SUPER-RESOLUTION FROM UNREGISTERED ALIASED IMAGES

THÈSE N^{O} 3591 (2006)

PRÉSENTÉE LE 21 JUILLET 2006
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
Laboratoire de communications audiovisuelles 1
SECTION SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Patrick VANDEWALLE

Master in Electrical Engineering, Katholieke Universiteit Leuven, Belgique
de nationalité belge

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. M. Vetterli, Prof. S. Süsstrunk, directeurs de thèse
Prof. P. Milanfar, rapporteur
Prof. P. Vandergheynst, rapporteur
Prof. J. Zerubia, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2006

Contents

Abstract v

Résumé vii

Acknowledgments ix

Frequently Used Terms, Abbreviations, and Notation xi

1 Introduction 1
- 1.1 What is resolution? 2
- 1.2 Super-resolution imaging 3
- 1.3 Application domains 3
- 1.4 Aliasing 4
- 1.5 Thesis outline and contributions 4

2 Problem Setup 7
- 2.1 Sampling methods 7
  - 2.1.1 General multichannel sampling 9
  - 2.1.2 Multichannel sampling of bandlimited functions 10
  - 2.1.3 Examples 12
  - 2.1.4 Sampling kernel 15
- 2.2 Aliasing 15
- 2.3 Super-resolution imaging 18
  - 2.3.1 Image registration 19
  - 2.3.2 Image reconstruction 21
  - 2.3.3 Variations on super-resolution 22
- 2.4 Uniqueness of the solution 22
  - 2.4.1 General case 22
  - 2.4.2 Bandlimited signals 23
  - 2.4.3 Polynomials 26
- 2.5 Conclusions 29

3 Registration of Partially Aliased Signals 31
- 3.1 Planar motion estimation 31
- 3.2 Rotation estimation 32
- 3.3 Shift estimation 34
- 3.4 Planar motion estimation for partially aliased images 35
- 3.5 Reconstruction method 36
- 3.6 Results 40

3.7 Conclusions 47

4 Application to Digital Cameras 49
4.1 Leica DC250 camera 49
4.1.1 Spatial frequency response 49
4.1.2 Super-resolution image reconstruction 52
4.2 Sigma SD10 camera 57
4.2.1 Spatial frequency response 57
4.2.2 Super-resolution image reconstruction 57
4.3 Conclusions 58

5 Registration of Totally Aliased Signals using Subspace Methods 65
5.1 Solution using matrix rank 65
5.1.1 Method 65
5.1.2 Discussion 67
5.2 Solution using projections 69
5.2.1 Method 69
5.2.2 Discussion 70
5.3 Practical Issues 72
5.3.1 Images and higher-dimensional signals 72
5.3.2 Minimization 73
5.3.3 Heuristic approaches 74
5.4 Complexity 75
5.5 Results 77
5.6 Conclusions 83

6 Registration of Totally Aliased Signals using Gröbner Bases 85
6.1 Multichannel sampling as a set of polynomial equations 86
6.2 Gröbner bases 87
6.2.1 Affine varieties and ideals 88
6.2.2 The Ideal Membership Problem 90
6.2.3 Buchberger's algorithm 91
6.2.4 Solution of polynomial equations 92
6.3 Multichannel Sampling using Gröbner Bases 93
6.4 Multichannel Sampling under Noisy Conditions 96
6.5 Complexity and optimizations 99
6.6 Conclusions 104

7 Conclusions 105
7.1 Thesis summary 105
7.2 Future research 107

A Reproducible Research 109
A.1 Motivation 110
A.2 Related work 111
A.3 Practical method 112
A.4 Results 113
A.5 Conclusions and future work 113

Bibliography 115

Curriculum Vitae 127

Abstract

Aliasing in images is often considered as a nuisance. Artificial low frequency patterns and jagged edges appear when an image is sampled at a too low frequency. However, aliasing also conveys useful information about the high frequency content of the image, which is exploited in super-resolution applications. We use a set of input images of the same scene to extract such high frequency information and create a higher resolution aliasing-free image. Typically, there is a small shift or more complex motion between the different images, such that they contain slightly different information about the scene.

Super-resolution image reconstruction can be formulated as a multichannel sampling problem with unknown offsets. This results in a set of equations that are linear in the unknown signal coefficients but nonlinear in the offsets. This thesis concentrates on the computation of these offsets, as they are an essential prerequisite for an accurate high resolution reconstruction. If a part of the image spectra is free of aliasing, the planar shift and rotation parameters can be computed using only this low frequency information. In such a case, the images can be registered pairwise to a reference image. Such a method is not applicable if the images are undersampled by a factor of two or larger. A higher number of images needs to be registered jointly. Two subspace methods are discussed for such highly aliased images. The first approach is based on a Fourier description of the aliased signals as a sum of overlapping parts of the spectrum. It uses a rank condition to find the correct offsets. The second one uses a more general expansion in an arbitrary Hilbert space to compute the signal offsets. The sampled signal is represented as a linear combination of sampled basis functions. The offsets are computed by projecting the signal onto varying subspaces.

Under certain conditions, in particular for bandlimited signals, the nonlinear super-resolution equations can be written as a set of polynomial equations. Using Buchberger's algorithm, the solution can then be computed as a Gröbner basis for the corresponding polynomial ideal. After a description of a standard algorithm, adaptations are made for the use with noisy measurements.

The techniques presented in this thesis are tested in simulations and practical experiments. The experiments are performed on sets of real images taken with a digital camera. The results show the validity of the algorithms: registration parameters are computed with subpixel precision, and aliasing is accurately removed from the resulting high resolution image.

This thesis is produced according to the concepts of reproducible research. All the results and examples used in this thesis are reproducible using the code and data available online.

Keywords: registration, aliasing, super-resolution, sampling.

Résumé

En imagerie numérique, le repliement spectral est souvent considéré comme un défaut. Quand une image est échantillonnée à une fréquence trop basse, des basses fréquences artificielles et des bords en escalier apparaissent. Cependant, le repliement spectral contient de l'information utile sur le contenu à haute fréquence de l'image, qui peut être exploité dans les applications de super-résolution. Une série d'images de la même scène est utilisée pour extraire cette information à haute fréquence et construire une image à résolution plus élevée, ne contenant plus de repliement spectral. Typiquement, les images diffèrent par un léger décalage ou un mouvement plus complexe et présentent des informations légèrement différentes sur la scène.

La reconstruction d'images super-résolution peut être formulée comme un problème d'échantillonnage multi-canaux avec décalages inconnus. Cette formulation fournit une série d'équations linéaires pour les coefficients inconnus du signal mais non linéaires pour les décalages. Cette thèse se concentre sur le calcul de ces décalages, car ils sont nécessaires pour une précise reconstruction à haute résolution. Si une partie du spectre de l'image ne contient pas de repliement spectral, les paramètres de décalage et de rotation planaires peuvent être obtenus en utilisant uniquement l'information à basse fréquence. Dans un tel cas, les images peuvent être alignées deux par deux par rapport à une image de référence. Une telle méthode n'est pas utilisable si les images sont sous-échantillonnées par un facteur de deux ou plus. Un plus grand nombre d'images doivent être alignées simultanément. Deux méthodes sont présentées qui utilisent des sous-espaces pour aligner ces images fortement sous-échantillonnées. La première méthode est fondée sur une description du signal sous-échantillonné dans la base de Fourier comme une somme de différentes parties superposées du spectre. Elle utilise le rang d'une matrice pour trouver les paramètres. La deuxième méthode utilise une expansion du signal dans un espace de Hilbert pour calculer les paramètres d'alignement. Le signal échantillonné est représenté comme une combinaison linéaire des fonctions de base elles aussi échantillonnées. Les paramètres d'alignement sont calculés en projetant le signal sur des sous-espaces.

Sous certaines conditions, les équations non-linéaires de super-résolution peuvent être écrites comme une série d'équations polynomiales. La solution peut alors être obtenue avec l'algorithme de Buchberger qui calcule une base de Gröbner pour l'idéal polynomial correspondant à ces équations. Après la description de l'algorithme de base, des adaptations sont proposées pour son utilisation lorsque les mesures sont bruitées.

Les techniques présentées dans cette thèse ont été testées par simulations et expériences pratiques faites sur des séries d'images prises avec un appareil photographique numérique. Les résultats prouvent la validité des algorithmes: les paramètres d'alignement sont calculés avec une précision sous-pixélique, et l'image reconstruite à haute résolution ne contient plus de repliement spectral.

Cette thèse est générée selon les principes de la recherche reproductible. Tous les résultats et les exemples dans cette thèse sont reproductibles avec le code et les données disponibles en ligne.

Mots-clés: alignement, repliement spectral, super-résolution, échantillonnage.

Acknowledgments

During my time as a Ph.D. student, I have received help and support from many people.

First of all, I would like to thank my two advisors, Sabine Süsstrunk and Martin Vetterli for their support. They introduced me to signal and image processing during my exchange year at EPFL, and gave me the opportunity to do a Ph.D. in their lab. I would like to thank them for the great freedom they gave me in choosing research topics, and for the numerous inspiring discussions and support. I also appreciate very much the help from Dr. Luciano Sbaiz in the increasingly mathematical topics of my thesis. My research was funded by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

I would also like to thank the members of my jury, Prof. Peyman Milanfar, Prof. Emre Telatar, Prof. Pierre Vandergheynst, and Prof. Josiane Zerubia, for being on my thesis committee, and for reading and commenting this thesis.

Next, I would like to thank Dr. Paul Hubel, and the entire image processing team at Foveon for giving me the opportunity to do an internship in their group. This was a very enriching experience, and I am very grateful for it.

I also thank Dr. Urs Schmid from Leica Microsystems AG, and Rudy Guttosch from Foveon for the cameras they provided for my various experiments. Dr. Sina Farsiu and Prof. Peyman Milanfar helped me a lot by giving me their super-resolution software to experiment and compare my work with.

I am very grateful to all my LCAV colleagues for their friendship and support. And last but not least, I would like to thank my family and friends in Belgium and at various other places around the world. I thank my parents, my brother and my sister, for their support in everything I undertake. Finally, I am especially thankful to Els, for her unconditional support and love.

Frequently Used Terms, Abbreviations, and Notation

Terms and abbreviations

megapixel: one million (10^6) pixels

A/D: analog to digital;

CCD: charge coupled device (type of digital camera sensor);

CFA: color filter array;

CMOS: complementary metal oxide semiconductor (type of digital camera sensor);

ISO: International Organization for Standardization;

MTF: modulation transfer function;

OECF: opto-electronic conversion function;

MSE: mean squared error;

PSF: point spread function;

SFR: spatial frequency response.

Notation and variables

vectors are bold lowercase letters ($\mathbf{y}, \boldsymbol{\alpha}$, etc);
matrices are bold uppercase letters ($\mathbf{D}, \mathbf{F}$, etc);

$\otimes$	denotes the Kronecker product;
$\mathcal{B}$	basis for the Hilbert space $\mathcal{H}$;
d	degree of a polynomial;
$\mathbf{D}_{t_m}$	$L \times L$ diagonal matrix with elements $\mathbf{D}_{t_m}(l,l) = z_m^l$;
$\mathbf{D}'_{t_m}$	$N \times N$ diagonal matrix with elements $\mathbf{D}'_{t_m}(l,l) = z_m^l$; $\mathbf{D}'_{t_m}$ is the central $N \times N$ part of $\mathbf{D}_{t_m}$;
$f(t)$	continuous-time signal;
$f(\boldsymbol{x})$	continuous-space image;
$\mathbf{F}$	$L \times N$ discrete Fourier transform matrix

	with elements $\mathbf{F}(k,l) = e^{-j2\pi kl/N}$;
$\mathbf{f}_l$	l-th row of the matrix $\mathbf{F}$;
$\mathbf{F}_N$	$N \times N$ discrete Fourier transform matrix with elements $\mathbf{F}_N(k,l) = e^{-j2\pi kl/N}$;
G	Gröbner basis;
$H(\alpha)$	frequency content of an image as a function of the angle α;
$\mathcal{H}$	Hilbert space to which $f(t)$ belongs;
I	polynomial ideal;
K	maximum Fourier expansion coefficient ($K = (L-1)/2$, with L odd);
L	number of unknown expansion coefficients α_l;
M	number of sets of samples;
N	number of samples in each set $\mathbf{y}_m$;
p	polynomial;
$\mathbf{P}_{\mathbf{A}}$	projection onto the space spanned by $\mathbf{A}$;
Q	number of error evaluations in a minimization algorithm;
r	remainder in (polynomial) division;
$\mathbf{R}$	rotation matrix (with rotation angle θ_m);
S	number of overlapping parts of the Fourier spectrum ($S = \lceil L/N \rceil$);
$S(p_0, p_1)$	S-polynomial of the polynomials p_0 and p_1;
$\mathcal{S}_i$	subset of the total set of polynomial equations;
t	continuous time variable ;
t_m	offset of the m-th set of samples with respect to the first one;
$\mathbf{t}$	length M vector of the offsets t_m between the different sets of samples;
$\mathcal{T}$	set of sampling times;
$\boldsymbol{u}$	two-dimensional continuous frequency vector $\begin{pmatrix} u_h & u_v \end{pmatrix}^T$ in cartesian coordinates or $\begin{pmatrix} u_r & u_\theta \end{pmatrix}^T$ in polar coordinates;
V	polynomial variety;
$\mathcal{V}_l$	subspace of the MN-dimensional complex vector space $\mathbb{C}^{MN}$;
W	principal N-th root of unity, $e^{j2\pi/N}$;
$\mathbf{x}$	vector of unknowns in a polynomial;
$\boldsymbol{x}$	two-dimensional coordinate vector ($\boldsymbol{x} = \begin{pmatrix} x_h & x_v \end{pmatrix}^T$);
$\boldsymbol{x}_m$	two-dimensional offset vector of the m-th image with respect to the first one ($\boldsymbol{x}_m = \begin{pmatrix} x_{m,h} & x_{m,v} \end{pmatrix}^T$);
$\mathbf{y}_m$	length N vector with the m-th set of samples;
$\mathbf{Y}_m$	length N vector with the DFT of the m-th set of samples;
$\mathbf{Y}_{\mathbf{t}}^D$	matrix containing the modified Fourier transforms of all the sets of samples ($\mathbf{Y}_{\mathbf{t}}^D = \begin{pmatrix} \mathbf{Y}_0 & \mathbf{D'}_{t_1}^{-1}\mathbf{Y}_1 & \cdots & \mathbf{D'}_{t_{M-1}}^{-1}\mathbf{Y}_{M-1} \end{pmatrix}$);
z_m	offset-dependent variable in a Fourier basis ($z_m = e^{j2\pi t_m/N}$);
$\boldsymbol{\alpha}$	length L vector of the expansion coefficients α_l to be reconstructed;
α_l	l-th expansion coefficient of $f(t)$ in the basis $\mathcal{B}$ corresponding

	to $\varphi_l(t)$;
θ	angle (degrees);
θ_m	rotation angle between the first image (reference) and the m-th image;
ρ	image radius, or half the image size;
$\sigma_{S+1}(\mathbf{Y}^D_{\hat{\mathbf{t}}})$	$S+1$-th singular value of the matrix $\mathbf{Y}^D_{\hat{\mathbf{t}}}$;
$\varphi_l(t)$	basis function for the Hilbert space $\mathcal{H}$;
$\mathbf{\Phi}_{t_m}$	basis matrix sampled with offset t_m;
$\mathbf{\Phi}_{\mathbf{t}}$	combined basis matrix with the different offsets $\mathbf{t}$;
$\boldsymbol{\phi}^l_{\mathbf{t}}$	l-th column of the matrix $\mathbf{\Phi}_{\mathbf{t}}$.

Fourier Transforms

Fourier transform (FT) The (continuous time) Fourier transform is defined for continuous signals $f(t)$ as

$$F(u) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ut} dt.$$

Its inverse is called the inverse Fourier transform (IFT), and can be written as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(u) e^{j2\pi ut} du.$$

Fourier series (FS) The (continuous time) Fourier series is defined for periodic, continuous signals $f(t)$ with period T as

$$Y(k) = \frac{1}{T} \int_{0}^{T} f(t) e^{-j2\pi kt/T}, \quad k = -\infty, ..., \infty.$$

Its inverse is called the inverse Fourier series (IFS):

$$f(t) = \sum_{k=-\infty}^{\infty} Y(k) e^{j2\pi kt/T}.$$

Discrete Fourier transform (DFT) The discrete Fourier transform is defined for finite length, discrete time signals $y(n)$ as

$$Y(k) = \sum_{n=0}^{N-1} y(n) W^{-kn}, \quad k = 0, ..., N-1,$$

with $W = e^{j2\pi/N}$. This can also be written in matrix notation as

$$\mathbf{Y} = \mathbf{F}_N \mathbf{y},$$

with $\mathbf{F}_N(k,l) = W^{-kl}$. A fast implementation is given by the fast Fourier transform (FFT) algorithm. Its inverse is called the inverse discrete Fourier transform (IDFT):

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) W^{kn},$$

or in matrix notation

$$\mathbf{y} = \mathbf{F}_N^* \mathbf{Y},$$

where $\mathbf{F}_N^*$ denotes the Hermitian transpose of $\mathbf{F}_N$.

Chapter 1

Introduction

Over the past ten years, digital cameras have gone through a fast evolution towards extremely compact models, containing sensors with a steadily increasing number of pixels. From about 0.3 megapixels (million pixels) in 1993, the number of pixels on the CCD or CMOS sensor in a digital camera has increased to 39 megapixels in some of the latest professional models. This pixel count has become the major selling argument for the different camera manufacturers.

The number of pixels in a digital image is also often referred to as the resolution of an image. The ever-increasing demand for more pixels, or higher resolution, in combination with the availability of more and more computational power, has generated a large interest in super-resolution imaging. The goal in super-resolution imaging is to take multiple 'low' resolution images of the same scene, and combine them to generate a 'higher' resolution image. In this way, a photographer could for example take a series of four images using a four megapixel camera, and combine them to obtain an image as if it would be taken with a sixteen megapixel camera. And who would not be interested in such a feature?

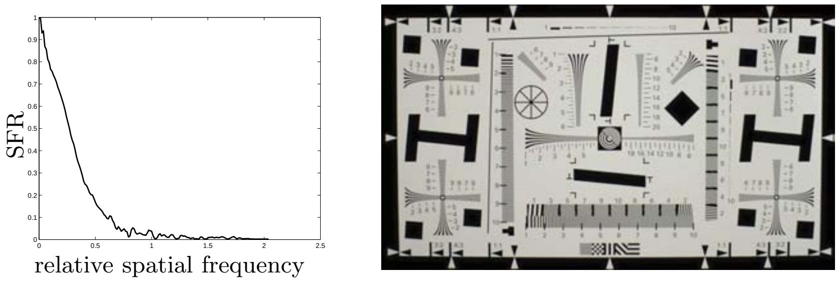
In practice, such a combination of information from multiple images is not trivial. There are two main problems that need to be solved in a super-resolution algorithm. First, all the input images need to be correctly aligned with each other on a common grid. Next, an accurate, sharp image has to be reconstructed from the gathered information. If one of these two steps is not well done, the resulting image is not good, and no gain in resolution is obtained. In this thesis, we mainly address the first problem, more specifically the alignment of aliased input images. An image is aliased if there are not enough sampling points (pixels) to represent the high frequencies in the scene. This typically results in artificial patterns or jagged edges in the image. If the images are not too severely aliased, we will show that it is possible to use the aliasing-free part of the images to align them one by one to a reference image. If there is severe aliasing, the different images need to be aligned jointly. In that case, the alignment is a highly nonlinear problem. Multiple solution methods for such an alignment are proposed. The presented image alignment methods can be applied to different application domains, such as consumer digital cameras, satellite imaging, etc.

1.1 What is resolution?

First of all, we need to define what we understand by the term ‘resolution’. If we take a single image, and multiply its size by four by repeating each pixel value four times, do we gain resolution? On the other hand, let us apply a blurring filter to an image. The resulting image still has the same size, but does it have the same resolution?

The above examples show that there is more to resolution than just counting the number of pixels that are present in the image. It is related to the ability to *distinguish details* in the image, in other words, to its *resolving power*.

The International Organization for Standardization (ISO) has described a precise method to *measure the resolution* of a digital camera [45]. The visual resolution can be measured as the highest frequency pattern of black and white lines where the individual black and white lines can still be visually distinguished in the image. It is expressed in line widths per picture height (LW/PH). The standard also describes a method to compute the spatial frequency response (SFR) of a digital camera. The spatial frequency response is the digital imaging equivalent of the modulation transfer function (MTF) used in analog imaging systems. It describes the variation between the maximum and minimum values that is visible as a function of the spatial frequency (the number of black and white lines per millimeter). It can be measured using an image of a slanted black and white edge, and is expressed in relative spatial frequencies (relative to the sampling frequency), line widths per picture height, or cycles per millimeter on the image sensor. An example of an SFR and the resolution chart that is used in the ISO standard are shown in Figure 1.1.



(a) (b)

Figure 1.1: Spatial frequency response (SFR). (a) Example of an SFR for a digital camera. A relative spatial frequency of 1 corresponds to the sampling frequency, to 1512 LW/PH, and to 55 cycles/mm on the image sensor. (b) ISO resolution chart used to compute the SFR of a digital camera.

Thus, we can define resolution as [45]:

Definition 1.1.1 (Resolution). Optical resolution is a measure of the ability of a camera system, or a component of a camera system, to depict picture detail. Image resolution is a measure of the amount of detail that is visible in an image.

1.2 Super-resolution imaging

If we want to increase the resolution of an image using super-resolution techniques, we essentially want to be able to distinguish more details in the final image. By adding images of the same scene, we try to add information to the reproduction. Typically this information is *high frequency* content of the scene.

There are different ways to add such high frequency information to an image. If we know that the image is of a certain type (faces, text, drawings, etc.), we can use that knowledge to add frequency content. Such an approach is called a model-based approach. For example, if we know that the images represent printed text, we can try to recognize characters, and replace them by sharp, high quality characters. The knowledge of the image model allows us to compute high frequency information. In this thesis, we will investigate abstract approaches to super-resolution. They use other information than a precise image model, and are therefore applicable to more general types of images. More specifically, we will compute the high frequency information from the aliasing that is present in the images.

Super-resolution techniques use a number of low resolution input images to generate a high resolution image. This assumes that there are some (small) differences between the input images. Most often, these differences are caused by small camera movements. In an ideal situation, we could assume that of four images taken, the second to fourth image have a horizontal, vertical, and diagonal shift of half a pixel compared to the first image. The pixels from the first image can then be interleaved with pixels from the three other images, and a double resolution image (in both dimensions) is obtained (see Figure 1.2).

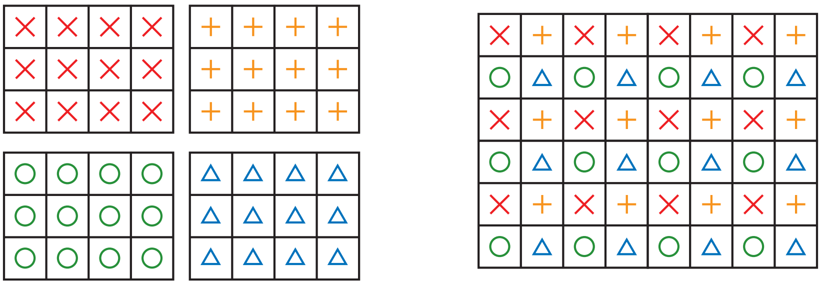


Figure 1.2: Ideal super-resolution setup. Four images are taken with relative shifts of half a pixel in horizontal, vertical, and diagonal directions (left). Their pixels can then be interleaved to generate a double resolution image (right).

In general, however, the shifts between the images are not exactly half a pixel, and can take any arbitrary value. Moreover, in most applications the motion parameters are unknown, and need to be computed first. In the next chapters, we will present methods to compute these motion parameters.

1.3 Application domains

Super-resolution techniques can be applied in various domains. As described above, in *consumer imaging*, one could imagine a digital camera that takes a burst of pictures instead of a single picture. From these images, which have typically small relative shifts due to the shaking of the user's hands, a high resolution image can be reconstructed. Similarly, in *satellite or spatial imaging*, a set of satellite images could be combined to display fine details that are not distinguishable in any of the input images [3]. Super-resolution methods can also be used to create high resolution still pictures or video from video sequences. In *surveillance cameras*, additional details can be revealed by combining multiple video frames to create a single high resolution image. The same techniques can also be applied to improve the resolution of existing (low resolution) video content for use in *high definition television* sets.

A similar approach for one-dimensional signals is used in *high rate analog-to-digital (A/D) converters.* If the rate at which the analog signal has to be sampled becomes too high, it is physically very difficult to build such converters. Instead of a single converter at a high rate, multiple converters at a lower rate are then used in parallel [41]. Each of the low rate converters has a small relative offset, such that the high rate signal can be reconstructed by combining the different low rate signals. In the ideal case, with for example two low rate converters, the samples of the second converter are taken exactly in the middle between the samples of the first converter. The two signals can then be interleaved to obtain a signal at twice the rate. However, the precise synchronization of such converters is very difficult [22]. This is exactly the same problem as the alignment of images in super-resolution imaging, but now for one-dimensional signals. The methods described in the following chapters can therefore also be applied to such problems.

1.4 Aliasing

We will mainly consider aliased images as input for our super-resolution algorithms. According to the sampling theorem, aliasing occurs when an image is sampled at a frequency lower than twice the maximum signal frequency present in the image [57, 75, 96, 124]. The high frequencies from the original signal are converted into low frequencies in the sampled image. Such aliased frequencies can no longer be distinguished from accurate low frequency signals.

In images, aliasing artifacts appear as artificial low frequency patterns or staircase effects along sharp edges. Illustrations of both effects are shown in Figure 1.3.

1.5 Thesis outline and contributions

In this thesis, we present super-resolution methods for unregistered, aliased images. In *Chapter 2*, we give a more formal description of the problem we are addressing. We first give an overview of existing sampling methods, and identify super-resolution image reconstruction as a multichannel sampling problem with unknown offsets. This can then be expressed as a set of equations that are linear in the unknown signal coefficients, but nonlinear in the offsets. Our new formulation allows the clear identification of aliasing, and forms a basis for the reconstruction algorithms proposed in Chapters 5 and 6. Our setup is first worked out for signals in an arbitrary Hilbert space and is then applied to bandlimited signals in a Fourier series setup. This will be the case that is

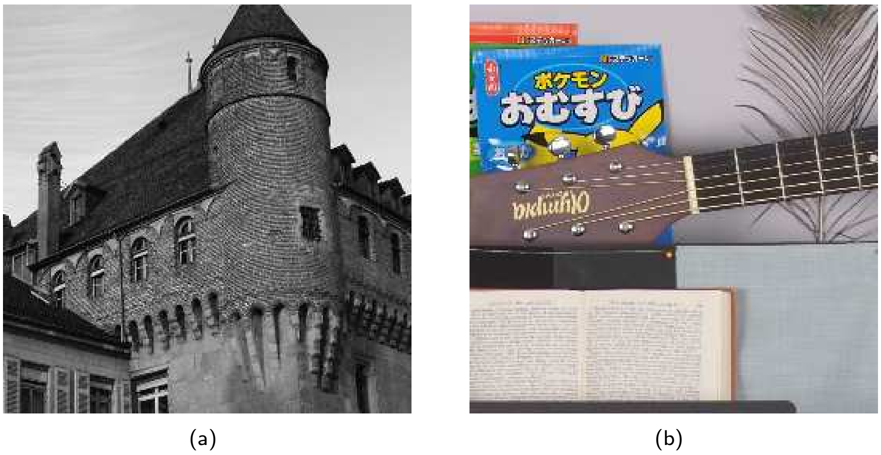

(a) (b)

Figure 1.3: Examples of aliasing in digital images. (a) Low frequency ringing artifacts appear in the walls of the building, replacing the original lines of the bricks. (b) Staircase effect in the guitar strings.

mostly used in practice. We also give a mathematical description of aliasing for bandlimited signals, and we extend it to signals in an arbitrary Hilbert space. Then, we give an overview of existing super-resolution methods. We treat the frequency domain registration methods in more detail, as they are closely related to our own methods. Finally, we discuss the uniqueness of a solution to the super-resolution problem with unregistered sets of samples.

In *Chapter 3*, we present a super-resolution method for partially aliased images. The main novelty of our method is in the image registration. We describe a new frequency domain algorithm for registration that is based on the aliasing-free part of the spectrum: the low frequencies. We use a planar motion model, and show that in frequency domain, planar shift and rotation in the image plane can be estimated separately. For the rotation estimation, we present a method that projects the frequency content along radial lines, and then computes a one-dimensional correlation between two images. The shift between a pair of images is then computed by fitting a plane through the phase difference between the two images. We use a bicubic interpolation method to reconstruct a high resolution image from the registered input images. We test our super-resolution algorithm in a number of simulations. The registration method is compared to other frequency and spatial domain image registration methods, and we can show that our algorithm outperforms the other algorithms.

This method is applied to consumer digital cameras in *Chapter 4*. We first measure the SFR for our two cameras (a Leica DC250 and a Sigma SD10) to verify that they allow aliasing. Then, we reconstruct high resolution images from sets of aliased input images using the algorithm from Chapter 3. Again, our method is compared to some other methods.

Three super-resolution methods for totally aliased images (i.e., where there is no aliasing-free part of the spectrum) are presented in Chapters 5 and 6. In such a case, the images can not be registered pairwise with a reference image anymore. The complete set of images has to be registered jointly. In *Chapter 5*, we present two methods that are based on a subspace approach derived from the formulation in Chapter 2. The first algorithm uses the dimensionality of the subspace to compute the correct offsets. It is a frequency domain approach, and is therefore only applicable to bandlimited signals. Next, we present an algorithm that uses projections onto lower-dimensional subspaces, and is applicable to signals in an arbitrary Hilbert space. These methods are tested in simulations on one-dimensional and two-dimensional signals, and their good behavior in noisy circumstances is shown. For the method described in *Chapter 6*, we first reformulate the equations from Chapter 2 as a set of polynomial equations in the unknown offsets and signal parameters. These equations can then be solved using Buchberger's algorithm for Gröbner bases. The basic results from the Gröbner basis theory are summarized, and this algorithm is applied to some examples. Because Buchberger's algorithm can only be used with exact equations with infinite precision, we also present an adaptation to this algorithm for noisy and finite precision measurements. Finally, we discuss the complexity of such methods.

Some of the mathematics in Chapters 3-6 are described on one-dimensional signals to simplify notations and give intuitions. They can be straightforwardly extended to two-dimensional signals.

Finally, *Chapter 7* concludes this thesis and discusses some ideas for future work. All the work that is presented in this thesis is also reproducible. An appendix on reproducible research is given in *Appendix A*. We discuss the importance of making research reproducible, and give a possible method for doing reproducible research.

Chapter 2

Problem Setup

In this chapter, we give a mathematical formulation of super-resolution image reconstruction in a sampling framework. First, we give a classification of existing sampling methods and write one-dimensional super-resolution reconstruction as a multichannel sampling problem with unknown offsets. Then, we give an interpretation of aliasing and aliased sampling using projection operators. This allows us to extend some of the frequency domain intuitions about aliasing to arbitrary Hilbert spaces. Next, we give an overview of existing super-resolution methods, and situate our approach in this context. And finally we show under which conditions there exists a unique solution to the super-resolution problem.

2.1 Sampling methods

Let us consider a finite dimensional[1] Hilbert space $\mathcal{H}$, for which we have a basis $\mathcal{B} = \{\varphi_l(t)\}_{l=0..L-1}$. That is, $\mathcal{H} = \text{span}\{\varphi_l(t)\}_{l=0..L-1}$. For simplicity, let us consider the functions $\varphi_l(t)$ defined on the interval $[0, 1]$. For periodic functions, we will assume the period to be 1, such that we consider one period. An arbitrary continuous-time signal $f(t)$ from this space can be expressed as

$$f(t) = \sum_{l=0}^{L-1} \alpha_l \varphi_l(t), \tag{2.1}$$

where α_l is the l-th expansion coefficient of $f(t)$ in the L-dimensional basis $\mathcal{B}$. Possible examples of spaces with associated bases include truncated Fourier series, wavelets, splines, etc.

Assume now that we sample $f(t)$ at times $\mathcal{T}$, resulting in the sampled signal $y(n)$:

$$y(n) = f(\mathcal{T}(n)). \tag{2.2}$$

Sampling methods can be classified into different categories, according to the way the sampling times $\mathcal{T}$ are chosen (see Figure 2.1). A recent overview of

---

[1] The extension to an infinite dimensional Hilbert space is theoretically possible, but of little practical interest since all data (one-dimensional signals and images) and algorithms are finite.

sampling methods is given by Unser [108]. If the samples are taken uniformly, at a constant rate N, we have *uniform sampling* (see Figure 2.2(a)):

$$\mathcal{T} = \begin{pmatrix} 0 & \frac{1}{N} & \frac{2}{N} & \cdots & \frac{N-1}{N} \end{pmatrix}. \tag{2.3}$$

The sampled signal can be written as

$$y(n) = f\left(\frac{n}{N}\right), \text{ for } 0 \leq n < N. \tag{2.4}$$

This is the standard sampling setup as it is most often used, and as it is also presented in the sampling theorem by Whittaker [124], Nyquist [75], Kotel'nikov [57] and Shannon [96] (see also [65] for a discussion about the origins of the sampling theorem).

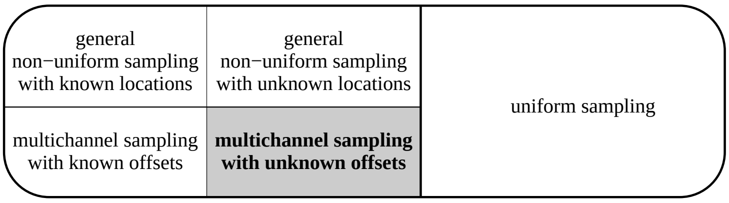
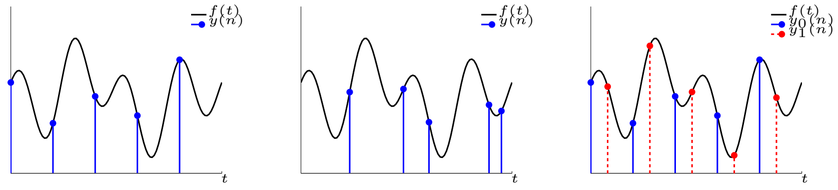

general non−uniform sampling with known locations
general non−uniform sampling with unknown locations
uniform sampling
multichannel sampling with known offsets
multichannel sampling with unknown offsets

Figure 2.1: Classification of sampling methods. Sampling methods can be divided into uniform and non-uniform methods. The non-uniform sampling methods can be subdivided depending on whether the locations are known and whether the samples are grouped in uniform sets with only unknown offsets. In super-resolution, we are interested in multichannel sampling methods with unknown offsets.



(a) Uniform sampling (b) Nonuniform sampling (c) Multichannel sampling

Figure 2.2: Illustration of the different sampling methods.

When the samples are not chosen uniformly, the sampling methods are logically called non-uniform (Figure 2.2(b)). Among the *non-uniform sampling* methods, a distinction needs to be made between methods where the sampling instants $\mathcal{T}$ are known [3, 67, 102], and other methods where the sampling locations are unknown. If the sampling locations are unknown and completely arbitrary, the problem cannot be solved. This can be shown using a simple counting argument. Assume that the signal to be reconstructed has L unknown parameters. For every additional sample, there is also an additional unknown (its location). Therefore the number of unknowns is always larger than the number of measurements, and this problem is unsolvable. However, for discrete signals, where the sampling locations can only take a finite number of values, a combinatorial solution can be found, as described by Marziliano and Vetterli [69].

2.1.1 General multichannel sampling

An important subset of the non-uniform sampling methods is formed by multichannel sampling methods. In these methods, the set of sampling instants $\mathcal{T}$ can be divided into M sets of uniformly spaced samples $\mathcal{T}_m$. Each of the sets of samples $\mathcal{T}_m$ is uniform, but the different sets have an arbitrary offset t_m (Figure 2.2(c)). Note that t_m is expressed in samples. In such a case, the m-th set of samples can be written as

$$\begin{aligned} y_m(n) &= f\left(\mathcal{T}_m\right) = f\left(\frac{n+t_m}{N}\right) \\ &= \sum_{l=0}^{L-1} \alpha_l \varphi_l \left(\frac{n+t_m}{N}\right), \text{ for } 0 \leq n < N. \end{aligned} \tag{2.5}$$

Papoulis described a solution for multichannel sampling with known sampling locations [77]. He showed that a bandlimited signal can be perfectly reconstructed from M sets of samples that are uniformly sampled at $1/M$ the Nyquist sampling rate. This result was extended by Unser and Zerubia in their generalized sampling approach [109, 110]. The problem with multiple sets and unknown sampling locations was solved for discrete-time signals by Marziliano and Vetterli [69]. They developed a combinatorial method to compute the discrete offsets between the different sets of samples. In this thesis, we will study the continuous-time case: multichannel sampling with unknown, real-valued offsets t_m.

