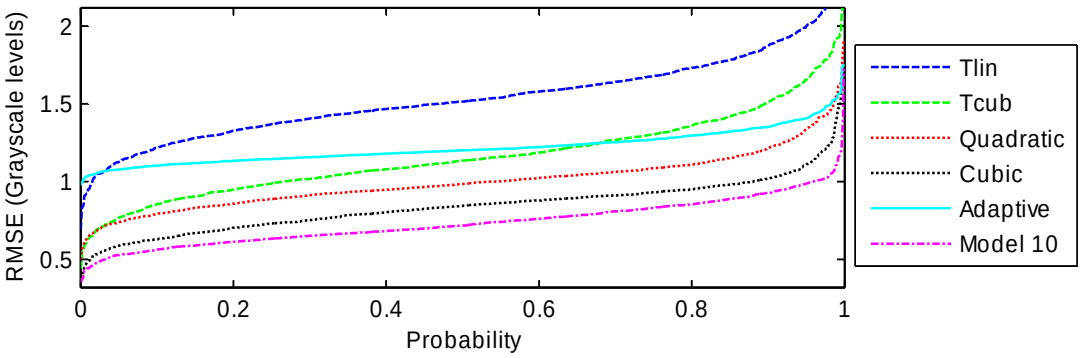


Figure 4.17 iCDF curves for image Face.

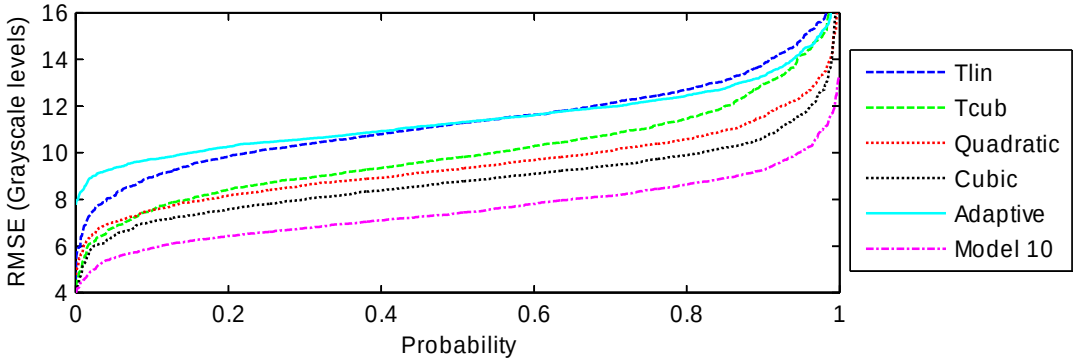


Figure 4.18 iCDF curves for image Text.

Figure 4.15 through to 4.18 show the iCDF plots of the results for each image. As expected, the triangulation methods did not perform as well as polynomial expansion methods. The planar fit also performed significantly worse than the cubic fit. The inverse distance weighted method performed significantly worse than other methods for all four test images and was left out of the comparison.

The adaptive quadratic method did not perform as well as expected. For every image, the results are worse than that of non-adaptive quadratic polynomial expansion. The reason for this could lie in the fact that the adaptive method requires an estimate of the image gradients. These may not be reliably estimated, as the low-resolution images are highly aliased. However, the adaptive method has shown to have lowest error variability for all images.

#### *Conclusions*

The near-optimal filter of order 10 outperforms all existing methods used for comparison for all four test images. Apart from the triangulation method with a planar fit, the near-optimal filter also has lowest computational complexity.

## *4.4. Effect of Noise on Near-optimal Interpolation*

The resampling process so far has been considered in the absence of noise. However, all real images contain some noise. This section explores the effect of noise on model-based near-optimal interpolation.

In the presence of noise it makes less sense to uphold the interpolation constraint of, equation (2.2) because it is known that the input data is noisy, so the data points will not coincide with the true surface. Methods that relax this constraint and apply some smoothing to filter out the noise are sometimes dubbed approximation, rather than interpolation. If the constraint is relaxed too much, the resulting image will be overly smooth. If the constraint is enforced too strongly, the resulting image will fit the noise, and consequently suffer from noise artefacts. It is desired to find the optimal trade-off between noise smoothing and data conformity.

Optimal interpolation does not enforce constraint (2.2) explicitly; it is inherited from the data itself (in the absence of noise). In the presence of noise, however, the interpolation constraint is automatically relaxed to minimise the square error. For noisy inputs, the optimal filter coefficients offer the ‘optimal’ amount of smoothing, because they result in the smallest possible square error for a given filter order. In

section 3.3.3 it was shown that in the presence of additive white Gaussian noise, an additional noise term added to the elements of matrix $\mathbf{F}^T\mathbf{F}$ and vector $\mathbf{F}^T\bar{\mathbf{g}}$. The effect of this as the noise increases is for the optimal filter coefficients to tend towards $1/R$, which is the optimal filter to average out the noise.

The same mechanism of noise reduction can be offered with model-based near-optimal interpolation. The elements of matrix **A** and vector **b** in equations (4.4) and (4.5) must be modified to include an additive term $2\ ^2$ for diagonal elements of **A** and $^2$ for non-diagonal elements of **A** and the elements of **b**. can be thought of as a smoothing parameter that offers a trade-off between conformity to the signal and smoothness of the output. When $=0$ there is no smoothing; as is increased, the amount of smoothing is increased and the filter approaches a mean of the pixels within the region of support. This is very similar to the concept of regularisation. In section 3.3.3, it was shown that for an optimal filter the value of that offers the 'optimal' amount of smoothing is equal to the noise standard deviation.

For model-based optimal interpolation, the situation is more complex. The optimal value of depends on the signal strength, as well as the noise. When a model is used to optimise the filter coefficients, the signal strength does not correspond to the image that will be resampled. Hence, to estimate the value of that results in optimal smoothing, both the signal and noise power must be estimated.

### 4.4.1. Experiment IX – Effect of the smoothing parameter on the performance of model-based near-optimal filters

#### *Aim*

To investigate the effect of varying the smoothing parameter on the performance of model-based near-optimal interpolation filters.

#### *Method*

The general experimental set-up is similar to the previous experiments. Low-resolution input images are generated in the same way from the same four source images. The same 1000 combinations of input image offsets are also used. The high-resolution output is reconstructed using model-based near-optimal interpolation from input images contaminated with additive white Gaussian noise with zero mean and

variance $^2$. The reconstruction is performed using $10^{th}$ order filters derived from the rectangular pulse model with the smoothing parameter set to various values between zero and 250. This is performed with the noise standard deviation ranging from zero to five.

***Results***

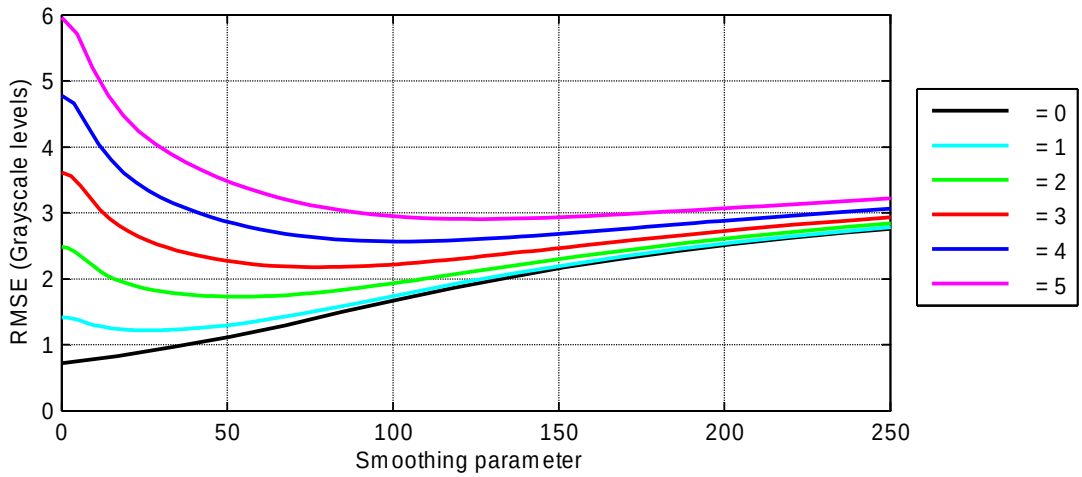


Figure 4.19 The effect of the smoothing parameter on median RMSE resulted from the use of near-optimal interpolation on image 'face'. Noise SD is in greyscale levels.

Figure 4.19 shows a plot of median RMSE versus the smoothing parameter for various noise levels obtained using image 'face'. As expected, the optimal value of depends strongly on the noise level. Setting to its optimal value can dramatically improve the error. Figure 4.21 shows the profile of image intensities from a cross section indicated by a white line in figure 4.20 for noise level $= 3$. Resampling with $= 0$ produces a noisy output, resulting in large error. Increasing increases the amount of smoothing. This reduces the error by averaging out some of the noise, but at the same time it increases the error by blurring sharp detail, such as between samples 40 and 50. This clearly shows the trade-off encountered with setting the smoothing parameter. The optimal value of is at the point where the error from noise is well balanced with the error from blurring the image, resulting in minimum overall error. Hence, the optimal value will depend not only on the noise power, but also on the content of the image, specifically the amount of sharp detail.

Figure 4.20 Crop of image ‘face’.

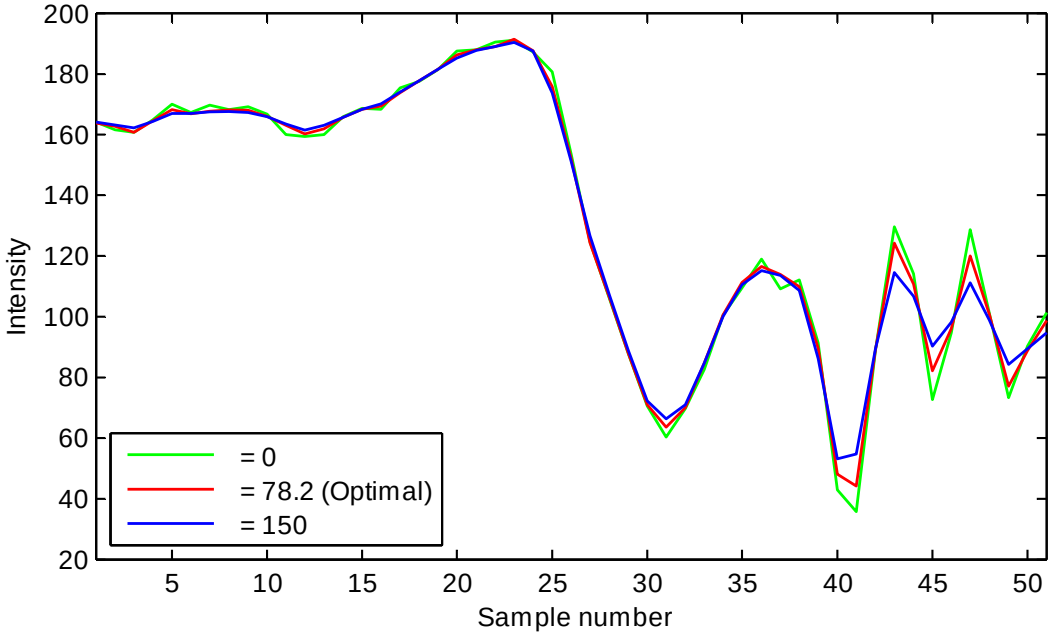


Figure 4.21 Cross section taken across image ‘face’, as indicated by a white line in figure 4.20.

In the absence of noise, setting $=0$ is best. For higher levels of noise, overestimating the smoothing parameter does not seem to have as much consequence as underestimating it. This is because the main source of degradation is noise and the optimal smoothing applies quite strong blurring. At this point, the high frequencies in the image are already highly attenuated and any additional blurring does not have much effect on the RMSE. Underestimating and applying less blurring, however, increases the amount of noise in the output image and consequently increases RMSE.

Results obtained from other images follow a similar pattern. The optimal values of are tabulated in table 4.1 for each noise level and each image under test. It is clear from these results that the optimal value of strongly depends on image content. When a filter is optimised using image data these factors are automatically taken into account. However, when a filter is optimised on the edge-based model, no information about the edge height or edge frequency is included.

Table 4.1 Optimal values of the smoothing parameter for selected noise levels.

| | = 0 | = 1 | = 2 | = 3 | = 4 | = 5 |
|---|---|---|---|---|---|---|
| 'bird' | 0 | 7.9 | 19.2 | 27.3 | 38.7 | 49.0 |
| 'cat' | 0 | 11.8 | 26.8 | 38.4 | 54.1 | 68.4 |
| 'face' | 0 | 27.2 | 53.7 | 78.2 | 101.2 | 126.5 |
| 'text' | 0 | 2.5 | 6.5 | 8.1 | 14.2 | 17.3 |

The optimal values of the smoothing parameter are plotted against the noise standard deviation in figure 4.22. The relationship between the optimal value of the smoothing parameter used with model-based filter optimisation and the input image noise level is approximately linear for any individual image. The slope depends on the content of each image, in particular how signal power in the image relates to that of the model.

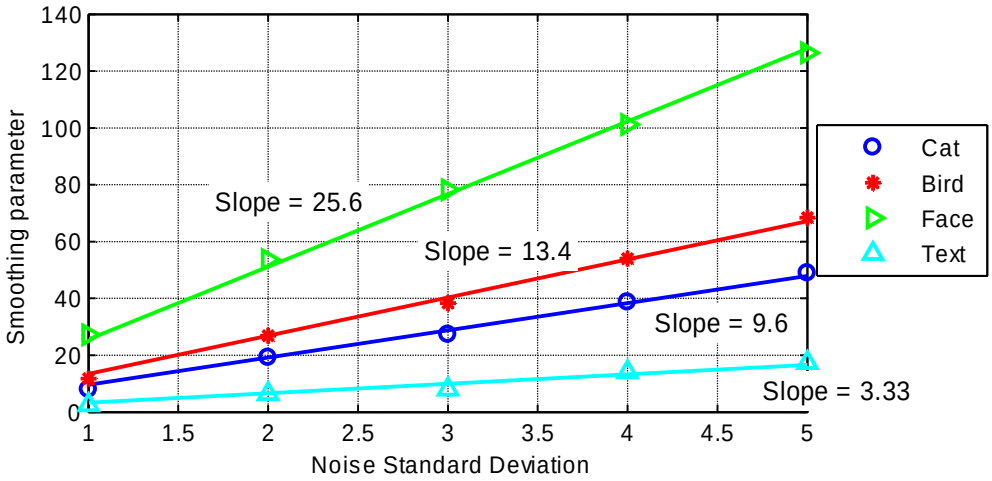


Figure 4.22 Optimal value of the smoothing parameter versus noise level for the four test images. Noise SD is in greyscale levels.

### *Conclusions*

The smoothing parameter can be used to adjust the amount of smoothing applied to the data. If too much smoothing is applied, the result is overly blurry. If not enough smoothing is applied, the result is overly noisy. The amount of smoothing depends on the image content, as well as the noise level. For a given image, the optimal value of the smoothing parameter is proportional to the noise standard deviation. However, the proportionality constant may be significantly different for different images.

This constant will depend on how the signal strength in the model relates to the signal strength in the image; hence, it may depend on the number of edges in the image, their sharpness and their amplitudes. Finding the value of this constant is

considered to be out of the scope of this thesis. However, the values obtained empirically by fitting a line to the data in figure 4.22 are used to demonstrate how well model-based near-optimal interpolation can perform in the presence of noise.

### 4.4.2. Experiment X – Effect of noise on performance

***Aim***

To assess the effect of noise on the performance of near-optimal interpolation.

***Method***

The general experimental set up is the same as in the previous experiment. The high-resolution output is reconstructed using $10^{th}$ order model-based near-optimal interpolation with the smoothing parameter determined from the linear relationship obtained in the previous experiment. The reconstruction error is compared with that of the $10^{th}$ order optimal filter. Noise with standard deviation between zero and 10 was employed in this experiment.

***Results***

Figure 4.23 shows how the performance of the proposed model-based filter compares to the performance of an optimal filter in the presence of various levels of noise. Figure 4.24 shows the difference in performance as a percentage of the optimal. The difference for image 'face' is very small. This can be attributed to the fact that the image is relatively easy to interpolate (lowest RMSE of all the test images), which means that noise is the primary degradation, rather than interpolation error. Both, optimal and near-optimal (with  set to optimal value) filters are equally good at attenuating the noise, resulting in very close RMSE.

In contrast to this, image 'text' is much harder to interpolate. The absolute difference between optimal and near-optimal stays relatively large throughout the noise range; however, the percentage difference decreases.

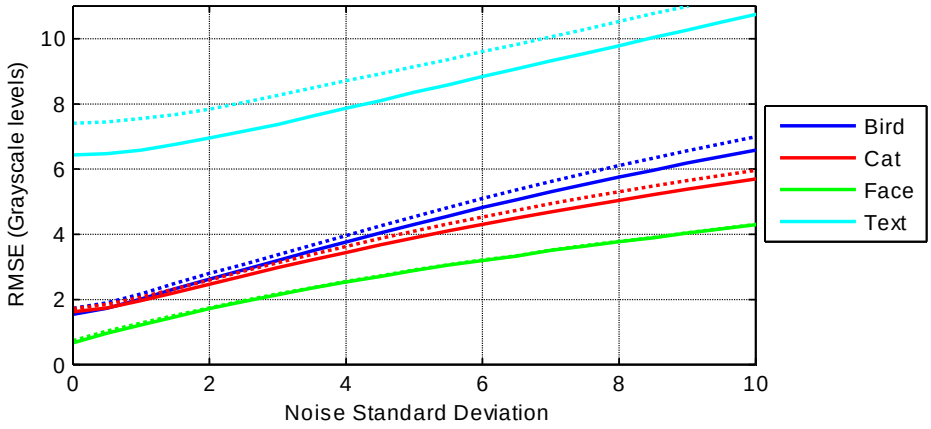


Figure 4.23 Median RMSE versus noise standard deviation for optimal and near-optimal (dotted line) filters of orders 10.

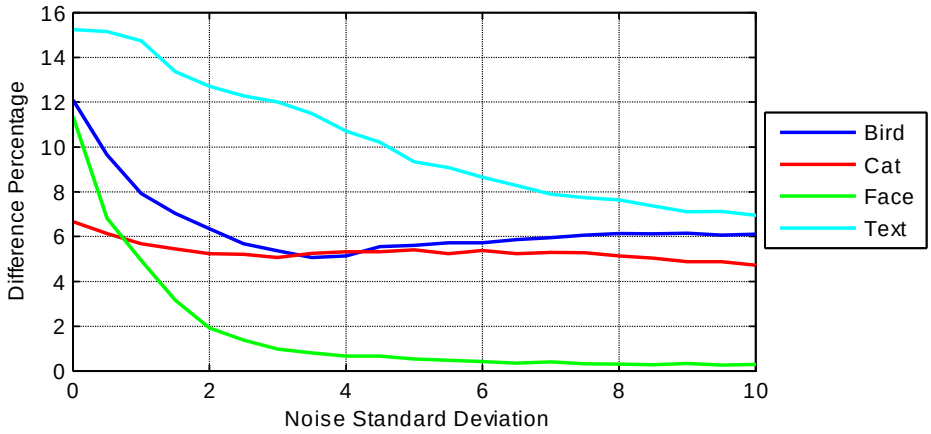


Figure 4.24 Difference between median optimal and median model-based near-optimal RMSE as a percentage of median optimal value for filters of order 10.