Using vector notation, (2.5) can be written more compactly as

$$\mathbf{y}_m = \mathbf{\Phi}_{t_m} \boldsymbol{\alpha}. \tag{2.6}$$

In this equation, $\mathbf{y}_m$ is the $N \times 1$ vector containing the m-th uniform set of samples, and $\boldsymbol{\alpha}$ is the $L \times 1$ vector of expansion coefficients. The $N \times L$ matrix $\mathbf{\Phi}_{t_m}$ contains the sampled basis functions that are uniformly sampled with an offset t_m.

Now, all the sets of samples $\mathbf{y}_m$ are combined into a single vector $\mathbf{y}$ and similarly the basis matrices $\mathbf{\Phi}_{t_m}$ are combined into $\mathbf{\Phi}_{\mathbf{t}}$, with $\mathbf{t} = \begin{pmatrix} t_0 & t_1 & \cdots & t_{M-1} \end{pmatrix}$ denoting the offset vector. This can be written as

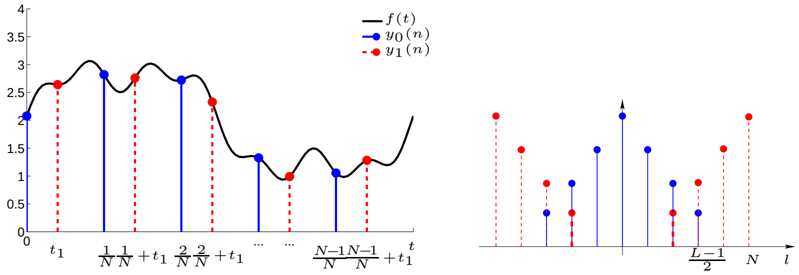
$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{M-1} \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi}_{t_0} \\ \mathbf{\Phi}_{t_1} \\ \vdots \\ \mathbf{\Phi}_{t_{M-1}} \end{pmatrix} \boldsymbol{\alpha} = \mathbf{\Phi}_{\mathbf{t}} \boldsymbol{\alpha}. \tag{2.7}$$

The matrix $\mathbf{\Phi}_{\mathbf{t}}$ has size $MN \times L$. Assuming that the total number of samples is larger than or equal to the number of expansion coefficients, or $MN \geq L$, this set of equations is in general well- or over-determined if $\mathbf{t}$ is known. If, additionally,

$$MN \geq L + M - 1, \tag{2.8}$$

the number of equations is also larger than or equal to the number of unknowns (L expansion coefficients and $M-1$ offsets), and it should be possible to remove the uncertainty of the unknown offsets. As we will show in the next chapters, these additional equations allow us in general to compute the unknown offsets. Note that the challenging part of the problem is that it is a *nonlinear problem in the unknown offsets and expansion coefficients.* In summary, the most important variables in this reconstruction problem are listed here (see also Figure 2.3):

- N : the number of samples in each set $\mathbf{y}_m$,
- $\mathbf{y}_m$: the length N vector of the m-th set of samples,
- L : the number of unknown expansion coefficients,
- $\boldsymbol{\alpha}$: the length L vector of the expansion coefficients α_l to be reconstructed,
- M : the number of sets of samples,
- $\mathbf{t}$: the length M vector of the offsets t_m between the different sets of samples.



(a) (b)

Figure 2.3: Illustration of the different variables with $M = 2$ and a Fourier basis. (a) Time domain representation of the signal $f(t)$ and its sets of samples $y_0(n)$ (—) and $y_1(n)$ (– –). (b) Frequency domain representation of the absolute values of the signal spectrum (—) and its aliased copies after sampling (– –).

The unknown variables are the expansion coefficients $\boldsymbol{\alpha}$ and the offsets $\mathbf{t}$. We assume that all the other variables are known. This is evident for the sets of samples $\mathbf{y}_m$, the number of samples per set N and the number of sets M, as they form the input of the problem. We will also require that the number of signal coefficients L, or at least an estimate for L, is available.

2.1.2 Multichannel sampling of bandlimited functions

As we will often use a Fourier basis, we specifically write out the equations (2.1)-(2.7) for this case. First we define periodic bandlimited signals.

Definition 2.1.1 (Bandlimited signal). A periodic signal $f(t)$ with period 1 is bandlimited if it can be expressed in the Fourier basis with a finite number $L = 2K + 1$ of coefficients α_l. It can be written as a Fourier series:

$$f(t) = \sum_{l=-K}^{K} \alpha_l e^{j2\pi lt}, \tag{2.9}$$

where α_l are the Fourier coefficients.

The coefficients are indexed from $-K$ to K (instead of 0 to $L-1$), because it is the usual way of indexing Fourier coefficients. For a real signal $f(t) \in \mathbb{R}$, the negative frequency coefficients are the complex conjugate of the positive coefficients: $\alpha_{-k} = \alpha_k^*$. We will assume here that L is odd, such that $K = (L-1)/2$.

The first set of samples can be written as

$$y_0(n) = f\left(\frac{n}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{ln}, \tag{2.10}$$

with $W = e^{j2\pi/N}$. We can also write this in matrix notation as

$$\mathbf{y}_0 = \begin{pmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ W^{-K} & \cdots & W^{-1} & 1 & W & \cdots & W^{K} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ W^{-(N-1)K} & \cdots & W^{-(N-1)} & 1 & W^{N-1} & \cdots & W^{(N-1)K} \end{pmatrix} \begin{pmatrix} \alpha_{-K} \\ \vdots \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_K \end{pmatrix} = \mathbf{F}^* \boldsymbol{\alpha}, \tag{2.11}$$

where $\mathbf{F}^*$ is an $N \times L$ inverse discrete Fourier transform (IDFT) matrix. The notation $\mathbf{F}^*$ is used for the Hermitian transpose of the forward transform matrix $\mathbf{F}$. Due to the undersampling ($N < L$), some of the columns in $\mathbf{F}^*$ are repeated. The principal N-th root of unity is used for W ($W = e^{j2\pi/N}$), which means that the n-th column is the same as the $n + iN$-th columns (with i any integer such that $n + iN \leq L$), for any n between 1 and N. It is important to note here that $\mathbf{F}^*$ is not part of a larger $L \times L$ Fourier matrix $\mathbf{F}_L^*$, but rather an $N \times N$ Fourier matrix $\mathbf{F}_N^*$ that is extended by repeating columns up to size $N \times L$.

Similarly, for the m-th set of samples $y_m(n)$, we obtain

$$y_m(n) = f\left(\frac{n + t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l e^{\frac{j2\pi l(n+t_m)}{N}} = \sum_{l=-K}^{K} \alpha_l W^{ln} z_m^l, \tag{2.12}$$

with $z_m = e^{j2\pi t_m/N}$. In matrix notation, this can be expressed as

$$\mathbf{y}_m = \mathbf{F}^* \mathbf{D}_{t_m} \boldsymbol{\alpha}, \tag{2.13}$$

with $\mathbf{F}^*$ the $N \times L$ IDFT matrix defined above, and $\mathbf{D}_{t_m}$ an $L \times L$ diagonal matrix with elements $\mathbf{D}_{t_m}(l, l) = z_m^l$ $(-K \leq l \leq K)$. If we put all the M sets of N samples together, we obtain the Fourier basis equivalent of (2.7):

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{M-1} \end{pmatrix} = \begin{pmatrix} \mathbf{F}^* \\ \mathbf{F}^*\mathbf{D}_{t_1} \\ \vdots \\ \mathbf{F}^*\mathbf{D}_{t_{M-1}} \end{pmatrix} \boldsymbol{\alpha}. \tag{2.14}$$

The DFT of a set of samples $\mathbf{y}_m$ can be written as

$$\mathbf{Y}_m = \frac{1}{N}\mathbf{F}_N \mathbf{y}_m = \frac{1}{N}\mathbf{F}_N \mathbf{F}^* \mathbf{D}_{t_m} \boldsymbol{\alpha}, \tag{2.15}$$

where $\mathbf{F}_N$ is a square $N \times N$ DFT matrix, and $\mathbf{F}^*$ is the $N \times L$ matrix defined in (2.11). The resulting vector $\mathbf{Y}_m$ has length N and is an aliased and phase shifted version of $\boldsymbol{\alpha}$. If we take for example $L = 3N$, we get

$$\begin{aligned} \mathbf{Y}_m &= \frac{1}{N}\mathbf{F}_N \mathbf{F}^* \mathbf{D}_{t_m} \boldsymbol{\alpha} = \frac{1}{N}\mathbf{F}_N \begin{pmatrix} \mathbf{F}_N^* & \mathbf{F}_N^* & \mathbf{F}_N^* \end{pmatrix} \mathbf{D}_{t_m} \boldsymbol{\alpha} \\ &= \begin{pmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \end{pmatrix} \mathbf{D}_{t_m} \boldsymbol{\alpha} = \sum_{i=-1}^{1} z_m^{iN} \mathbf{D}'_{t_m} \boldsymbol{\alpha}_i, \end{aligned} \tag{2.16}$$

where $\mathbf{D}'_{t_m}$ is the $N \times N$ central part of the $L \times L$ matrix $\mathbf{D}_{t_m}$, and $\boldsymbol{\alpha}_i$ is the i-th block of length N from the vector $\boldsymbol{\alpha}$. In general, if L is not a multiple of N, we can still perform the same decomposition by adding zeros to $\boldsymbol{\alpha}$ up to the next multiple of N.

2.1.3 Examples

We will now illustrate the description given above with some small examples for bandlimited signals, polynomial signals, and piecewise polynomial signals. By piecewise linear or polynomial signals we understand in this thesis periodic and continuous functions with period 1 that are piecewise linear or polynomial, respectively, on uniform intervals.

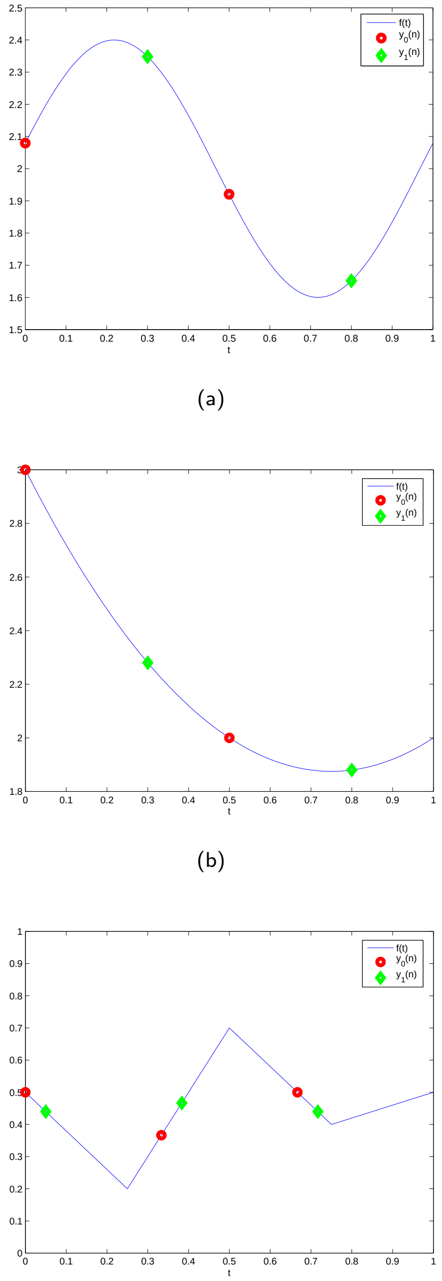
Example 2.1.1 (Bandlimited functions). Take $f(t)$ a bandlimited function (see Figure 2.4(a)), and $M = 2$, $N = 2$, and $L = 3$. Using (2.7), we can write

$$\mathbf{y} = \left(\begin{array}{c} y_0(0) \\ y_0(1) \\ \hline y_1(0) \\ y_1(1) \end{array}\right) = \left(\begin{array}{ccc} 1 & 1 & 1 \\ -1 & 1 & -1 \\ \hline z_1^{-1} & 1 & z_1 \\ -z_1^{-1} & 1 & -z_1 \end{array}\right) \begin{pmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \boldsymbol{\Phi}_{\mathbf{t}} \boldsymbol{\alpha}. \tag{2.17}$$

The unknown offset in the matrix $\boldsymbol{\Phi}_{\mathbf{t}}$ is multiplied with the expansion coefficients in $\boldsymbol{\alpha}$ and makes the problem nonlinear. As in (2.13), we can write $\boldsymbol{\Phi}_{t_1}$ (the lower part of $\boldsymbol{\Phi}_{\mathbf{t}}$) as an IDFT matrix multiplied with a diagonal matrix:

$$\boldsymbol{\Phi}_{t_1} = \begin{pmatrix} z_1^{-1} & 1 & z_1 \\ -z_1^{-1} & 1 & -z_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_1^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & z_1 \end{pmatrix} = \mathbf{F}^* \mathbf{D}_{t_1}. \tag{2.18}$$

□



(a)

(b)

(c)

Figure 2.4: Examples of the general setup. (a) Bandlimited function with two sets of two samples and unknown offset between the sets. (b) Second-degree polynomial with two sets of two samples and unknown offset between the sets. (c) One period of a piecewise linear function with two sets of three samples and unknown offset between the sets.

Example 2.1.2 (Second order polynomials). Let $f(t)$ be a second degree polynomial that is sampled using two sets of two samples (see Figure 2.4(b)). We can write $f(t)$ as $f(t) = \alpha_0 t^2 + \alpha_1 t + \alpha_2$. In this case, we have $M = 2$, $N = 2$, and $L = 3$. Note that here, we do not consider a periodic signal, but the samples are still taken in the interval $[0, 1]$. The rest of the setup is exactly as described above, and (2.7) becomes

$$\mathbf{y} = \left(\begin{array}{c} y_0(0) \\ y_0(1) \\ \hline y_1(0) \\ y_1(1) \end{array}\right) = \left(\begin{array}{ccc} 0 & 0 & 1 \\ 0.25 & 0.5 & 1 \\ \hline 0.25t_1^2 & 0.5t_1 & 1 \\ (0.5+0.5t_1)^2 & 0.5+0.5t_1 & 1 \end{array}\right) \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \mathbf{\Phi_t}\boldsymbol{\alpha}. \tag{2.19}$$

We can again see that the unknown offset t_1 appears in the matrix $\mathbf{\Phi_t}$, and is multiplied with the unknown signal coefficients in $\boldsymbol{\alpha}$, making the equations nonlinear. □

Example 2.1.3 (Piecewise linear functions). Consider a continuous, periodic function with period 1 that is piecewise linear on uniform intervals $[l/L, (l+1)/L]$ (with $0 \le l < L$). This is the space $\mathcal{H} = \operatorname{span}\{\varphi_{l,\mathrm{circ}}(t)\}$ for $0 \le l < L$, with

$$\varphi(t) = \begin{cases} -L+1+Lt & 1-1/L < t < 1 \\ 1-Lt & 0 \le t < 1/L \\ 0 & \text{otherwise,} \end{cases} \qquad \text{with } t \in [0,1) \tag{2.20}$$

and $\varphi_{l,\mathrm{circ}}(t)$ the periodic extension of $\varphi(t - l/L)$ with period 1. An example is given in Figure 2.4(c) for $L = 4$. In this case, we can write the signal $f(t)$ as

$$f(t) = \alpha_0\varphi_0(t) + \alpha_1\varphi_1(t) + \alpha_2\varphi_2(t) + \alpha_3\varphi_3(t). \tag{2.21}$$

If we take two sets of three samples, we obtain the following equations:

$$\left(\begin{array}{c} y_0(0) \\ y_0(1) \\ y_0(2) \\ \hline y_1(0) \\ y_1(1) \\ y_1(2) \end{array}\right) = \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 2/3 & 1/3 & 0 \\ 0 & 0 & 1/3 & 2/3 \\ \hline 1-t_1/3 & t_1/3 & 0 & 0 \\ 0 & 2/3-t_1/3 & 1/3+t_1/3 & 0 \\ 0 & 0 & 1/3-t_1/3 & 2/3+t_1/3 \end{array}\right) \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}. \tag{2.22}$$

We assume that the second set of samples is taken in the same $[l/L, (l+1)/L]$ interval as the first set (in this example $0 < t_1 \le 1/4$). Other values of t_1 do not cause any fundamental changes, they simply change the above equations because the basis function $\varphi(t)$ is described by different equations on the intervals $[1-1/L, 1]$, $[0, 1/L]$ and outside of these intervals, as can be seen in (2.20). Because of the finite support of the basis functions, the matrices $\mathbf{\Phi}_{t_m}$ have a banded structure. □

From the above examples, we see that the unknown offsets and signal coefficients are mixed up, which makes the problem hard to solve. Once the offsets are known, (2.7) is reduced to a set of linear equations in the unknown signal coefficients. This can be solved straightforwardly using a least squares method.

2.1.4 Sampling kernel

In our sampling model, we will not consider the sampling kernel, also often known as the point spread function (PSF) in two-dimensional (imaging) systems. We will assume that the signals are sampled using Diracs. Although this is not very realistic in practical applications, it is an approximation that is often made to simplify the analysis. In frequency domain, it is in most cases not very difficult to take a different sampling kernel into account. The sampling operation can be considered as a convolution with a (generally lowpass filtering) sampling kernel, followed by the actual sampling. This convolution can be seen as a multiplication in frequency domain with the Fourier transform of the sampling kernel. Therefore, as long as the sampling kernel does not remove frequencies, the reconstructed function can always be divided by the sampling kernel function again (in Fourier domain). This should cancel the effect of the sampling kernel. Of course, this supposes that the sampling kernel is known, space/time invariant, and not too ill-conditioned. For space-varying sampling kernels, spatial domain super-resolution methods are better suited, such as the methods presented by Tekalp et al. [103] or Farsiu et al. [31].

2.2 Aliasing

Aliasing is most often considered in a *Fourier domain setup.* A sampled signal is aliased if the sampling frequency is lower than twice the maximum signal frequency, or in our setup $N < 2K$ [96]. Due to the periodicity of the sampled spectrum, frequencies above half the sampling frequency are replicated in the base spectrum of the sampled signal. In this way, a frequency above half the sampling frequency is mapped onto a frequency below this limit after sampling and the two cannot be distinguished anymore from their samples. Or, using (2.10), the k-th Fourier coefficient of $y_0(n)$ can be written as:

$$\begin{aligned} Y_0(k) &= \frac{1}{N}\sum_{n=0}^{N-1} y_0(n)W^{-kn} = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{l=-K}^{K}\alpha_l W^{ln}W^{-kn} \\ &= \frac{1}{N}\sum_{l=-K}^{K}\alpha_l \sum_{n=0}^{N-1} W^{(l-k)n} = \sum_i \alpha_{k+iN}, \qquad \text{with } 0 \leq k < N. \end{aligned} \tag{2.23}$$

Similarly, we can write the Fourier coefficient vector $\mathbf{Y}_0$ using the notation from (2.16) as a sum of the overlapping parts $\boldsymbol{\alpha}_i$ of the spectrum:

$$\mathbf{Y}_0 = \frac{1}{N}\mathbf{F}_N\mathbf{F}^*\boldsymbol{\alpha} = \sum_i \boldsymbol{\alpha}_i. \tag{2.24}$$

Note that there are no phase coefficients z_m in the above equations, as we do not consider any offsets (we consider a single set of samples, namely the base set $\mathbf{y}_0$).

As already discussed in the previous section, we will consider not only bandlimited functions, but *more generally* functions that belong to a finite-dimensional Hilbert space $\mathcal{H}$. In this case, the interpretation of frequency replication is not valid anymore, and the idea of aliasing needs to be broadened to this new signal space. In order to do this, we show that the sampling process is a projection. A projection operator is defined as follows:

Definition 2.2.1 (Projection operator [64,125]). An operator P is a projection operator if it is a linear transformation that is idempotent, that is, $P^2 = P$. The projection is orthogonal if, additionally, it is self-adjoint.

Now that we have defined projections, we can show that sampling can be seen as a projection operation [108]. For this, we actually need to consider the combination of sampling with interpolation, such that the resulting signal is in the same space as the original signal. We will follow a different approach here than the one given by Unser [108], as we consider sampling of a function in an arbitrary Hilbert space with Dirac functions. In other words, we use a Dirac sampling kernel, while Unser uses other sampling kernels, depending on the signal space.

Theorem 2.2.1 (Sampling as a projection). Assume a signal $f(t)$ in a periodic finite-dimensional Hilbert space $\mathcal{H}$ with basis $\{\varphi_l(t)\}_{l=0..L-1}$ and period 1:

$$f(t) = \sum_{l=0}^{L-1} \alpha_l \varphi_l(t), \tag{2.25}$$

with α_l the expansion coefficient corresponding to $\varphi_l(t)$. We sample $f(t)$ uniformly at a rate N, resulting in the sample vector

$$\mathbf{y} = \mathbf{\Phi}_0 \boldsymbol{\alpha}, \tag{2.26}$$

where $\mathbf{\Phi}_0$ is an $N \times L$ matrix with the sampled basis functions $\varphi_l(n/N)$ as its columns ($\mathbf{\Phi}_0(i,j) = \varphi_j(i/N)$). The signal coefficients are interpolated again using

$$\boldsymbol{\alpha}' = \begin{cases} (\mathbf{\Phi}_0^* \mathbf{\Phi}_0)^{-1} \mathbf{\Phi}_0^* \mathbf{y} & \text{if } N \geq L \\ \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{y} & \text{if } N < L. \end{cases} \tag{2.27}$$

The overall operator $\mathbf{P}_{\mathbf{\Phi}_0}(\boldsymbol{\alpha})$ described by

$$\mathbf{P}_{\mathbf{\Phi}_0} = \begin{cases} (\mathbf{\Phi}_0^* \mathbf{\Phi}_0)^{-1} \mathbf{\Phi}_0^* \mathbf{\Phi}_0 = \mathbf{I} & \text{if } N \geq L \\ \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0 & \text{if } N < L, \end{cases} \tag{2.28}$$

is an orthogonal projection. It describes the sampling and interpolation of $f(t)$.

Proof: The proof is quite straightforward. For $N \geq L$, the operator is identity, so it is clearly a projection of the signal onto itself. Let us therefore concentrate on the case where $N < L$. The linearity of $\mathbf{P}_{\mathbf{\Phi}_0}$ is obvious, as it is described by a matrix, which in itself is a linear operator. We need to prove that $\mathbf{P}_{\mathbf{\Phi}_0}$ is idempotent, or $\mathbf{P}_{\mathbf{\Phi}_0}(\mathbf{P}_{\mathbf{\Phi}_0}(\boldsymbol{\alpha})) = \mathbf{P}_{\mathbf{\Phi}_0}(\boldsymbol{\alpha})$. This can be shown as follows:

$$\mathbf{P}_{\mathbf{\Phi}_0}^2 = \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0 \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0 = \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0 = \mathbf{P}_{\mathbf{\Phi}_0}. \tag{2.29}$$

Therefore, the sampling operator $\mathbf{P}_{\mathbf{\Phi}_0}$ is a projection. To show that it is an orthogonal projection, we also need to prove that it is self-adjoint ($\mathbf{P}_{\mathbf{\Phi}_0}^* = \mathbf{P}_{\mathbf{\Phi}_0}$), or

$$\mathbf{P}_{\mathbf{\Phi}_0}^* = \left(\mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0\right)^* = \mathbf{\Phi}_0^* (\mathbf{\Phi}_0 \mathbf{\Phi}_0^*)^{-1} \mathbf{\Phi}_0 = \mathbf{P}_{\mathbf{\Phi}_0}. \tag{2.30}$$

This proves that our sampling followed by interpolation operator $\mathbf{P}_{\mathbf{\Phi}_0}$ is an orthogonal projection (for any value of N). ■

If $N \geq L$, the expansion coefficients $\boldsymbol{\alpha}$ can be exactly reconstructed from the samples $\mathbf{y}$, and the projection operator $\mathbf{P}_{\mathbf{\Phi}_0}$ is identity. If the measurements $\mathbf{y}$ are noisy, (2.27) computes the least squares estimate of the parameters $\boldsymbol{\alpha}$. It performs an orthogonal projection onto the space spanned by the column vectors of $\mathbf{\Phi}_0$, which is the signal space $\mathcal{H}$.

If $N < L$, the sampled signal is aliased. We can compute a minimum norm estimate for $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \mathbf{\Phi}_0^*(\mathbf{\Phi}_0\mathbf{\Phi}_0^*)^{-1}\mathbf{y}, \tag{2.31}$$

which performs a projection onto the rows of $\mathbf{\Phi}_0$. However, perfect reconstruction from a single set of samples is no longer possible. Different input signals are projected onto the same set of samples. This interpretation of sampling and aliasing as a projection is valid for signals in arbitrary finite-dimensional Hilbert spaces, including the bandlimited functions from the more restricted analysis.

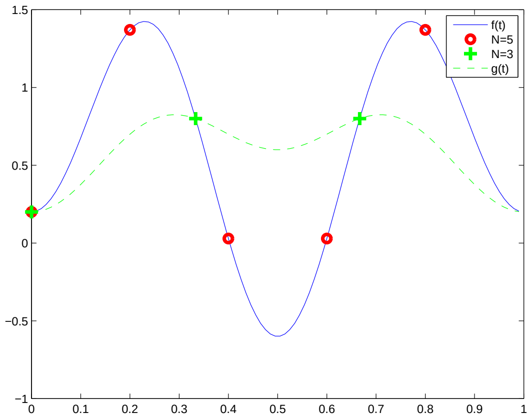


Figure 2.5: Example of a signal $f(t)$ with $L = 5$ coefficients (—) and its projections using $N = 5$ and $N = 3$ samples. Using 5 samples, the signal can be perfectly reconstructed, while with 3 samples, it is aliased. Another signal $g(t)$ is reconstructed from the projection (– –).

Example 2.2.1 (Bandlimited signals). Let us illustrate this with an example for a signal in the Fourier basis. Take a bandlimited signal with $L = 5$ and Fourier coefficients $\boldsymbol{\alpha} = \begin{pmatrix} -2 & 1 & 3 & 1 & -2 \end{pmatrix}^T$. If we take $N = 5$, we obtain the sample vector $\mathbf{y} = \begin{pmatrix} 0.2 & 1.371 & 0.029 & 0.029 & 1.371 \end{pmatrix}^T$. From this sample vector, we can perfectly reconstruct the signal coefficients using (2.27):

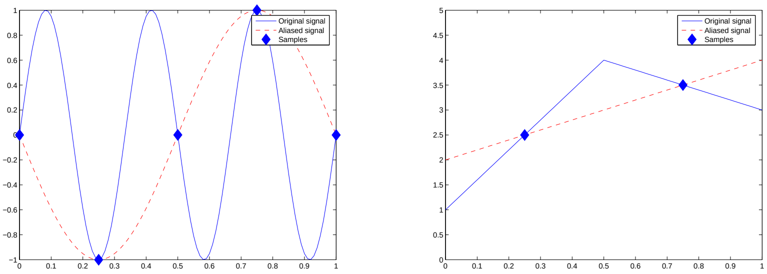
$$\boldsymbol{\alpha}' = \begin{pmatrix} -2 & 1 & 3 & 1 & -2 \end{pmatrix}^T = \boldsymbol{\alpha}. \tag{2.32}$$

However, if we take only $N = 3$ samples, we obtain the sample vector $\mathbf{y} = \begin{pmatrix} 0.2 & 0.8 & 0.8 \end{pmatrix}^T$. The signal is aliased, and the Fourier coefficients cannot be perfectly reconstructed anymore:

$$\boldsymbol{\alpha}' = \begin{pmatrix} -0.5 & -0.5 & 3 & -0.5 & -0.5 \end{pmatrix}^T \neq \boldsymbol{\alpha}. \tag{2.33}$$

The signal and its different sets of samples are shown in Figure 2.5. $\square$

Some more examples of aliasing for different types of signals are shown in Figure 2.6.



(a) (b)

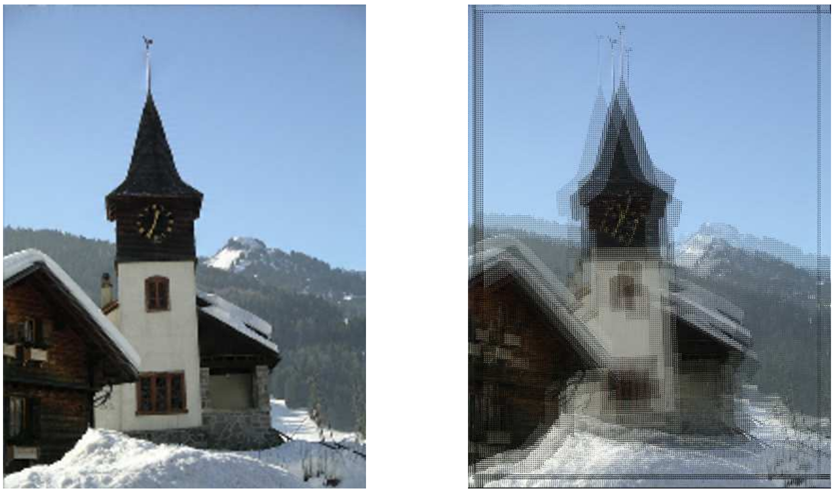
Figure 2.6: Examples of aliasing for different types of signals. (a) For a bandlimited signal, input signals of different frequencies map into the same low frequency after sampling. (b) For a piecewise linear signal, different input signals map into the same set of samples (and reconstruction) after sampling.

2.3 Super-resolution imaging

To our knowledge, the term super-resolution was introduced by Papoulis in 1968 [76, p. 453-454]. He used the term for the computation of additional high frequency information from a single image using the bandlimitedness and the finite extent of the image. In 1984, Tsai and Huang presented a first super-resolution algorithm that uses multiple input images [107]. They describe a frequency domain method to increase the resolution of satellite images from a set of input images. The registration parameters between the different acquisitions are computed by searching the parameter values for which the reconstructed signal is bandlimited to a certain frequency. It is shown that for (small) deviations from these optimal values, the signal always has a larger bandwidth. The Fourier coefficients of the high resolution image can then be computed efficiently using an FFT-based algorithm.

Good overviews of existing super-resolution methods are given by Borman and Stevenson [9], and Park et al. [79]. Recently, special issues on super-resolution imaging appeared in IEEE Signal Processing Magazine (edited by Kang and Chaudhuri [50]), and EURASIP Journal on Applied Signal Processing (edited by Ng et al. [73]). Two books by Chaudhuri discussing motionless super-resolution and various other topics in super-resolution imaging were also published in the past few years [19,20]. Finally, Matlab implementations with a graphical user interface for certain super-resolution algorithms were released by Farsiu et al. [30] and Vandewalle et al. [120].

Most super-resolution algorithms can be decomposed into *two parts*: an *image registration* part followed by a *reconstruction* part. Very high accuracy is required in the registration (up to subpixel level) to be able to reconstruct the high resolution image correctly. If the registration parameters are incorrectly estimated, it is typically better to interpolate a single image to the desired size, than to combine (incorrectly) the information from the different images (see Figure 2.7). Once the images are registered, a robust reconstruction method is needed to build a high resolution image from the set of irregularly spaced samples (pixels) and undo the blur caused by the optical system. We will mainly study the first problem, as it is an essential requirement for a high precision reconstruction afterwards.



(a) (b)

Figure 2.7: If images are badly registered, it is better to interpolate a single image (a), than to combine (incorrectly) the information from the different images (b).

2.3.1 Image registration

Surveys of image registration methods are given by Brown [10], and by Zitová and Flusser [133]. We summarize some of the most frequently used *spatial domain* techniques here, before we proceed to a more detailed overview of the frequency domain methods. One popular technique is to compute salient features and find a mapping between corresponding features in different images [16, 17, 33]. Another approach is the use of a Taylor series approximation of the images. The motion parameters are the unknowns in the approximation, and they can be computed from the set of equations that can be derived from this approximation. Because Taylor series only give a good approximation for small offsets, these registration methods are generally applied iteratively using a Gaussian pyramid [46, 51]. Irani et al. present a multiresolution approach to compute multiple, possibly occluding motions using an iterative method [47]. Other methods compute an optical flow field [35] and obtain therefore a motion vector for every pixel of the image.

Methods based on the *correlation* between shifted images can be performed either in spatial or in frequency domain. Such methods are based on the fact that the correlation between two images is maximum if they are correctly aligned. In frequency domain, a correlation can be computed efficiently using a multiplication of the two spectra. An early method performing the frequency domain operations is presented by Anuta [4]. A spatial correlation-based method is presented by Zokai and Wolberg [134]. They apply a log-polar transform to the spatial domain images to allow for large rotations and scale changes.

The *frequency domain* methods are limited to global motion in a plane parallel to the image plane. A planar shift can be described in frequency domain as a linear phase shift, while a rotation (in the image plane) in spatial domain corresponds to a rotation over the same angle in frequency domain [76, p. 90-91]. Other, more general motion models are difficult to describe in frequency domain. However, frequency domain registration methods can be computationally very efficient, and offer a good framework to model aliasing.

If the input images are not undersampled, and their motion can be approximated by a *planar shift*, this shift can be computed as the linear phase difference between the two images (Kim and Su [54], Stone et al. [100], Vandewalle et al. [118]). The horizontal and vertical components of the shift can be separated using a singular value decomposition (Hoge [42]). This frequency domain approach has the advantage that continuous values can be estimated for the shifts. However, it suffers from another problem: phase wrapping at $-\pi$ and π. If we compute the inverse Fourier transform of the ratio between the two frequency domain images, we obtain a Dirac at the correct shift position in spatial domain. The peak in the inverse Fourier transform occurs at a discrete position, and its precision is therefore limited by the resolution of the input. Foroosh et al. [34] showed that instead of a single Dirac, the signal power in the phase correlation corresponds to a polyphase transform of a filtered unit impulse. They applied this idea to obtain the subpixel part of the shifts.

Planar rotation can be added to such an algorithm, and is represented in frequency domain by a rotation over the same angle. The shift and rotation parameters can be estimated separately, because shift only affects the phase information, while rotation affects both phase and amplitude of the Fourier transform. Reddy and Chatterji [87] and Marcel et al. [66] applied this in their image registration algorithm. To estimate the rotation, they transform the Fourier domain image into polar coordinates, such that the rotation is transformed into a shift. Lucchese and Cortelazzo [63] present another rotation estimation method that is based on the property that the magnitude of the Fourier transform of an image and the mirrored version of the magnitude of the Fourier transform of a rotated image have a pair of orthogonal zero-crossing lines. The angle that these lines make with the axes is equal to half the rotation angle between the two images. The shift is estimated in the same way as Marcel et al.

If the input images are aliased, these simple relations do not hold anymore. However, if the sampling frequency is sufficiently high, part of the spectrum is still free of aliasing. The registration can then be performed using specifically this part of the spectrum [54, 100, 119]. Such a method will be presented in Chapter 3. A frequency domain method will be described to estimate planar motion (shift and rotation) using the aliasing-free part of the spectrum. Our rotation estimation method uses radial projections to reduce computational complexity. Another projection-based registration method using a Radon transform is presented by Robinson and Milanfar [88]. On one hand, projection-based methods reduce the available information by projecting the images along a certain direction, but on the other hand, this also reduces computational complexity and increases robustness to noise [2, 15]. If the entire frequency spectrum is aliased, all the images have to be registered jointly. Several approaches to solve this problem will be presented in Chapter 5 and 6: two subspace-based methods in Chapter 5, and a method based on Gröbner bases in Chapter 6.

2.3.2 Image reconstruction

The second part of the super-resolution methods is the reconstruction, in which a high resolution image is reconstructed from the registered low resolution images. As described above, a frequency domain algorithm was presented by Tsai and Huang [107]. We will use a similar approach for the reconstruction in Chapter 5. Kim et al. [53] extended this algorithm to include blur and noise models. More recently several other, mostly spatial domain techniques have been developed.