#### *Conclusions*

The proposed method can perform close to optimal (for that filter size) in the presence of moderate amounts of noise.

## *4.5. Summary and Conclusions*

Registered low-resolution input images can be fused into a compound, non-uniformly sampled image. The higher sampling density of the compound image allows it to represent higher detail; however, to make use of this, the image must be resampled on a regular grid. This problem is not trivial because the compound image is not uniformly sampled, nor it is separable; hence, traditional methods of image interpolation and resampling may not be applied.

The proposed method for performing this resampling is based on rectangular pulse model-based near-optimal interpolation, as described in chapter 2. This chapter has extended the ideas of model-based near-optimal interpolation to non-uniformly

sampled data. The proposed method offers a flexible trade-off between computational complexity and quality of reconstruction in the form of filter order. The performance of filters of order two to 14 was shown to follow closely to the performance of optimal resampling filters of the same order.

In an experimental test, the proposed method was compared to the performance of a number of linear methods from three classes popular in the literature: inverse distance weighted methods, triangulation methods and local polynomial expansion methods. Model based near-optimal filter of order 10 was shown to outperform all of these methods on all four test images.

When input images are contaminated by noise, the interpolation constraint can be relaxed to offer some noise smoothing. Optimal resampling filters achieve this automatically, as the filters are optimised on the data to yield the smallest square error. For model-based near-optimal interpolation it is also possible to offer some noise smoothing, controlled by a smoothing parameter. It was shown that given a good estimate of the smoothing parameter, the model-based near-optimal interpolation method can achieve noise smoothing similar to that offered by an optimal resampling filter.

# Chapter 5

# Image Super-resolution

A novel image registration technique and a novel image fusion and resampling technique were proposed in chapters 3 and 4 and shown to perform well in comparison to other methods in the literature. Direct image super-resolution combines these two procedures to produce a single high-resolution image from a number of low-resolution input images. However, with non-iterative methods, any errors from the registration would propagate and compound the errors introduced by the image fusion stage.

## 5.1.1. Experiment XI – Effects of registration error on image fusion

### *Aim*

To investigate the effect of registration error on the fusion and resampling stage.

### *Method*

A 2592×3888 pixels source image was area-sampled to produce four 125×190 low-resolution images with random sub-pixel offsets in both directions and a 250×380 ground truth high-resolution image. Registration error was simulated by randomly perturbing the *x* and *y* offsets of each low-resolution image from its true position with zero mean and standard deviation $_r$. These perturbed images were resampled using a $10^{th}$ order rectangular pulse model-based filter. The root mean square error was computed between the resampled output and the ground truth high-resolution image. To gauge average performance, this process was repeated for a thousand different sub-pixel offsets of the input images and the median RMSE was computed. The experiment was repeated using the same one thousand combinations of offsets but with a different standard deviation of offset perturbation $_r$, varied between zero and 0.06 in steps of 0.0025.

To investigate how random offset perturbation affects the reconstruction in the presence of image noise, the above was repeated with input images contaminated by additive white Gaussian noise with zero mean and standard deviation $_n$. The

resampling filter's smoothing parameter was set to the optimal value, computed from figure 4.22 (the slope of the fitted lines).

In a similar manner to experiment VII, four source images pictured in figure 2.2 were used for this experiment.

### *Results*

The results are displayed in figure 5.1. As expected, the error in the fusion stage is affected by misregistration. For images 'bird', 'cat' and 'face' the error is doubled at perturbation standard deviation of around 0.06 pixels. For image 'text' the error is already quite large and the error increased by only 40% at the same perturbation standard deviation.

Resampling of noisy data is less sensitive to registration error – the exact positions of samples is not as important, as a result of smoothing. Image 'face' is least sensitive to misregistration out of the four images, because the optimal value of the smoothing parameter is largest for this image. Image 'text' is the most sensitive to misregistration, as the optimal value for the smoothing parameter is smallest for this image.

Experiments in chapter 3 suggested that an average registration error of better than 0.01 pixels may be achieved. Splitting this evenly into the errors in $x$ and $y$ components results in RMS error of 0.007 in each component. In the absence of noise, the fusion and resampling stage error is expected to increase by around 4, 2, 4 and 1 per cent for images 'bird', 'cat', 'face' and 'text' respectively (computed from the difference between median RMSE at $_r = 0$ and $_r = 0.007$ ).

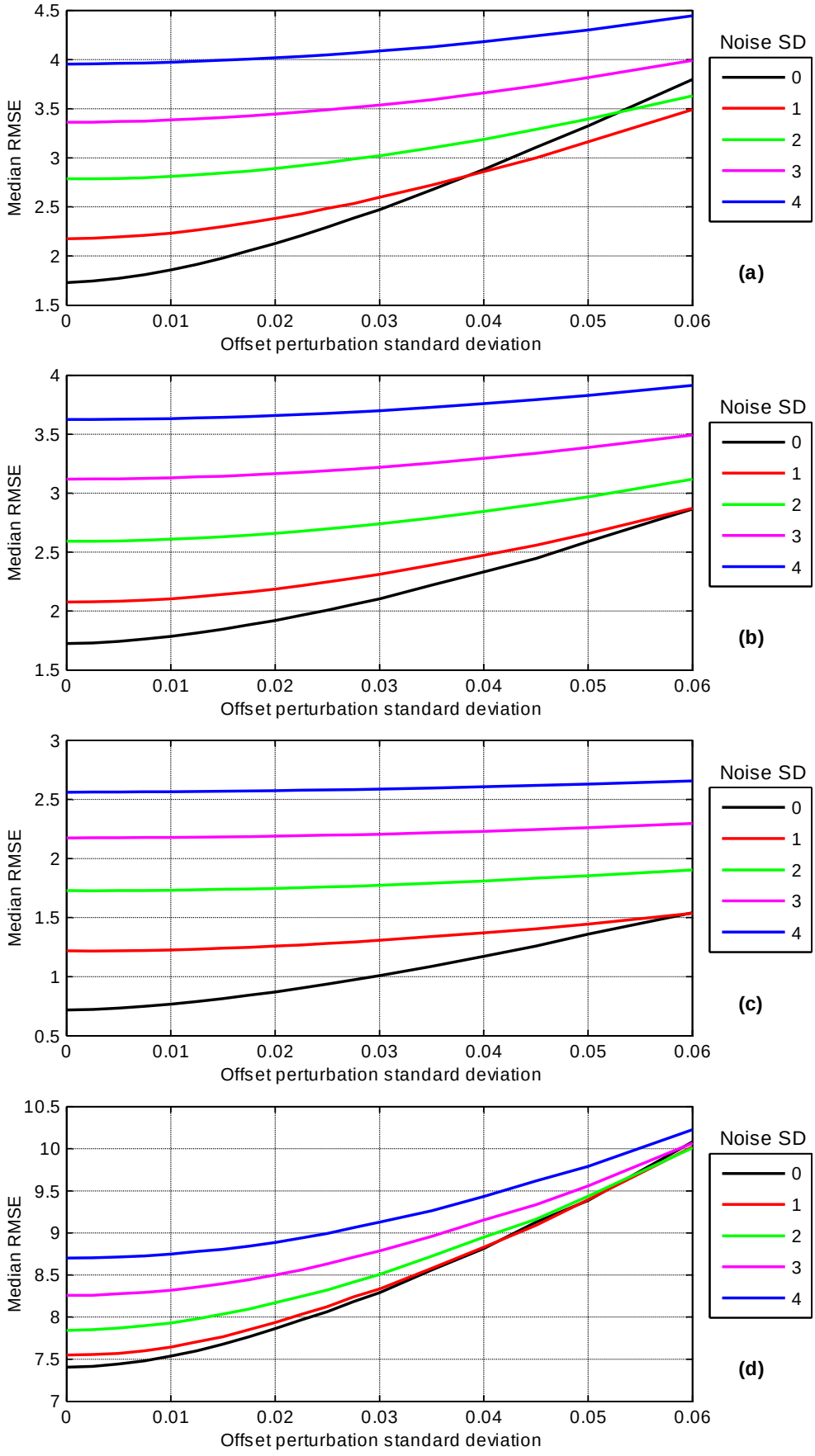


Figure 5.1 The effects of misregistration on resampling error for different levels of image noise: (a) Bird, (b) Cat, (c) Face and (d) Text. Noise SD in the legend is stated in grayscale levels.

#### *Conclusions*

The results indicate that in the absence of noise, accurate sub-pixel registration is required to achieve good resampling results. In the presence of moderate noise, high registration accuracy is not as important role because of the smoothing required to filter out the noise. In a low noise situation, there should only be a small increase in the error for a $10^{th}$ order resampling filter when images are registered using the method proposed in chapter 3 over using ground truth offsets.

### 5.1.2. Experiment XII – Effect of misregistration on resampling filters of different order

#### *Aim*

To investigate whether the sensitivity of resampling to misregistration differs for filters of different order.

#### *Method*

To perform this investigation, the previous experiment was repeated with filters of order 6 and 14 for noise standard deviation of one.

#### *Results*

Figure 5.2 shows how the resampling error deteriorates with increasing registration error for filters of order six, 10 and 14. For all four images, there does not seem to be any difference in the sensitivity of resampling to misregistration between higher and lower order filters.

#### *Conclusions*

Filters of all orders are affected in a similar way. Thus, there does not seem to be any advantages or disadvantages in the use of higher/lower order filters in terms of sensitivity to misregistration. For a given registration method with certain accuracy, higher order resampling filters achieve better results.