A good overview of reconstruction methods is given by Park et al. [79]. First of all, there are a set of *nonuniform interpolation* methods [24, 74, 77, 84, 102] that reconstruct a bandlimited signal from an irregular set of samples. In Chapter 3, we will use such a method. A triangulation is first computed from the irregular set of samples using the Qhull algorithm [6, 70]. Next, the image values are interpolated on a uniform grid using cubic interpolation and the computed triangulation. Another category of reconstruction algorithms is based on the *POCS* method, like for example the method by Patti et al. [80]. These are iterative algorithms that project the reconstruction successively onto different convex sets representing both the known measurements and assumptions about the high resolution image. After a number of iterations, the reconstructed image converges to the image at the intersection of the different sets. Similarly, the *iterative backprojection* algorithms [46, 51, 104] use an image formation model to estimate the low resolution images that would be created when capturing the current high resolution reconstruction estimate. The estimate is then corrected with a term relative to the difference between these estimated low resolution images and the real ones. Zomet et al. [135] improved the results obtained with typical iterative backprojection algorithms by taking the median of the errors in the different backprojected images. This proves to be more robust in the presence of outliers. Farsiu et al. [31] proposed a new and robust super-resolution algorithm that replaces the standard L_2 minimization by an L_1 minimization. This results in sharper images. They also added a regularization term to stabilize the reconstruction. They showed that this approach performs very well in combination with the algorithm by Zomet et al. [135]. A spatio-temporal FIR filtering approach is presented by Goldberg et al. [39]. They also perform a 2D + time analysis of the image sequence in frequency domain to check whether (next to the individual images) the global sequence is aliased. An adaptive filtering approach using LMS and RLS algorithms for super-resolution video reconstruction is presented by Elad and Feuer [27]. Finally, a *maximum a posteriori* (MAP) approach can also be used for reconstruction [16, 17, 28, 94]. It allows to easily add different priors on the reconstructed image into the model. Elad and Feuer [26] present a hybrid approach combining a maximum likelihood (ML) estimator with a POCS method.

This gives only a very concise overview of the wide variety of existing super-resolution reconstruction algorithms. For a more detailed overview, the reader is referred to the review articles mentioned above. In the next section, we will describe some different approaches to super-resolution, and related problems.

2.3.3 Variations on super-resolution

Next to the above methods that are based on relative motion between the images, other methods use images with different *zoom* to create a super-resolution image [20,48,86]. Shekarforoush et al. [97] present a method to reconstruct not only a super-resolution image, but also the surface albedo and height of the scene objects using a Bayesian approach. Super-resolution methods can also be applied to *demosaicing*: instead of a single image, multiple images sampled with a Bayer CFA pattern are used to compute a single (high resolution) image with red, green and blue values at every pixel location [29]. Recently, some *bounds* to the possible increase in resolution have been computed. Lin and Su [61] showed that under practical conditions, reconstruction-based algorithms can improve the resolution by at most a factor 1.6. They use perturbation theory to compute these limits. Baker and Kanade [5] perform an analysis of the point spread function (PSF) to show limits of some super-resolution methods. They also present a method to 'break' those limits using prior knowledge about the image class. Cramer-Rao bounds for super-resolution and image registration problems are computed by Robinson and Milanfar [89, 90].

2.4 Uniqueness of the solution

In this section, we discuss the existence and uniqueness of a solution to the super-resolution problem with unregistered sets of samples that was described in Section 2.1. It follows straightforwardly from this description that a solution exists (at least in the ideal, noiseless case). The uniqueness of such a solution is less trivial.

2.4.1 General case

Let us start with an intuitive statement for the general problem. A unique solution exists if there is a unique mapping from each set of samples to a single space generated by the columns of $\mathbf{\Phi_t}$ (see (2.7)). In other words, the intersection of the spaces generated for two different offset vectors $\mathbf{t}$ only contains the trivial zero vector, except for possibly some degenerate cases. This argument follows the one made by Marziliano for discrete sampling with unknown locations [68]. More formally, we have the following lemma.

Lemma 2.4.1 (Uniqueness of the general problem). If the L-dimensional subspaces (in the MN-dimensional measurement space) generated by the columns of $\mathbf{\Phi}_{\mathbf{t}}$ for any pair of different offset vectors $\mathbf{t}$ and $\mathbf{t}'$ have a low-dimensional intersection, then a unique signal $f(t)$ belonging to the finite-dimensional Hilbert space $\mathcal{H}$ can be reconstructed from the sample vector $\mathbf{y}$, except if the sample vector $\mathbf{y}$ is an element of the intersection.

Proof: We prove this by contradiction. Assume that there are two different offset vectors $\mathbf{t}$ and $\mathbf{t}'$, and that the sample vector $\mathbf{y}$ is not in the intersection between the spaces generated by the columns of $\mathbf{\Phi}_{\mathbf{t}}$ and $\mathbf{\Phi}_{\mathbf{t}'}$. We can then write the sample vector $\mathbf{y}$ in two ways, namely:

$$\mathbf{y} = \mathbf{\Phi}_{\mathbf{t}}\boldsymbol{\alpha} = \mathbf{\Phi}_{\mathbf{t}'}\boldsymbol{\alpha}'. \tag{2.34}$$

Because the vector $\mathbf{y}$ is not in the intersection of the spaces spanned by $\mathbf{\Phi}_{\mathbf{t}}$ and $\mathbf{\Phi}_{\mathbf{t}'}$, this cannot be true. Our assumption was wrong, and $\mathbf{t} = \mathbf{t}'$. ■

In general, the space of sample vectors that belong to the intersection is low-dimensional and thus, the set of sample vectors for which a unique solution exists, is dense in the Hilbert space $\mathcal{H}$. We shall therefore ignore the degenerate cases. Nevertheless, we will specifically calculate examples of such degenerate cases for polynomial signals in Section 2.4.3.

Unfortunately, this theorem does not give any specific conditions for the uniqueness of the solution. We will therefore analyze a few specific signal types to gain understanding.

2.4.2 Bandlimited signals

First of all, we consider the space of bandlimited functions with the Fourier basis. As we discussed in Section 2.1.2, the matrix $\mathbf{\Phi}_{t_m}$ can then be written as the product of an $N \times L$ IDFT matrix $\mathbf{F}^*$ with an $L \times L$ diagonal matrix $\mathbf{D}_{t_m}$. For aliased signals, some of the columns in $\mathbf{F}^*$ are repeated, and the corresponding columns of $\mathbf{\Phi}_{t_m}$ only differ by their multiplicative factor z_m^l. These columns correspond to overlapping frequencies in the sampled spectrum.

If all the sets of samples are considered together, the l-th column $\boldsymbol{\phi}_{\mathbf{t}}^l$ of the matrix $\mathbf{\Phi}_{\mathbf{t}}$ is the Hermitian transpose of a repetition of M times the same basis vector $\mathbf{f}_l$, the l-th row of the $L \times N$ forward DFT matrix $\mathbf{F}$, multiplied by the different factors z_m^{-l}:

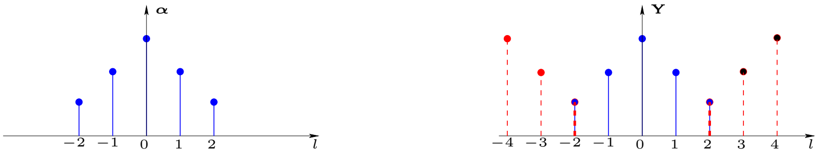
$$\left(\boldsymbol{\phi}_{\mathbf{t}}^l\right)^* = \begin{pmatrix} \mathbf{f}_l & z_1^{-l}\mathbf{f}_l & \cdots & z_{M-1}^{-l}\mathbf{f}_l \end{pmatrix}. \tag{2.35}$$

Example 2.4.1. For example, if we take $M = 2$, $L = 5$, and $N = 4$, there are two overlapping frequency components (see Figure 2.8). We have

$$\mathbf{\Phi}_{\mathbf{t}} = \begin{pmatrix} \mathbf{\Phi}_0 \\ \mathbf{\Phi}_{t_1} \end{pmatrix} = \begin{pmatrix} \mathbf{F}^* \\ \mathbf{F}^*\mathbf{D}_{t_1} \end{pmatrix} = \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ W^2 & W^3 & 1 & W & W^2 \\ 1 & W^2 & 1 & W^2 & 1 \\ W^2 & W & 1 & W^3 & W^2 \\ \hline z_1^{-2} & z_1^{-1} & 1 & z_1 & z_1^2 \\ z_1^{-2}W^2 & z_1^{-1}W^3 & 1 & z_1 W & z_1^2 W^2 \\ z_1^{-2} & z_1^{-1}W^2 & 1 & z_1 W^2 & z_1^2 \\ z_1^{-2}W^2 & z_1^{-1}W & 1 & z_1 W^3 & z_1^2 W^2 \end{array}\right), \tag{2.36}$$

with $W = e^{j2\pi/4}$. The first and the fifth column of $\mathbf{F}^*$ are equal, such that the corresponding columns of $\mathbf{\Phi_t}$ only differ by their factors z_1^{-2} and z_1^2 from $\mathbf{D}_{t_1}$.

□



(a) (b)

Figure 2.8: Signal spectrum before (a) and after sampling (b), for $L = 5$ and $N = 4$. The base spectrum (—) and aliased spectrum (– –) overlap for the first and last spectral component, corresponding to the first and fifth column in (2.36).

Because the Fourier basis row vectors $\mathbf{f}_l$ of the $L \times N$ matrix $\mathbf{F}$ are orthogonal (note that we use $W = e^{j2\pi/N}$ in this matrix, and not $e^{j2\pi/L}$), the vectors $\boldsymbol{\phi}_\mathbf{t}^l$ are also orthogonal to each other for any set of offset values $\mathbf{t}$. Only the vectors corresponding to overlapping spectrum coefficients (like the first and fifth column in Example 2.4.1) are not orthogonal, because they are composed by the same Fourier vector $\mathbf{f}_l$ but have different coefficients. Each vector describes a trajectory in $\mathbb{C}^{MN}$ for varying values of $\mathbf{t}$. This trajectory is defined more precisely in the following theorem:

Theorem 2.4.1. For varying $\mathbf{t}$, any vector $\boldsymbol{\phi}_\mathbf{t}^l$ describes a trajectory in the M-dimensional subspace $\mathcal{V}_l$ of $\mathbb{C}^{MN}$:

$$\mathcal{V}_l = \operatorname{span}\{\mathbf{A} \otimes \mathbf{f}_l^*\} \quad \text{with } \mathbf{A} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & -1 & & 1 \\ \vdots & & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{pmatrix}, \tag{2.37}$$

where $\mathbf{A}$ is an $M \times M$ matrix, and $\otimes$ represents the Kronecker product. The vectors $\boldsymbol{\phi}_\mathbf{t}^{l+iN}$ corresponding to overlapping spectrum coefficients belong to the same space $\mathcal{V}_l$, while other vectors $\boldsymbol{\phi}_\mathbf{t}^k$ belong to orthogonal subspaces $\mathcal{V}_k \perp \mathcal{V}_l$.

Proof: The trajectory of $\boldsymbol{\phi}_\mathbf{t}^l$ as a function of $\mathbf{t}$ is in $\mathcal{V}_l$ iff we can write any arbitrary $\boldsymbol{\phi}_\mathbf{t}^l$ as a linear combination of the columns of $\mathbf{A} \otimes \mathbf{f}_l^*$. From (2.35) and (2.37), we can write the vector $\boldsymbol{\phi}_\mathbf{t}^l$ as a linear combination of the columns of $\mathbf{A} \otimes \mathbf{f}_l^*$ by solving the set of linear equations

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & -1 & & 1 \\ 1 & & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{M-1} \end{pmatrix} = \begin{pmatrix} 1 \\ z_1^l \\ \vdots \\ z_{M-1}^l \end{pmatrix}. \tag{2.38}$$

As this set of M equations is of full rank, it always has a unique solution, and our vector is therefore part of $\mathcal{V}_l$. Vectors corresponding to overlapping Fourier coefficients are composed from the same Fourier basis vectors $\mathbf{f}_l$, and therefore belong to the same space $\mathcal{V}_l$.

The orthogonality between two subspaces $\mathcal{V}_k$ and $\mathcal{V}_l$ for $k \neq l$ can easily be seen. As each of the subspaces is generated by the columns of the matrices defined in (2.37), it is sufficient that we prove that any arbitrary column of $\mathbf{A} \otimes \mathbf{f}_k^*$ is orthogonal to any column of $\mathbf{A} \otimes \mathbf{f}_l^*$. Denoting the i-th column of $\mathbf{A}$ as $\mathbf{A}^i$, we can write the inner product between two such vectors as

$$\left\langle \mathbf{A}^i \otimes \mathbf{f}_k^*, \mathbf{A}^j \otimes \mathbf{f}_l^* \right\rangle = \sum_{n=0}^{M-1} (\pm 1) \left\langle \mathbf{f}_k^*, \mathbf{f}_l^* \right\rangle = 0 \text{ if } k \neq l. \tag{2.39}$$

This is valid because $\langle \mathbf{f}_k^*, \mathbf{f}_l^* \rangle = 0$ from the orthogonality of the Fourier basis.

■

From this analysis, we can see that the problem can be considered in each of the N subspaces separately by projecting $\mathbf{y}$ onto the different M-dimensional subspaces. The projection of $\mathbf{y}$ onto the subspace $\mathcal{V}_l$ is called

$$\mathbf{y}^{(l)} = \mathbf{P}_{\mathcal{V}_l} \mathbf{y}. \tag{2.40}$$

An illustration for two overlapping vectors in a three-dimensional space is sketched in Figure 2.9.

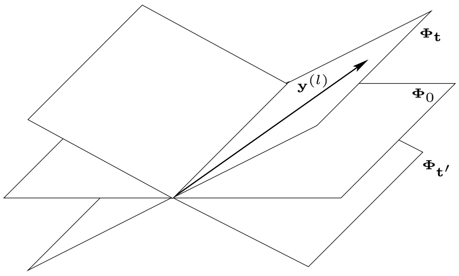


Figure 2.9: Illustration of the trajectory of the span of two columns $\boldsymbol{\phi}_{\mathbf{t}}^l$ (the planes in this drawing) in a three-dimensional space $\mathcal{V}_l$ for different offset values (0, $\mathbf{t}$ and $\mathbf{t}'$). Only for the correct offsets $\mathbf{t}$, the vector $\mathbf{y}^{(l)}$ belongs to the space spanned by the two columns.

Assuming that $MN \geq L + M - 1$, at most M columns of $\mathbf{\Phi_t}$ belong to the same subspace $\mathcal{V}_l$. If there are M vectors $\boldsymbol{\phi}_{\mathbf{t}}^{l+iN}$ from $\mathbf{\Phi_t}$ in a particular subspace $\mathcal{V}_l$, these vectors form a basis for $\mathcal{V}_l$. As the vector $\mathbf{y}^{(l)}$ is also in $\mathcal{V}_l$, it belongs to the subspace spanned by these vectors $\boldsymbol{\phi}_{\mathbf{t}}^{l+iN}$ for any value of $\mathbf{t}$. However, for the subspaces containing less than M vectors $\boldsymbol{\phi}_{\mathbf{t}}^{l+iN}$, the span of these vectors does not cover the entire space $\mathcal{V}_l$. Hence, the projection of $\mathbf{y}$ onto $\mathcal{V}_l$ generally only belongs to $\text{span}(\boldsymbol{\phi}_{\mathbf{t}}^{l+iN})_{0 \leq i < M-1}$ for a single offset vector $\mathbf{t}$. The offset vector $\mathbf{t}$ can then be uniquely determined from these subspaces, and the signal coefficients $\boldsymbol{\alpha}$ are thus also uniquely determined.

Using the Fourier basis, there are certain cases for which it is not sufficient to require $MN \geq L + M - 1$. The vectors in the space $\mathcal{V}_0$ (the columns $l = iN$) do not depend on $\mathbf{t}$, and therefore, no information about the offsets can be derived from these vectors. So if only the space $\mathcal{V}_0$ contains less than M vectors, it is not possible to derive the offsets from the sets of samples. This happens when the number of overlapping parts of the frequency spectrum is even (and therefore also the number of sets of samples M is even). In that case, we need $MN \geq L + M$.

Example 2.4.2. Looking back at Example 2.4.1, we can see that the first and fifth column of $\mathbf{\Phi_t}$ span the subspace $\mathcal{V}_2$, except in the degenerate case when $t_1 \in \mathbb{Z}$. The projection of $\mathbf{y}$ onto these two columns will therefore always be the same as its projection onto $\mathcal{V}_2$. No information about t_1 can be derived from this subspace. However, for $\mathcal{V}_{-1}$ and $\mathcal{V}_1$, there is only one column of $\mathbf{\Phi_t}$ in each of these subspaces, and the projections onto that column or onto the space $\mathcal{V}_l$ only coincide for the correct value of t_1. The part of $\mathbf{y}$ in the subspace $\mathcal{V}_0$ does not depend on t_1 ($\mathcal{V}_0$ is a one-dimensional subspace), and is therefore of no use in the computation of the offset. We need at least one subspace containing only 1 column from $\mathbf{\Phi_t}$ in order to be able to compute t_1. As we do not want this to be the zero frequency, we require that $2N \geq L + 2$, or $N = 4$. □

This leads to the following theorem. Its proof follows from the above argument.

Theorem 2.4.2 (Bandlimited signals). If $MN \geq L + M - 1$ for M odd, or $MN \geq L + M$ for M even, a bandlimited signal with L Fourier coefficients can be uniquely reconstructed from M uniform sets of N samples with unknown offsets.

2.4.3 Polynomials

Let us now also analyze the uniqueness for polynomials, using the same setup as in Example 2.1.2. For second order polynomials, we can prove the following theorem.

Theorem 2.4.3 (Second order polynomials). When an arbitrary second order polynomial is sampled at four non-coinciding points $\{0, 0.5, 0.5t_1, 0.5t_1 + 0.5\}$, then this polynomial $ax^2 + bx + c$ and the shift t_1 can be uniquely retrieved, except in the particular case $at_1 + a + 2b = 0$, when an infinite number of solutions exist.

Proof: Let us call the two sets of samples $\mathbf{y}_0$ and $\mathbf{y}_1$. The samples satisfy

$$\left(\begin{array}{c} \mathbf{y}_0 \\ \hline \mathbf{y}_1 \end{array}\right) = \left(\begin{array}{c} y_0(0) \\ y_0(1) \\ \hline y_1(0) \\ y_1(1) \end{array}\right) = \left(\begin{array}{ccc} 0 & 0 & 1 \\ 0.25 & 0.5 & 1 \\ \hline 0.25t_1^2 & 0.5t_1 & 1 \\ 0.25(1+t_1)^2 & 0.5(1+t_1) & 1 \end{array}\right) \left(\begin{array}{c} a \\ b \\ c \end{array}\right). \tag{2.41}$$

Let us assume that, next to the desired solution $\{a, b, c, t_1\}$, there exists another solution $\{a', b', c', t_1'\}$ resulting in the same samples. We can therefore also write (2.41) for this solution, and obtain the following set of equations:

$$\begin{cases} c = c' \\ 0.25a + 0.5b + c = 0.25a' + 0.5b' + c' \\ 0.25t_1^2 a + 0.5t_1 b + c = 0.25t_1'^2 a' + 0.5t_1' b' + c' \\ 0.25(1+t_1)^2 a + 0.5(1+t_1)b + c = 0.25(1+t_1')^2 a' + 0.5(1+t_1')b' + c'. \end{cases} \tag{2.42}$$

Eliminating c and c' from the equations, and rearranging the terms, we get

$$\begin{cases} c' = c \\ b' = 0.5a + b - 0.5a' \\ 0.5t_1^2 a - 0.5t_1'^2 a' + t_1 b - t_1' b' = 0 \\ 0.5t_1^2 a - 0.5t_1'^2 a' + t_1 a - t_1' a' + 0.5a - 0.5a' + bt_1 - b't_1' + b - b' = 0. \end{cases} \tag{2.43}$$

From the second equation, we can compute b' as a function of a' and the terms of the first solution $\{a, b, c, t_1\}$. Similarly, after simplification, a' can be computed from the fourth equation:

$$a' = \frac{at_1}{t_1'}. \tag{2.44}$$

We assume that $t_1' \neq 0$, which is a reasonable assumption, because this would mean that the two sets of samples $\mathbf{y}_0$ and $\mathbf{y}_1$ coincide, and there are only two distinct samples. After filling in a' and b', the third equation can be factorized as follows:

$$0.5(t_1 - t_1')(at_1 + a + 2b) = 0. \tag{2.45}$$

From this equation, we can see that either $at_1 + a + 2b = 0$, or $t_1 = t_1'$. Because we assumed that the two solutions $\{a, b, c, t_1\}$ and $\{a', b', c', t_1'\}$ are different, the second solution can be discarded (it involves immediately $a' = a$, $b' = b$, and $c' = c$).

It is therefore only when $at_1 + a + 2b = 0$ that other solutions exist, which are given by

$$\begin{cases} a' = \frac{at_1}{t_1'} \\ b' = 0.5a + b - 0.5\frac{at_1}{t_1'} \\ c' = c, \end{cases} \tag{2.46}$$

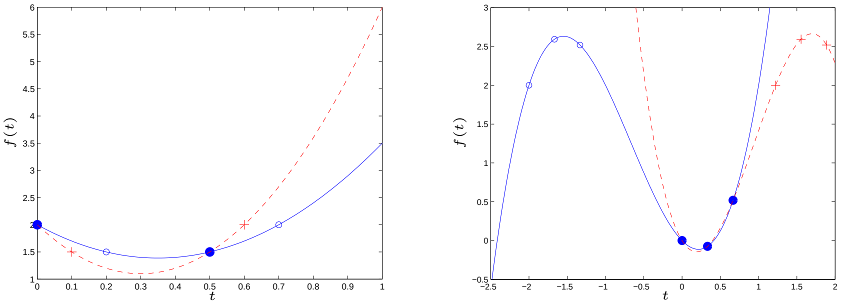
with t_1' a free parameter. In all the other cases, the solution $\{a, b, c, t_1\}$ is unique. ■

An example of multiple solutions for second order polynomials with $at_1 + a + 2b = 0$ is given in Figure 2.10(a). A similar theorem can be shown for polynomials of third order.

Theorem 2.4.4 (Third order polynomials). When an arbitrary third order polynomial is sampled at six non-coinciding points $\{0, 1/3, 2/3, t_1/3, (t_1 + 1)/3, (t_1 + 2)/3\}$, then this polynomial $ax^3 + bx^2 + cx + d$ and the shift t_1 can be uniquely retrieved, except in the particular case $t_1 a + 2a + 2b = 0$, when an extra parasitic solution exists.

Proof: Let us call the two sets of samples $\mathbf{y}_0$ and $\mathbf{y}_1$. The samples satisfy

$$\left(\frac{\mathbf{y}_0}{\mathbf{y}_1}\right) = \left(\begin{array}{c} y_0(0) \\ y_0(1) \\ y_0(2) \\ \hline y_1(0) \\ y_1(1) \\ y_1(2) \end{array}\right) = \left(\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 1/27 & 1/9 & 1/3 & 1 \\ 8/27 & 4/9 & 2/3 & 1 \\ \hline t_1^3/27 & t_1^2/9 & t_1/3 & 1 \\ (1+t_1)^3/27 & (1+t_1)^2/9 & (1+t_1)/3 & 1 \\ (2+t_1)^3/27 & (2+t_1)^2/9 & (2+t_1)/3 & 1 \end{array}\right) \left(\begin{array}{c} a \\ b \\ c \\ d \end{array}\right). \tag{2.47}$$



(a) (b)

Figure 2.10: Example of parasitic solutions for second and third order polynomials. (a) The second order polynomial $f(x) = 5x^2 - 3.5x + 2$ (—) is sampled at times $\{0, 0.5\}$ and $\{0.2, 0.7\}$ ($t_1 = 0.4$, plotted with 'o'). From the two sets of samples, other second degree polynomials can also be reconstructed using other offsets t_1. For example, setting $t_1 = 0.2$, and thus shifting the second set of samples to the left (plotted with '+'), we would obtain the polynomial $g(x) = 10x^2 - 6x + 2$ (– –). (b) The third order polynomial $f(x) = x^3 + 2x^2 - x$ (—) is sampled at times $\{0, 1/3, 2/3\}$ and $\{-2, -5/3, -4/3\}$ ($t_1 = -6$, plotted with 'o'). From the two sets of samples, another third degree polynomial $g(x) = -1.64x^3 + 4.64x^2 - 1.59x$ (– –) can be reconstructed using $t_1 = 3.66$ as offset. The shifted samples are shown with '+'.

Let us assume that, next to the desired solution $\{a, b, c, d, t_1\}$, there exists another solution $\{a', b', c', d', t'_1\}$ resulting in the same samples. We can therefore also write (2.47) for this solution, and obtain the following set of equations:

$$\begin{cases} d = d' \\ a/27 + b/9 + c/3 + d = a'/27 + b'/9 + c'/3 + d' \\ 8a/27 + 4b/9 + 2c/3 + d = 8a'/27 + 4b'/9 + 2c'/3 + d' \\ t_1^3 a/27 + t_1^2 b/9 + t_1 c/3 + d = t_1'^3 a'/27 + t_1'^2 b'/9 + t'_1 c'/3 + d' \\ (1+t_1)^3 a/27 + (1+t_1)^2 b/9 + (1+t_1)c/3 + d = (1+t'_1)^3 a'/27 \\ \qquad +(1+t'_1)^2 b'/9 + (1+t'_1)c'/3 + d' \\ (2+t_1)^3 a/27 + (2+t_1)^2 b/9 + (2+t_1)c/3 + d = (2+t'_1)^3 a'/27 \\ \qquad +(2+t'_1)^2 b'/9 + (2+t'_1)c'/3 + d'. \end{cases} \tag{2.48}$$

Eliminating d and d' from the equations, and rearranging the terms, we get

$$\begin{cases} d - d' = 0 \\ a - a' + 3(b - b') + 9(c - c') = 0 \\ 4(a - a') + 6(b - b') + 9(c - c') = 0 \\ t_1^3 a - t_1'^3 a' + 3t_1^2 b - 3t_1'^2 b' + 9t_1 c - 9t'_1 c' = 0 \\ t_1 a - t'_1 a' + t_1^2 a - t_1'^2 a' + 2(t_1 b - t'_1 b') = 0 \\ 2(t_1 a - t'_1 a') + t_1^2 a - t_1'^2 a' + 2(t_1 b - t'_1 b') = 0. \end{cases} \tag{2.49}$$

From the second and third equation, we can compute b' and c' as a function of a' and the terms of the first solution $\{a, b, c, d, t_1\}$. Similarly, we can compute

t_1' using the fifth and sixth equation:

$$\begin{cases} d' = d \\ b' = a - a' + b \\ c' = c - 2(a - a')/9 \\ t_1^3 a - t_1'^3 a' + 3t_1^2 b - 3t_1'^2 b' + 9t_1 c - 9t_1' c' = 0 \\ t_1^2 a - t_1'^2 a' + 2t_1 b - 2t_1' b' = 0 \\ t_1' = t_1 a/a'. \end{cases} \tag{2.50}$$

The fourth and fifth equations can be factorized as follows:

$$\begin{cases} d' = d \\ b' = a - a' + b \\ c' = -2(a - a')/9 + c \\ (t_1^2 aa' + t_1^2 a^2 + 3t_1 ba' + 3t_1 ba + 3t_1 a^2 + 9a'c - 2aa')(a - a')t_1 = 0 \\ (t_1 a + 2b + 2a)(a - a')t_1 = 0 \\ t_1' = t_1 a/a'. \end{cases} \tag{2.51}$$

From these equations, we can see that either $a' = -(3t_1a^2 + 3t_1ba + t_1^2a^2)/(t_1^2a + 3t_1b + 9c - 2a)$, or $a' = a$. Because we assumed that the two solutions $\{a, b, c, d, t_1\}$ and $\{a', b', c', d', t_1'\}$ are different, the second solution can be discarded (it involves immediately $b' = b$, $c' = c$, $d' = d$, and $t_1' = t_1$).

It is therefore only when $t_1a + 2a + 2b = 0$ that a second (parasitic) solution exists, which is given by

$$\begin{cases} a' = -\frac{3t_1a^2+3t_1ba+t_1^2a^2}{t_1^2a+3t_1b+9c-2a} \\ b' = a + b + \frac{3t_1a^2+3t_1ba+t_1^2a^2}{t_1^2a+3t_1b+9c-2a} \\ c' = c - \frac{2a}{9} - \frac{6t_1a^2+6t_1ba+2t_1^2a^2}{9t_1^2a+27t_1b+81c-18a} \\ d' = d \\ t_1' = -\frac{t_1^2a+3t_1b+9c-2a}{3a+3b+t_1a}. \end{cases} \tag{2.52}$$

In all the other cases, the solution $\{a, b, c, d, t_1\}$ is unique. ■

An example of a polynomial with $t_1a + 2a + 2b = 0$ and its parasitic solution is given in Figure 2.10(b). With increasing order, the proofs by contradiction like the proofs given above become very complex. Even for higher-order polynomials, however, the solution can typically be assumed unique except for a lower-dimensional set of degenerate cases. Proofs for such other cases can also be obtained using the Gröbner basis computation that is discussed more in detail in Chapter 6.

2.5 Conclusions

In this chapter, we have presented super-resolution image reconstruction in a multichannel sampling framework. We have given a mathematical description, both for signals in a general Hilbert space and for bandlimited signals. Next, we have shown that sampling can be written as a projection operator, and we have presented an interpretation of aliasing in such a setup. We have also given an overview of existing super-resolution algorithms, with an emphasis on the frequency domain methods for image registration. Finally, we showed that the solution to the super-resolution problem formulated in Section 2.1 is generally unique if enough samples are taken. In the next chapters, we will present different methods to compute this solution, mainly to find the correct alignment parameters between a set of images.

Chapter 3

Registration of Partially Aliased Signals

In this chapter, we present a method for the registration of partially aliased images [112,117,118,119]. The registration parameters between a pair of images are computed using the aliasing-free part of the spectrum. As we explicitly use the Fourier transform, our method is limited to bandlimited signals described in the Fourier basis. Although this limits the applicability of such a method, it is a very reasonable assumption in practice. The optical system of a digital camera typically acts as a lowpass filter, and attenuates or blocks all high frequencies. The captured image will therefore be an essentially bandlimited signal.

Our method computes the planar shift and rotation parameters between a pair of images. The separability of the shift and rotation estimation is shown in Section 3.1, and frequency domain rotation and shift estimation methods for aliasing-free images are described in Section 3.2 and Section 3.3, respectively. A modification of these algorithms for the registration of partially aliased images is presented in Section 3.4. A super-resolution reconstruction method is presented in Section 3.5, and simulation results are shown in Section 3.6.

3.1 Planar motion estimation

As described in Chapter 2, Fourier based image registration methods only allow global motion in a plane parallel to the image plane. In such a case, the motion between two images can be described as a function of three parameters that are all continuous variables: horizontal and vertical shifts $x_{1,h}$ and $x_{1,v}$ and a planar rotation angle θ_1.

A frequency domain approach allows us to estimate the horizontal and vertical shift and the (planar) rotation separately. Assume we have a continuous two-dimensional reference signal $f_0(\boldsymbol{x})$ and its shifted and rotated version $f_1(\boldsymbol{x})$:

$$f_1(\boldsymbol{x}) = f_0(\mathbf{R}(\boldsymbol{x} + \boldsymbol{x}_1)), \tag{3.1}$$

$$\text{with } \boldsymbol{x} = \begin{pmatrix} x_h \\ x_v \end{pmatrix}, \boldsymbol{x}_1 = \begin{pmatrix} x_{1,h} \\ x_{1,v} \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix}.$$

This can be expressed in Fourier domain as

$$\begin{aligned} F_1(\boldsymbol{u}) &= \iint_{\boldsymbol{x}} f_1(\boldsymbol{x}) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} \\ &= \iint_{\boldsymbol{x}} f_0(\mathbf{R}(\boldsymbol{x} + \boldsymbol{x}_1)) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} \\ &= e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}', \end{aligned} \tag{3.2}$$

with $F_1(\boldsymbol{u})$ the two-dimensional Fourier transform of $f_1(\boldsymbol{x})$ and the coordinate transformation $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{x}_1$. After another transformation $\boldsymbol{x}'' = \mathbf{R}\boldsymbol{x}'$, the relation between the amplitudes of the Fourier transforms can be computed as

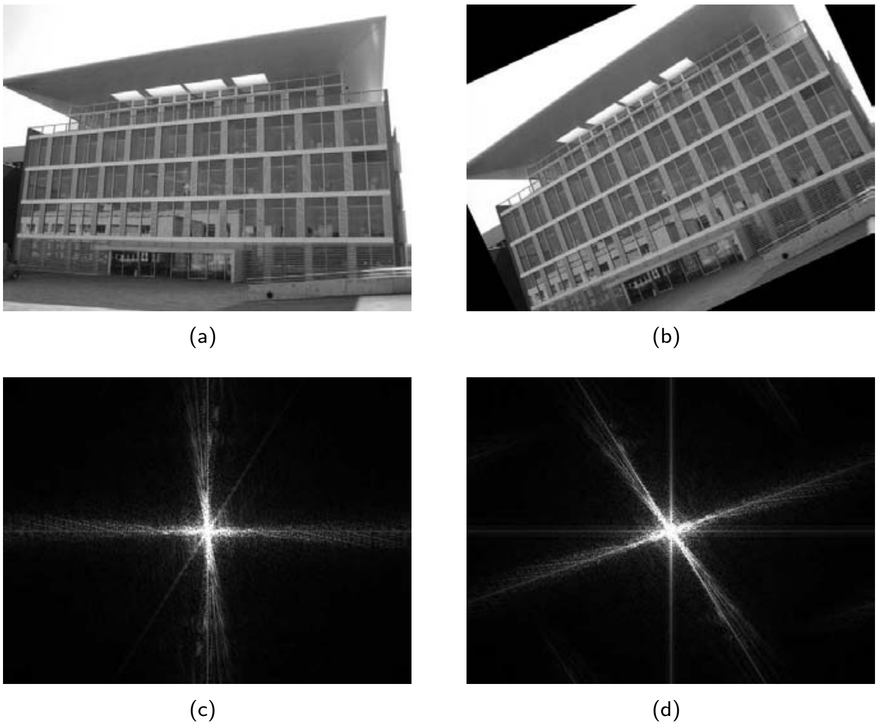
$$\begin{aligned} |F_1(\boldsymbol{u})| &= \left| e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}' \right| \\ &= \left| \iint_{\boldsymbol{x}'} f_0(\mathbf{R}\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}' \right| \\ &= \left| \iint_{\boldsymbol{x}''} f_0(\boldsymbol{x}'') e^{-j2\pi \boldsymbol{u}^T (\mathbf{R}^T \boldsymbol{x}'')} d\boldsymbol{x}'' \right| \\ &= \left| \iint_{\boldsymbol{x}''} f_0(\boldsymbol{x}'') e^{-j2\pi (\mathbf{R}\boldsymbol{u})^T \boldsymbol{x}''} d\boldsymbol{x}'' \right| \\ &= \left| F_0(\mathbf{R}\boldsymbol{u}) \right| . \end{aligned} \tag{3.3}$$

We can see that $|F_1(\boldsymbol{u})|$ is a rotated version of $|F_0(\boldsymbol{u})|$ over the same angle θ_1 as the spatial domain rotation (see Figure 3.1). $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$ do not depend on the shift values $\boldsymbol{x}_1$, because the spatial domain shifts only affect the phase values of the Fourier transforms. Therefore we can first estimate the rotation angle θ_1 from the amplitudes of the Fourier transforms $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$. After compensation for the rotation, the shift $\boldsymbol{x}_1$ can be computed from the phase difference between $F_0(\boldsymbol{u})$ and $F_1(\boldsymbol{u})$.

3.2 Rotation estimation

The rotation angle between $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$ can be computed as the angle θ_1 for which the Fourier transform of the reference image $|F_0(\boldsymbol{u})|$ and the rotated Fourier transform of the image to be registered $|F_1(\mathbf{R}\boldsymbol{u})|$ have maximum correlation. This implies the computation of a rotation of $|F_1(\boldsymbol{u})|$ for every evaluation of the correlation, which is computationally heavy and thus practically difficult.

If $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$ are transformed in polar coordinates, the rotation over the angle θ_1 is reduced to a (circular) shift over θ_1. We can compute the Fourier transform of the polar spectra $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$, and compute θ_1 as the phase shift between the two (as it was done by Marcel et al. [66] and Reddy and Chatterji [87]). This requires a transformation of the spectrum to polar coordinates. The data from the uniform u_h, u_v-grid need to be interpolated to obtain a uniform u_r, u_θ-grid. Mainly for the low frequencies, which generally contain most of the energy, the interpolations are based on very few function values and thus introduce large approximation errors. An implementation of



(a) (b) (c) (d)

Figure 3.1: The amplitude of the Fourier transform of an image is rotated over the same angle ($\theta_1 = 25°$) as the spatial domain image. (a) Original image. (b) Rotated image. (c) Fourier transform amplitude of the original image. (d) Fourier transform amplitude of the rotated image.

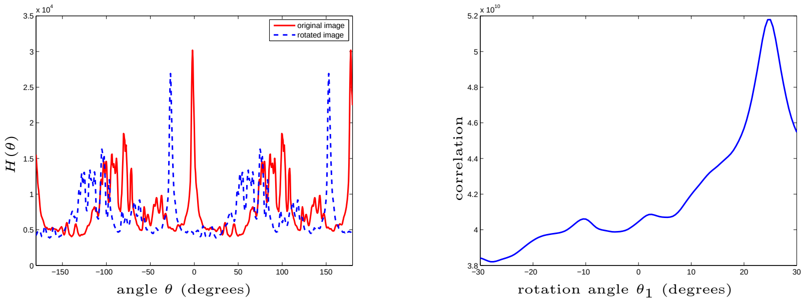
this method is also computationally intensive.

Our approach is computationally much more efficient than the two methods described above. First of all, we compute the frequency content H as a function of the angle θ by integrating over radial lines:

$$H(\theta) = \int_{\theta-\Delta\theta/2}^{\theta+\Delta\theta/2} \int_{0}^{\infty} |F(u_r, u_\theta)| du_r du_\theta. \tag{3.4}$$

In practice, $|F(u_r, u_\theta)|$ is a discrete signal. Different methods exist to relate discrete directions to continuous directions, like for example digital lines [121]. Here, we compute the discrete function $H(\theta)$ as the average of the values on the rectangular grid that have an angle $\theta - \Delta\theta/2 < u_\theta < \theta + \Delta\theta/2$. As we want to compute the rotation angle with a precision of 0.1 degrees, $H(\theta)$ is computed every 0.1 degrees. To get a similar number of signal values $|F(u_r, u_\theta)|$ at every angle, the average is only evaluated on a circular disc of values for which $u_r < \rho$ (where ρ is the image radius, or half the image size). Finally, as the values for low frequencies are very large compared to the other values and are very coarsely sampled as a function of the angle, we discard the values for which $u_r < \epsilon\rho$, with $\epsilon = 0.1$. Thus, $H(\theta)$ is computed as the average of the frequency values on a discrete grid with $\theta - \Delta\theta/2 < u_\theta < \theta + \Delta\theta/2$ and $\epsilon\rho < u_r < \rho$.

This results in a function $H(\theta)$ for both $|F_0(\boldsymbol{u})|$ and $|F_1(\boldsymbol{u})|$ (Figure 3.2). The exact rotation angle can then be computed as the value for which their correlation reaches a maximum. Note that only a one-dimensional correlation has to be computed, as opposed to the two-dimensional correlation approaches in [66] and [87].



(a) (b)

Figure 3.2: Rotation estimation. (a) Average Fourier domain amplitude as a function of the angle $H(\theta)$ for the two images from Figure 3.1. (b) Correlation between $H_0(\theta)$ and $H_1(\theta)$, with a maximum at the rotation angle of $\theta_1 = 25°$.

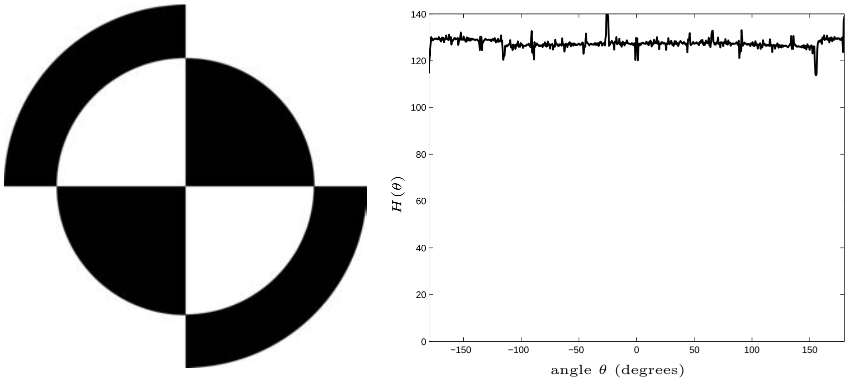
Of course, the use of such a radial projection also reduces the available information, and might introduce ambiguities in the estimation. For example, for the Fourier transform image shown in Figure 3.3(a), the function $H(\theta)$ is a constant, and remains the same for any applied rotation angle. It is therefore impossible to estimate the rotation for such images using our projection-based method, and the full, two-dimensional information is required. Our algorithm works best if some strong directionality is present in the images, as will also be shown in Section 3.6.

3.3 Shift estimation

A shift of the image parallel to the image plane can be expressed in Fourier domain as a linear phase shift:

$$
\begin{aligned}
F_1(\boldsymbol{u}) &= \iint_{\boldsymbol{x}} f_1(\boldsymbol{x}) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} = \iint_{\boldsymbol{x}} f_0(\boldsymbol{x} + \boldsymbol{x}_1) e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}} d\boldsymbol{x} \\
&= e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} \iint_{\boldsymbol{x}'} f_0(\boldsymbol{x}') e^{-j2\pi \boldsymbol{u}^T \boldsymbol{x}'} d\boldsymbol{x}' = e^{j2\pi \boldsymbol{u}^T \boldsymbol{x}_1} F_0(\boldsymbol{u}).
\end{aligned}
\tag{3.5}
$$

It is well known that the shift parameters $\boldsymbol{x}_1$ can thus be computed as the slope of the phase difference $\angle(F_1(\boldsymbol{u})/F_0(\boldsymbol{u}))$ (see also [34, 42, 54, 63, 66, 87, 100]). To make the solution less sensitive to noise, we fit a plane through the phase differences using a least squares method.



(a) (b)

Figure 3.3: Ambiguity in rotation estimation due to projections. For certain images, like the one with Fourier transform shown in (a), the rotation angle can not be estimated. The frequency content as a function of the angle $H(\theta)$ is almost constant (b). The variations that are visible in (b) are due to the discrete grid of pixels.

3.4 Planar motion estimation for partially aliased images

If the low resolution images are aliased, the methods described in the previous sections do not result in precise registration anymore. Due to the aliasing, the relations (3.2), (3.3) and (3.5) between the Fourier transforms of the images are no longer valid. Different parts of the spectrum overlap, and the Fourier transforms of two shifted images can now differ by more than a linear phase shift.

Let us first analyze this on a one-dimensional periodic, bandlimited signal $f(t)$ with maximum frequency K and period $T = 1$ (see Figure 3.4(a)). Its Fourier series coefficients are given by α_k $(-K \leq k \leq K)$. As described in Chapter 2, we sample $f(t)$ with two sets of N samples, resulting in

$$\begin{aligned} y_0(n) &= f\left(\frac{n}{N}\right) \\ y_1(n) &= f\left(\frac{n+t_1}{N}\right). \end{aligned} \tag{3.6}$$

The shift between the two sets of samples is denoted t_1. If $N > 2K$, the sampled signals are not aliased, and their discrete Fourier transforms can be written as (see also 2.23)

$$\begin{aligned} Y_0(k) &= \frac{1}{N}\sum_{n=0}^{N-1} y_0(n)W^{-kn} = \alpha_k \\ Y_1(k) &= \frac{1}{N}\sum_{n=0}^{N-1} y_1(n)W^{-kn} = \alpha_k z_1^k. \end{aligned} \tag{3.7}$$

As discussed above, the shift t_1 can be directly derived from $\mathbf{Y}_1/\mathbf{Y}_0$. However, when the sampled signals are aliased, or $N \leq 2K$, these simple relations do not hold anymore. Instead, we have now (2.23):

$$\begin{aligned} Y_0(k) &= \frac{1}{N}\sum_{n=0}^{N-1} y_0(n)W^{-kn} = \sum_i \alpha_{k+iN} \\ Y_1(k) &= \frac{1}{N}\sum_{n=0}^{N-1} y_1(n)W^{-kn} = \sum_i \alpha_{k+iN} z_1^{k+iN}. \end{aligned} \tag{3.8}$$

Aliasing terms modify the linear phase relation between $\mathbf{Y}_0$ and $\mathbf{Y}_1$. If $N \leq K$, all frequencies are aliased. Methods to reconstruct the signal from such samples will be discussed in Chapter 5 and 6. In this chapter, we consider $N > K$, which means that not all the frequencies are aliased (Figure 3.4(b)). For the low frequencies k, with $|k| < N - K$, only a single term in the sums in (3.8) is non-zero, and $Y_1(k)/Y_0(k)$ is still linear phase. The shift can therefore be estimated from the low, aliasing-free frequencies. A lowpass filter could also be applied to the spatial domain signals, such that the aliased components are removed from the sampled signals (Figure 3.4(c)). The shifts can then be estimated using for example a correlation operation.

An extension to two dimensions is straightforward. The two sampled signals $y_0(\boldsymbol{n})$ and $y_1(\boldsymbol{n})$ are first lowpass filtered with cutoff frequency $\boldsymbol{N} - \boldsymbol{K}$ ($N_h - K_h$ and $N_v - K_v$ in horizontal and vertical dimension, respectively). The filtered images are identical up to their registration parameters, and can be registered using the methods described in Section 3.2 and Section 3.3. As both methods are applied in the Fourier domain, the filtering step can be avoided by applying the registration algorithms immediately to the low frequencies. The rotation estimation is then based on the frequencies for which $\epsilon\rho < u_r < \rho_{max}$ (with $\rho_{max} = \min_{h,v}((\boldsymbol{N} - \boldsymbol{K})/\boldsymbol{N})$), and the horizontal and vertical shifts are estimated from the phase differences for $|\boldsymbol{k}| < \boldsymbol{N} - \boldsymbol{K}$.

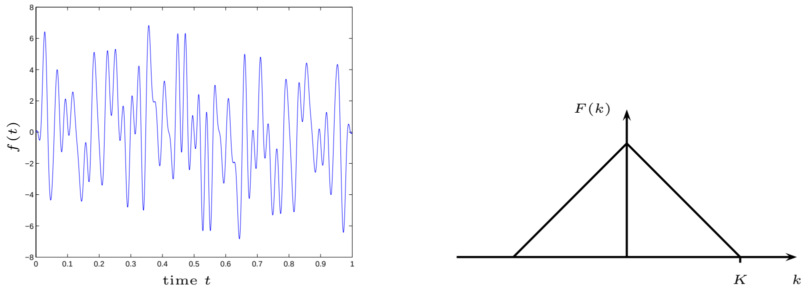
High frequency components of a signal offer a higher precision for registration than low frequency components (the signal changes more rapidly). This motivates the extra emphasis on the high frequency part of the signal in the registration method by Reddy and Chatterji [87]. However, the high frequency part of a signal typically has a lower signal-to-noise ratio, and in undersampled signals, they also suffer most from aliasing. This is why we discard the high frequency components in our registration algorithm. Similar methods for shift estimation on aliased images were presented by Kim and Su [54] and Stone et al. [100].

3.5 Reconstruction method

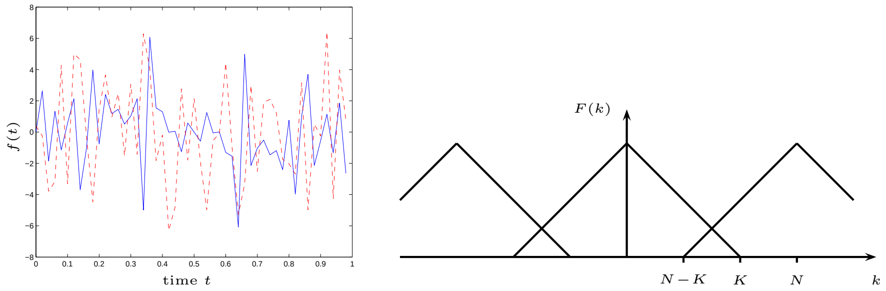
When the low resolution images are accurately registered, the samples of the different images can be combined to reconstruct a high resolution image. In our reconstruction algorithm, the samples of the different low resolution images are first expressed in the coordinate frame of the reference image. Then, based on these known samples, a Delaunay triangulation of the available data is made using the Qhull algorithm [6, 70]. This triangulation is then used to interpolate the image values on a uniform high resolution grid. We chose bicubic interpolation because of its low computational complexity and good results. This completes our super-resolution algorithm. An overview of the complete algorithm is given in Algorithm 3.1. In this algorithm, the parameters $\epsilon = 0.1$ and $\rho_{max} = 0.6$ are chosen based on the fact that we want to discard the lowest, sparsely sampled frequencies, and on the model of partially aliased signals that we used. The precision of the rotation estimation (0.1 degrees), and the extent of the possible rotations (-30 to 30 degrees) are chosen empirically and depend on the specific application.

Algorithm 3.1: Super-resolution imaging on partially aliased images. A high resolution image f_{HR} (with Fourier transform F_{HR}) is reconstructed from a set of M low resolution, partially aliased images $f_{LR,m}$ $(m = 0, 1, \ldots, M-1)$ with Fourier transform $F_{LR,m}$.

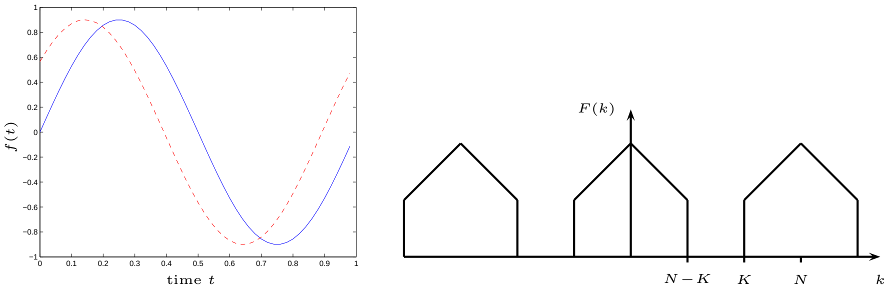
1. Multiply the images $f_{LR,m}$ by a Tukey window to make them circularly symmetric. The windowed images are called $f_{LR,w,m}$.
2. Compute the Fourier transforms $F_{LR,w,m}$ of all low resolution images.
3. *Rotation estimation:* the rotation angle between every image $f_{LR,w,m}$ $(m = 1, \ldots, M-1)$ and the reference image $f_{LR,w,0}$ is estimated.
    (a) Compute the polar coordinates $(u_r,\ u_\theta)$ of the Fourier transform samples.
    (b) For every angle θ, compute the average value $H_m(\theta)$ of the Fourier coefficients for which $\theta - 1 < u_\theta < \theta + 1$ and $0.1\rho < u_r < \rho_{max}$. The angles are expressed in degrees and $H_m(\theta)$ is evaluated every 0.1 degrees. A typical value used for ρ_{max} is 0.6.
    (c) Find the maximum of the correlation between $H_0(\theta)$ and $H_m(\theta)$ between -30 and 30 degrees. This is the estimated rotation angle θ_m.
    (d) Rotate image $f_{LR,w,m}$ by $-\theta_m$ to cancel the rotation.
4. *Shift estimation:* the horizontal and vertical shifts between every image $f_{LR,w,m}$ $(m = 1, \ldots, M-1)$ and the reference image $f_{LR,w,0}$ are estimated.
    (a) Compute the phase difference between image m and the reference image as $\angle(F_{LR,w,m}/F_{LR,w,0})$.
    (b) For all frequencies $|\boldsymbol{k}| < \boldsymbol{N} - \boldsymbol{K}$ write the linear equation describing a plane through the computed phase difference with unknown slopes $\boldsymbol{x}_m$.
    (c) Find the shift parameters $\boldsymbol{x}_m$ as the least squares solution of the equations.
5. *Image reconstruction:* a high resolution image f_{HR} is reconstructed from the registered images $f_{LR,m}$ $(m = 0, \ldots, M-1)$.
    (a) For every image $f_{LR,m}$, compute the coordinates of its pixels in the coordinate frame of $f_{LR,0}$ using the estimated registration parameters.
    (b) From these known samples, interpolate the values on a uniform high resolution grid using bicubic interpolation.



(a) Original continuous-time signal in time and frequency domain.



(b) Sampled signal in time and frequency domain, with aliasing.



(c) Lowpass filtered sampled signal in time and frequency domain.

Figure 3.4: In the presence of (partial) aliasing, the shift between two sampled signals cannot be found directly. However, after lowpass filtering, the shift can be easily determined.

Number of Images The optimal number of low resolution images to use for a super-resolution algorithm depends on many parameters: registration accuracy, imaging model, total frequency content, etc. Intuitively, there is a trade-off between two effects. On one hand, the more images there are, the better the reconstruction should be. On the other hand, there is a limit to the improvements that can be obtained: even from a very large number of very low resolution images of a scene, it will not be possible to reconstruct a sharp, high resolution image. Blur, noise, and inaccuracies in the signal model limit the increase in resolving power that can be obtained. In our case, the motion estimation algorithm is limited to subsampling by a factor less than two in both dimensions (because our algorithm needs an aliasing-free part of the spectrum, see also Section 3.4). Therefore, the resolution can only be really increased by (almost) a factor of four. Any supplementary increase in number of pixels can as well be performed by upsampling one of the signals and applying lowpass interpolation. It does not result in an increase of resolving power, but noise can be reduced. Figure 3.5 shows the mean squared error (MSE) of the reconstruction using our algorithm versus the number of images used. The performance increases rapidly with the first six images, but the improvement is marginal beyond that.

In the simulations in the next section, we will use four images as input to the super-resolution algorithms. Assuming the low resolution images were subsampled by almost two, this is the theoretical limit for which our algorithm should be able to reconstruct an image of almost double resolution. In other words, four images are a minimum to have a well-determined system when upsampling by two.

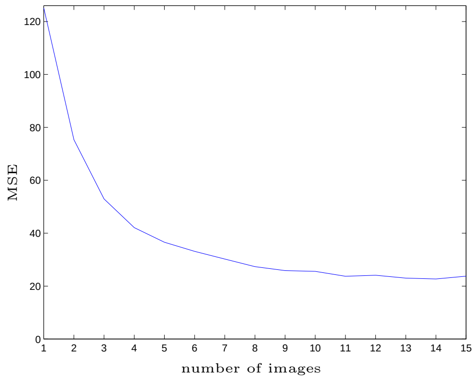
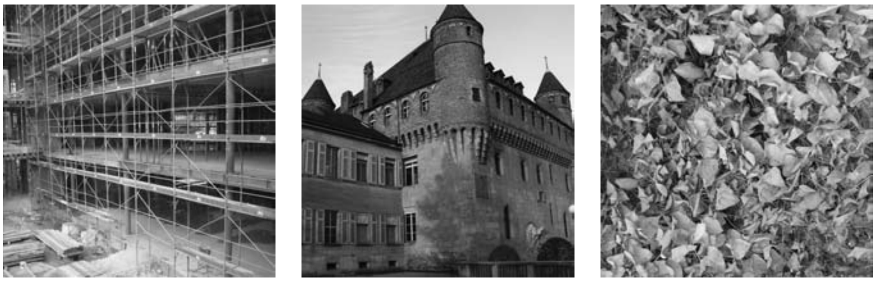


Figure 3.5: Mean squared error (MSE) of the reconstructed image as a function of the number of images used in the super-resolution algorithm from Algorithm 3.1. Six images form a good trade-off between performance and computational complexity. We performed 25 simulations on the images shown in Figure 3.6(a) and 3.6(b) for each number of input images.

3.6 Results

The super-resolution algorithm described above is tested with numerical simulations. Practical experiments will be described in Chapter 4. A simulation gives complete control over the setup and gives exact knowledge of the registration parameters. It enables us to test the performance of the registration and the reconstruction algorithms separately. The three images that were used in the simulations are shown in Figure 3.6.



(a) Building (b) Castle (c) Leaves

Figure 3.6: High resolution images used in the simulations.

We compared our registration algorithm to the frequency domain algorithms by Marcel et al. [66] and by Lucchese and Cortelazzo [63]. Because Lucchese and Cortelazzo use the same phase correlation method as Marcel et al., the method by Lucchese and Cortelazzo is not included in the simulations where only shifts are used. Next, we also compared it to the spatial domain method based on Taylor expansions by Keren et al. [51]. In the simulations using only shifts, our registration method was also compared to the algorithm by Bergen et al. [8], as it was implemented in the super-resolution imaging software by Farsiu et al. [30]. This was only done for the case of horizontal and vertical shifts, because image rotations are not (yet) implemented in this software.

We started from a high resolution image (3536×3536 pixels), which we consider as equivalent to continuous space. This image was then multiplied by a Tukey window (Figure 3.7(a)) to make it circularly symmetric and thus avoiding all boundary effects. Next, three shifted and rotated copies are created from this high resolution image. Gaussian zero-mean random variables are used for the shift (pixels) and rotation (degrees) parameters. For the shifts, a standard deviation of 5 is used, while the rotation angles have a standard deviation of 0.5. The different images are then lowpass filtered using an ideal lowpass filter with cutoff frequency $0.12\boldsymbol{N}_{HR}$ (with $\boldsymbol{N}_{HR}$ the sampling frequency of the high resolution image) to achieve the setup specified in Section 3.4 ($\boldsymbol{K} < \boldsymbol{N} < 2\boldsymbol{K}$). The first of these images (not-moved reference image) will be the reconstruction target for the super-resolution algorithm (Figure 3.7(b)). And finally, the four images are downsampled by a factor eight. This results in four low resolution (442×442 pixels), shifted and rotated images that can be used as input for the super-resolution algorithm (Figure 3.8(a)). They are aliasing-free in the frequency band $(-0.04\boldsymbol{N}, 0.04\boldsymbol{N})$, and are aliased in the rest of the spectrum as discussed in Section 3.4. By construction, all shifts are multiples of 0.125, but this information is not used in any of the registration algorithms to keep them generally applicable. An example of a reconstructed image is shown in Figure 3.8(b).

The results using the different algorithms are summarized in Table 3.1. The registration results with our algorithm are better than the frequency domain algorithm by Marcel et al. [66] and similar to the algorithm by Lucchese and Cortelazzo [63]. Our algorithm performs worse on the rotation estimation, but an order of magnitude better in the shift estimation. Also, when the variance of the rotations is increased, our algorithm has similar behavior as in Table 3.1, while the other frequency domain algorithms degrade rapidly. The algorithm by Keren et al. [51] performs best in the rotation estimation and similar to our algorithm in the shift estimation.

Table 3.1: Shift and rotation estimation. Comparison of the average absolute error (μ) and the standard deviation of the error (σ) for the shift and rotation parameters in the different algorithms. 150 simulations were performed for each of the images (Figure 3.6).

Parameter	Our algorithm		Marcel et al.		Lucchese et al.		Keren et al.	
	μ	σ	μ	σ	μ	σ	μ	σ
shift (pixels)	0.030	0.041	0.565	0.801	0.318	0.402	0.011	0.016
rot. angle (deg)	0.139	0.216	0.378	0.462	0.081	0.110	0.024	0.033

Another simulation was also made with the same setup, but only using horizontal and vertical shifts (no rotations). The results of this simulation are listed in Table 3.2. Our algorithm outperformed the other methods and computed the parameters with very high precision. The algorithm by Marcel et al. [66] has clearly lower precision than the other algorithms. The spatial domain algorithms by Keren et al. [51] and by Bergen et al. [8] (as implemented by Farsiu et al. [30]) outperform the frequency domain algorithm by Marcel et al., but have lower precision than our algorithm.

Table 3.2: Shift estimation. Comparison of the average absolute error (μ) and the standard deviation of the error (σ) for the shift parameter in the different algorithms. 150 simulations with only horizontal and vertical shifts (no rotations) were performed for each of the images (Figure 3.6).

Parameter	Our algorithm		Marcel et al.		Keren et al.		Bergen et al.	
	μ	σ	μ	σ	μ	σ	μ	σ
shift (pixels)	3.1e-5	2.0e-4	0.314	0.377	4.1e-3	5.9e-3	5.5e-3	7.9e-3

Finally, a simulation was done using the same setup as for Table 3.1, but without the windowing operation from Figure 3.7(a). This means that the input images are no longer periodic, an assumption that is made in any Fourier domain approach. However, such a setup is also more realistic, as real digital images are rarely periodic. The results of these simulations are shown in Table 3.3.

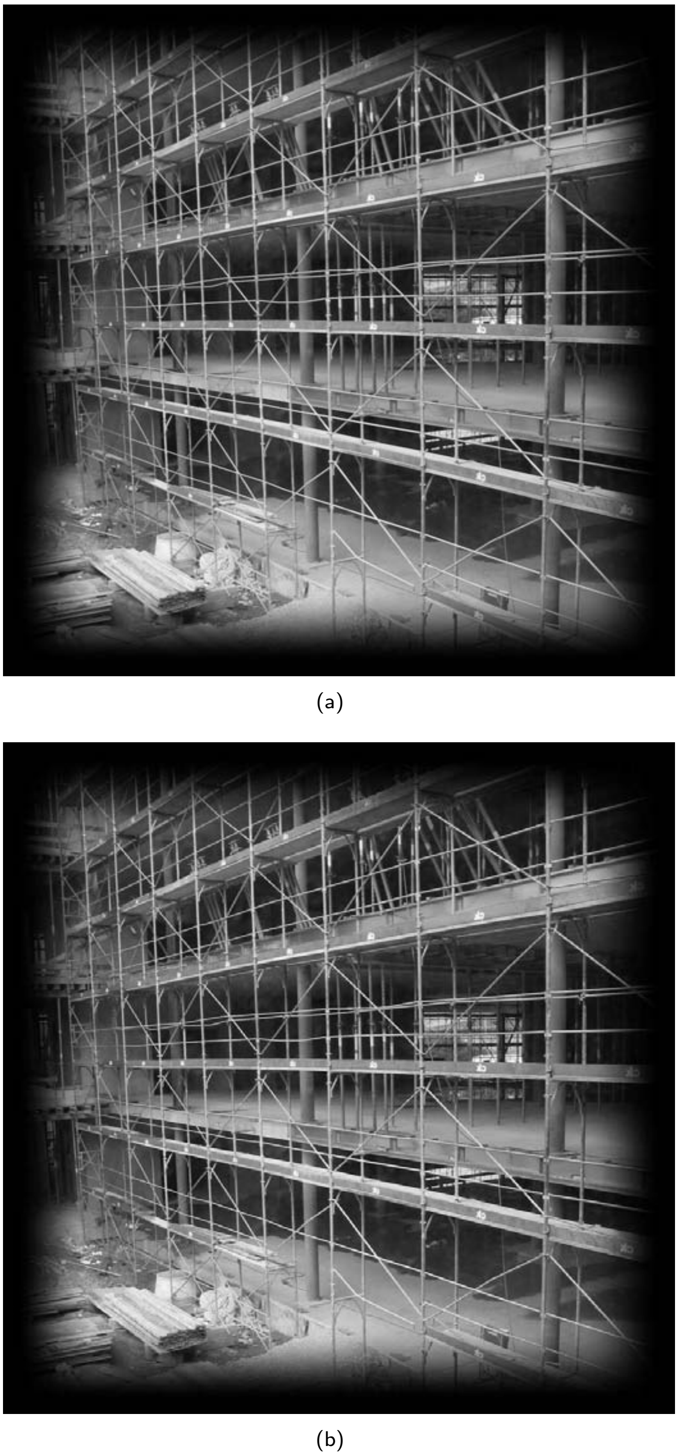
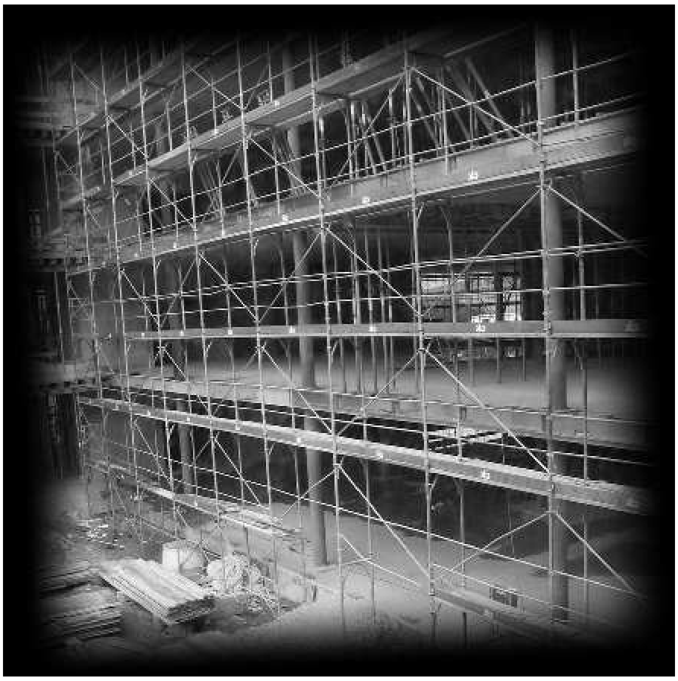
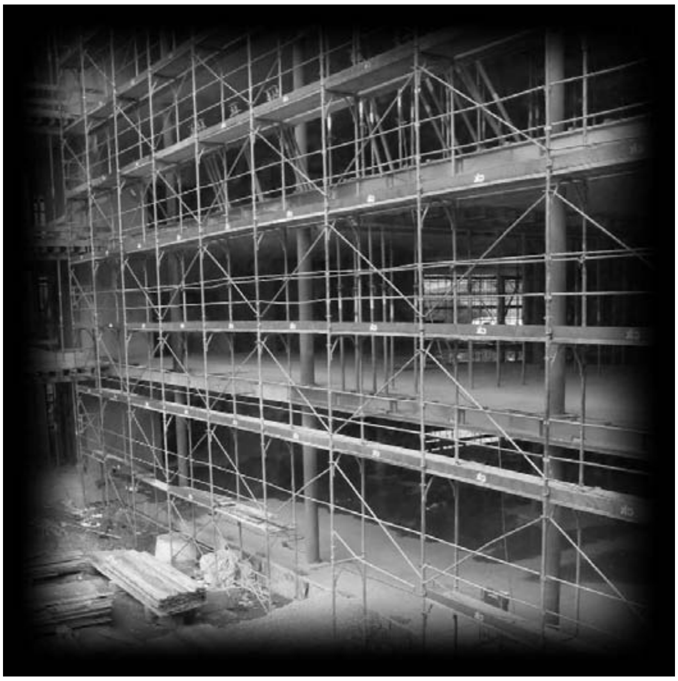


Figure 3.7: Simulation setup. (a) Original image multiplied by a window to make it circularly symmetric. (b) Lowpass filtered image to satisfy the reconstruction conditions. This image is used as reconstruction target.



(a)



(b)

Figure 3.8: Simulation setup. (a) Low resolution image used as input to the super-resolution algorithm. (b) Reconstructed high resolution image.

As expected, the performance of the Fourier domain methods decreases. Our algorithm performs better now than the other frequency domain methods. More surprisingly, the performance of the spatial domain algorithm by Keren et al. is also lower than in the windowed case. This is mainly due to the border effects: the shifts and rotations now cause differences between the images at their borders. It still outperforms all the other algorithms, though.

Table 3.3: Shift and rotation estimation without windowing. Comparison of the average absolute error (μ) and the standard deviation of the error (σ) for the shift and rotation parameters in the different algorithms. No windowing operation was applied to make the images periodic. 150 simulations were performed for each of the images (Figure 3.6).

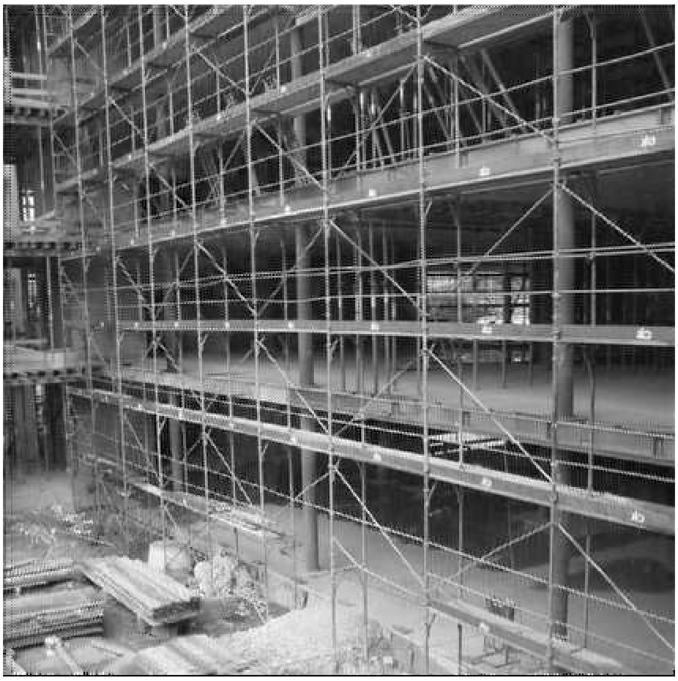
Parameter	Our algorithm		Marcel et al.		Lucchese et al.		Keren et al.	
	μ	σ	μ	σ	μ	σ	μ	σ
shift (pixels)	0.484	0.702	0.815	4.841	47.025	80.788	0.135	0.275
rot. angle (deg)	0.237	0.324	1.173	16.923	13.558	27.242	0.033	0.038

In order to find the same motion parameters in the registration as the parameters that were used to create the images, we need to reverse the order in the registration. In other words, because we first shifted the images and then rotated them in the simulation setup, we need to undo the rotation first and then the shifts. Otherwise, a conversion would have to be made before comparing the two.

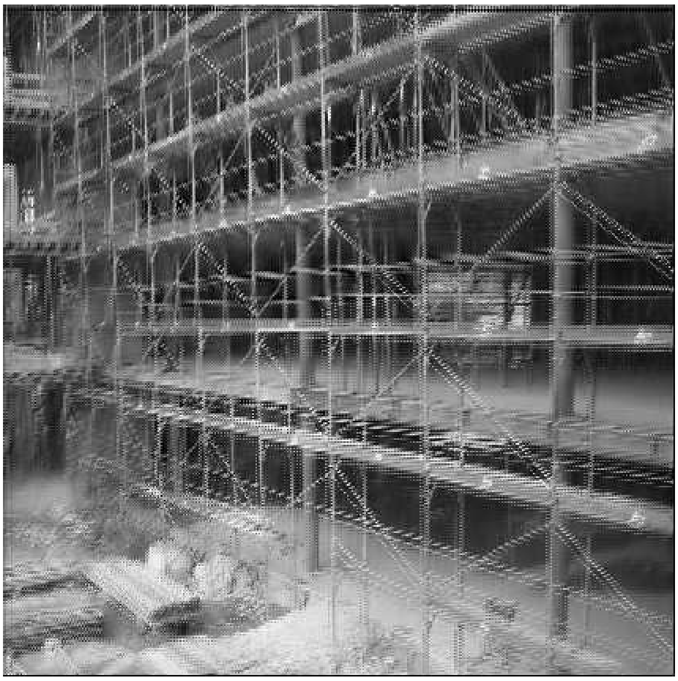
A very precise registration algorithm is required for any super-resolution algorithm to work. From the comparison in the simulations, it is clear that our frequency domain algorithm and the spatial domain algorithm by Keren et al. [51] are accurate enough to improve resolution and remove aliasing artifacts. The other frequency domain algorithms by Marcel et al. [66] and Lucchese and Cortelazzo [63] perform worse.

We can also observe that a bad image registration is fatal for the reconstruction. In such cases, it would be better to reconstruct a larger image from only one of the low resolution images using interpolation, even though this does not increase the resolution. The artifacts due to bad motion estimation are visually very noticeable (see Figure 3.9 and 3.10).

Directionality Our algorithm works best on images with strong frequency content in certain directions (Figure 3.11(a) and (b)). In that case, our algorithm outperforms all other algorithms, including the spatial domain algorithm by Keren et al. [51] (mainly for large rotation angles). The accuracy of our rotation estimation (and consequently also of the shift estimation) depends on the presence of some strong directionality in the images. This can be observed in Table 3.4, where the results from Table 3.1 for our algorithm are displayed per image. If such frequency directions are not present (Figure 3.11(c) and (d)), the registration performance decreases. The results with our rotation estimation algorithm are then worse than with the algorithm by Keren et al., but still comparable to or better than those using the other frequency domain algorithms.



(a)



(b)

Figure 3.9: If the images are badly registered, it is better to interpolate a single image than to reconstruct from all the images together. (a) Reconstruction after accurate registration. (b) Reconstruction after bad registration.

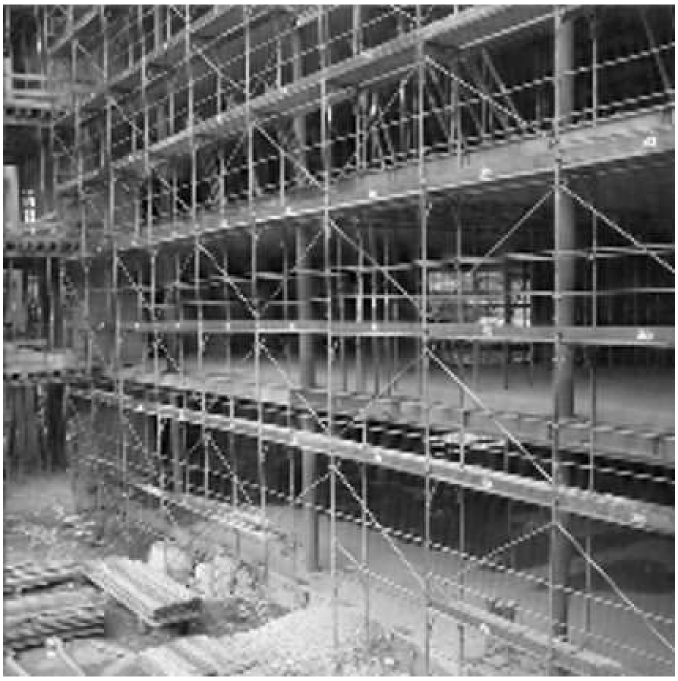


Figure 3.10: If the images are badly registered, it is better to interpolate a single image than to reconstruct from all the images together. Bicubic interpolation on a single image.