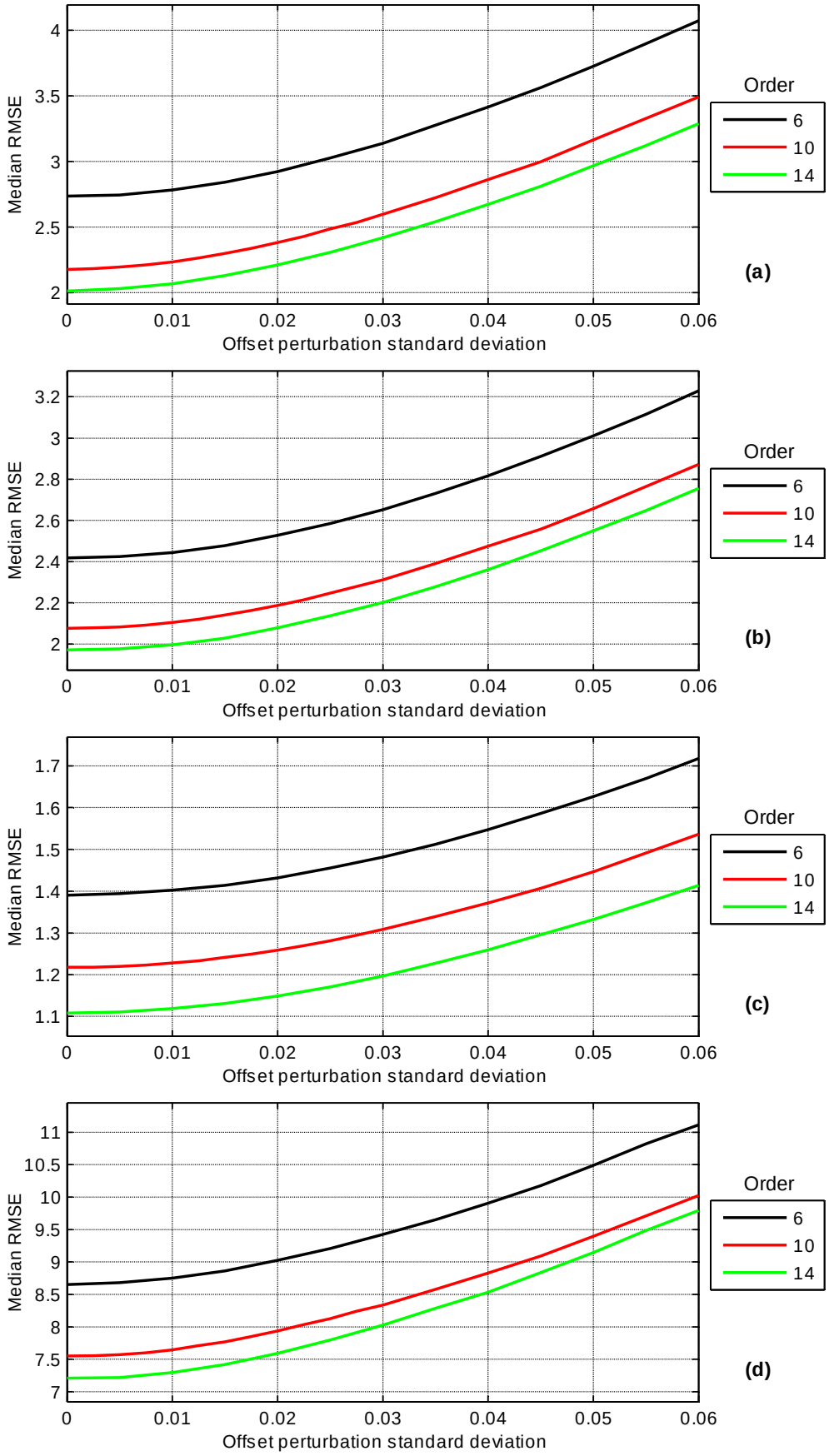


Figure 5.2 The effect of misregistration on resampling error for filters of different order for image (a) Bird, (b) Cat, (c) Face and (d) Text.

### 5.1.3. Experiment XIII – Comparison to existing methods

***Aim***

To investigate the performance of the proposed image super-resolution method, consisting of optimal interpolation based image registration and image resampling.

***Method***

The experimental set-up is the same as experiment VIII, apart from the following differences. The registration parameters are not assumed to be known *a priori*, instead these are determined from the low-resolution images using the registration procedure based on a 4×4 optimal filter, as described in chapter 3. This same registration process is used with all of the resampling methods. The input images are contaminated by additive white Gaussian noise with zero mean and standard deviation ranging between zero and five grayscale levels. The proposed rectangular pulse model-based reconstruction (Model 10) is compared with the triangulation method with a cubic polynomial fit (Tcub) and local polynomial expansion method using cubic (Cubic) and quadratic (Quadratic) polynomials.

***Results***

The results are shown in figure 5.3. The results of resampling using the proposed method and ground truth registration parameters are also shown for comparison (Model 10 GT).

Based on the assumption that registration accuracy is around one hundredth of a pixel, it was estimated in experiment XI that the increase in RMSE due to misregistration will be around 4, 2, 4 and 1 per cent for images ‘bird’, ‘cat’, ‘face’ and ‘text’ respectively. The actual increase was measured at 2.7, 0.3, 1.2 and 0.9 per cent respectively, which is significantly below the estimated error. For comparison, reducing the resampling filter order by one would result in RMSE increase of 4.6, 2.4, 5.6 and 1.7 per cent respectively.

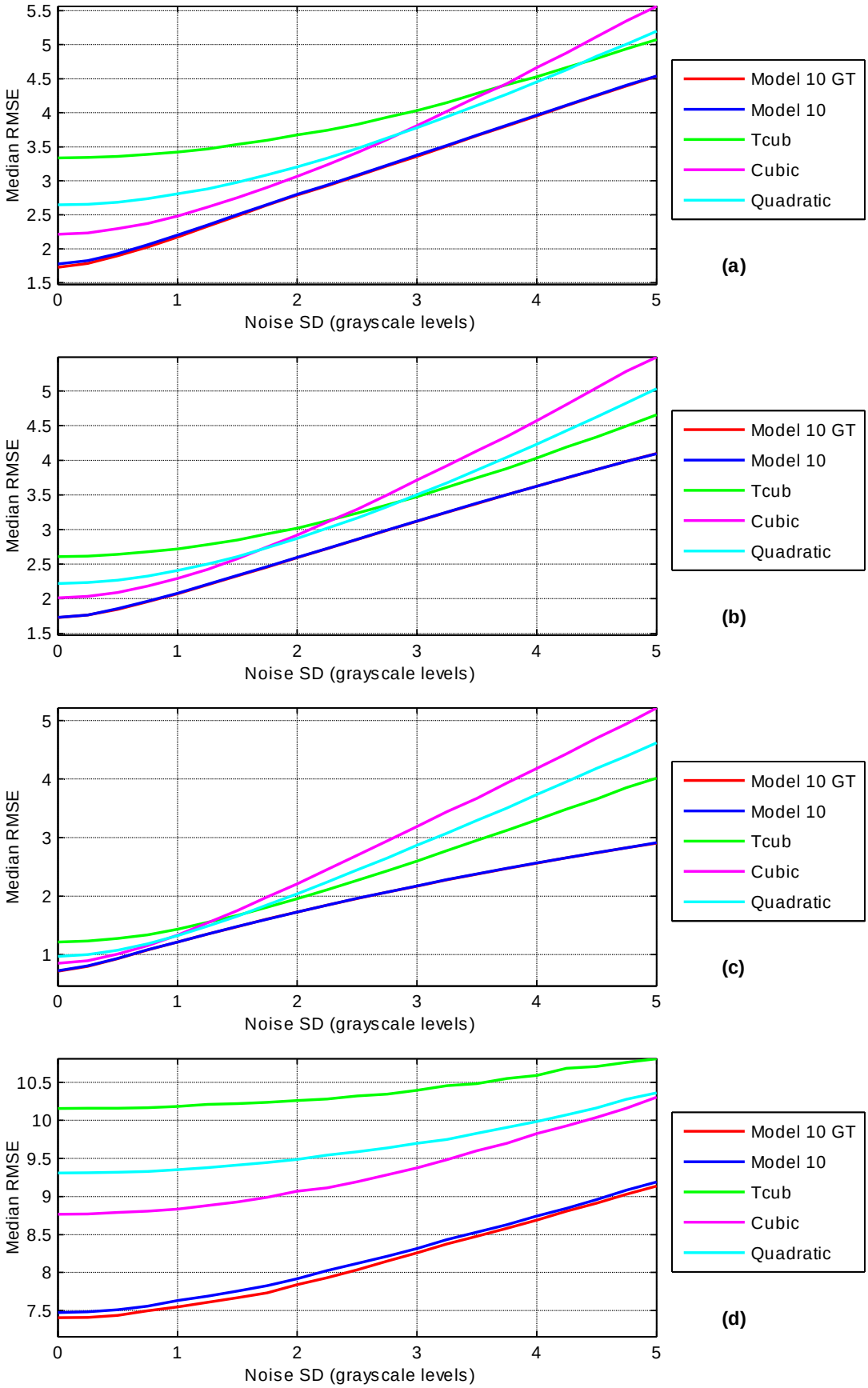


Figure 5.3 Plot of median RMSE versus noise standard deviation for a range of methods. Results obtained from image (a) Bird, (b) Cat, (c) Face and (d) Text.

### *Conclusions*

The proposed method is shown to outperform other direct methods for all four images.

Figure 5.4 shows cropped versions of some example output images of the proposed super-resolution method, as well as results of interpolation of a single image and the actual high-resolution image for comparison. Input images were noise free and the offsets that yielded median RMSE in experiment XIII were used. It is clear in considering the images in rows C and D that the edges are significantly smoother and have less aliasing. This shows that the method is achieving super-resolution.