This dependence on directionality is related to the projection along radial lines in our rotation estimation algorithm. This highly reduces the computational complexity of the algorithm, as only a one-dimensional correlation is required instead of the regular two-dimensional correlations. However, because of the projection, it is also more subject to errors if there are no strong directions in the image.

Table 3.4: Rotation estimation works better on images with strong directionality. Comparison of the average absolute error (μ) and the standard deviation of the error (σ) for the shift and rotation parameters on different images. 150 simulations were performed for each of the images (Figure 3.6).

Parameter	image 3.6(a)		image 3.6(b)		image 3.6(c)	
	μ	σ	μ	σ	μ	σ
shift (pixels)	0.036	0.044	0.027	0.030	0.028	0.045
rotation angle (deg)	0.045	0.031	0.070	0.038	0.304	0.363

Image Size Next to the presence of directional frequency content, the size of the low resolution images also constrains the precision of our rotation estimation algorithm. As the frequency values have to be averaged over a small angle (typically a few degrees), the number of values to be averaged will be very

Table 3.5: Better performance using larger images. Comparison of the average absolute error (μ) and the standard deviation of the error (σ) for the shift and rotation parameters for different image sizes using our algorithm. 150 simulations were performed for each of the three images (Figure 3.6).

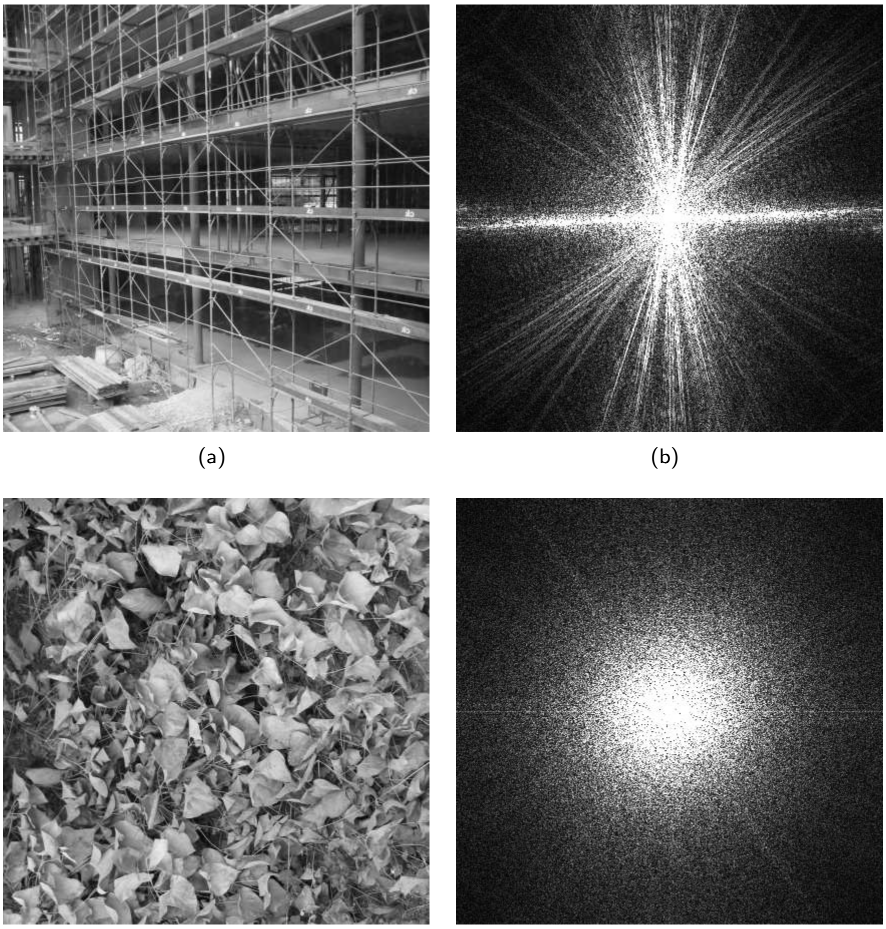
Input image size	221x221 pixels		442x442 pixels	
	μ	σ	μ	σ
shift (pixels)	0.2022	0.6022	0.0303	0.0412
rotation angle (degrees)	0.2539	0.3211	0.1395	0.2167

limited for small images. This number of values also varies for different angles (e.g. more values around 0 and 90 degrees, less in between), which biases the computed functions. In Table 3.5, simulation results with our algorithm are compared for different image sizes. This explains also why we consider a large disc for the rotation estimation, as estimates based on the aliasing-free part alone are not accurate enough.

3.7 Conclusions

In this chapter we described a frequency domain method for the registration of planarly shifted and rotated images. The rotation estimation is based on a radial projection of the Fourier transforms of the images. The rotation angle is then found as the maximum of the (one-dimensional) correlation between two such projections. Shifts are estimated from the (linear) phase difference between the two Fourier transforms. This method was extended to partially aliased images by using only the frequencies that are not aliased. A super-resolution method based on this registration algorithm was given, and its performance tested in a number of simulations. Our method outperforms most other existing registration methods. It works best if the input images are large enough, and have some strong directionality. This sensitivity to image size and directionality is due to the projections in the rotation estimation.

All the algorithms described above (except the one by Bergen et al.) are also implemented in the Matlab program with GUI available online [120].



(a) (b)

(c) (d)

Figure 3.11: Our algorithm works best on images with strong frequency content in certain directions ((a) and its Fourier transform (b)). If the energy is homogeneously spread among all possible directions (as can be seen in (c) and its Fourier transform (d)), the performance of the motion estimation algorithm decreases due to the projections onto a one-dimensional signal performed in our method.

Chapter 4

Application to Digital Cameras

In this chapter we apply the super-resolution techniques described in Chapter 3 to some sets of real images taken with a digital camera. Such experiments allow us to test our algorithms on some real measurement data. However, as the true motion parameters and the exact high resolution image are unknown, the results can only be judged visually. First, we measure the spatial frequency response of the digital cameras, to verify that frequencies above half the sampling frequency are captured. Then we apply Algorithm 3.1 to the sequences of images taken with these cameras and reconstruct an image with almost twice the original resolution. All aliasing artifacts are removed in the resulting images. We applied these experiments both to a Leica DC250 and a Sigma SD10 digital camera.

4.1 Leica DC250 camera

We ran a first experiment with a Leica DC250 black and white digital camera. We used a Nikon 85mm lens in this experiment that was mounted on the Leica camera using a C mount to Nikon adapter. A picture of the camera with optical system is shown in Figure 4.1. This camera produces images of 1280×1024 pixels, and is often used in microscopy.

4.1.1 Spatial frequency response

The spatial frequency response (SFR) of the camera is measured following the ISO 12233:2000 standard on resolution measurements [45]. We measure first the opto-electronic conversion function (OECF) of the camera to linearize the camera response function. We use the 15 patches of the ISO 15739 noise test chart for this (see Figure 4.3(a)), and measure for each patch the luminance and the average camera response. This results in 15 values of the OECF. In between, the function is linearly interpolated such that we obtain its value for all the 256 possible gray levels. The OECF for the Leica DC250 is shown in Figure 4.2.

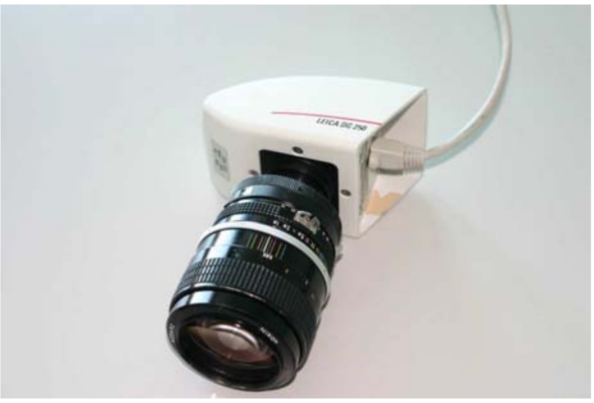


Figure 4.1: Leica DC250 digital camera with a Nikon 85mm optical system used in the first experiment.

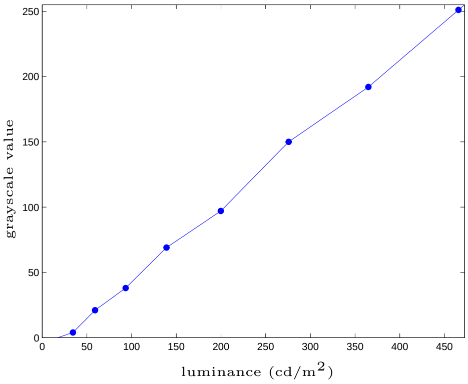


Figure 4.2: Opto-electronic conversion function (OECF) for the Leica DC250 digital camera. Values are linearly interpolated between the 15 measured values.

Next, four images are taken of the ISO 12233:2000 resolution test chart to measure the spatial frequency response (see Figure 4.3(b)). The spatial frequency responses are computed for both horizontal and vertical frequencies using the slanted black to white and white to black edges at the center of the chart. This can be done using the SFRwin application. It computes the SFR based on the interpolated edge point at each row or column. The results are then averaged over the four images, and give the final SFR measurement (see Figure 4.4(a)).

The computed spatial frequency response uses relative frequencies that are normalized with respect to the sampling frequency. With an image height of the resolution chart of 1017 pixels, and a pixel pitch[1] of 6.7 μm, this corresponds to 1017 line widths per picture height and 75 cycles/mm[2] on the image sensor.

---

[1] The pixel pitch is defined as the distance between the centers of two adjacent pixels.

[2] The number of cycles/mm specifies the resolution characteristics in terms of the response of an imaging system to a linear radiance sine wave input, as a function of the frequency of the sine wave [45].

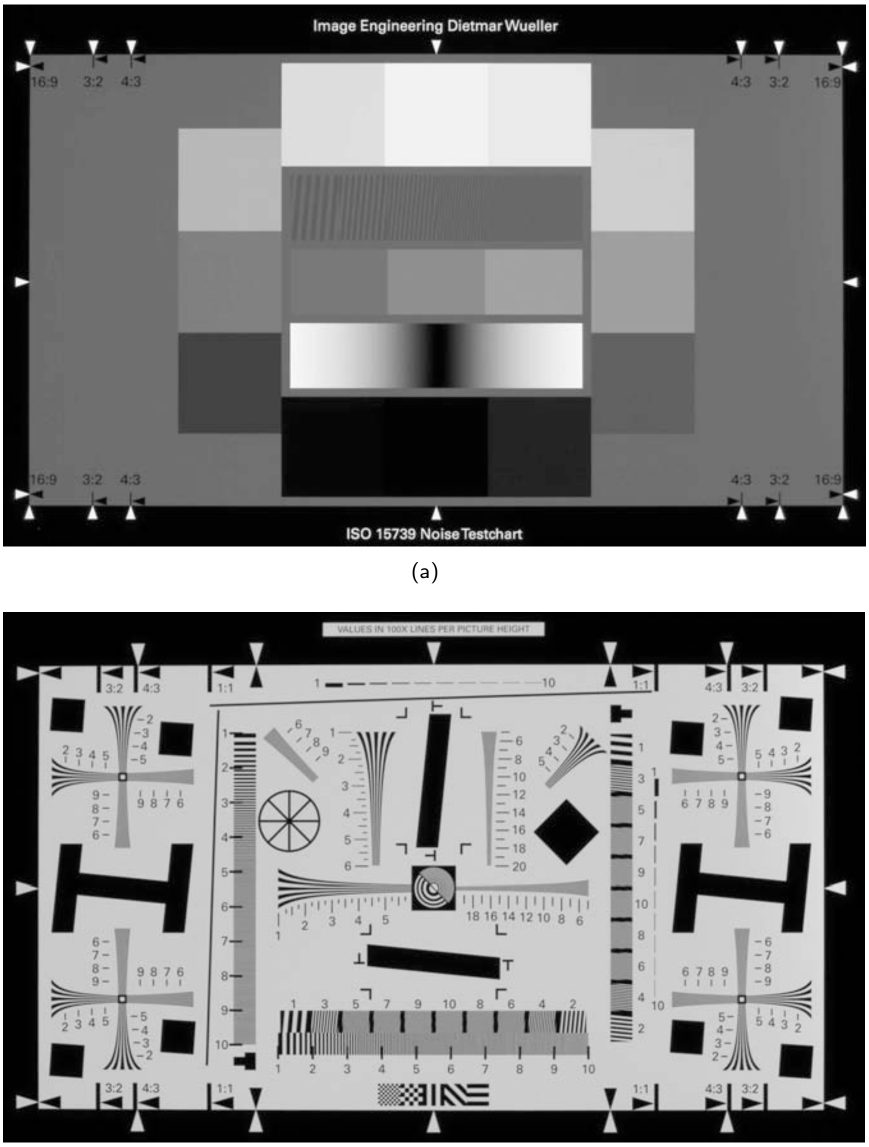
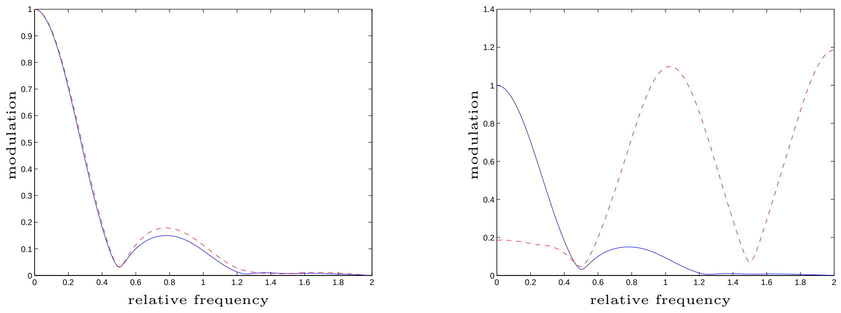


Image Engineering Dietmar Wueller
16:9
3:2
4:3
ISO 15739 Noise Testchart
VALUES IN 100X LINES PER PICTURE HEIGHT
1:1
3:2
4:3

(a)

(b)

Figure 4.3: (a) ISO 15739 noise test chart used to measure the camera OECF. (b) ISO 12233 resolution test chart used to measure the camera SFR.



(a) (b)

Figure 4.4: (a) Horizontal (– –) and vertical (—) relative spatial frequency response (SFR) for the Leica DC250 digital camera using a Nikon 85mm lens. A relative spatial frequency of 1 corresponds to the sampling frequency, or 1017 line widths per picture height and 75 cycles/mm on the image sensor. (b) Vertical relative spatial frequency response (—) and its aliased versions (– –) after sampling.

From Figure 4.4(a) we can see that the spatial frequency response is non-zero well beyond the limit of half the sampling frequency. The resulting images will therefore be aliased, as it is also shown in Figure 4.4(b). This figure shows the SFR with its aliased components after sampling.

4.1.2 Super-resolution image reconstruction

As shown in Figure 4.4, the images taken with this camera setup are aliased. This means that we can apply our super-resolution techniques to these images and reconstruct a higher resolution image by removing the aliasing. Although there is no aliasing-free part in the spectrum, the signal-to-aliasing ratio is relatively high for low frequencies, and we can still apply Algorithm 3.1.

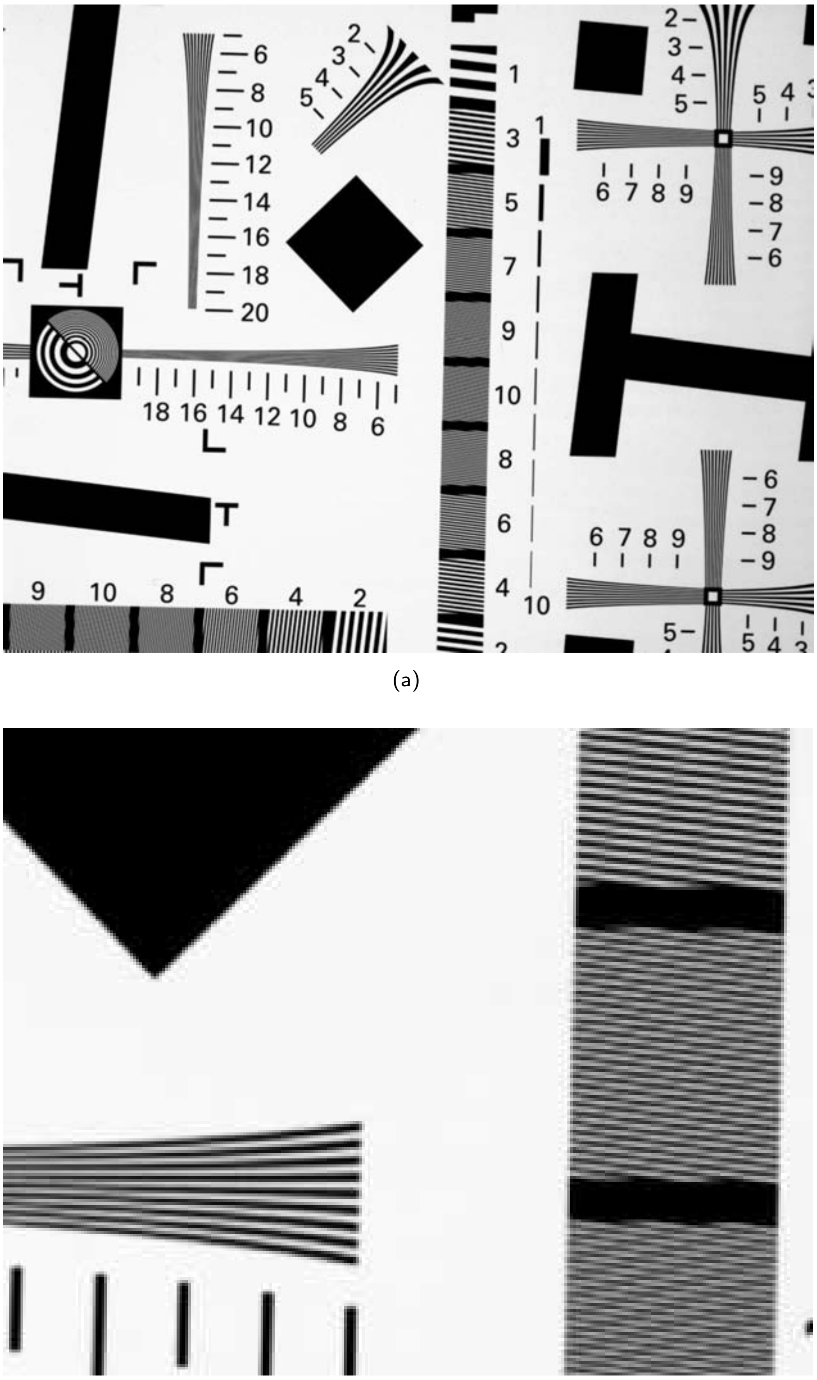
For the actual experiment, we fixed the camera firmly on a stable tripod that allows only horizontal and vertical shifts and planar rotations parallel to the image plane. Using this setup, we took four shifted and rotated images of a planar scene (Figure 4.5). The planar scene is a resolution test chart in a plane parallel to the image plane of the camera. Algorithm 3.1 can then be applied to these four images, and an image with almost double resolution is reconstructed (see Figure 4.6). The aliasing is accurately removed in this high resolution image and the original patterns are visible again. The computed registration parameters are given in Table 4.1, together with the parameters found using the algorithms by Marcel et al. [66], Lucchese and Cortelazzo [63], and Keren et al. [51]. These are the same algorithms as the ones used for comparison in Chapter 3. As the actual motion parameters are unknown, we can only compare the results after reconstruction of the high resolution image (see Figure 4.7 and 4.8). For the reconstruction of the high resolution images, we used the bicubic interpolation method described in Chapter 3. The results shown in Figure 4.6 - 4.8 are very good for both Algorithm 3.1 and the algorithm by Keren et al. From the results obtained with the algorithms by Marcel et al. and by Lucchese and Cortelazzo, we can see that at least one of the images

Table 4.1: Registration parameters for the four Leica images of a resolution chart using Algorithm 3.1 and the algorithms by Marcel et al., Lucchese and Cortelazzo, and Keren et al.

image pairs	Algorithm 3.1			Marcel et al.		
	$x_{m,h}$	$x_{m,v}$	θ_m	$x_{m,h}$	$x_{m,v}$	θ_m
image 2 - image 1	9.24	-3.84	0.9	11.2	-4.2	1.06
image 3 - image 1	9.74	-2.21	1.2	12.4	0.2	1.39
image 4 - image 1	10.32	-5.00	1.2	12.4	-5.0	1.39
	Lucchese et al.			Keren et al.		
	$x_{m,h}$	$x_{m,v}$	θ_m	$x_{m,h}$	$x_{m,v}$	θ_m
image 2 - image 1	9.00	-0.50	1.21	9.27	-3.86	0.92
image 3 - image 1	10.00	2.00	1.68	9.86	-2.29	1.14
image 4 - image 1	11.25	-0.75	1.63	10.37	-5.06	1.17

was badly aligned. These results are clearly worse than our reconstruction and perform no improvement over the lower resolution input images.

In this experiment, the maximum signal frequency is unknown. As shown in Figure 4.4, the entire spectrum is aliased, so no aliasing-free part can be determined. We fixed the part of the spectrum from which shifts are estimated to the central 5% of the frequencies, as this is also the part with the highest signal-to-aliasing ratio. For the rotation estimation, a disc with $\rho = 0.6$ is used in order to get sufficient precision. This is also approximately the area in which the aliased component of the sampled spectrum is smaller than the base component. The low performance of the algorithm by Marcel et al. and the shift estimation part of the algorithm by Lucchese and Cortelazzo (which is the same as the shift estimation by Marcel et al.) can be partially explained by their memory requirements. Due to the size of the input images, the required additional upsampling and interpolation in these methods uses too much memory to be performed on a regular computer. The upsampling is therefore only performed at a lower rate, and the shifts are consequently also computed at lower precision.



(a)

(b)

Figure 4.5: One of the four images taken with the Leica DC250 camera as input to Algorithm 3.1 (a), with a detail that clearly shows the aliasing in the image (b).

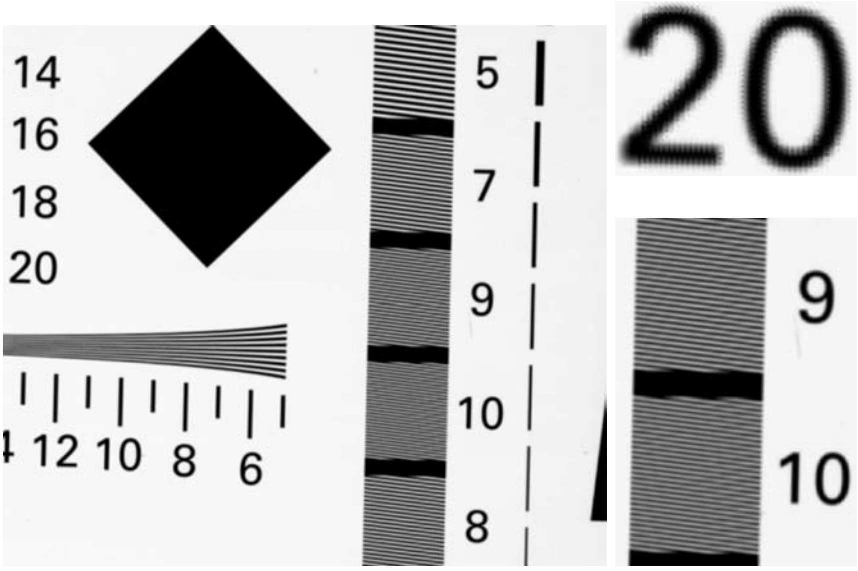


Figure 4.6: High resolution image reconstructed from four Leica DC250 input images using Algorithm 3.1. Zoomed images of the central part are shown to display the differences better.

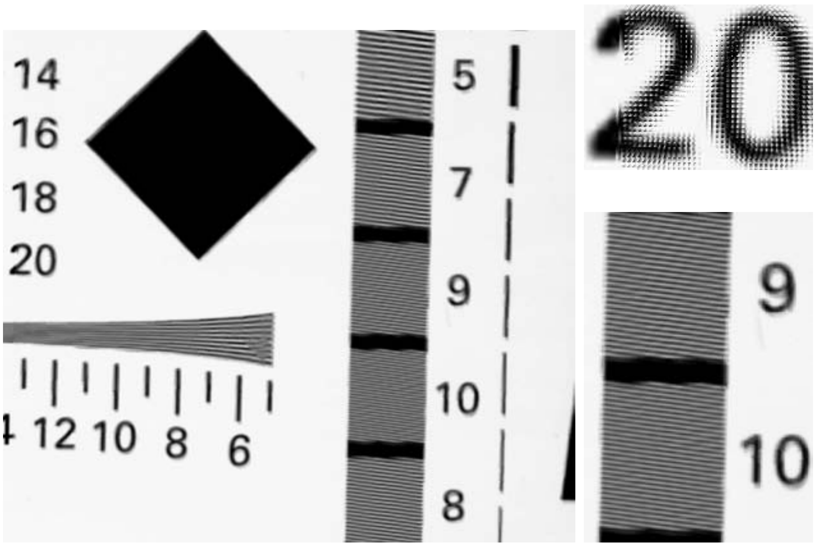
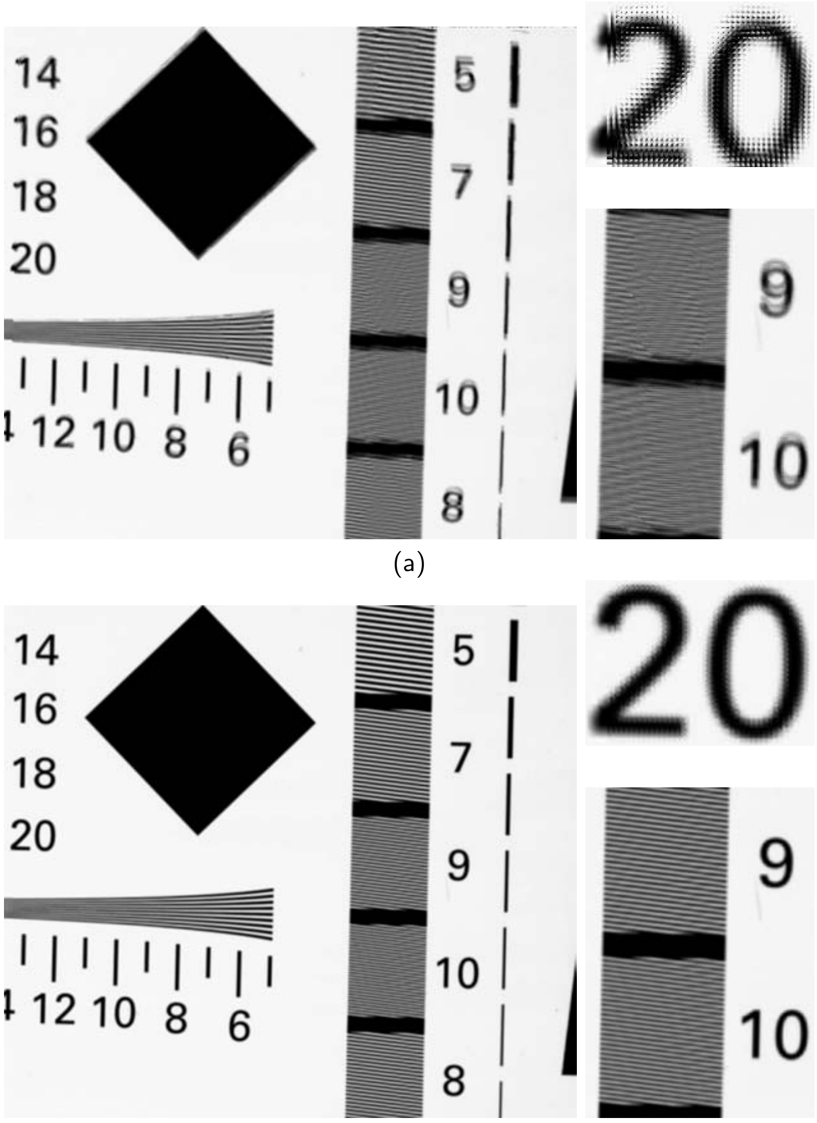


Figure 4.7: Results of super-resolution reconstruction using the registration algorithm by Marcel et al. on the Leica DC250 images of the resolution chart. Zoomed images of the central part are displayed to show the differences better.



(a)

(b)

Figure 4.8: Results of other super-resolution algorithms on the Leica DC250 images of the resolution chart. Zoomed images of the central part are displayed to show the differences better. (a) Registration algorithm by Lucchese and Cortelazzo. (b) Registration algorithm by Keren et al.

4.2 Sigma SD10 camera

In a second experiment, we used a Sigma SD10 digital camera (see Figure 4.9). This camera uses a Foveon X3 sensor, which has three photodetectors (for red, green, and blue) at every pixel location. We used a Sigma 18-50 mm lens with this camera. The images captured are 2268 × 1512 pixels.

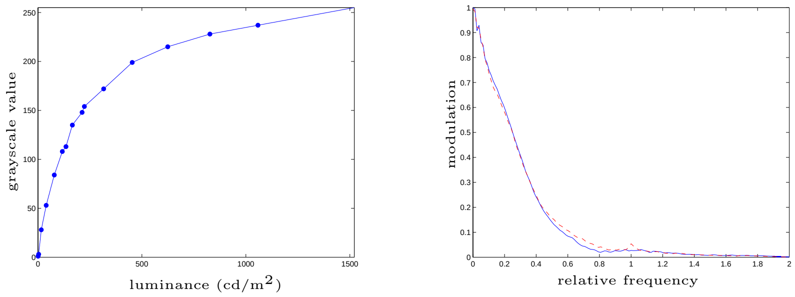
Figure 4.9: Sigma SD10 digital camera with a Sigma 18-50 mm lens used in the second experiment.

4.2.1 Spatial frequency response

First, we measured the camera's spatial frequency response using the same methodology as described in the previous section. Its OECF and SFR are shown in Figure 4.10(a) and 4.10(b), respectively. With a pixel pitch of 9.12 μm and an image height of 1512 pixels, the sampling frequency corresponds to 1512 line widths per picture height, or 55 cycles per millimeter on the image sensor. We can again see that the modulation is non-zero above half the sampling frequency, indicating that aliasing artifacts can occur in images taken with this camera.

4.2.2 Super-resolution image reconstruction

Four pictures were taken with this camera in an outdoor environment. During the image capture, the camera was held manually in approximately the same position. This causes small shifts of the order of ten pixels and rotations of a fraction of a degree between the images. Although these movements are not guaranteed to be parallel to the image plane, we can make this approximation, as the movements are small and the objects are at a large distance. There was no motion in the scene between the different images. A uniform motion would pose no problems, and is equivalent to a camera motion. However, if there would be non-uniform motion of different objects in different directions, our method would fail because it fits a single set of motion parameters to the whole image. One of these input images is shown in Figure 4.11, with a detail



(a) (b)

Figure 4.10: (a) Opto-electronic conversion function (OECF) for the Sigma SD10 digital camera. The function is linearly interpolated between the 15 measurements. (b) Horizontal (– –) and vertical (—) relative spatial frequency response (SFR) for the Sigma SD10 digital camera using a Sigma 18-50 mm optical system that was set to focus at 35 mm. A relative spatial frequency of 1 corresponds to the sampling frequency, or 1512 line widths per picture height and 55 cycles/mm on the image sensor.

where aliasing is visible in Figure 4.13. Aliasing artifacts can be seen in the high frequency horizontal ventilation grids.

We apply Algorithm 3.1 to these four images, and obtain a double resolution image without any aliasing artifacts (Figure 4.12(a) and 4.14). As in the previous experiment, this result is compared to the results obtained with the algorithms by Marcel et al. [66], Lucchese and Cortelazzo [63], and Keren et al. [51] (see Figure 4.12 and 4.15 - 4.17). The registration parameters for the different algorithms are given in Table 4.2. Because the images are even larger than those used in the first experiment, no more interpolation could be performed in the algorithms by Marcel et al. and the shift estimation by Lucchese and Cortelazzo. The parameters are estimated up to integer precision only. Like in the first experiment, the algorithms by Marcel et al. and by Lucchese and Cortelazzo do not give satisfactory results. The results obtained with Algorithm 3.1 and the algorithm by Keren et al. are comparable, and accurately remove the aliasing.

4.3 Conclusions

In two practical experiments, we have shown the good performance of the algorithm presented in Chapter 3. In both experiments, we used a sequence of four images taken with a digital camera to reconstruct an image with almost double resolution. We showed that the aliasing present in the input images was accurately removed using Algorithm 3.1.

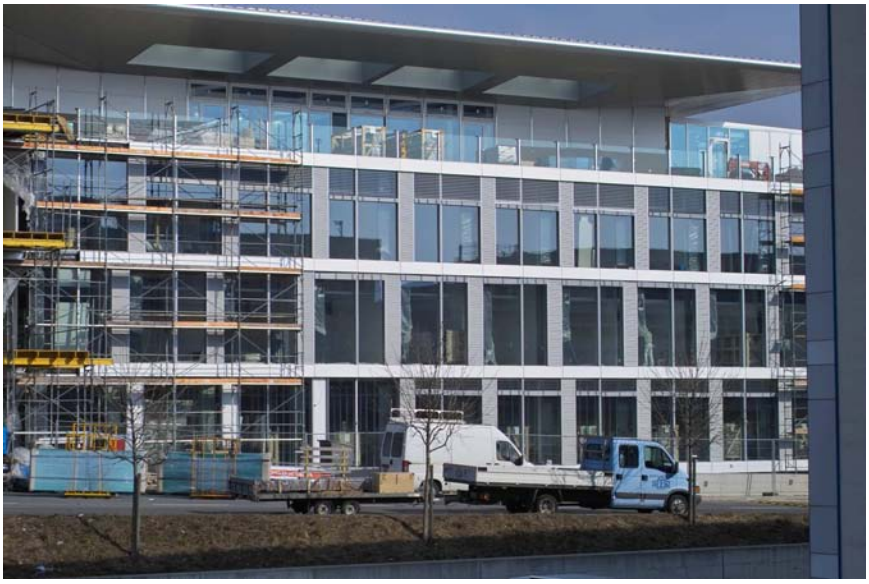
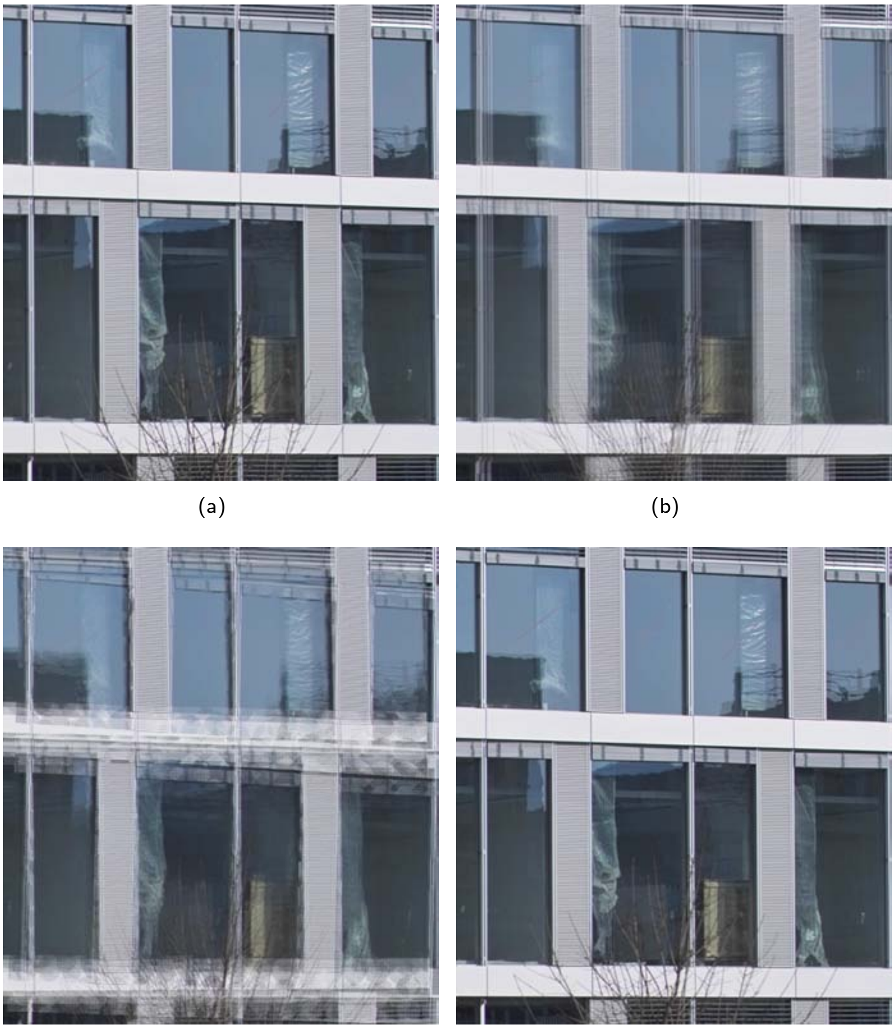


Figure 4.11: One of the four images taken with the Sigma SD10 camera as input to Algorithm 3.1. A detail is shown in Figure 4.13.

Table 4.2: Registration parameters for the four Sigma images of an outdoor scene using Algorithm 3.1 and the algorithms by Marcel et al., Lucchese and Cortelazzo, and Keren et al.

image pairs	Algorithm 3.1			Marcel et al.		
	$x_{m,h}$	$x_{m,v}$	θ_m	$x_{m,h}$	$x_{m,v}$	θ_m
image 2 - image 1	-12.75	-10.34	-0.1	-17	-10	0
image 3 - image 1	14.65	12.96	0	22	13	0
image 4 - image 1	-12.08	1.54	-0.1	-18	2	0

	Lucchese et al.			Keren et al.		
	$x_{m,h}$	$x_{m,v}$	θ_m	$x_{m,h}$	$x_{m,v}$	θ_m
image 2 - image 1	-15	-4	-0.53	-12.51	-10.43	-0.01
image 3 - image 1	11	36	1.66	15.22	13.08	0.01
image 4 - image 1	-14	-6	-0.63	-12.76	1.74	-0.09



(a) (b)

(c) (d)

Figure 4.12: Results of the different super-resolution algorithms on the Sigma SD10 images of the outdoor scene. Details of the central part of the images are displayed in Figure 4.14 - 4.17 to show the differences better. (a) Algorithm 3.1. (b) Registration algorithm by Marcel et al. (c) Registration algorithm by Lucchese and Cortelazzo. (d) Registration algorithm by Keren et al.

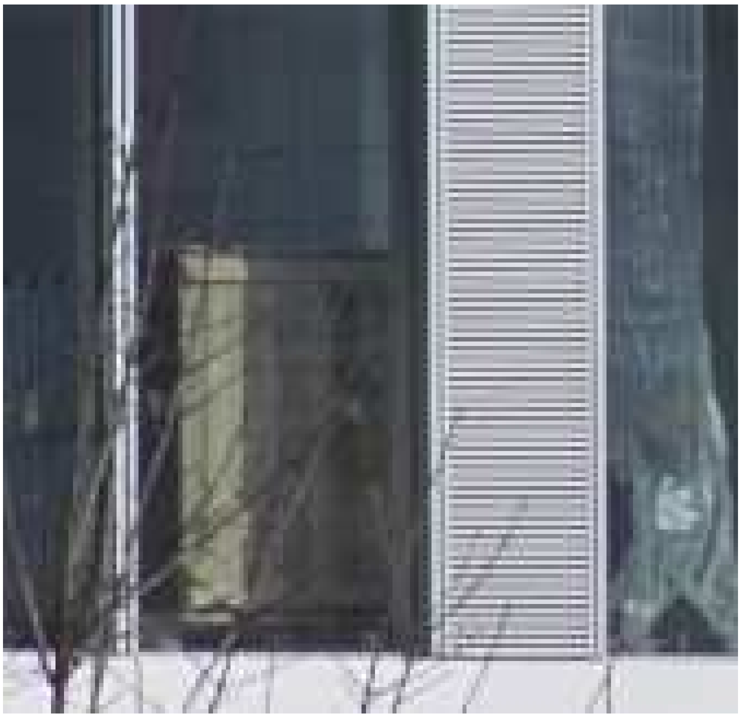


Figure 4.13: Detail of Figure 4.11 where aliasing is well visible.

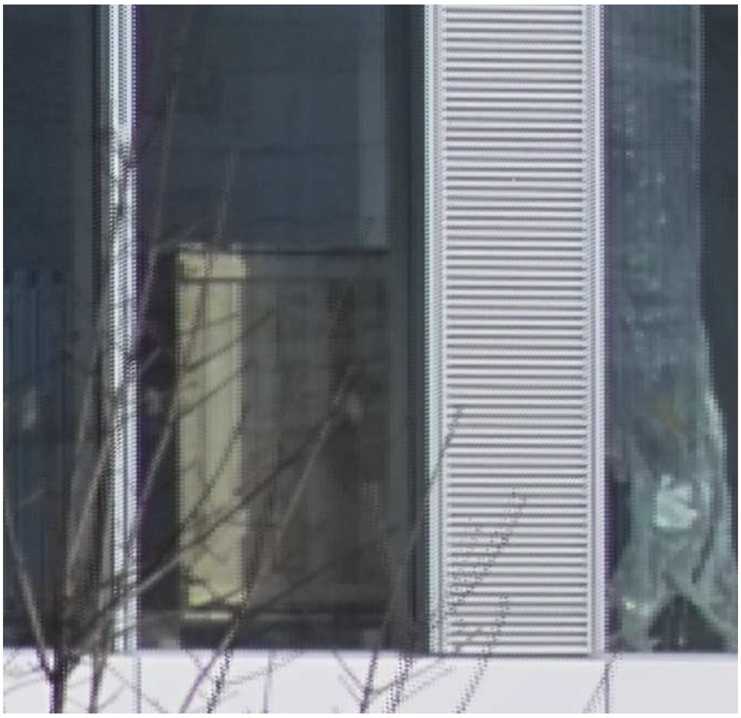


Figure 4.14: Detail of Figure 4.12(a). Algorithm 3.1.

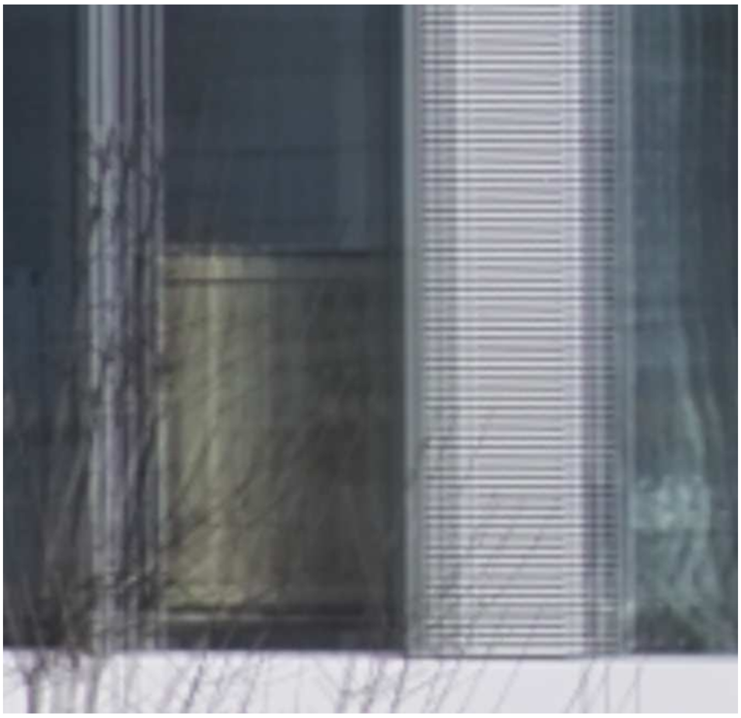


Figure 4.15: Detail of Figure 4.12(b). Algorithm by Marcel et al.

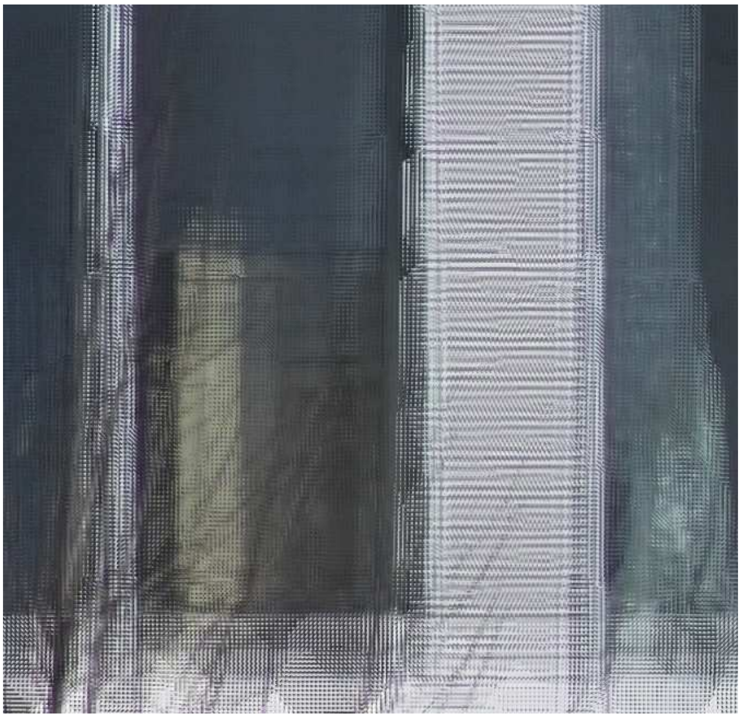


Figure 4.16: Detail of Figure 4.12(c). Algorithm by Lucchese and Cortelazzo.

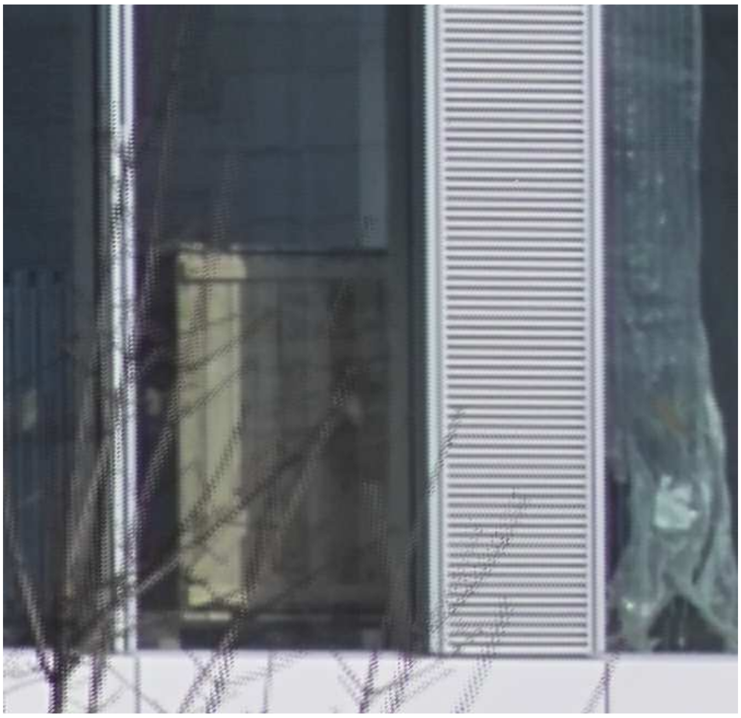


Figure 4.17: Detail of Figure 4.12(d). Algorithm by Keren et al.

Chapter 5

Registration of Totally Aliased Signals using Subspace Methods

If the entire spectrum of the sampled signals is aliased, or $N < K$, a pairwise registration of the signals is not possible anymore. All the signals need to be registered jointly, before a high resolution reconstruction can be made. We need to solve directly the set of nonlinear equations that is described in Chapter 2. In this chapter, we will use a subspace analysis of this setup to reconstruct the high resolution signal [114]. We will first consider one-dimensional signals that have an unknown shift. This approach is then generalized to two-dimensional signals (images), where we still only consider (planar) shifts for the motion.

Section 5.1 presents a method using matrix rank to compute the registration parameters [113]. It is based on an analysis of the Fourier series description of the signals, and is therefore only applicable to bandlimited signals. An approach for signals in an arbitrary Hilbert space is then described in Section 5.2 [116]. It uses projections onto a lower-dimensional Hilbert space to compute the shifts. Some practical issues about these two subspace-based methods are discussed in Section 5.3. The complexity of the different methods is given in Section 5.4, and simulation results are shown in Section 5.5.

5.1 Solution using matrix rank

In this section, we describe a first solution method that uses specific properties of the Fourier transform. It is therefore only applicable to bandlimited functions.

5.1.1 Method

As shown in (2.15), the discrete Fourier transform $\mathbf{Y}_m$ of the m-th set of samples can be written as

$$\mathbf{Y}_m = \frac{1}{N}\mathbf{F}_N\mathbf{F}^*\mathbf{D}_{t_m}\boldsymbol{\alpha}, \tag{5.1}$$

with $\mathbf{F}_N$ a square $N \times N$ DFT matrix, $\mathbf{F}^*$ an $N \times L$ IDFT matrix, and $\mathbf{D}_{t_m}$ an $L \times L$ diagonal matrix with elements $\mathbf{D}_{t_m}(l,l) = z_m^l$ $(-K \leq l \leq K)$. If we extend $\boldsymbol{\alpha}$ to a length that is a multiple of N, we can split the Fourier coefficient vector $\boldsymbol{\alpha}$ in blocks $\boldsymbol{\alpha}_i$ of length N, and obtain (see also (2.16)):

$$\mathbf{Y}_m = \mathbf{D}'_{t_m} \sum_{i=\lceil -(S-1)/2 \rceil}^{\lceil (S-1)/2 \rceil} z_m^{iN} \boldsymbol{\alpha}_i, \tag{5.2}$$

where $\mathbf{D}'_{t_m}$ is the $N \times N$ central part of the $L \times L$ matrix $\mathbf{D}_{t_m}$. The vectors $\boldsymbol{\alpha}_i$ represent the overlapping parts of the Fourier spectrum due to undersampling, and there are $S = \lceil L/N \rceil$ such parts. This means that for any set of samples, $\mathbf{D'}_{t_m}^{-1} \mathbf{Y}_m$ is a linear combination of the S parts of the Fourier spectrum $\boldsymbol{\alpha}_i$:

$$\mathbf{D'}_{t_m}^{-1} \mathbf{Y}_m = \sum_{i=\lceil -(S-1)/2 \rceil}^{\lceil (S-1)/2 \rceil} z_m^{iN} \boldsymbol{\alpha}_i. \tag{5.3}$$

In other words, each vector $\mathbf{D'}_{t_m}^{-1} \mathbf{Y}_m$ belongs to the S-dimensional subspace $\operatorname{span}(\{\boldsymbol{\alpha}_i\}_{i=\lceil -(S-1)/2 \rceil .. \lceil (S-1)/2 \rceil})$. If we have more than S sets of samples ($M > S$), and $N \geq S$, the rank of the matrix containing all the sets of samples should therefore be S:

$$\operatorname{rank}\left(\mathbf{Y}_{\mathbf{t}}^D\right) = S \qquad \text{with } \mathbf{Y}_{\mathbf{t}}^D = \begin{pmatrix} \mathbf{Y}_0 & \mathbf{D'}_{t_1}^{-1} \mathbf{Y}_1 & \cdots & \mathbf{D'}_{t_{M-1}}^{-1} \mathbf{Y}_{M-1} \end{pmatrix}. \tag{5.4}$$

Example 5.1.1. Let us illustrate this for a small example. Assume we have a bandlimited signal with $L = 5$ unknown Fourier coefficients $\boldsymbol{\alpha} = (\alpha_{-2} \ \ \alpha_{-1} \ \ \alpha_0 \ \ \alpha_1 \ \ \alpha_2)^T$. We sample this signal with three sets of four samples $\mathbf{y}_0$, $\mathbf{y}_1$, and $\mathbf{y}_2$ ($M = 3$, $N = 4$). There are $S = \lceil L/N \rceil = 2$ overlapping parts of the spectrum, and the Fourier transforms of the sampled signals can be written as

$$\begin{aligned}
\mathbf{Y}_0 &= \begin{pmatrix} \alpha_{-2} + \alpha_2 \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \alpha_{-2} \\ \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} + \begin{pmatrix} \alpha_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1, \\
\mathbf{Y}_1 &= \begin{pmatrix} z_1^{-2}\alpha_{-2} + z_1^2\alpha_2 \\ z_1^{-1}\alpha_{-1} \\ \alpha_0 \\ z_1\alpha_1 \end{pmatrix} = \begin{pmatrix} z_1^{-2}\alpha_{-2} \\ z_1^{-1}\alpha_{-1} \\ \alpha_0 \\ z_1\alpha_1 \end{pmatrix} + z_1^4 \begin{pmatrix} z_1^{-2}\alpha_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{D}'_{t_1}\boldsymbol{\alpha}_0 + z_1^4 \mathbf{D}'_{t_1}\boldsymbol{\alpha}_{\mathbf{1}} \\
\mathbf{Y}_2 &= \begin{pmatrix} z_2^{-2}\alpha_{-2} + z_2^2\alpha_2 \\ z_2^{-1}\alpha_{-1} \\ \alpha_0 \\ z_2\alpha_1 \end{pmatrix} = \begin{pmatrix} z_2^{-2}\alpha_{-2} \\ z_2^{-1}\alpha_{-1} \\ \alpha_0 \\ z_2\alpha_1 \end{pmatrix} + z_2^4 \begin{pmatrix} z_2^{-2}\alpha_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{D}'_{t_2}\boldsymbol{\alpha}_0 + z_2^4 \mathbf{D}'_{t_2}\boldsymbol{\alpha}_{\mathbf{1}}.
\end{aligned} \tag{5.5}$$

We can then multiply each of these Fourier transform vectors with their corresponding inverse diagonal matrix $\mathbf{D'}_{t_m}^{-1}$, and combine them into the matrix

$\mathbf{Y}^D_{\mathbf{t}}$ as

$$\mathbf{Y}^D_{\mathbf{t}} = \begin{pmatrix} \alpha_{-2} + \alpha_2 & \alpha_{-2} + z_1^4 \alpha_2 & \alpha_{-2} + z_2^4 \alpha_2 \\ \alpha_{-1} & \alpha_{-1} & \alpha_{-1} \\ \alpha_0 & \alpha_0 & \alpha_0 \\ \alpha_1 & \alpha_1 & \alpha_1 \end{pmatrix} \tag{5.6}$$

$$= \begin{pmatrix} \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_0 + z_1^4 \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_0 + z_2^4 \boldsymbol{\alpha}_1 \end{pmatrix},$$

which is of rank 2, as all columns are linear combinations of the same coefficient vectors $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_1$. □

In general, if the estimated offsets $\hat{\mathbf{t}}$ do not have the correct values, $\operatorname{rank}(\mathbf{Y}^D_{\hat{\mathbf{t}}}) > S$. For most offset values $\hat{\mathbf{t}}$, the matrix $\mathbf{Y}^D_{\hat{\mathbf{t}}}$ will be of full rank M (and $M > S$). The correct values of the offsets $\mathbf{t}$ can then be found as the values for which the matrix rank becomes S. A schematic description of the complete reconstruction algorithm built on this method is given in Figure 5.1.

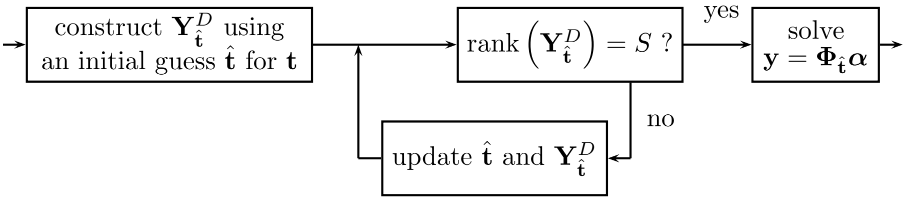

construct $\mathbf{Y}^D_{\hat{\mathbf{t}}}$ using an initial guess $\hat{\mathbf{t}}$ for $\mathbf{t}$
rank $\left(\mathbf{Y}^D_{\hat{\mathbf{t}}}\right) = S$?
yes
solve $\mathbf{y} = \boldsymbol{\Phi}_{\hat{\mathbf{t}}} \boldsymbol{\alpha}$
no
update $\hat{\mathbf{t}}$ and $\mathbf{Y}^D_{\hat{\mathbf{t}}}$

Figure 5.1: Signal reconstruction algorithm using the matrix rank method from Section 5.1. The estimate for the offsets $\hat{\mathbf{t}}$ and the corresponding sample matrix $\mathbf{Y}^D_{\hat{\mathbf{t}}}$ are updated iteratively. Once the estimate is good enough, the signal parameters $\boldsymbol{\alpha}$ are reconstructed.

5.1.2 Discussion

As there are S overlapping parts in the sampled Fourier spectrum, we need at least $M = S + 1 = \lceil L/N \rceil + 1$ sets of samples to find the offsets using this method. If more sets of samples are available, we still have $\operatorname{rank}\left(\mathbf{Y}^D_{\mathbf{t}}\right) = S$, and this adds some robustness to both the offset estimation and the signal reconstruction. However, each new set of samples also adds a new unknown offset, thereby increasing the complexity of the estimation.

Remark that the operation performed by multiplying a sample vector $\mathbf{Y}_m$ with a matrix $\mathbf{D}'^{-1}_{t_m}$ does not change its norm $\|\mathbf{Y}_m\|_2^2$. It merely performs a rotation to align the sample vector into the S-dimensional subspace.

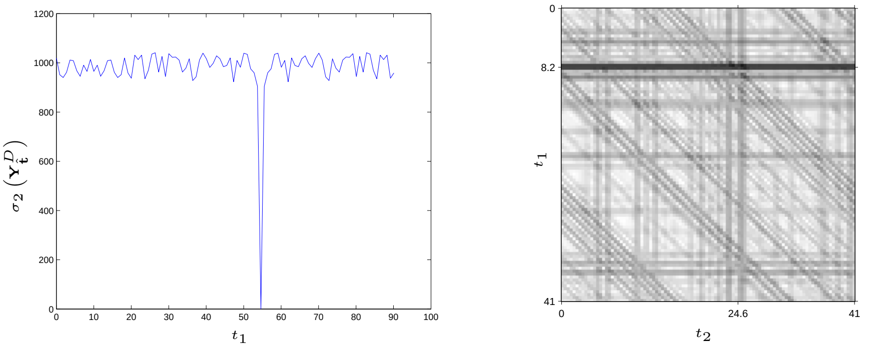
The computation of the rank of a matrix has a rather 'binary' outcome: either it is S, or it has an integer value larger than S. Hence, if the measurements are noisy, this test is very likely to fail even for the correct values of $\mathbf{t}$. It is therefore much better to evaluate the $S + 1$-th singular value of the matrix $\mathbf{Y}^D_{\hat{\mathbf{t}}}$, or the determinant of the square matrix $\mathbf{Y}^{D*}_{\hat{\mathbf{t}}} \mathbf{Y}^D_{\hat{\mathbf{t}}}$. This can also give an indication about the quality of the current approximation. While the determinant requires less computations, it is also numerically less stable than a singular value decomposition. The $S+1$-th singular value could be computed using the inverse power method described by Strang [101]. This is an iterative method to approximate the smallest singular value of a matrix. It requires the solution of a linear $M \times M$ system of equations in each iteration. The number of iterations needed will generally be very small, because the $S+1$-th singular value is typically much smaller than the first S singular values.

The objective function based on (5.4) can thus be written as

$$\min_{\hat{\mathbf{t}}} \sigma_{S+1}\left(\mathbf{Y}^{D}_{\hat{\mathbf{t}}}\right), \tag{5.7}$$

where the operation $\sigma_{S+1}(\mathbf{Y}^{D}_{\hat{\mathbf{t}}})$ stands for computing the $S+1$-th singular value of the matrix $\mathbf{Y}^{D}_{\hat{\mathbf{t}}}$.

Example 5.1.2 (Bandlimited functions). Let us consider a bandlimited function with $L = 81$ unknown Fourier coefficients. It is sampled with two sets of 91 samples, with offsets $\mathbf{t} = \begin{pmatrix} 0 & 54.6 \end{pmatrix}$. The objective function from (5.7) is shown as a function of t_1 in Figure 5.2(a). Similarly, we sample the same function with $M = 3$ sets of $N = 41$ samples and offsets $\mathbf{t} = \begin{pmatrix} 0 & 8.2 & 24.6 \end{pmatrix}$. The objective function is shown in Figure 5.2(b) as a function of the offsets t_1 and t_2. Smaller values are represented by darker graylevels. ☐



(a) (b)

Figure 5.2: Examples of the objective function in (5.7). Next to the global minimum, it also contains many local minima. (a) Two sets of 91 samples, with 81 unknown coefficients. The exact offset is $t_1 = 54.6$. (b) Three sets of 41 samples, with 81 unknown coefficients. The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels.

In the above examples, the objective function (5.7) is very 'flat', except for a small region around the optimal offset values. At an arbitrary value away from the correct solution, it is therefore not possible to predict what would be a better approximation. Standard minimization algorithms such as gradient descent will generally not converge to the right solution for such functions. However, from the example with three sets of samples, we can see that the objective function is not completely arbitrary. Horizontal, vertical and diagonal lines appear. They correspond to relative alignments of two out of the three sets of samples. It is therefore interesting to use this information in the search for the minimum. Such a heuristic algorithm will be described in Section 5.3.

5.2 Solution using projections

In this section, we describe a method that is applicable to any kind of signals in a finite-dimensional Hilbert space defined in Section 2.1. It uses projections onto a subspace basis to find the relative offsets. This method shows some similarities to the separable nonlinear least squares methods [40, 92], which consider a nonlinear least squares problem where the unknowns can be split in two sets of variables. One of the sets is then eliminated first (typically the linear variables), and a nonlinear minimization over a smaller set of variables is performed.

5.2.1 Method

From equation (2.7), we can see that $\mathbf{y}$ belongs to span($\mathbf{\Phi_t}$). However, this is only true if the matrix $\mathbf{\Phi_t}$ is constructed using the correct values for the offset vector $\mathbf{t}$ (except for degenerate cases). For another (incorrect) set of offset values $\hat{\mathbf{t}}$, (2.7) no longer holds and the sample vector $\mathbf{y}$ is not in span($\mathbf{\Phi}_{\hat{\mathbf{t}}}$) anymore. Using Lemma 2.4.1, one can test the correctness of the offset vector by verifying if the projection $\hat{\mathbf{y}}$ of the sample vector $\mathbf{y}$ onto span($\mathbf{\Phi}_{\hat{\mathbf{t}}}$) gives the sample vector $\mathbf{y}$ again. Or mathematically:

$$\begin{cases} \hat{\mathbf{y}} &= \mathbf{P}_{\mathbf{\Phi}_{\hat{\mathbf{t}}}}\mathbf{y} = \mathbf{\Phi}_{\hat{\mathbf{t}}}(\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{\Phi}_{\hat{\mathbf{t}}})^{-1}\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{y} = \mathbf{y}, \quad \text{for } \hat{\mathbf{t}} = \mathbf{t} \\ \hat{\mathbf{y}} &= \mathbf{P}_{\mathbf{\Phi}_{\hat{\mathbf{t}}}}\mathbf{y} \neq \mathbf{y}, \quad \text{for } \hat{\mathbf{t}} \neq \mathbf{t}. \end{cases} \tag{5.8}$$

Therefore, this can be used to build an optimization problem for finding the correct offsets $\mathbf{t}$. We search for the value of $\hat{\mathbf{t}}$ such that the following function is minimized:

$$\min_{\hat{\mathbf{t}}} \|\mathbf{y} - \hat{\mathbf{y}}\|^2. \tag{5.9}$$

Using this method to find the offsets between the sets of samples $\mathbf{y}_m$, we can derive an algorithm to reconstruct the signal $f(t)$ from its combined sets of samples $\mathbf{y}$. A block diagram of such an algorithm is given in Figure 5.3.

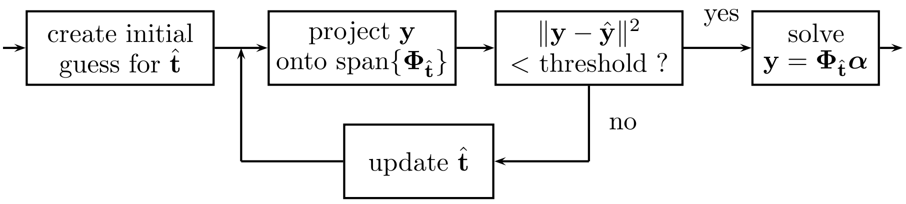

create initial guess for $\hat{\mathbf{t}}$
project $\mathbf{y}$ onto span$\{\mathbf{\Phi}_{\hat{\mathbf{t}}}\}$
$\|\mathbf{y} - \hat{\mathbf{y}}\|^2$ < threshold ?
yes
solve $\mathbf{y} = \mathbf{\Phi}_{\hat{\mathbf{t}}}\boldsymbol{\alpha}$
no
update $\hat{\mathbf{t}}$

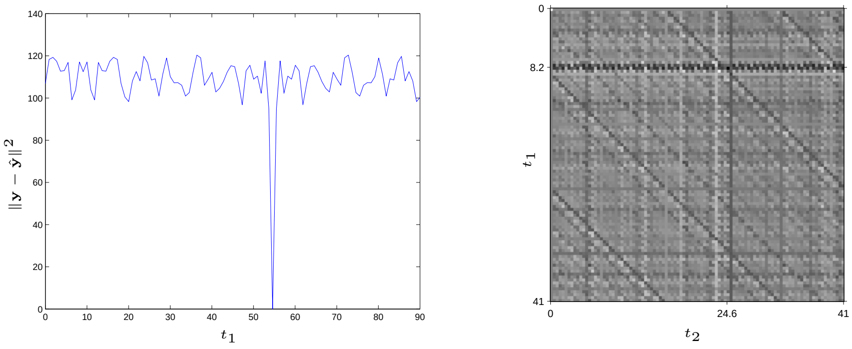
Figure 5.3: Signal reconstruction algorithm using the projection method from Section 5.2. The estimate for the offsets $\hat{\mathbf{t}}$ is updated iteratively. Once the estimate is considered good enough, the signal parameters $\boldsymbol{\alpha}$ are reconstructed.

5.2.2 Discussion

Let us study some examples of the objective function $\|\mathbf{y}-\hat{\mathbf{y}}\|^2$ (see (5.9)) as a function of the offset values $\hat{\mathbf{t}}$.

Example 5.2.1 (Bandlimited functions). A bandlimited function with 81 unknown Fourier coefficients is sampled with two sets of 91 samples. The offset vector that we used is $\mathbf{t} = \begin{pmatrix} 0 & 54.6 \end{pmatrix}$. The objective function $\|\mathbf{y}-\hat{\mathbf{y}}\|^2$ from (5.9) is shown as a function of the offset t_1 in Figure 5.4(a). It is 0 at the correct offset value ($t_1 = 54.6$), and is larger at other values of $\hat{t}_1$. Next to this global minimum, the function also has many local minima.

Similarly, we also take three sets of 41 samples from a bandlimited function with again 81 Fourier coefficients. The offset vector for this example is $\mathbf{t} = \begin{pmatrix} 0 & 8.2 & 24.6 \end{pmatrix}$. Figure 5.4(b) shows the objective function as a function of the offsets t_1 and t_2. Smaller values are represented by darker graylevels. Again, the minimum can clearly be seen, and is at the intersection of the dark horizontal, vertical and diagonal lines corresponding to pairwise alignments between the sets of samples. However, the objective function also has many local minima. □

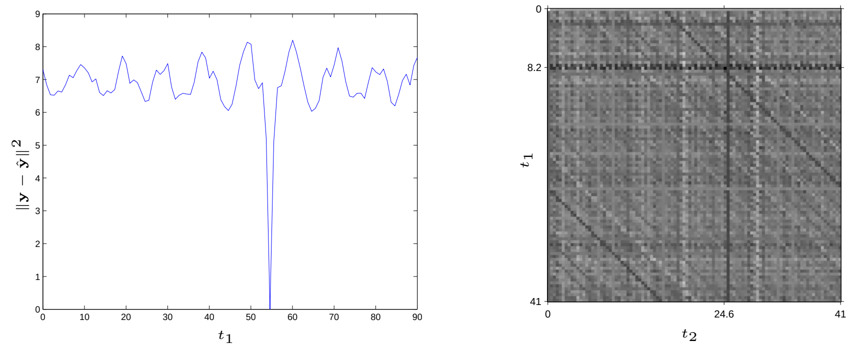


(a) (b)

Figure 5.4: Examples of the objective function in (5.9) where $\mathcal{H}$ is the Fourier space. Next to the global minimum, it also contains many local minima. (a) Two sets of 91 samples, with 81 unknown coefficients. The exact offset is $t_1 = 54.6$. (b) Three sets of 41 samples, with 81 unknown coefficients. The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels.

Example 5.2.2 (piecewise linear functions). A similar example is repeated for piecewise linear functions, as described in Example 2.1.3. The same setup as in Example 5.2.1 is used, with first $M = 2$, $N = 91$ and $L = 81$ (Figure 5.5(a)), and next $M = 3$, $N = 41$, and $L = 81$ (Figure 5.5(b)). The results are similar to those obtained in the bandlimited case. They also have many local minima, and are zero only for the correct offsets $\hat{\mathbf{t}} = \mathbf{t}$. □

From these examples, it is clearly visible that (5.9) is not easy to minimize (just like in Section 5.1). Next to the global minimum, the objective function has many local minima. We cannot use a standard algorithm like gradient



(a)

(b)

Figure 5.5: Examples of the objective function in (5.9) on a piecewise linear function. Next to the global minimum, it also contains many local minima. (a) Two sets of 91 samples, with 81 unknown coefficients. The exact offset is $t_1 = 54.6$. (b) Three sets of 41 samples, with 81 unknown coefficients. The exact offsets are $t_1 = 8.2$ and $t_2 = 24.6$. Small values are represented by dark pixels.

descent for minimization. However, as can also be seen from Figure 5.4(b) and Figure 5.5(b), the objective function shows the same structure of horizontal, vertical and diagonal lines as in Figure 5.2(b).

Let us now consider specifically bandlimited functions, with the Fourier basis, to give some further interpretation of the above results. As it was discussed in Section 2.4, the MN-dimensional sample space can be divided into N orthogonal subspaces of dimension M. It is therefore possible to split the function from (5.9) into independent terms according to these M-dimensional subspaces:

$$\|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \sum_{n=0}^{N-1} \|\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\|^2. \tag{5.10}$$

In this equation, $\mathbf{y}^{(n)}$ is the projection of $\mathbf{y}$ onto the subspace $\mathcal{V}_n$ spanned by the n-th vector for different values of $\hat{\mathbf{t}}$, and $\hat{\mathbf{y}}^{(n)}$ is the projection of $\mathbf{y}$ onto the vectors $\boldsymbol{\phi}_{\hat{\mathbf{t}}}^{n+iN}$ that belong to this space. An example of such a decomposition is given in Figure 5.6.

As we also described in Section 2.4, for the frequencies n that have M overlapping spectrum coefficients, the spaces $\mathcal{V}_n$ and $\mathrm{span}(\boldsymbol{\phi}_{\hat{\mathbf{t}}}^{n+iN})$ are the same, regardless of the value of $\hat{\mathbf{t}}$. For these frequencies, $\|\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\|^2 = 0$, and they do not contribute to the objective function in (5.10). The other terms each contribute a periodic term to the global objective function. The minimum of (5.10) can therefore be found by minimizing the different components individually (see Figure 5.6). For each of the components, we can now minimize over an M-dimensional subspace instead of the original MN-dimensional space.

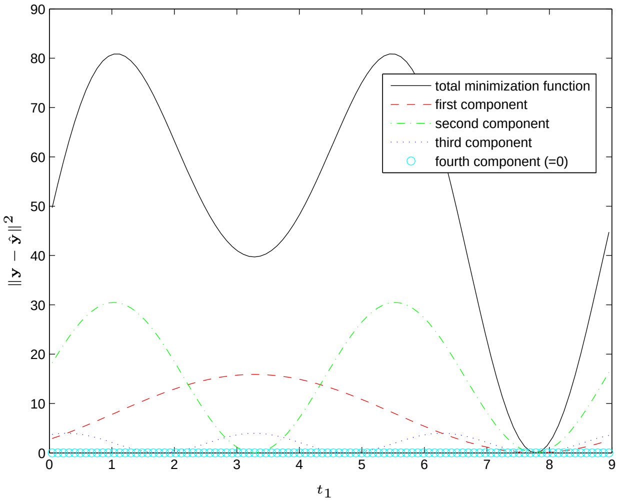


Figure 5.6: Example of the decomposition of the objective function into its different components belonging to orthogonal M-dimensional subspaces.