The images in rows C and D are sampled at twice the sampling rate of A. The additional samples allow to represent additional information in the form of higher frequencies. In images in row C (the output of super-resolution), these higher frequencies are reconstructed from aliasing contained in the low-resolution input images. However, it is clear that these images appear overly smooth and must be deblurred. Performing debluring is beyond the scope of this thesis; however, it is an integral part of image super-resolution and for completeness the results of applying blind deconvolution, as implemented in Matlab 2008a is also shown for each image. The default parameters were used with a starting PSF of [0.25 0.25; 0.25 0.25]. The deblurred images appear noticeably sharper.

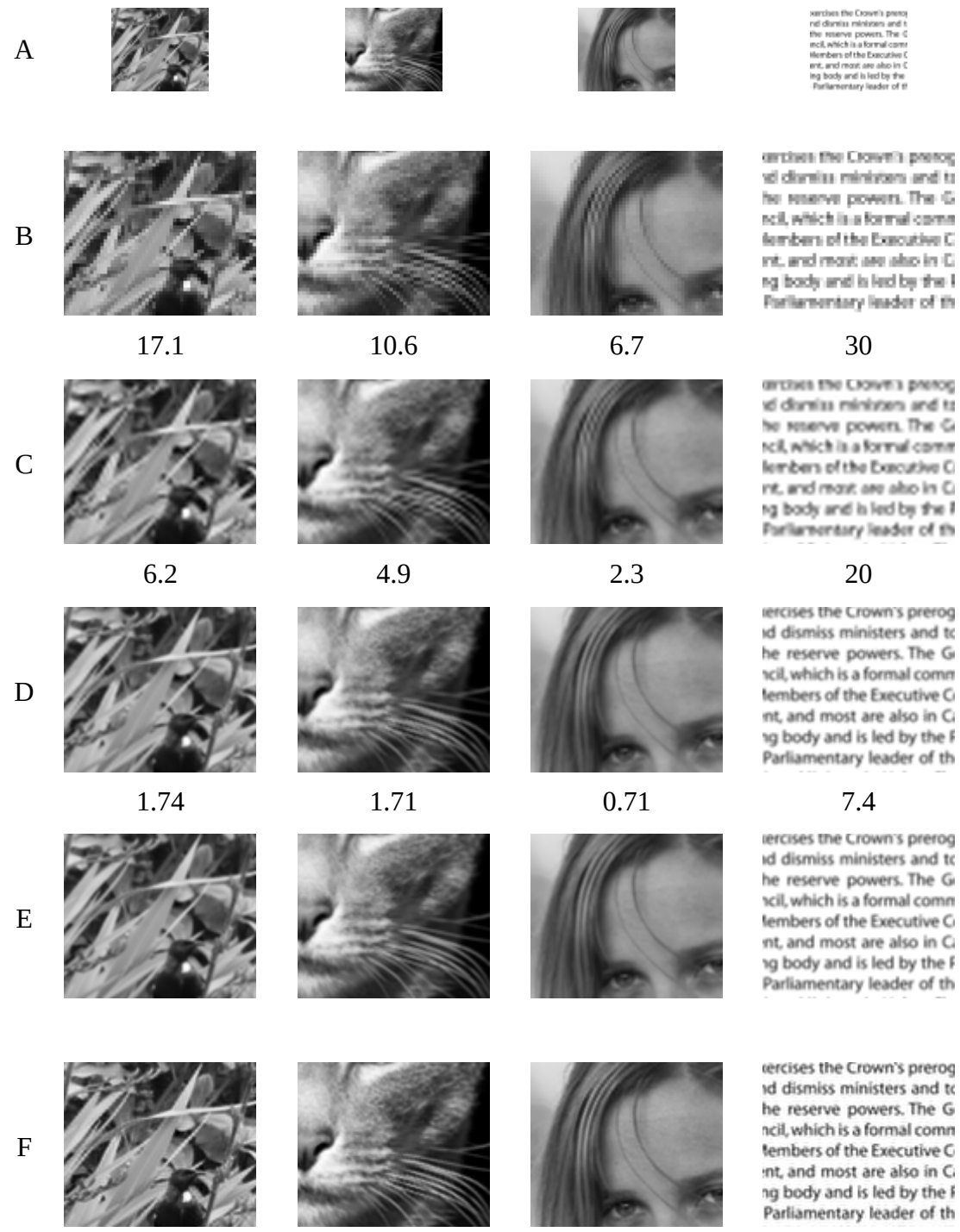


Figure 5.4 Example outputs: A – one low-resolution input image; B – nearest neighbour interpolation of A; C – pulse model-based near-optimal interpolation of the single low-resolution image in A; D – super-resolution output from four low-resolution images using the proposed method (order 10); E – high-resolution image; F – the result of deblurring D. Numbers under B, C and D indicate the RMSE between these images and E.

## 5.2. Summary and Conclusions

The combined processes of registration and fusion/resampling compose the bulk of direct image super-resolution. Because these steps are executed in series, any errors in registration will affect the resampling step and compound with the resampling errors. Accurate sub-pixel registration is required to achieve the results that are possible with the registration parameters known *a priori*. A perturbation with standard deviation of 0.06 pixels has been shown to double the error for most of the images for the $10^{th}$ order resampling filter. Misregistration has been shown to affect the resampling stage in the same way for different order filters. Consequently, higher order filters should still yield better result than lower order filters.

In the absence of noise, using optimal interpolation-based registration increased the error of resampling using a $10^{th}$ order filter by 2.7, 0.3, 1.2 and 0.9 per cent for images 'bird', 'cat', 'face' and 'text' respectively. In the presence of noise the increases were even smaller. The resulting RMSE was still well below that of other methods for noise standard deviation between zero and five. The error was shown to increase approximately linearly with noise.

For each test image, the error achieved from applying super-resolution to four input images is smaller by a factor of approximately three in comparison to the error achieved by interpolating a single input image (interpolating each one of the four input images resulted in a similar error). The resulting output images have also showed that there is noticeably less aliasing in super-resolution output in comparison to interpolation of a single image. This clearly demonstrates that image super-resolution has been achieved.

# Chapter 6

# Summary, Conclusions and Future Directions

## *6.1. Summary*

This thesis has considered the problem of interpolation based on minimising the least-squares error, and has applied it to the problem of fast, direct image super-resolution, particularly the problems of registration and image fusion.

Interpolation may be defined as model-based recovery of continuous signal from discrete data samples. Conventional interpolation methods employ predefined interpolation kernels; however, these are generally based on arbitrary functions, such as piecewise polynomials or the sinc function. The sinc function may be appropriate in the case of band-limited signals; however, it is not appropriate for aliased data.

Chapter 2 introduced the concept of least-squares optimal interpolation and optimal resampling filters, where the filter weights (and the interpolation kernel) are derived from the actual image data by globally minimising the sum of square differences between the interpolated image and the desired output. Minimising the square error has the advantages that the filter weights may be found using a closed-form solution with no need for iteration, and the resulting filter is unique and globally optimal in a least-squares sense. This means that no other spatially invariant resampling filter with the same region of support can achieve lower error when the error measure is based on the square error, for example mean square error or peak signal to noise ratio.

The problem with computing the ‘optimal’ output is that it requires the ideal output, which makes the method impractical for interpolation. However, the concept is valuable because it offers a flexible framework for deriving new kernels, based on various image models, making interpolation image specific. One such model, the pulse model, was presented in the chapter. This model is based on the observation that the optimal kernel is strongly influenced by step-edges. Resampling filters optimised on this model were shown to perform comparably to (or better than) a range of existing methods on a set of four images with a wide range of image statistics.

Chapter 3 considered the idea of using optimal interpolation for translation-only image registration. Image registration is traditionally a non-linear optimisation problem and conventional solutions are iterative. The advantage of using optimal

interpolation for registration is the way its use can linearise the problem, resulting in direct solutions.

An experimental analysis of optimal interpolation-based registration showed that it outperforms existing methods, both iterative and direct. It offers reasonable registration results for both noisy and aliased images and offers flexibility in terms of the size of the region of support, offering a good trade-off in terms of computational complexity and accuracy of registration.

Analytical analysis of bias in estimating registration parameters using optimal interpolation showed that the bias is a function of the noise level as well as the signal. These two components can have opposite signs in some circumstances, and therefore the interaction between the two results in partial bias cancellation. This provides one explanation for the unexpected observation that registration sometimes improves when the images are contaminated with more noise (another is that noise can act as a regulariser).

Chapter 4 extended the pulse model-based near-optimal interpolation from chapter 2 to the problem of resampling non-uniform data onto a uniform grid. The resulting method offers a flexible trade-off between computational complexity and the quality of reconstruction, in the form of the filter order. The performance of pulse model-based filters of different order are investigated and shown to follow closely to the performance of optimal filters (derived from the actual image data) of the same order.

The performance of the proposed method of resampling non-uniform image data has been compared to a number of existing methods belonging to three classes: inverse distance weighted methods, triangulation methods and local polynomial expansion methods. The pulse model-based near-optimal filter of order 10 was shown to outperform all of these methods for all of the test images.

Another advantage of the proposed method is its ability to relax the interpolation constraint to offer some degree of noise smoothing. This is controlled by a single variable – the smoothing parameter. Increasing this parameter increases the amount of smoothing. Decreasing the smoothing parameter increases conformity of the result to the input data.

Image super-resolution is concerned with constructing a higher-resolution representation of a scene from a number of low-resolution images of that scene.

Chapter 5 demonstrates that the image registration method proposed in chapter 3 and the image resampling method proposed in chapter 4 can be cascaded to solve the problem of image super-resolution. Because both of these proposed methods are direct and computationally inexpensive, they result in a fast and direct image super-resolution. Errors in the registration step are shown to have little effect on the resampling root mean square error, averaging across all four test images at around 1.3 per cent increase from resampling using ground truth offsets.

## 6.2. Original Contributions

The key original contributions of this work, as discussed in section 6.1, are as follows:

- Formulation of the model-based linear least squares optimal framework for image interpolation and resampling
- Creation of a step-edge-based image model for interpolation of aliased images using the above framework
- Application of linear least squares interpolation framework to image registration
- Analysis of bias in registration estimation using the above methodology

## 6.3. Conclusions

The main objective of the thesis was to show that optimal interpolation can be used to perform fast, direct and effective super-resolution. Optimal interpolation-based registration in combination with $10^{th}$ order pulse model-based resampling filter were compared experimentally to a number of existing interpolation-based methods and shown to outperform the existing methods in terms of root mean square error for a range of noise levels. The results of increasing resolution by a factor of two from four low-resolution images using the proposed method visually show a noticeable decrease in aliasing over interpolation of a single low-resolution image, demonstrating that super-resolution is performed. Applying a deblurring algorithm to the output images noticeably increases their sharpness.

The computational complexity of the image super-resolution method proposed is this thesis is $\mathcal{O}(N^2)$, where $N$ is length (or width) of the (square) input image, i.e., the cost is linear in the number of pixels. This can be seen by considering the two parts of the algorithm (registration and fusion) separately. Computation of the image

registration requires the inversion of a relatively small matrix (15x15 for the 4x4 window considered in the experiments). Matrix inversion is an $\mathcal{O}(n^3)$ time cost for an $n \times n$ matrix, but here for the constant window size we consider it is a constant cost as it is independent of the image size. Additionally, it is performed only once for any number of target images registered against a single reference image. The cost of the computation comes from filling in the elements of this matrix, which is $\mathcal{O}(N^2)$. The vector-matrix multiplication that follows the inversion of this matrix is also $\mathcal{O}(N^2)$, but must be computed for every target images. Thus, the overall complexity of the proposed registration algorithm is $\mathcal{O}(N^2)$ per image. Image fusion requires computation of the resampling filter coefficients, based on the result of the registration algorithm. This is performed using a closed-form expression and can be considered a constant cost. Performing the filtering operation itself has complexity $\mathcal{O}(\mathrm{N}^2)$.

Overall, this thesis has demonstrated that least-squares optimal interpolation can be efficiently implemented and used for fast, direct and effective image super-resolution. Along with the development of the methods for uniform and non-uniform interpolation, and image fusion, experiments have been presented demonstrating that the method compares favourably with methods in the literature.

## 6.4. Future work directions

This thesis is a good initial study into the use of least-squares optimal interpolation for image super-resolution. However, there are some questions that have not been addressed by this work and more questions that come to mind as a result of this work.

The proposed registration method is limited in the types of relative motion between the frames it can handle – only global translational motion. Even for a hand-held camera, there may be some rotation present between sequentially captured images. The algorithm could be improved to handle small amounts of rotation by sub-dividing the images into smaller blocks and registering these. Rotation could be computed from the translations of smaller blocks. An investigation must be performed to find out the relationship between registration accuracy and block size to determine into how many blocks an image can be sub-divided. It is also necessary to examine the effects of rotation on the accuracy of a pure translation-based estimate of registration.

The performance of near-optimal interpolation can be investigated with other image models. These models should also take into consideration the effects of lens blur as well as just area-sampling.

A method of determining a good value for the smoothing parameter must also be investigated to offer good noise smoothing. For optimal interpolation, this relates directly to the noise level. For model-based interpolation, the optimum value will depend on the signal to noise ratio. However the relationship is more complex because it depends on how the image models relates to the actual image content. For example the usual definition of image variance divided by noise variance does not relate directly to the important edge features. A more appropriate measure of signal strength would be the average step-edge height, which would require a suitable definition.

For super-resolution, it was observed that the error in the output image also depends on the position of the high-resolution resampling grid relative to the locations of the input samples. Further investigation is required to determine the optimum placement of the resampling grid for a given set of input sample locations.

Another possible direction is to attempt to incorporate deblurring into the resampling. The proposed method employs area-sampling of the model to produce input low-resolution and output high-resolution pixel values. However, the pixel area of the input and output is assumed to be the same. By sampling the model with appropriately reduced pixels (for the high-resolution output), it may be possible to achieve some degree of deblurring. This will also require revisiting the trade-off between computation and kernel size as a larger kernel will be more effective at deblurring.

# References


[1] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: A technical overview,” *IEEE Signal Processing Magazine,* vol. 20, no. 3, pp. 21-36, May, 2003.

[2] P. Askey. "Canon EOS-1Ds Mark III Preview," 15 May 2008; http://www.dpreview.com/previews/CanonEOS1DsMarkIII/.

[3] T. Komatsu, K. Aizawa, T. Igarashi *et al.*, “Signal-Processing Based Method for Acquiring Very High-Resolution Images with Multiple Cameras and Its Theoretical-Analysis,” *IEE Proceedings-I Communications Speech and Vision,* vol. 140, no. 1, pp. 19-25, February, 1993.

[4] Z. Zalevsky, and D. Mendlovic, *Optical superresolution*, Tel Aviv ; New York: Springer, 2004.

[5] K. Irie, A. E. McKinnon, K. Unsworth *et al.*, “A technique for evaluation of CCD video-camera noise,” *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 18, no. 2, pp. 280-284, February, 2008.

[6] G. E. Healey, and R. Kondepudy, “Radiometric CCD Camera Calibration and Noise Estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 3, pp. 267-276, March, 1994.

[7] M. Unser, “Sampling-50 years after Shannon,” *Proceedings of the IEEE,* vol. 88, no. 4, pp. 569-587, 2000.

[8] S. Baker, and T. Kanade, “Limits on super-resolution and how to break them,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 9, pp. 1167-1183, September, 2002.

[9] D. Keren, S. Peleg, and R. Brada, “Image sequence enhancement using sub-pixel displacements,” in Computer Society Conference on Computer Vision and Pattern Recognition, 1988, pp. 742-746.

[10] N. Nguyen, P. Milanfar, and G. Golub, “A Computationally Efficient Superresolution Image Reconstruction Algorithm,” *IEEE Transactions on Image Processing,* vol. 10, no. 4, pp. 573-583, April, 2001.

[11] A. J. Patti, and Y. Altunbasak, “Artifact Reduction for Set Theoretic Super Resolution Image Reconstruction With Edge Adaptive Constraints and

Higher-Order Interpolants," *IEEE Transactions on Image Processing,* vol. 10, no. 1, pp. 179-186, January, 2001.

[12] S. Peleg, D. Keren, and L. Schweitzer, "Improving Image-Resolution Using Subpixel Motion," *Pattern Recognition Letters,* vol. 5, no. 3, pp. 223-226, March, 1987.

[13] B. C. Tom, and A. K. Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," in Proceedings of International Conference on Image Processing, 1995, pp. 539-542.

[14] R. Y. Tsai, and T. S. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing,* vol. 1, pp. 317-339 1984.

[15] H. Ur, and D. Gross, "Improved Resolution from Subpixel Shifted Pictures," *CVGIP-Graphical Models and Image Processing,* vol. 54, no. 2, pp. 181-186, March, 1992.

[16] N. K. Bose, H. C. Kim, and H. M. Valenzuela, "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes," in IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993, pp. 269-272.

[17] E. Kaltenbacher, and R. C. Hardie, "High resolution infrared image reconstruction using multiple, low resolution, aliased frames," in IEEE National Aerospace and Electronics Conference, 1996, pp. 702-709.

[18] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive Reconstruction of High-Resolution Image From Noisy Undersampled Multiframes," *IEEE Transactions on Acoustics Speech and Signal Processing,* vol. 38, no. 6, pp. 1013-1027, June, 1990.

[19] S. Rhee, and M. G. Kang, "Discrete Cosine Transform Based Regularized High-Resolution Image Reconstruction Algorithm," *Optical Engineering,* vol. 38, no. 8, pp. 1348-1356, August, 1999.

[20] S. P. Kim, and W.-Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images," *IEEE Transactions on Image Processing,* vol. 2, no. 4, pp. 534-539, October, 1993.

[21] M. C. Chiang, and T. E. Boult, "Efficient Super-Resolution Via Image Warping," *Image and Vision Computing*, vol. 18, no. 10, pp. 761-771, July, 2000.

[22] S. P. Kim, and N. K. Bose, "Reconstruction of 2-D bandlimited discrete signals from nonuniform samples," *Radar and Signal Processing, IEE Proceedings F*, vol. 137, no. 3, pp. 197-204, 1990.

[23] Y. Koo, and W. Kim, "An Image Resolution Enhancing Technique Using Adaptive Sub-Pixel Interpolation for Digital Still Camera System," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 118-123, February 1999, 1999.

[24] M. S. Alam, J. G. Bognar, R. C. Hardie *et al.*, "Infrared Image Registration and High-Resolution Reconstruction Using Multiple Translationally Shifted Aliased Video Frames," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 915-923, October, 2000.

[25] D. Rajan, and S. Chaudhuri, "Generalized Interpolation and Its Application in Super-Resolution Imaging," *Image and Vision Computing*, vol. 19, no. 13, pp. 957-969, November, 2001.

[26] S. Lertrattanapanich, and N. K. Bose, "High resolution image formation from low resolution frames using Delaunay triangulation," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1427-1441, Dec, 2002.

[27] T. Q. Pham, L. J. van Vliet, and K. Schutte, "Robust fusion of irregularly sampled data using adaptive normalized convolution," *EURASIP Journal on Applied Signal Processing*, 2006.

[28] M. Irani, and S. Peleg, "Super resolution from image sequences," in 10th International Conference on Pattern Recognition, 1990, pp. 115-120.

[29] M. Irani, and S. Peleg, "Improving Resolution by Image Registration," *CVGIP-Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231-239, May, 1991.

[30] B. Cohen, and I. Dinstein, "Polyphase Back-Projection Filtering for Image Resolution Enhancement," *IEE Proceedings-Vision Image and Signal Processing*, vol. 147, no. 4, pp. 318-322, August, 2000.