5.3 Practical Issues

5.3.1 Images and higher-dimensional signals

The above approaches were described for one-dimensional functions. However, all the methods can be extended to higher-dimensional functions. Nothing in the descriptions is limited to a one-dimensional signal, so the algorithms can be directly applied to images.

Assume we have a periodic, bandlimited image

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{k_1,k_2=-K}^{K} \alpha_{\boldsymbol{k}} e^{j2\pi\boldsymbol{k}^T\boldsymbol{x}} \quad \text{with } \boldsymbol{k} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix},\ \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{5.11}$$

It is sampled with M images at horizontal and vertical frequencies N_1 and N_2:

$$\begin{aligned} \boldsymbol{y}_m(\boldsymbol{n}) &= \boldsymbol{f}\left(\frac{n_1}{N_1}+t_{m,1}, \frac{n_2}{N_2}+t_{m,2}\right) \\ &= \sum_{k_1,k_2=-K}^{K} \alpha_{\boldsymbol{k}} e^{j2\pi\left(k_1\left(\frac{n_1}{N_1}+t_{m,1}\right)+k_2\left(\frac{n_2}{N_2}+t_{m,2}\right)\right)}. \\ &= \sum_{k_1,k_2=-K}^{K} \alpha_{\boldsymbol{k}} e^{j2\pi(k_1 t_{m,1}+k_2 t_{m,2})} e^{j2\pi\left(k_1\frac{n_1}{N_1}+k_2\frac{n_2}{N_2}\right)}. \end{aligned} \tag{5.12}$$

The discrete Fourier transform (DFT) of such an image can be expressed as

$$\begin{aligned}
\boldsymbol{Y}_m(\boldsymbol{l}) &= \sum_{n_1,n_2=0}^{\boldsymbol{N}-1} \boldsymbol{y}_m(\boldsymbol{n}) e^{-j2\pi\left(l_1 \frac{n_1}{N_1} + l_2 \frac{n_2}{N_2}\right)} \\
&= \sum_{n_1,n_2=0}^{\boldsymbol{N}-1} \sum_{k_1,k_2=-K}^{K} \alpha_{\boldsymbol{k}} e^{j2\pi(k_1 t_{m,1} + k_2 t_{m,2})} e^{j2\pi\left(k_1 \frac{n_1}{N_1} + k_2 \frac{n_2}{N_2}\right)} e^{-j2\pi\left(l_1 \frac{n_1}{N_1} + l_2 \frac{n_2}{N_2}\right)} \\
&= \sum_{k_1,k_2=-K}^{K} \alpha_{\boldsymbol{k}} e^{j2\pi(k_1 t_{m,1} + k_2 t_{m,2})} \sum_{n_1,n_2=0}^{\boldsymbol{N}-1} e^{j2\pi\left((k_1 - l_1)\frac{n_1}{N_1} + (k_2 - l_2)\frac{n_2}{N_2}\right)} \\
&= \sum_{i_1,i_2} \alpha_{l_1 + i_1 N_1, l_2 + i_2 N_2} e^{j2\pi((l_1 + i_1 N_1) t_{m,1} + (l_2 + i_2 N_2) t_{m,2})}.
\end{aligned} \tag{5.13}$$

In other words, $\boldsymbol{Y}_m(\boldsymbol{l})$ is a weighted sum of the overlapping Fourier coefficients at frequency $\boldsymbol{l}$. Combining all the DFT coefficients $\boldsymbol{Y}_m(\boldsymbol{l})$ into a vector $\mathbf{Y}_m$, we can write (see also (2.16)):

$$\mathbf{Y}_m = \sum_{i_1,i_2} e^{j2\pi(i_1 N_1 t_{m,1} + i_2 N_2 t_{m,2})} \mathbf{D}'_{t_m} \boldsymbol{\alpha}_{i_1,i_2}, \tag{5.14}$$

with now $\mathbf{Y}_m(l_1 + N_1 l_2) = \boldsymbol{Y}_m(l_1, l_2)$, $\mathbf{D}'_{t_m}(l_1 + Nl_2, l_1 + Nl_2) = e^{j2\pi(l_1 t_{m,1} + l_2 t_{m,2})}$, and $\boldsymbol{\alpha}_{i_1,i_2}(k_1, k_2) = \boldsymbol{\alpha}(k_1 + i_1 N_1, k_2 + i_2 N_2)$. The analysis from Section 5.1 is therefore still valid for 2D signals.

Similarly, we can combine all the samples of an image into a vector $\mathbf{y}_m$ (with $\mathbf{y}_m(n_1 + N_1 n_2) = \boldsymbol{y}_m(n_1, n_2)$):

$$\mathbf{y}_m = \mathbf{\Phi}_{t_m} \boldsymbol{\alpha}, \tag{5.15}$$

with $\mathbf{\Phi}_{t_m}$ an $N_1 N_2 \times L_1 L_2$ sampled basis matrix with elements $\mathbf{\Phi}_{t_m}(n_1 + N_1 n_2, l_1 + L_1 l_2) = e^{j2\pi(n_1 l_1/N_1 + n_2 l_2/N_2)}$. We can then repeat the analysis from Section 5.2 using this equation instead of (2.15).

Of course, the complexity increases fast with the dimensionality. While a minimum of $\lceil L/N \rceil + 1$ sets of samples are required for a 1D signal using the rank-based method, and $\lceil (L-1)/(N-1) \rceil$ sets of samples for the projection-based method, we need at least $\lceil L/N \rceil^2 + 1$ two-dimensional images for the rank-based method, and $\lceil (L^2 - 2)/(N^2 - 2) \rceil$ images for the projection-based method. As signals can be shifted along each of their dimensions, the number of offsets (which is also the dimensionality of the search space) also increases rapidly. Examples of the above methods on images are shown in Section 5.5.

5.3.2 Minimization

The main difficulty with the methods described in Sections 5.1 and 5.2 is to find the global minimum of the objective functions. A good approximation of the offsets $\mathbf{t}$ is required for an approach like gradient descent to converge to the global minimum. Such an approximation could be obtained by evaluating the error functions from (5.7) or (5.9) on a uniform (dense) grid of possible values $\hat{\mathbf{t}}$. The global minimum can then be found with high probability close to the smallest value obtained on this uniform grid. However, this is a computationally very intensive method. It only makes sense if the number of sets of samples M - and therefore also the number of offsets - is small, or when a first estimate of the offsets is available. Although this may sound like moving the problem to obtaining such a first estimate, it is a reasonable assumption. For example in super-resolution imaging, the shifts between the images are generally very small. Therefore, our search could be restricted to a small area around the origin (for example 10×10 pixels). Similarly, a first estimate could also be obtained from another registration method that does not use the aliasing. Such a method is described in the next subsection. Other approaches are heuristic methods that rely on the lines that can be seen in Figure 5.2(b), 5.4(b) and 5.5(b). They will also be described in the next subsection. Simulation results using these methods will be shown in Section 5.5.

During the minimization, it would be interesting to know whether the global minimum has been found, or whether the current optimum is a local minimum. Such an indication can be obtained by evaluating the objective functions (5.7) or (5.9) for the current value of $\hat{\mathbf{t}}$. Even for noisy measurements, there is a large difference between the average values of these functions and their value at the global minimum. This can therefore be used to check whether the algorithm has converged, and whether the result is reliable.

5.3.3 Heuristic approaches

The high computational complexity of the algorithms from Sections 5.1 and 5.2 is mainly due to the coupling between the offsets, i.e. the need to search the different offsets between the signals jointly. The joint minimum is not necessarily located at the intersection of the minima from individual optimizations.

Algorithm 5.3.1 (Hierarchical approach). If the sets of samples are images, we know that their coefficients are not arbitrary Gaussian random variables. In general, the amplitude of the Fourier transform of a natural image decays like $1/f$ [105]. We can therefore assume that a good estimate for the offsets can be obtained from the low frequencies of the sampled sets. For these frequencies, the aliased coefficients are much smaller than the base spectrum coefficients, and can be neglected in a first estimate. We used the method from Chapter 3 for this. Once we have such an initial estimate, we can use the methods from Section 5.1 or 5.2 using a gradient descent algorithm to obtain a more precise estimate for the motion parameters, taking the aliasing into account. □

Algorithm 5.3.2 (Search for lines in the objective function). From the horizontal, vertical and diagonal lines that are often visible in figures like Figure 5.2(b), 5.4(b) and 5.5(b), it can be seen that an independent pairwise alignment often works. Such lines correspond to pairwise registrations of the first and second, first and third, and second and third signals, respectively. A heuristic algorithm has therefore been developed to search for these pairwise alignments without evaluating the complete M-dimensional grid. A horizontal line for example is searched by evaluating the objective function on five different vertical lines, and searching the minimum of the average values. The estimated offset can then be found at the intersection of the different lines. Even if, in a setup like the one shown in Figure 5.4(b), only one line can easily be found, this still results in a considerable reduction of the complexity. As it can also be seen from the results in Section 5.5, this method performs very well in a setup with low noise values, but degrades rapidly when more noise is added. □

Algorithm 5.3.3 (Keeping the P best pairwise alignments). Another heuristic approach based on this independence, is to search a fixed number P of local minima for the pairwise registration between the first set of samples and each of the other sets. When performing such a pairwise registration between the first and the m-th set of samples, all the other offset values are kept constant. For each pair, a vector of P possible offset values $\begin{pmatrix} \hat{t}_{m_1} & \hat{t}_{m_2} & \cdots & \hat{t}_{m_P} \end{pmatrix}$ is obtained. Next, the best combination $\hat{\mathbf{t}}$ of these pairwise local minima is searched among all possible combinations. The global minimum can then be searched in the neighbourhood of this value of $\hat{\mathbf{t}}$. Keeping only a single minimum for each of the pairwise alignments ($P = 1$) is generally not sufficient, due to the different approximations made by pairwise registration. However, by trying all the combinations of the best $P = 5$ pairwise alignments, the algorithm typically converges to the correct result. □

5.4 Complexity

In this section, we discuss the computational complexity of the different methods presented in this chapter. For each of the methods described above, the complexity of computing the offsets $\mathbf{t}$ can be written as the number of times the objective function has to be evaluated multiplied with the number of operations required to evaluate the objective function. In this analysis, we will assume that the variables N and L grow at the same rate, and M grows at a lower rate. The number of operations required for the reconstruction is the same for all the algorithms.

Let us start by analyzing the complexity of the error function evaluation both for the matrix rank method (Section 5.1) and the projection method (Section 5.2). In the matrix rank method, the smallest singular value of the matrix $\mathbf{Y}_{\hat{\mathbf{t}}}^D$ has to be computed for every error evaluation. The construction of $\mathbf{Y}_{\hat{\mathbf{t}}}^D$ for a specific set of offsets $\hat{\mathbf{t}}$ requires $O((M-1)N)$ operations. To compute the smallest singular value, we use the inverse power method on $\mathbf{Y}_{\hat{\mathbf{t}}}^{D*}\mathbf{Y}_{\hat{\mathbf{t}}}^D$ [101]. This matrix multiplication involves $O(M^2N)$ operations, and the inverse power method itself requires $O(M^3 + aM^2)$ operations, where a is the number of iterations required for the method to converge. As we assume that the difference with the second smallest singular value is large, convergence will be fast. So the total number of operations becomes

$$C_{\text{rank}} = O((M-1)N + M^2N + M^3 + aM^2) = O(M^3 + M^2N). \tag{5.16}$$

For the projection method, we need to compute the projection of the sample vector $\mathbf{y}$ onto the space spanned by the columns of $\boldsymbol{\Phi}_{\hat{\mathbf{t}}}$. This can be done using (5.8). In the general case, for an arbitrary basis, such a projection has complexity

$$C_{\text{proj}} = O(MNL^2 + L^3). \tag{5.17}$$

However, if enough storage space is available, these projection matrices can be precomputed, because they do not depend on the actual signal. In that case, the complexity is reduced to

$$C'_{\text{proj}} = O(M^2N^2). \tag{5.18}$$

For specific types of bases, the complexity can be strongly reduced.

Example 5.4.1 (Fourier basis). For the Fourier basis, the blocks $\mathbf{\Phi}_{\hat{t}_m}$ of the matrix $\mathbf{\Phi}_{\hat{\mathbf{t}}}$ can be divided into an $N \times L$ IDFT inverse Fourier transform block $\mathbf{F}^*$, multiplied by an $L \times L$ diagonal offset matrix $\mathbf{D}_{t_m}$: $\mathbf{\Phi}_{\hat{t}_m} = \mathbf{F}^*\mathbf{D}_{t_m}$ (see also (2.14)). Using this decomposition, we can simplify the projection formula

$$\hat{\mathbf{y}} = \mathbf{P}_{\mathbf{\Phi}_{\hat{\mathbf{t}}}}\mathbf{y} = \mathbf{\Phi}_{\hat{\mathbf{t}}}(\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{\Phi}_{\hat{\mathbf{t}}})^{-1}\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{y}. \tag{5.19}$$

The multiplication $\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{y}$ can be decomposed into M multiplications $\mathbf{D}_{\hat{t}_m}^*\mathbf{F}\mathbf{y}$, where $\mathbf{F}\mathbf{y}$ is the Fourier transform of $\mathbf{y}$. This Fourier transform can be computed in advance, because it is independent of the offsets $\mathbf{t}$. Only the multiplications with the diagonal matrices $\mathbf{D}_{\hat{t}_m}$ remain, which require ML operations. The matrix $\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{\Phi}_{\hat{\mathbf{t}}}$ has nonzero elements only on the main diagonal and on the iN-th diagonals, due to the orthogonality of the Fourier vectors. Its inverse, $(\mathbf{\Phi}_{\hat{\mathbf{t}}}^*\mathbf{\Phi}_{\hat{\mathbf{t}}})^{-1}$, can therefore be computed efficiently in $O(L^3/N^2) = O(S^2L)$ operations. Finally, the multiplication with the first matrix $\mathbf{\Phi}_{\hat{\mathbf{t}}}$ can be decomposed into M multiplications with diagonal matrices, and M inverse DFTs. As the error can as well be computed in the Fourier domain (using Parseval's theorem), we only need to multiply with the diagonal matrices, which requires again ML operations. The overall complexity can then be approximated as

$$C_{\text{proj},F} = O(ML + S^2L + ML) = O(S^2L), \tag{5.20}$$

where S is the maximum number of overlapping spectral components, which is typically of the same order as M. ☐

For the standard algorithm described in Section 5.3.2, the error function needs to be evaluated first on a uniform grid. Assuming that no estimate of the offsets is available, this requires N^{M-1} error evaluations. Next, a standard minimization algorithm is applied near the minimum value that was obtained from the first part. Let us call the number of error evaluations in this part Q (typical values for the examples given above are around $Q = 30$). This is negligible compared to the number of evaluations on the uniform grid, and the total number of evaluations with this standard algorithm can therefore be approximated as

$$O(N^{M-1} + Q) = O(N^{M-1}). \tag{5.21}$$

For the hierarchical method from Algorithm 5.3.1, we need only Q evaluations. Of course, the complexity C_{init} for computing the initial estimate needs to be added to this.

With the heuristic line search of Algorithm 5.3.2, the number of function evaluations is difficult to predict, as multiple functions are executed consecutively to find the minimum along horizontal, vertical, or diagonal lines. When the value of the error function is below a certain threshold, the algorithm is stopped. The maximum number of evaluations is

$$O(21N + (27 + 7^{M-1} + N)Q) = O((7^{M-1} + N)Q). \tag{5.22}$$

Algorithm 5.3.3, using pairwise alignments, requires

$$O((M-1)N + 5^{M-1} + Q) = O(MN + 5^{M-1}) \qquad (5.23)$$

error function evaluations. The complexity of the different algorithms is summarized in Table 5.1.

Table 5.1: Computational complexity for the different methods on 1D signals. The evaluation on a uniform grid from Section 5.3.2 is compared to the hierarchical method (Algorithm 5.3.1), the line search method (Algorithm 5.3.2), and the pairwise alignment method (Algorithm 5.3.3). The total complexity is obtained by multiplying the number of function evaluations with the complexity of a single function evaluation. A worst case scenario was used for Alg. 5.3.2. The cost to obtain the initial estimate in Alg. 5.3.1 is denoted as C_{init}.

	matrix rank method M^3+M^2N	**projection method** MNL^2+L^3
uniform grid N^{M-1} eval.	$N^{M-1}(M^3+M^2N)$	$N^{M-1}(MNL^2+L^3)$
hier. method Q eval. + C_{init}	$(M^3+M^2N)Q+C_{\text{init}}$	$(MNL^2+L^3)Q+C_{\text{init}}$
line search $(7^{M-1}+N)Q$ eval.	$(7^{M-1}+N)(M^3+M^2N)Q$	$(7^{M-1}+N)(MNL^2+L^3)Q$
pairw. al. $MN+5^{M-1}$ eval.	$(MN+5^{M-1})(M^3+M^2N)$	$(MN+5^{M-1})(MNL^2+L^3)$

	proj. method (Fourier) S^2L
uniform grid N^{M-1} eval.	$N^{M-1}S^2L$
hier. method Q eval. + C_{init}	S^2LQ+C_{init}
line search $(7^{M-1}+N)Q$ eval.	$(7^{M-1}+N)S^2LQ$
pairw. al. $MN+5^{M-1}$ eval.	$(MN+5^{M-1})S^2L$

For the reconstruction, the set of linear equations from (2.7) has to be solved. The complexity of this operation is $O(MNL^2 + L^3)$.

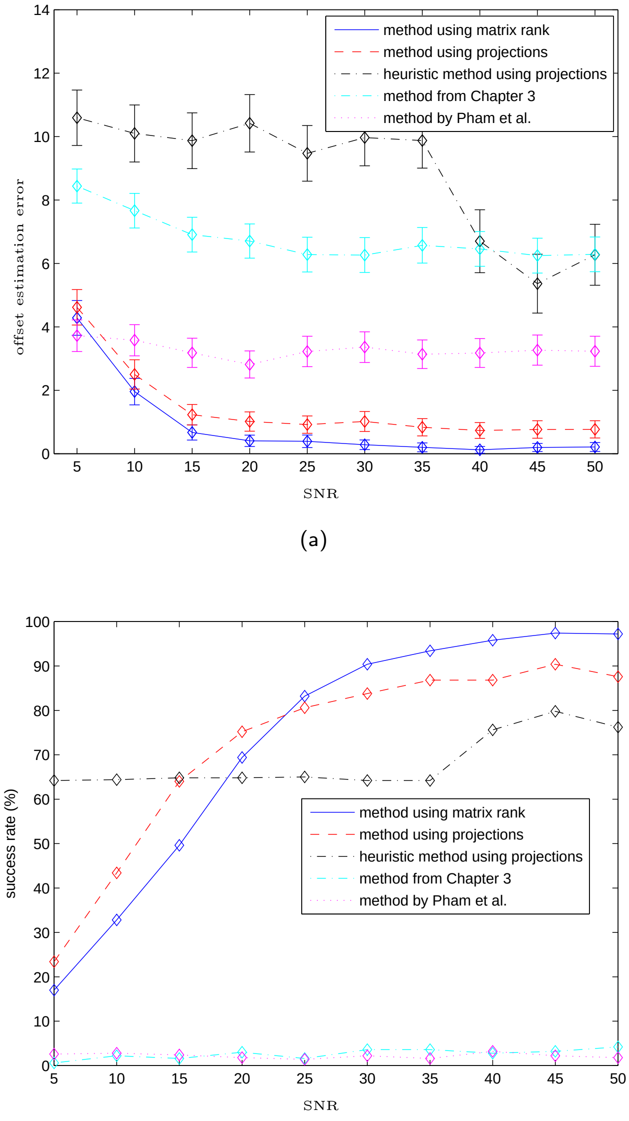
5.5 Results

The above algorithms are tested in a number of simulations and compared to the algorithm from Chapter 3 and the algorithm by Pham et al. [83]. The algorithm by Pham et al. is an iterative Taylor-based spatial domain algorithm. We used the implementation from [111]. The simulations are performed on one-dimensional random bandlimited signals. To be able to compare both methods directly, we use a Fourier basis. The Fourier coefficients of the signals are generated as a white Gaussian random process. Hence, the resulting (time domain) signals also form white Gaussian random processes. A number of simulations are performed with different random offset values, and different amounts of additive white Gaussian noise. The performance of the different algorithms on 1D signals is compared in Figures 5.7, 5.8 and 5.9. Figures 5.7(a) and (b) show the mean absolute error in the shift estimates and the success rate as a function of the SNR, respectively. The success rate of the methods is defined as the relative number of simulations in which the error on the registration is smaller than 10^{-3}, which we consider a sufficient precision for accurate signal reconstruction. All the results were averaged over 250 simulations. Parameter values of $M = 3$, $L = 81$, and $N = 41$ were used in all the simulations. An offset value of 1 corresponds to a shift over the sampling period $1/N$. We can see that the method using matrix rank from Section 5.1 performs slightly better than the method using projections described in Section 5.2. The heuristic line search method (Algorithm 5.3.2) performs clearly worse than the two methods that sample a uniform grid first and then perform a minimization around the minimum value. It works well for high SNR values, but breaks down for SNRs around 35 dB. It can also be clearly seen that both the rank-based and the projection based algorithm perform much better on totally aliased signals than the methods that do not use aliasing, such as the method from Chapter 3 and [83].

The absolute error in the shift estimates and the success rate are plotted as a function of the number of samples per set N in Figure 5.8. In these simulations, parameters $M = 3$ and $L = 81$ were used, and for N, values from 25 to 45 were taken. No noise was added for these simulations. Again, the results were averaged over 250 simulations. The performance of the different algorithms increases with increasing number of samples. This is what we expected, as an increasing number of samples per set gives increasing amounts of information for the same number of unknowns. For the matrix rank algorithm from Section 5.1, the minimum number of samples per set that are required in this setup is $N = 41$, while for the projection algorithm from Section 5.2, $N = 28$ samples per set are needed. This explains the better performance of the projection algorithm for $28 \leq N < 41$. With more than 41 samples per set, both algorithms perform very well. When $N < 28$ samples are available per set, all algorithms have low accuracy.

The simulations shown in Figure 5.7 were repeated on random one-dimensional signals with a $1/f$ frequency behavior, also known as pink noise [60,127]. This is a model that is often used for natural images [105]. The results are shown in Figure 5.9. We can see that in such a case, the influence of the aliasing is decreased, and the methods from Chapter 3 and [83] perform much better than in the previous simulation. The method by Pham et al. [83] even has a better average performance than our methods described in this chapter. However, from Figure 5.9(b), we can see that our algorithm still computed the offsets up to a precision of 10^{-3} in a larger proportion of the cases. The heuristic line search algorithm now performs worse for all SNR values.

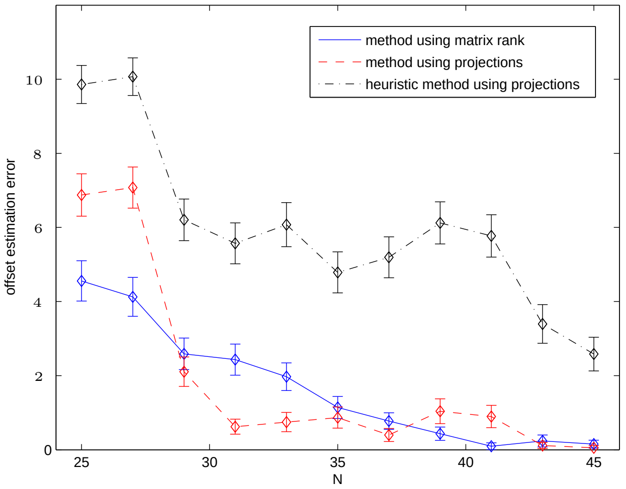
Our algorithms are also applied to 2D signals (images). In this case, the signals were undersampled by only two, because of the high complexity (see the discussion in Section 5.3). The results can be seen in Figure 5.10.



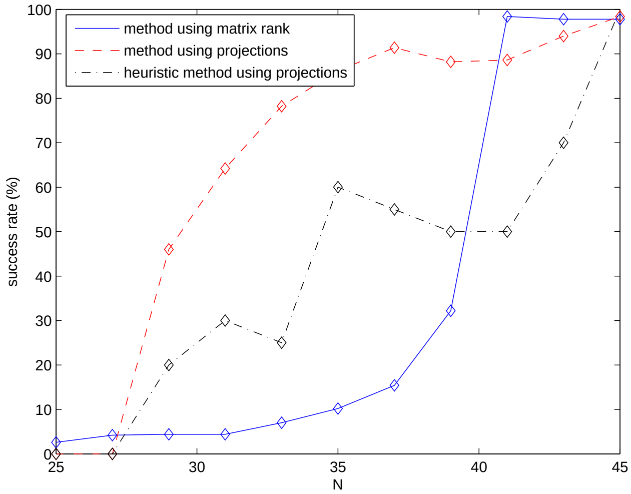
(a)

(b)

Figure 5.7: Results of the different algorithms as a function of the signal-to-noise ratio (SNR). The matrix rank algorithm from Section 5.1 performs slightly better than the projection algorithm from Section 5.2. The heuristic line search method (Algorithm 5.3.2) only performs well for high SNRs. Both the matrix rank and the projection algorithms perform clearly better than the algorithms that do not use the aliasing for registration (methods from Chapter 3 and [83]). Parameter values of $M = 3$, $L = 81$, and $N = 41$ were used. (a) Offset estimation error as a function of the SNR of the sampled signals. An offset error of 1 corresponds to a shift over one sampling period. (b) Success rate in percentage of the registration and reconstruction as a function of the SNR.

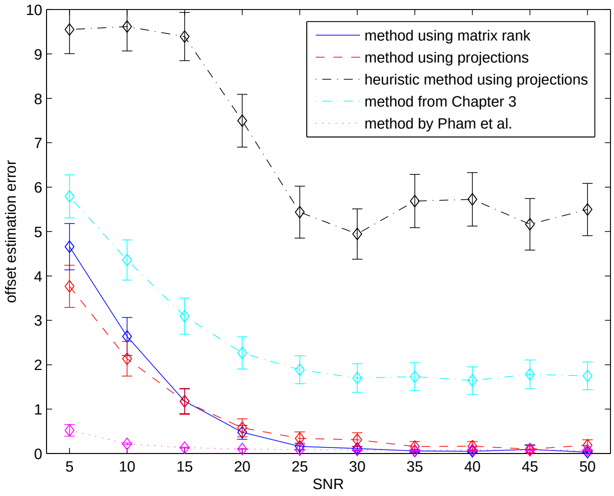


(a)

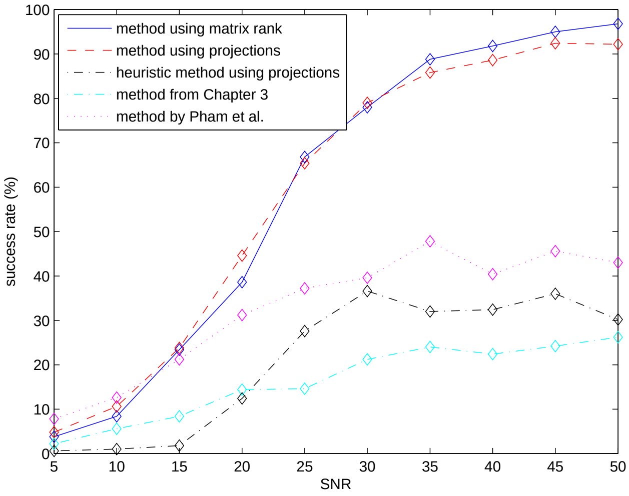


(b)

Figure 5.8: Results of the different algorithms as a function of the number of samples N. Parameter values $M = 3$ and $L = 81$ were used in all simulations, and for N, values from 25 to 45 were used. No noise was added in this setup. The algorithm from Section 5.2 performs better than the algorithm from Section 5.1 if the number of samples per set N is low. The heuristic line search method (Algorithm 5.3.2) performs worse than the other algorithms. (a) Offset estimation error as a function of the number of samples per set N. An offset error of 1 corresponds to a shift over one sampling period. (b) Success rate in percentage of the registration and reconstruction as a function of the number of samples per set N.



(a)



(b)

Figure 5.9: Results of the different algorithms as a function of the signal-to-noise ratio (SNR) on pink noise. The matrix rank algorithm from Section 5.1 performs slightly better than the projection algorithm from Section 5.2. The algorithm from Chapter 3 still performs worse than the algorithms for totally aliased signals presented here. The algorithm from [83] has a lower average absolute error than the other algorithms, but its success rate is smaller than the algorithms presented in this chapter. The heuristic line search method (Algorithm 5.3.2) has always a lower performance than the other algorithms. Parameter values of $M = 3$, $L = 81$, and $N = 41$ were used. (a) Offset estimation error as a function of the SNR of the sampled signals. (b) Success rate in percentage of the registration and reconstruction as a function of the SNR.

A double resolution image is accurately reconstructed (up to Matlab precision) from a set of low resolution images, both using the rank-based algorithm from Section 5.1 (Figures 5.10(a) and 5.10(b)) and the projection-based algorithm from Section 5.2 (Figures 5.10(c) and 5.10(d)).

In all the simulations, we use periodic signals. This is not the case in most real applications, but we can generally assume that the shift between the sets of samples is small. The differences between the signals due to their aperiodicity are therefore small, and can be neglected.

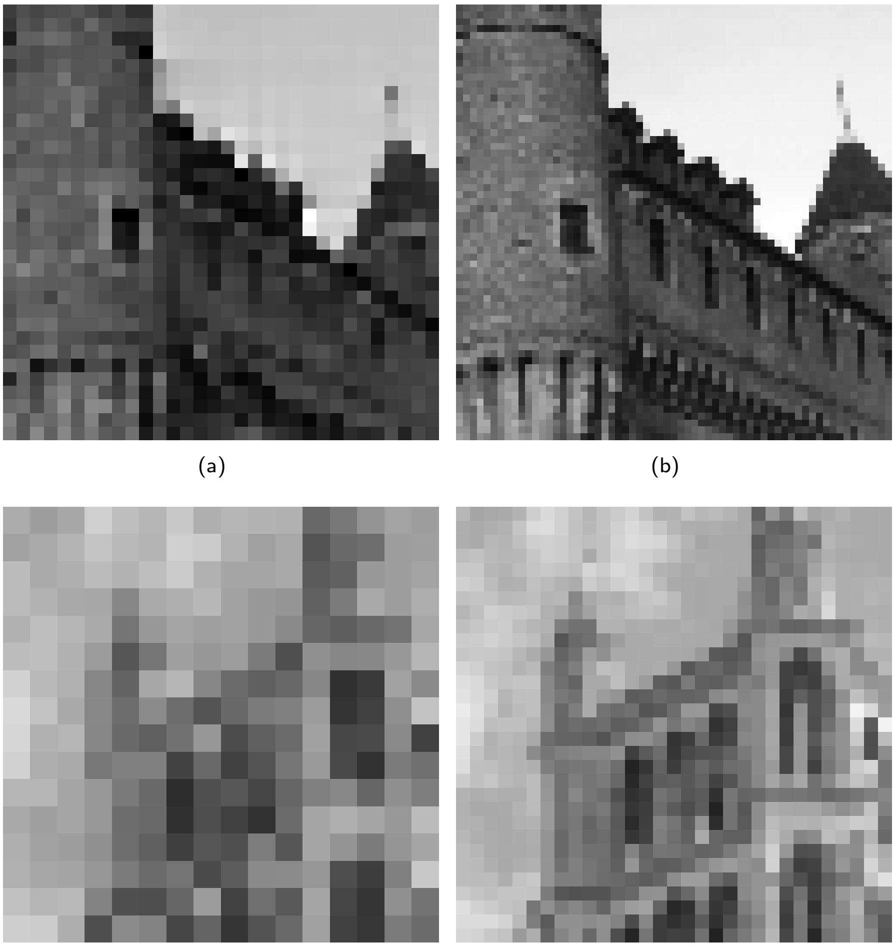

(a) (b)
(c) (d)

Figure 5.10: Simulation results of the different algorithms on images (noiseless). (a) One of the five 32×32 images used as input for Algorithm 5.3.3, with the rank-based method from Section 5.1. (b) Reconstructed 63×63 image from the images in (a). (c) One of the five 16×16 images used as input for Algorithm 5.3.1, with the projection-based method from Section 5.2. (d) Reconstructed 31×31 image from the images in (c).

5.6 Conclusions

In this chapter, we presented two subspace methods to reconstruct a high resolution signal from multiple sets of totally aliased samples. In the first method, the rank of a modified sample matrix is checked. The second method uses projections onto subspaces to compute the offsets between the different sets of samples. The first method is only applicable to bandlimited signals, while the second is applicable to any type of signals in a finite-dimensional Hilbert space. Both methods can be used for the reconstruction of one or two-dimensional signals from multiple sets of aliased samples. The main limitation of these methods is their computational complexity. They are therefore mainly applicable in domains that do not require real-time reconstruction.

Chapter 6

Registration of Totally Aliased Signals using Gröbner Bases

In this chapter, we show that, in many cases, the multichannel sampling problem with unknown offsets can be written as a set of polynomial equations in both the unknown signal coefficients and the offsets. The solution can then be computed using Gröbner bases [93, 115]. In any practical setting, the samples are corrupted by noise, and then there is no algebraic solution. Thus, the next step is to address this noisy version of the problem, and to show how a good approximation can be obtained from multiple Gröbner bases for subsets of samples. Due to the memory requirements of Gröbner basis algorithms, these methods will only be applied to one-dimensional signals.

Gröbner basis theory is a very powerful tool from algebraic geometry. The theory was originally introduced by Buchberger in 1965 [11], and can be found in some very good text books, like for example the book by Cox et al. [21], as well as in many free (Macaulay2, Singular) and commercial (Magma, Maple, Mathematica) software packages. Gröbner bases have also found their way into many applications in signal processing and system theory [12,13]. Examples can be found in filter bank design [18, 32, 49, 78], multichannel deconvolution [132], or motion estimation [43]. In this last paper, Holt et al. use algebraic geometry to determine the number of solutions and uniqueness for certain problems in three-dimensional motion estimation. They analyze the 3D motion of a rigid link moving in a plane where one endpoint is known, and the extraction of 3D motion from 2D optical flow information. We will consider here shifts of one-dimensional signals, which can be extended to global planar shifts of images in the image plane.

This chapter is structured as follows. The multichannel sampling problem with unknown offsets is formulated mathematically as a set of polynomial equations in Section 6.1. Section 6.2 gives an overview of Gröbner basis theory, and more particularly the main ideas that we will use for our reconstruction problem. Gröbner bases are then applied to the multichannel sampling problem in Section 6.3. Section 6.4 presents a solution for noisy measurements. The complexity of such an algorithm is discussed in Section 6.5, and some optimizations are presented that take advantage of the particular structure of the polynomials. Finally, Section 6.6 concludes this chapter.

6.1 Multichannel sampling as a set of polynomial equations

Using the setup from Chapter 2, we can write the sample vector as in (2.7):

$$\mathbf{y} = \mathbf{\Phi_t} \boldsymbol{\alpha}. \tag{6.1}$$

We illustrate this again with an example for second order polynomials like in Example 2.1.2. Now we fill in the signal parameters for a specific signal.

Example 6.1.1 (Second degree polynomials). Consider the space $\mathcal{H}$ defined as the span of the functions $\varphi_l(t) = t^l$, $l = 0, \ldots, L-1$, with $L = 3$. Assume that we take two sets of two samples, i.e. $M = 2$, $N = 2$. If we consider the signal parameters $\boldsymbol{\alpha} = \begin{pmatrix} 64 & -24 & -4 \end{pmatrix}^T$ and offsets $\mathbf{t} = \begin{pmatrix} 0 & 1/4 \end{pmatrix}$, the two sets of samples are $\mathbf{y}_0 = \begin{pmatrix} -4 & 0 \end{pmatrix}^T$ and $\mathbf{y}_1 = \begin{pmatrix} -6 & 6 \end{pmatrix}^T$. The signal and its samples are shown in Figure 6.1. In this case, (6.1) becomes

$$\begin{pmatrix} 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{2} & 1 \\ \frac{1}{4}t_1^2 & \frac{1}{2}t_1 & 1 \\ (\frac{1}{2}+\frac{1}{2}t_1)^2 & \frac{1}{2}+\frac{1}{2}t_1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} -4 \\ 0 \\ -6 \\ 6 \end{pmatrix}. \tag{6.2}$$

We can clearly see that the unknown offset t_1 appears together in the equations with the unknown signal coefficients α_0, α_1 and α_2. □

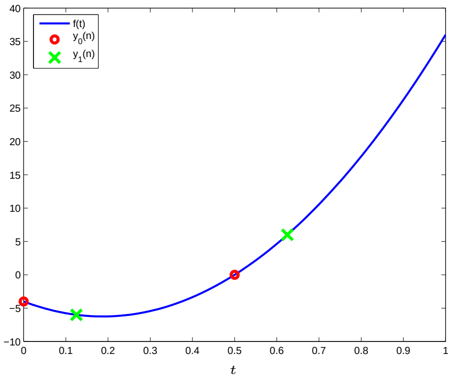


Figure 6.1: Second degree polynomial signal used in Example 6.1.1. The signal $f(t) = 64t^2 - 24t - 4$ is sampled with two sets of two samples $\mathbf{y}_0 = (-4 \quad 0)^T$ and $\mathbf{y}_1 = (-6 \quad 6)^T$ with offset $t_1 = 1/4$.