[31] N. R. Shah, and A. Zakhor, "Resolution Enhancement of Color Video Sequences," *IEEE Transactions on Image Processing*, vol. 8, no. 6, pp. 879-885, June, 1999.

[32] N. K. Bose, S. Lertrattanapanich, and J. Koo, “Advances in superresolution using L-curve,” in IEEE International Symposium on Circuits and Systems, 2001, pp. 433-436 vol. 2.

[33] R. C. Hardie, K. J. Barnard, J. G. Bognar *et al.*, “High-Resolution Image Reconstruction From a Sequence of Rotated and Translated Frames and Its Application to an Infrared Imaging System,” *Optical Engineering,* vol. 37, no. 1, pp. 247-260, January, 1998.

[34] M. K. Ng, R. H. Chan, T. F. Chan *et al.*, “Cosine Transform Preconditioners for High Resolution Image Reconstruction,” *Linear Algebra and Its Applications,* vol. 316, no. 1-3, pp. 89-104, September, 2000.

[35] N. Nguyen, P. Milanfar, and G. Golub, “Efficient Generalized Cross-Validation With Applications to Parametric Image Restoration and Resolution Enhancement,” *IEEE Transactions on Image Processing,* vol. 10, no. 9, pp. 1299-1308, September, 2001.

[36] S. Farsiu, D. Robinson, M. Elad *et al.*, “Fast and robust super-resolution,” in International Conference on Image Processing, 2003, pp. 291-294.

[37] M. Elad, and A. Feuer, “Restoration of a Single Superresolution Image From Several Blurred, Noisy, and Undersampled Measured Images,” *IEEE Transactions on Image Processing,* vol. 6, no. 12, pp. 1646-1658, December, 1997.

[38] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Super-Resolution Reconstruction of Compressed Video Using Transform-Domain Statistics,” *IEEE Transactions on Image Processing,* vol. 13, no. 1, pp. 33-43, January, 2004.

[39] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, “Joint MAP Registration and High-Resolution Image Estimation Using a Sequence of Undersampled Images,” *IEEE Transactions on Image Processing,* vol. 6, no. 12, pp. 1621-1633, December, 1997.

[40] R. Molina, M. Vega, J. Abad *et al.*, “Parameter Estimation in Bayesian High-Resolution Image Reconstruction With Multisensors,” *IEEE Transactions on Image Processing,* vol. 12, no. 12, pp. 1655-1667, December, 2003.

[41] M. C. Pan, “A Novel Blind Super-Resolution Technique Based on the Improved Poisson Maximum a Posteriori Algorithm,” *International Journal of Imaging Systems and Technology,* vol. 12, no. 6, pp. 239-246, 2002.

[42] D. Rajan, and S. Chaudhuri, “An MRF-Based Approach to Generation of Super-Resolution Images From Blurred Observations,” *Journal of Mathematical Imaging and Vision,* vol. 16, no. 1, pp. 5-15, January, 2002.

[43] C. A. Segall, A. K. Katsaggelos, R. Molina *et al.*, “Bayesian Resolution Enhancement of Compressed Video,” *IEEE Transactions on Image Processing,* vol. 13, no. 7, pp. 898-911, July, 2004.

[44] R. R. Schultz, and R. L. Stevenson, “Extraction of High-Resolution Frames From Video Sequences,” *IEEE Transactions on Image Processing,* vol. 5, no. 6, pp. 996-1011, June, 1996.

[45] H. Stark, and P. Oskoui, “High-Resolution Image Recovery From Image-Plane Arrays, Using Convex Projections,” *Journal of the Optical Society of America A-Optics Image Science and Vision,* vol. 6, no. 11, pp. 1715-1726, November, 1989.

[46] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, “High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration,” in IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992, pp. 169-172.

[47] N. Nguyen, “Numerical algorithms for image superresolution,” Ph.D., Stanford University, 2000.

[48] D. G. Bailey, and L. T. H., “Image registration methods for resolution improvement,” in Proceedings of Image and Vision Computing New Zealand, Christchurch, New Zealand, 1999, pp. 91-96.

[49] D. G. Bailey, “Predictive Interpolation for Registration,” in Proceedings of Image and Vision Computing Conference NZ, Hamilton, NZ, 2000, pp. 240-245.

[50] A. Lukin, A. S. Krylov, and A. Nasonov, “Image Interpolation by Super-Resolution,” in International Conference on Computer Graphics and Vision, Novosibirsk Akademgorodok, Russia, 2006, pp. 239-242.

[51] J. C. Gillett, T. M. Stadtmiller, and R. C. Hardie, “Aliasing Reduction in Staring Infrared Imagers Utilizing Subpixel Techniques,” *Optical Engineering,* vol. 34, no. 11, pp. 3130-3137, November, 1995.

[52] S. E. Reichenbach, and S. K. Park, “Small Convolution-Kernels for High-Fidelity Image-Restoration,” *IEEE Transactions on Signal Processing,* vol. 39, no. 10, pp. 2263-2274, Oct, 1991.

[53] R. C. Gonzalez, and P. A. Wintz, *Digital image processing*, 2nd ed., Reading, Mass.: Addison-Wesley, 1987.

[54] W. K. Pratt, *Digital image processing*, 2nd ed., New York: Wiley, 1991.

[55] A. K. Katsaggelos (Ed.), *Digital image restoration*, Heidelberg: Springer-Verlag, 1991.

[56] M. I. Sezan, *Selected papers on digital image restoration*, Bellingham, Wash., USA: SPIE Optical Engineering Press, 1992.

[57] H. C. Andrews, and B. R. Hunt, *Digital image restoration*, Englewood Cliffs, N.J.: Prentice-Hall, 1977.

[58] E. Meijering, "A chronology of interpolation: From ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319-342, Mar, 2002.

[59] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049-1075, Nov, 1999.

[60] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of medical imaging*, pp. 393-420, Orlando, FL, USA: Academic Press, Inc., 2000.

[61] T. M. Lehmann, C. Gonner, and K. Spitzer, "B-spline interpolation in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 660-665, Jul, 2001.

[62] S. E. Reichenbach, and F. Geng, "Two-dimensional cubic convolution," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 857-865, Aug, 2003.

[63] S. E. Reichenbach, and J. Shi, "Two-dimensional cubic convolution for one-pass image restoration and reconstruction," in IEEE International Symposium on Geoscience and Remote Sensing, 2004, pp. 2074-2076.

[64] J. Z. Shi, and S. E. Reichenbach, "Image interpolation by two-dimensional parametric cubic convolution," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1857-1870, Jul, 2006.

[65] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel B-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228-244, 1997.

[66] C. Vazquez, E. Dubois, and J. Konrad, "Reconstruction of nonuniformly sampled images in spline spaces," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 713-725, Jun, 2005.

[67] L. Youming, "Irregular sampling for spline wavelet subspaces," *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 623-627, 1996.

[68] D. F. Watson, *Contouring : a guide to the analysis and display of spatial data : with programs on diskette*, 1st ed., Oxford ; New York: Pergamon Press, 1992.

[69] L. Yaroslavsky, "Boundary effect free and adaptive discrete signal sinc-interpolation algorithms for signal and image resampling," *Applied Optics*, vol. 42, no. 20, pp. 4166-4175, Jul 10, 2003.

[70] J. K. Han, and S. U. Baek, "Parametric cubic convolution scaler for enlargement and reduction of image," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 2, pp. 247-256, May, 2000.

[71] X. Li, and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521-1527, Oct, 2001.

[72] G. Ramponi, "Warped distance for space-variant linear image interpolation," *IEEE Transactions on Image Processing*, vol. 8, no. 5, pp. 629-639, May, 1999.

[73] K. P. Hong, J. K. Paik, H. J. Kim *et al.*, "An edge-preserving image interpolation system for a digital camcorder," *IEEE Transactions on Consumer Electronics*, vol. 42, no. 3, pp. 279-284, Aug, 1996.

[74] K. Jensen, and D. Anastassiou, "Subpixel Edge Localization and the Interpolation of Still Images," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 285-295, Mar, 1995.

[75] J. W. Hwang, and H. S. Lee, "Adaptive image interpolation based on local gradient features," *Signal Processing Letters, IEEE*, vol. 11, no. 3, pp. 359-362, 2004.

[76] S. G. Mallat, *A wavelet tour of signal processing*, 2nd ed., San Diego: Academic Press, 1999.

[77] G. H. Golub, and C. F. Van Loan, *Matrix computations*, 3rd ed., Baltimore: Johns Hopkins University Press, 1996.

[78] R. R. Schultz, and R. L. Stevenson, “A Bayesian Approach to Image Expansion for Improved Definition,” *IEEE Transactions on Image Processing,* vol. 3, no. 3, pp. 233-242, May, 1994.

[79] L. Chen, and K.-H. Yap, “Regularized interpolation using Kronecker product for still images,” in IEEE International Conference on Image Processing, 2005, pp. 1014-17.

[80] S. E. El-Khamy, M. M. Hadhoud, M. I. Dessouky *et al.*, “Efficient implementation of image interpolation as an inverse problem,” *Digital Signal Processing,* vol. 15, no. 2, pp. 137-152, 2005.

[81] W. Y. V. Leung, P. J. Bones, and R. G. Lane, “Statistical interpolation of sampled images,” *Optical Engineering,* vol. 40, no. 4, pp. 547-553, 2001.

[82] J. H. Shin, J. H. Jung, and J. K. Paik, “Regularized iterative image interpolation and its application to spatially scalable coding,” *IEEE Transactions on Consumer Electronics,* vol. 44, no. 3, pp. 1042-1047, Aug, 1998.

[83] S. K. Mitra, *Digital signal processing : a computer-based approach*, 2nd ed., Boston: McGraw-Hill/Irwin, 2001.

[84] C. E. Shannon, “Communication In The Presence Of Noise,” *Proceedings of the IEEE,* vol. 86, no. 2, pp. 447-457, 1998.

[85] F. J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proceedings of the IEEE,* vol. 66, no. 1, pp. 51-83, 1978.

[86] G. Wolberg, *Digital image warping*, Los Alamitos, Calif.: IEEE Computer Society Press, 1992.

[87] K. Turkowski, "Filters for common resampling tasks," *Graphics gems*, pp. 147-165: Academic Press Professional, Inc., 1990.

[88] A. Aldroubi, M. Unser, and M. Eden, “Cardinal spline filters: stability and convergence to the ideal sinc interpolator,” *Signal Process.,* vol. 28, no. 2, pp. 127-138, 1992.

[89] I. Schoenberg, “Notes on spline functions III: On the convergence of the interpolating cardinal splines as their degree tends to infinity,” *Israel Journal of Mathematics,* vol. 16, no. 1, pp. 87-93, 1973.

[90] I. J. Schoenberg, “Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. Part B - On the Problem of

Osculatory Interpolation - a 2nd Class of Analytic Approximation Formulae," *Quarterly of Applied Mathematics,* vol. 4, no. 2, pp. 112-141, 1946.

[91] I. J. Schoenberg, "Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. Part A - On the Problem of Smoothing or Graduation - a 1st Class of Analytic Approximation Formulae," *Quarterly of Applied Mathematics,* vol. 4, no. 1, pp. 45-99, 1946.

[92] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of Interpolating Methods for Image Resampling," *Medical Imaging, IEEE Transactions on,* vol. 2, no. 1, pp. 31-39, 1983.

[93] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal-Processing. Part 1 - Theory," *IEEE Transactions on Signal Processing,* vol. 41, no. 2, pp. 821-833, Feb, 1993.

[94] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal-Processing. Part 2 - Efficient Design and Applications," *IEEE Transactions on Signal Processing,* vol. 41, no. 2, pp. 834-848, Feb, 1993.

[95] M. Unser, "Splines - A perfect fit for signal and image processing," *IEEE Signal Processing Magazine,* vol. 16, no. 6, pp. 22-38, Nov, 1999.

[96] R. G. Keys, "Cubic Convolution Interpolation for Digital Image-Processing," *IEEE Transactions on Acoustics Speech and Signal Processing,* vol. 29, no. 6, pp. 1153-1160, 1981.

[97] F. N. Fritsch, and R. E. Carlson, "Monotone Piecewise Cubic Interpolation," *Siam Journal on Numerical Analysis,* vol. 17, no. 2, pp. 238-246, 1980.

[98] D. P. Mitchell, and A. N. Netravali, "Reconstruction filters in computer-graphics," *SIGGRAPH Comput. Graph.,* vol. 22, no. 4, pp. 221-228, 1988.

[99] J. Z. Shi, and S. E. Reichenbach, "Interpolation by asymmetric, two-dimensional cubic convolution," in IEEE International Conference on Image Processing, 2005, pp. 994-997.

[100] B. J. C. Baxter, "The interpolation theory of radial basis functions," PhD Dissertation, Trinity College, Cambridge University, 1992.

[101] G. Wahba, *Spline models for observational data*, Philadelphia, Pa.: Society for Industrial and Applied Mathematics, 1990.

[102] A. Bab-Hadiashar, D. Suter, and R. Jarvis, "Optic Flow Computation Using Interpolating Thin-Plate Splines," in Second Asian Computer Vision Conference, Singapore, 1995, pp. 452-456.

[103] J. Romberg, "Imaging via Compressive Sampling [Introduction to compressive sampling and recovery via convex programming]," *Signal Processing Magazine, IEEE,* vol. 25, no. 2, pp. 14-20, 2008.

[104] P. J. Huber, "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Annals of Statistics,* vol. 1, no. 5, pp. 799-821, 1973.

[105] C. L. Lawson, and R. J. Hanson, *Solving least squares problems*, Englewood Cliffs, N.J.,: Prentice-Hall, 1974.

[106] C. Tomasi, and R. Manduchi, "Bilateral filtering for gray and color images," in Sixth International Conference on Computer Vision, 1998, pp. 839-846.

[107] V. Patanavijit, and S. Jitapunkul, "A Robust Iterative Multiframe Super-Resolution Reconstruction using a Bayesian Approach with Tukey's Biweigth," in 8th International Conference on Signal Processing, Beijing, China, 2006, pp. 1099-1102.

[108] M. J. Black, G. Sapiro, D. H. Marimont *et al.*, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing,* vol. 7, no. 3, pp. 421-432, March, 1998.

[109] V. Patanavijit, and S. Jitapunkul, "A Lorentzian Bayesian Approach for Robust Iterative Multiframe Super-Resolution Reconstruction with Lorentzian-Tikhonov Regularization," in International Symposium on Communications and Information Technologies, 2006, pp. 1044-1049.

[110] J. Z. Shi, S. E. Reichenbach, and J. D. Howe, "Small-kernel superresolution methods for microscanning imaging systems," *Applied Optics,* vol. 45, no. 6, pp. 1203-1214, Feb, 2006.

[111] X.-Q. Chen, J. Zhang, and L.-N. Wu, "Improvement of a nonlinear image interpolation method based on heat diffusion equation," in International Conference on Machine Learning and Cybernetics, 2003, pp. 2911-2914.

[112] N. A. Dodgson, "Quadratic interpolation for image resampling," *IEEE Transactions on Image Processing,* vol. 6, no. 9, pp. 1322-1326, Sep, 1997.

[113] E. Doi, and M. S. Lewicki, "Relations between the statistical regularities of natural images and the response properties of the early visual system," in SIG P&P, Kyoto University, Japan, 2005.

[114] A. Torralba, and A. Oliva, "Statistics of natural image categories," *Network-Computation in Neural Systems,* vol. 14, no. 3, pp. 391-412, August, 2003.

[115] L. G. Brown, “A Survey of Image Registration Techniques,” *Computing Surveys,* vol. 24, no. 4, pp. 325-376, Dec, 1992.

[116] B. Zitova, and J. Flusser, “Image Registration Methods: a Survey,” *Image and Vision Computing,* vol. 21, no. 11, pp. 977-1000, 2003 Oct 1, 2003.

[117] D. Capel, *Image mosaicing and super-resolution*, London ; New York: Springer, 2004.

[118] D. Scharstein, and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” *International Journal of Computer Vision,* vol. 47, no. 1, pp. 7-42, 2002.

[119] S. P. Boyd, and L. Vandenberghe, *Convex optimization*, Cambridge, UK ; New York: Cambridge University Press, 2004.

[120] S. Baker, and I. Matthews, “Lucas-Kanade 20 years on: A unifying framework,” *International Journal of Computer Vision,* vol. 56, no. 3, pp. 221-255, Feb-Mar, 2004.

[121] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of Optical-Flow Techniques,” *International Journal of Computer Vision,* vol. 12, no. 1, pp. 43-77, Feb, 1994.

[122] R. Hartley, and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge, UK ; New York: Cambridge University Press, 2000.

[123] B. D. Lucas, and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” in Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981, pp. 674-679.

[124] A. K. Jain, *Fundamentals of digital image processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[125] E. O. Brigham, *The fast Fourier transform*, Englewood Cliffs, N.J.,: Prentice-Hall, 1974.

[126] H. Stone, M. Orchard, and E.-C. Chang, “Subpixel registration of images,” in Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, 1999, pp. 1446-1452.

[127] H. Stone, M. T. Orchard, E. C. Chang *et al.*, “A fast direct fourier-based algorithm for subpixel registration of images,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 39, no. 10, pp. 2235-2243, Oct, 2001.

[128] S. Baker. "Lucas-Kanade 20 Years on," February 1, 2008; http://www.ri.cmu.edu/projects/project_515.html.

[129] B. Jon Louis, “Multidimensional binary search trees used for associative searching,” *Commun. ACM,* vol. 18, no. 9, pp. 509-517, 1975.

[130] G. R. Shorack, *Probability for statisticians*, New York: Springer, 2000.

[131] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in Proceedings of the 1968 23rd ACM national conference, 1968.

[132] R. Franke, “Scattered Data Interpolation: Tests of Some Methods,” *Mathematics of Computation,* vol. 38, no. 157, pp. 181-200, 1982.

[133] B. N. Delaunay, “Sur la Sphère Vide,” *Izvestia Akademia Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk,* vol. 7, pp. 793-800, 1934.

[134] J. O'Rourke, *Computational geometry in C*, Cambridge ; New York: Cambridge University Press, 1994.

[135] R. Clough, and J. Tocher, “Finite element stiffness matrices for analysis of plates in bending,” in Proceedings of Conference on Matrix Methods in Structural Analysis, 1965, pp. 515-545.

[136] I. Amidror, “Scattered data interpolation methods for electronic imaging systems: a survey,” *Journal of Electronic Imaging,* vol. 11, no. 2, pp. 157-176, Apr, 2002.

[137] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Trans. Math. Softw.,* vol. 22, no. 4, pp. 469-483, 1996.

[138] C. L. Lawson, "Software for C1 Surface Interpolation," *Mathematical Software III*, J. R. Rice, ed., pp. 161-194, New York: Academic Press, 1977.

[139] R. M. Haralick, and L. Watson, “A Facet Model for Image Data,” *Computer Graphics and Image Processing,* vol. 15, no. 2, pp. 113-129, 1981.

[140] G. Farnebäck, “Polynomial expansion for orientation and motion estimation,” PhD, Linkopings Universitet (Sweden), Sweden, 2002.

[141] J. Van de Weijer, and R. Van den Boomgaard, “Least squares and robust estimation of local image structure,” *International Journal of Computer Vision,* vol. 64, no. 2-3, pp. 143-155, Sep, 2005.