In the above example, we obtain a set of nonlinear polynomial equations. The equations are linear in the unknown signal coefficients $\boldsymbol{\alpha}$. Thanks to the specific choice of a polynomial basis $\{\varphi_l(t) = t^l\}$, the equations are polynomials in the offsets $\mathbf{t}$. Note that for an arbitrary basis $\{\varphi_l(t)\}$, this is not valid. However, for certain bases, we can rewrite the equations (6.1) as a set of polynomial equations using a change of variables. This is possible when the basis is a set of functions $\varphi_l(t) = h(t)^l$, with $h(t)$ an invertible function.

Probably the most important and practically useful example of such a basis is when $h(t) = e^{j2\pi t}$, that is, the Fourier series. In fact, consider the case of a complex signal of the form

$$f(t) = \sum_{l=-K}^{K} \alpha_l \varphi_l(t), \tag{6.3}$$

with $\varphi_l(t) = e^{j2\pi lt}$. The samples are given by

$$y_m(n) = f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} e^{\frac{j2\pi l t_m}{N}} \quad \text{for } 0 \leq n < N, \tag{6.4}$$

with $W = e^{j2\pi/N}$. By setting $z_m = e^{j2\pi t_m/N}$, we obtain

$$y_m(n) = f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^l. \tag{6.5}$$

We multiply (6.5) with z_m^K to eliminate negative exponents:

$$z_m^K y_m(n) = z_m^K f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^{l+K}. \tag{6.6}$$

For each sample, this can be rewritten as a polynomial constraint

$$p_{nN+m} = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^{l+K} - z_m^K y_m(n) = 0. \tag{6.7}$$

In this equation, the unknowns are the signal parameters α_l and the offset-dependent variables z_m. As in Example 6.1.1, the equations are linear in the signal parameters and polynomial in the offset variables z_m. We will now introduce Gröbner bases and Buchberger's algorithm, which provide an elegant method to solve such a set of polynomial equations.

6.2 Gröbner bases

We present here the main results related to our multichannel sampling problem and we refer to Cox et al. [21] and Buchberger [12, 13] for a complete presentation of algebraic geometry and Gröbner bases. This section is intended as a quick introduction and overview of key results that are necessary to our solution method. It can be skipped by readers familiar with Gröbner bases.

6.2.1 Affine varieties and ideals

We consider polynomials in the n complex variables, $x_0, \ldots, x_{n-1}$. A polynomial p can then be written compactly as

$$p = \sum_{\mathbf{d}} a_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}, \qquad a_{\mathbf{d}} \in \mathbb{C}, \tag{6.8}$$

where the sum is over a finite number of n-tuples $\mathbf{d} = (d_0, \ldots, d_{n-1})$ and $\mathbf{x}^{\mathbf{d}}$ is a compact notation for $x_0^{d_0} \ldots x_{n-1}^{d_{n-1}}$. Each term of the sum in (6.8) is called a *monomial.* In the following, we will denote $\mathbb{C}[x_0, \ldots, x_{n-1}]$ the set of (complex) polynomials in the variables $x_0, \ldots, x_{n-1}$.

The basic objects of algebraic geometry are affine varieties:

Definition 6.2.1 (Affine Variety). Consider the polynomials $p_0, \ldots, p_{s-1}$ in the n variables $x_0, \ldots, x_{n-1} \in \mathbb{C}$. Then we set,

$$V(p_0, \ldots, p_{s-1}) = \{(c_0, \ldots, c_{n-1}) \in \mathbb{C}^n : p_i(c_0, \ldots, c_{n-1}) = 0, \ \forall\, 0 \le i < s\}. \tag{6.9}$$

We call $V(p_0, \ldots, p_{s-1})$ the *affine variety* defined by $p_0, \ldots, p_{s-1}$. The elements of an affine variety are the points for which the polynomials $p_0, \ldots, p_{s-1}$ are all zero.

The determination of the affine variety is trivial in the linear case, since the polynomial p_i has the simple form

$$p_i(x_0, \ldots, x_{n-1}) = a_{i0} x_0 + \ldots + a_{i(n-1)} x_{n-1} + b_i \qquad i = 0, \ldots, s-1 \tag{6.10}$$

and the points of the variety $V(p_0, \ldots, p_{s-1})$ are those that satisfy the system

$$\mathbf{A}\mathbf{x} + \mathbf{b} = 0, \tag{6.11}$$

with $\{\mathbf{A}\}_{i,j} = a_{ij}$ and $\mathbf{b} = (b_0, \ldots, b_{s-1})^T$. The solution can be easily computed by using Gaussian elimination. Recall that Gaussian elimination consists in computing linear combinations of the rows of (6.11) in order to remove progressively the variables. The method is based on a certain ordering of the variables. For example, with the ordering $x_0, x_1, \ldots, x_{n-1}$, we obtain a system

$$\tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{b}} = \mathbf{0}. \tag{6.12}$$

The i-th row of $\tilde{\mathbf{A}}$ has the form

$$\begin{pmatrix} 0 & \ldots & 0 & \tilde{a}_{ij_i} & \tilde{a}_{ij_i+1} & \ldots & \tilde{a}_{i(n-1)} \end{pmatrix}. \tag{6.13}$$

The leading zeros in each row correspond to the positions of the variables that have been eliminated from the previous equations. Therefore, we have (possibly with an initial reordering of the equations)

$$j_1 < j_2 < \ldots < j_l < n, \tag{6.14}$$

and the rows $l+1$ to s are all zero. That is, at least one of the variables is eliminated at each step (and possibly more than one). Note that, after the l-th equation, all the variables are eliminated. If $\tilde{b}_l = \ldots = \tilde{b}_{s-1} = 0$,

$\mathrm{rank}(\tilde{\mathbf{A}} \mid \tilde{b}) = \mathrm{rank}(\tilde{\mathbf{A}}) = l$ and the system admits a solution. The solution of the system is obtained by back substitution.

The procedure of Gaussian elimination can be extended to the case of polynomial equations. This extension is known as Buchberger's algorithm and the set of equations obtained after elimination is called a Gröbner basis. In order to give an overview of the algorithm, we recall the theoretical background and show the analogy with Gaussian elimination. We refer to the bibliography for the details and formal proofs.

As in the linear case, we need to define an ordering of the terms of (6.8), i.e. the monomials of $x_0, \ldots, x_{n-1}$. Since the variables may appear with different exponents, there are different ways to order monomials according to the variables and the exponents. A common choice is *lexicographic (lex) ordering.*

Definition 6.2.2 (Lexicographic ordering). Let $\mathbf{d} = (d_0, \ldots, d_{n-1})$ and $\mathbf{d}' = (d'_0, \ldots, d'_{n-1})$ be two n-tuples representing positive integer exponents of the monomials $\mathbf{x}^{\mathbf{d}}$, $\mathbf{x}^{\mathbf{d}'}$. We say that $\mathbf{d} >_{\mathrm{lex}} \mathbf{d}'$ if, in the vector difference $\mathbf{d} - \mathbf{d}' \in \mathbb{Z}^n$, the left-most nonzero entry is positive. We will write $\mathbf{x}^{\mathbf{d}} >_{\mathrm{lex}} \mathbf{x}^{\mathbf{d}'}$ if $\mathbf{d} >_{\mathrm{lex}} \mathbf{d}'$.

Note that, next to the type of ordering, we also need to define the order between the different variables. In the following, we will assume that the terms of each polynomial are ordered in descending order according to lex ordering, and with $x_0 > x_1 > \ldots > x_{n-1}$. We define the *multidegree* of a polynomial p, $\mathrm{multideg}(p)$ as the largest exponent of the monomials of p according to the lex ordering. We call *leading term*, $\mathrm{LT}(p)$ the term of p with the largest exponent. The *total degree* of a polynomial is defined as the maximum sum of the exponent vectors $\mathbf{d}$ of its terms.

Example 6.2.1. Let us consider a polynomial

$$p = 2x_0^3x_1^2 + 5x_0x_1^3x_2^3 + 3x_1^4x_2. \tag{6.15}$$

Using lex ordering, and $x_0 > x_1 > x_2$, we have $x_0^3x_1^2 > x_0x_1^3x_2^3 > x_1^4x_2$, and (6.15) is ordered in descending lexicographic order. Its multidegree is $\mathrm{multideg}(p) = (3, 2, 0)$, and the leading term $\mathrm{LT}(p) = 2x_0^3x_1^2$. The total degree is $1 + 3 + 3 = 7$. □

In the procedure of Gaussian elimination, the equations of the system correspond to a set of vectors generating a subspace. The aim of elimination is to determine a new basis for such a subspace with the structure given by (6.12). In the case of polynomials, the equations can be combined using polynomial coefficients. The set of all polynomials that can be constructed from an original set has the algebraic structure of an ideal of the ring of polynomials.

Definition 6.2.3 (Ideal). A subset $I \subset \mathbb{C}[x_0, \ldots, x_{n-1}]$ is an *ideal* if it satisfies:

1. $0 \in I$.
2. If $p, q \in I$, then $p + q \in I$.
3. If $p \in I$ and $a \in \mathbb{C}[x_0, \ldots, x_{n-1}]$, then $ap \in I$.

If $p_0, \dots, p_{s-1}$ are polynomials, then we set

$$I = \langle p_0, \dots, p_{s-1} \rangle = \left\{ \sum_{i=0}^{s-1} a_i p_i : a_i \in \mathbb{C}[x_0, \dots, x_{n-1}] \right\}. \tag{6.16}$$

We call I the ideal generated by $p_0, \dots, p_{s-1}$.

6.2.2 The Ideal Membership Problem

A key problem in algebra is to determine whether a given element p of a ring belongs to a given ideal I or not. In terms of polynomials, the problem is equivalent to testing if a given polynomial p can be written as a linear combination of the polynomial generators of I, $p_0, \dots, p_{s-1}$, using polynomial coefficients $a_0, \dots, a_{s-1}$. Such a problem is known as the *Ideal Membership Problem.*

If we think of an ideal generated by a single polynomial in one variable, the problem has a simple solution. In fact, we can apply the algorithm of polynomial division and write p as

$$p = a_0 p_0 + r. \tag{6.17}$$

The quotient a_0 and the remainder r are uniquely determined under the condition that $\deg(r) < \deg(p_0)$. In this case, the ideal membership problem has a simple solution: if $r = 0$, p belongs to $\langle p_0 \rangle$, otherwise not.

In the case of multiple polynomials in multiple variables, we can extend the algorithm of polynomial division. The goal is to write p as

$$p = a_0 p_0 + \ldots + a_{s-1} p_{s-1} + r. \tag{6.18}$$

The division algorithm consists in considering the monomials of p in decreasing order. For each monomial, if the leading term of one of the p_i's is a divisor, then the corresponding quotient a_i is updated together with the remaining monomials of p. Otherwise, the monomial is moved to the remainder r. The following theorem can be proven for polynomial division [21, § 2.3, Theorem 3].

Theorem 6.2.1. Fix a monomial order and let $P = (p_0, \dots, p_{s-1})$ be an ordered s-tuple of polynomials in $x_0, \dots, x_{n-1}$. Then every polynomial p can be written as in (6.18), where either $r = 0$ or r is a linear combination of monomials, none of which is divisible by any of $\mathrm{LT}(p_0), \dots, \mathrm{LT}(p_{s-1})$. Furthermore, we have

$$\mathrm{multideg}(p) \geq \mathrm{multideg}(a_i p_i) \quad i = 0, \dots, s-1. \tag{6.19}$$

A crucial point of the algorithm is that the result of the division depends on the order that we consider for the divisors $p_0, \dots, p_{s-1}$.

Example 6.2.2. Let $p_0 = x_0 x_1 + 1$, $p_1 = x_1^2 - 1$ be two polynomials in x_0, x_1 and assume we use the lex order with $x_0 > x_1$. If we divide $p = x_0 x_1^2 - x_0$ by $P = (p_0, p_1)$ the result is

$$x_0 x_1^2 - x_0 = x_1 \cdot (x_0 x_1 + 1) + 0 \cdot (x_1^2 - 1) + (-x_0 - x_1). \tag{6.20}$$

With $P = (p_1, p_0)$, however, we have

$$x_0 x_1^2 - x_0 = x_0 \cdot (x_1^2 - 1) + 0 \cdot (x_0 x_1 + 1) + 0. \tag{6.21}$$

□

Therefore, the result of division is not unique. Moreover, the remainder of division may be nonzero, even if $p \in \langle p_0, p_1 \rangle$. In the following, we will denote $\overline{p}^P$ the remainder r of the division of p by the s-tuple of polynomials P.

There are some cases where the s-tuple of polynomials has a particular structure that allows to solve the ambiguity. A set with such a property is called a Gröbner basis.

Definition 6.2.4 (Gröbner basis). Let $G = \{g_0, \dots, g_{u-1}\}$ be a basis for the ideal I. If for all $p \in I$ the remainder of the division $\overline{p}^G = 0$ then G is called a *Gröbner basis* for I.

Gröbner bases have several interesting properties, including a generalization of the structure of the system (6.12). However, the most surprising result is given by the following theorem [21, § 2.5, Theorem 4]:

Theorem 6.2.2 (Hilbert Basis Theorem). Every ideal I of the ring of polynomials of n variables has a finite generating set. That is, $I = \langle g_0, \dots, g_{u-1} \rangle$ for some $g_0, \dots, g_{u-1} \in I$. In particular, it is always possible to choose $g_0, \dots, g_{u-1}$ so that they form a Gröbner basis.

6.2.3 Buchberger's algorithm

The key step of Gaussian elimination was to combine two rows of the matrix (i.e. two equations) in order to cancel the entry corresponding to the variable of highest order. This concept is extended to polynomials by introducing S-polynomials.

Definition 6.2.5 (S-polynomial). Let p_0, p_1 be two non-zero polynomials in $x_0, \dots, x_{n-1}$. If $\text{multideg}(p_0) = \mathbf{d}$ and $\text{multideg}(p_1) = \mathbf{d}'$, then let $\mathbf{d}'' = (d_0'', \dots, d_{n-1}'')$, where $d_i'' = \max(d_i, d_i')$. The *S-polynomial of p_0 and p_1* is defined as the linear combination

$$S(p_0, p_1) = \frac{\mathbf{x}^{\mathbf{d}''}}{\text{LT}(p_0)} p_0 - \frac{\mathbf{x}^{\mathbf{d}''}}{\text{LT}(p_1)} p_1. \tag{6.22}$$

Using S-polynomials, we can easily verify if a basis G is a Gröbner basis. In fact, we have the following theorem [21, § 2.6, Theorem 6]:

Theorem 6.2.3. Let I be a polynomial ideal. Then a basis $G = \{g_0, \dots, g_{u-1}\}$ is a Gröbner basis for I if and only if, for all pairs $i \neq j$, the remainder on division of $S(g_i, g_j)$ by G (listed in some order) is zero.

There is a main difference between the linear and the polynomial case when we combine equations. In the linear case, if we combine p_0 and p_1 we obtain an equation of the form

$$h = a p_0 + b p_1, \qquad a, b \in \mathbb{C} \tag{6.23}$$

and this equation can be used to replace p_0 or p_1, i.e.

$$\langle p_0, p_1 \rangle = \langle p_0, h \rangle = \langle h, p_1 \rangle. \tag{6.24}$$

In the polynomial case, equations are combined using polynomial coefficients, i.e. the terms a and b are polynomials in the variables $x_0, \ldots, x_{n-1}$. Since the set of polynomials has the structure of a ring, it is not always possible to find an inverse of the coefficients. This means that, for example, it is not always possible to compute p_1 from p_0 and h. For this reason, to construct a Gröbner basis, one has to increase initially the number of elements of the basis. Such an extension ends when the conditions given by Theorem 6.2.3 are satisfied. This algorithm is due to Buchberger and is given in the following algorithm [21, § 2.7, Theorem 2].

Algorithm 6.1: Buchberger's algorithm for the computation of a Gröbner basis.

Let $I = \langle p_0, \ldots, p_{s-1} \rangle \neq 0$ be a polynomial ideal. Then a Gröbner basis for I can be constructed in a finite number of steps by the following algorithm:

```
Input: $P = (p_0, \ldots, p_{s-1})$
Output: a Gröbner basis $G = (g_0, \ldots, g_{u-1})$ for $I$, with $P \subseteq G$
$G := P$
Repeat
    $G' := G$
    For each pair $(p, q)$, $p \neq q$ in $G'$ do
        $S := \overline{S(p,q)}^{G'}$
        If $S \neq 0$ then $G := G \cup S$
until $G = G'$.
```

Algorithm 6.1 is not a very practical way to compute a Gröbner basis. Several improvements are possible. Moreover, Gröbner bases computed in this way are often bigger than necessary. For this reason, unneeded generators are eliminated by using Theorem 6.2.3 or similar tests.

6.2.4 Solution of polynomial equations

We can now show that a Gröbner basis corresponding to a system of polynomial equations and built using lex ordering simplifies the system and allows to compute the solution by back substitution. Remember that we defined the ideal I as the set of all polynomials that can be derived from the initial set using polynomial coefficients. We can also define the *elimination ideal* I_k as the set of all polynomials that can be deduced from the original system and contain only the variables $x_k, \ldots, x_{n-1}$,

$$I_k = I \cap \mathbb{C}[x_k, \ldots, x_{n-1}]. \tag{6.25}$$

If we can find a basis for each one of the sets I_k, $k = 1, \ldots, n-1$, we can determine the solutions of the original system using back substitution. In fact, we clearly have that for any $k \geq 1$, $I_{k+1} \subseteq I_k$. Therefore, if we have a solution of the system of equations associated to I_{k+1}, we can extend it to the system associated to I_k by computing the values of the variable x_k. This can be done by computing the zeros of a polynomial in the variable x_k. An important property of Gröbner bases is that they solve easily the problem of determining the ideals I_k, $k = 1, \ldots, n-1$. Namely, the Gröbner bases of all the ideals I_k, $k = 1, \ldots, n-1$ can be determined from the Gröbner basis of I. The result is given by the elimination theorem [21, § 3.1, Theorem 2]:

Theorem 6.2.4 (Elimination Theorem). Let $I \subset [x_0, \ldots, x_{n-1}]$ be an ideal and let G be a Gröbner basis of I with respect to lex order where $x_0 > x_1 > \ldots > x_{n-1}$. Then, for every $1 \leq k < n$, the set

$$G_k = G \cap \mathbb{C}[x_k, \ldots, x_{n-1}] \tag{6.26}$$

is a Gröbner basis of the k-th elimination ideal I_k.

Using this theorem, we can compute the different variables from a Gröbner basis using back substitution. To summarize, we can solve a set of polynomial equations in multiple variables as follows. First, we compute a Gröbner basis for the ideal corresponding to the set of equations using Buchberger's algorithm. The solution can then be obtained from this Gröbner basis using back substitution.

6.3 Multichannel Sampling using Gröbner Bases

We can now use Gröbner bases and Buchberger's algorithm to solve the equations from (6.1). After a possible change of variables to write the equations in polynomial form, we can directly apply Buchberger's algorithm. This results in a Gröbner basis for the ideal defined by the set of equations. The signal parameters can then be easily extracted from this Gröbner basis using the elimination theorem. This is summarized in Algorithm 6.2. We will illustrate this algorithm with two examples for polynomial signals and signals described by Fourier series.

Algorithm 6.2: Algorithm for multichannel sampling with unknown offsets using Gröbner bases.

1. Write out the equations from (6.1) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of polynomial equations.
3. Compute a Gröbner basis for the set of polynomial equations using Buchberger's algorithm.
4. Use back substitution to compute the offsets and signal parameters from the Gröbner basis.
5. If necessary, eliminate solutions that are not valid (e.g. offset values not on the unit circle in the Fourier case).

Example 6.3.1 (Polynomial signals). First, we reconsider the equations obtained in Example 6.1.1. That is, we consider a second degree polynomial signal with two sets of two samples ($L = 3$, $M = 2$, and $N = 2$, see also Figure 6.1). We can represent the set of solutions of (6.2) as the points of the affine variety defined by the set of polynomials:

$$\begin{aligned}
&p_0 = \alpha_2 + 4,\\
&p_1 = \tfrac{1}{4}\alpha_0 + \tfrac{1}{2}\alpha_1 + \alpha_2,\\
&p_2 = \tfrac{1}{4}\alpha_0 t_1^2 + \tfrac{1}{2}\alpha_1 t_1 + \alpha_2 + 6,\\
&p_3 = \tfrac{1}{4}\alpha_0 t_1^2 + \tfrac{1}{2}\alpha_0 t_1 + \tfrac{1}{4}\alpha_0 + \tfrac{1}{2}\alpha_1 t_1 + \tfrac{1}{2}\alpha_1 + \alpha_2 - 6,
\end{aligned} \tag{6.27}$$

in the variables α_0, α_1, α_2 and t_1. We fix the ordering of variables as $\alpha_0 > \alpha_1 > \alpha_2 > t_1$ and we use lex ordering for monomials.

At the first step of Buchberger's algorithm, we find that

$$\begin{aligned}
S(p_0,p_1) &= 4\alpha_0 - 2\alpha_1\alpha_2 - 4\alpha_2^2 = (-2\alpha_1 - 4\alpha_2)p_0 + 16p_1,\\
S(p_0,p_2) &= \alpha_0 t_1^2 - \tfrac{1}{2}\alpha_1\alpha_2 t_1 - \alpha_2^2 - 6\alpha_2\\
&= (-\tfrac{1}{2}\alpha_1 t_1 - \alpha_2 - 4t_1^2 - 2)p_0 + 4t_1^2 p_1 - 2\alpha_1 t_1^2 + 2\alpha_1 t_1 + 16t_1^2 + 8,
\end{aligned} \tag{6.28}$$

$$\begin{aligned}
S(p_0,p_3) &= -\tfrac{1}{2}\alpha_0\alpha_2 t_1 - \tfrac{1}{4}\alpha_0\alpha_2 + \alpha_0 t_1^2 - \tfrac{1}{2}\alpha_1\alpha_2 t_1 - \tfrac{1}{2}\alpha_1\alpha_2 - \alpha_2^2 + 6\alpha_2\\
&= (-\tfrac{1}{2}\alpha_0 t_1 - \tfrac{1}{4}\alpha_0 - \tfrac{1}{2}\alpha_1 t_1 - \tfrac{1}{2}\alpha_1 - \alpha_2 - 4t_1^2 - 8t_1 + 6)p_0\\
&\quad +(4t_1^2 + 8t_1 + 4)p_1 - 2\alpha_1 t_1^2 - 2\alpha_1 t_1 + 16t_1^2 + 32t_1 - 24,\\
S(p_1,p_2) &= \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 + \tfrac{1}{4}\alpha_2 t_1^2 - \tfrac{1}{4}\alpha_2 - \tfrac{3}{2}\\
&= (\tfrac{1}{4}t_1^2 - \tfrac{1}{4})p_0 + \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 - t_1^2 - \tfrac{1}{2},\\
S(p_1,p_3) &= -\tfrac{1}{8}\alpha_0 t_1 - \tfrac{1}{16}\alpha_0 + \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 - \tfrac{1}{8}\alpha_1 + \tfrac{1}{4}\alpha_2 t_1^2 - \tfrac{1}{4}\alpha_2 + \tfrac{3}{2}\\
&= (\tfrac{1}{4}t_1^2 + \tfrac{1}{2}t_1)p_0 + (-\tfrac{1}{2}t_1 - \tfrac{1}{4})p_1 + \tfrac{1}{8}\alpha_1 t_1^2 + \tfrac{1}{8}\alpha_1 t_1 - t_1^2 - 2t_1 + \tfrac{3}{2},\\
S(p_2,p_3) &= -\tfrac{1}{2}\alpha_0 t_1 - \tfrac{1}{4}\alpha_0 - \tfrac{1}{2}\alpha_1 + 12 = (2t_1 + 1)p_0 + (-2t_1 - 1)p_1\\
&\quad +\alpha_1 t_1 - 8t_1 + 8.
\end{aligned} \tag{6.29}$$

Therefore, we add the remainders that are non-zero to the basis:

$$\begin{aligned}
p_4 &= \overline{S(p_0,p_2)}^G = -2\alpha_1 t_1^2 + 2\alpha_1 t_1 + 16t_1^2 + 8,\\
p_5 &= \overline{S(p_0,p_3)}^G = -2\alpha_1 t_1^2 - 2\alpha_1 t_1 + 16t_1^2 + 32t_1 - 24,\\
p_6 &= \overline{S(p_2,p_3)}^G = \alpha_1 t_1 - 8t_1 + 8.
\end{aligned} \tag{6.30}$$

The remainders of $S(p_1,p_2)$ and $S(p_1,p_3)$ are not added, because they are the same as polynomials p_4 and p_5, respectively. Following the same procedure, in the second iteration, we find that only $S(p_2,p_6)$ and $S(p_4,p_6)$ give a distinct, non-zero remainder. We add the polynomials

$$\begin{aligned}
p_7 &= \overline{S(p_2,p_6)}^G = -2\alpha_1 - 48,\\
p_8 &= \overline{S(p_4,p_6)}^G = 32t_1 - 8
\end{aligned} \tag{6.31}$$

to the basis. In the following iteration all remainders are zero and by Theorem 6.2.3 we conclude that $p_0, \ldots, p_8$ is a Gröbner basis. Applying again Theorem 6.2.3 we can try to reduce the elements of the basis. In this case, we have that p_2, p_3, p_4, p_5, p_6 can be removed and the final basis is given by $\{p_0, p_1, p_7, p_8\}$. In order to apply the elimination theorem, we rename the elements of the basis as:

$$\begin{aligned} g_0 &= \tfrac{1}{4}\alpha_0 + \tfrac{1}{2}\alpha_1 + \alpha_2, \\ g_1 &= -2\alpha_1 - 48, \\ g_2 &= \alpha_2 + 4, \\ g_3 &= 32t_1 - 8. \end{aligned} \tag{6.32}$$

The elimination ideals are $I_1 = \langle g_1, g_2, g_3\rangle$, $I_2 = \langle g_2, g_3\rangle$, and $I_3 = \langle g_3\rangle$. The solution of the problem can be obtained by computing the points of the affine variety associated to I_3 and extending it by back substitution to I_2, I_1 and I. We easily find that the unique solution is given by $t_1 = \frac{1}{4}$, $\alpha_2 = -4$, $\alpha_1 = -24$, and $\alpha_0 = 64$. ☐

The procedure described in the above example can be applied to any multichannel sampling problem in the polynomial space $\mathcal{H}$. For any value of the variables L, M, and N, the equations in (6.1) form a set of polynomial equations and we can therefore compute the parameter values by calculating a Gröbner basis for the corresponding ideal. Similarly, the same algorithm can be applied to Fourier series, using the change of variables given in Section 6.1. This is a very interesting case from a practical point of view, as signals and images are often bandlimited or can be considered to be so.

Example 6.3.2 (Fourier series). Assume $K = 2$ (and therefore $L = 5$), i.e. the input signal is represented by the parameter vector $\boldsymbol{\alpha} = (\alpha_{-2}, \ldots, \alpha_2)$, where each entry is a complex value. For this example, we assume

$$\boldsymbol{\alpha} = \begin{pmatrix} 3 & 2-j & 1 & 2+j & 3 \end{pmatrix}^T.$$

We suppose that $M = 2$ sets of $N = 4$ samples are taken from the input signal, with the displacements $\mathbf{t} = \begin{pmatrix} 0 & 1/2 \end{pmatrix}$. In this case, the two sets of measurements are

$$\begin{aligned} \mathbf{y}_0 &= \begin{pmatrix} 11 & -7 & 3 & -3 \end{pmatrix}^T, \\ \mathbf{y}_1 &= \begin{pmatrix} 1+\sqrt{2} & 1-3\sqrt{2} & 1-\sqrt{2} & 1+3\sqrt{2} \end{pmatrix}^T. \end{aligned} \tag{6.33}$$

The signal and its samples are shown in Figure 6.2. Applying (6.7), we obtain 8 polynomials that represent the constraints imposed by the measurements:

$$\begin{aligned} p_0 &= \alpha_2 + \alpha_1 + \alpha_0 + \alpha_{-1} + \alpha_{-2} - 11, \\ p_1 &= -\alpha_2 + j\alpha_1 + \alpha_0 - j\alpha_{-1} - \alpha_{-2} + 7, \\ p_2 &= \alpha_2 - \alpha_1 + \alpha_0 - \alpha_{-1} + \alpha_{-2} - 3, \\ p_3 &= -\alpha_2 - j\alpha_1 + \alpha_0 + j\alpha_{-1} - \alpha_{-2} + 3, \\ p_4 &= \alpha_2 z_1^4 + \alpha_1 z_1^3 + \alpha_0 z_1^2 + \alpha_{-1} z_1 + \alpha_{-2} - (1+\sqrt{2}) z_1^2, \\ p_5 &= -\alpha_2 z_1^4 + j\alpha_1 z_1^3 + \alpha_0 z_1^2 - j\alpha_{-1} z_1 - \alpha_{-2} - (1-3\sqrt{2}) z_1^2, \\ p_6 &= \alpha_2 z_1^4 - \alpha_1 z_1^3 + \alpha_0 z_1^2 - \alpha_{-1} z_1 + \alpha_{-2} - (1-\sqrt{2}) z_1^2, \\ p_7 &= -\alpha_2 z_1^4 - j\alpha_1 z_1^3 + \alpha_0 z_1^2 + j\alpha_{-1} z_1 - \alpha_{-2} - (1+3\sqrt{2}) z_1^2, \end{aligned} \tag{6.34}$$

where the complex variable $z_1 = e^{j2\pi t_1/4}$ represents the displacement. Again, by using Buchberger's algorithm, we obtain a Gröbner basis. Assuming the

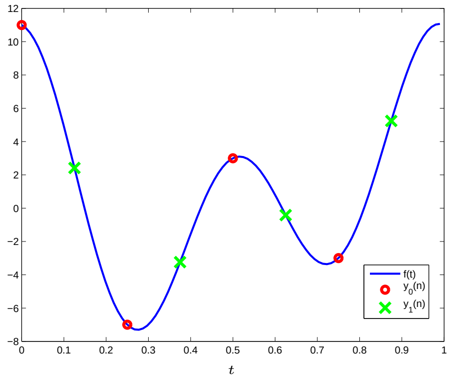


Figure 6.2: Fourier series signal used in Example 6.3.2. The signal with Fourier series coefficients $\mathbf{x} = (\,3 \quad 2-j \quad 1 \quad 2+j \quad 3\,)^T$ is sampled with two sets of four samples $\mathbf{y}_0 = (\,11 \;\; -7 \;\; 3 \;\; -3\,)^T$ and $\mathbf{y}_1 = (\,1+\sqrt{2} \;\; 1-3\sqrt{2} \;\; 1-\sqrt{2} \;\; 1+3\sqrt{2}\,)^T$ with offset $t_1 = 1/2$.

ordering $\alpha_2 > \alpha_1 > \ldots > \alpha_{-2} > z_1$, we obtain

$$\begin{aligned}
g_0 &= 2\alpha_2 - 3j\sqrt{2}z_1 + 3\sqrt{2}z_1 - 12,\\
g_1 &= \alpha_1 - 2 - j,\\
g_2 &= \alpha_0 - 1,\\
g_3 &= \alpha_{-1} - 2 + j,\\
g_4 &= 2\alpha_{-2} + 3j\sqrt{2}z_1 - 3\sqrt{2}z_1,\\
g_5 &= 2z_1^2 - \sqrt{2}(1+j)z_1 = 2z_1(z_1 - \tfrac{\sqrt{2}}{2}(1+j)).
\end{aligned} \tag{6.35}$$

In the last polynomial of the basis, g_5, all variables but z_1 are eliminated. Therefore we can compute the solutions for the displacement variable, $z_1 = 0$ and $z_1 = e^{j\pi/4}$, from g_5. Clearly, $z_1 = 0$ is discarded since it does not belong to the unit circle, while the second solution corresponds to the correct displacement $t_1 = 1/2$. By back substitution, one can compute the signal parameters. □

To sum up, in the above examples we have MN polynomial equations with maximum total degree L. The equations are linear in the signal coefficients α_l, and polynomial of order at most $L-1$ in the offsets t_m. The computed Gröbner basis is linear in Example 6.3.1, and contains a second degree polynomial in Example 6.3.2. This is much lower than the theoretical double exponential bound that will be discussed in Section 6.5.

6.4 Multichannel Sampling under Noisy Conditions

The computation of a Gröbner basis is typically performed with infinite precision. A Gröbner basis is defined as a set of polynomials that generates the same variety as the original set of polynomial equations. The solution that is computed using Gröbner bases is therefore an exact solution to the set of polynomial equations.

Moreover, concepts such as projections or distance do not have any meaning over the ring of polynomials. As we are now working with polynomials, the distance between two such polynomials does not have any meaning. It is not possible to compute a 'least squares solution' to a set of equations with Gröbner bases. Hence, if the measurements are noisy, or known with limited precision, Buchberger's algorithm would generally conclude that there is no solution. As there are usually more equations than unknowns (see Example 6.3.2), the errors on the sample values make the equations from (6.1) incoherent. There has already been a lot of research on the stability of Gröbner basis computation, and various solutions have been proposed [98, 99, 106].

We propose to solve this problem by dividing the complete set of polynomial equations into multiple (overlapping) critical subsets. By critical we mean that there is a finite, non empty set of solutions (typically when the number of equations is equal to the number of unknowns). We could use all the critical subsets that can be derived from the original set of equations, or select only a limited number of them to limit the computational time. We can now compute a Gröbner basis for each subset, and obtain a set of parameter values using back substitution. The final solution can then be defined as a (weighted) average of the different solutions from the subsets. This method is summarized in Algorithm 6.3. Let us now analyze an example.

Algorithm 6.3: Algorithm for multichannel sampling from noisy samples.

1. Write out the equations from (6.1) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of polynomial equations.
3. Divide these equations into at most $\binom{MN}{L+M-1}$ critical subsets of equations $\mathcal{S}_i$.
4. Compute a Gröbner basis for each set $\mathcal{S}_i$. Use back substitution to obtain the offsets and the signal parameters.
5. Eliminate solutions that are not valid (e.g. offset values not on the unit circle in the Fourier case).
6. Compute the weighted average of the offsets corresponding to the remaining solutions (typically one per set $\mathcal{S}_i$).
7. Fill in the offsets in the equations from (6.1) and solve the set of linear equations for the unknown signal parameters.

Example 6.4.1 (Fourier series with noisy measurements). Consider a signal that is represented by its $L = 5$ Fourier series coefficients, given by

$$\boldsymbol{\alpha} = \begin{pmatrix} 5-j & -3j & -6 & 3j & 5+j \end{pmatrix}^T. \tag{6.36}$$

The signal is sampled with two sets of four samples ($M = 2, N = 4$), with an offset vector $\mathbf{t} = \begin{pmatrix} 0 & 24/11 \end{pmatrix} = \begin{pmatrix} 0 & 2.1818 \end{pmatrix}$. In a noiseless case, this would result in the following two sets of samples:

$$\begin{aligned}\mathbf{y}_0 &= \begin{pmatrix} 4 & -22 & 4 & -10 \end{pmatrix}^T, \\ \mathbf{y}_1 &= \begin{pmatrix} 3.0217 & -7.5743 & -0.3591 & -19.0882 \end{pmatrix}^T.\end{aligned} \tag{6.37}$$

The second set of samples is given numerically, because the exact expressions are quite complicated. Now we add white Gaussian noise to these samples with mean zero and standard deviation 1, resulting in the noisy sample values

$$\begin{aligned}\mathbf{y}_0 &= \begin{pmatrix} 3.4845 & -21.2468 & 3.6672 & -9.5310 \end{pmatrix}^T, \\ \mathbf{y}_1 &= \begin{pmatrix} 2.0917 & -7.4480 & 0.7300 & -19.3078 \end{pmatrix}^T,\end{aligned} \tag{6.38}$$

for one particular realization (see also Figure 6.3). We obtain a similar set of polynomials as in (6.34), with just different sample values. As we have 8 equations in 6 unknowns (5 signal parameters and an offset), we compute a Gröbner basis for all $\binom{8}{6} = 28$ subsets $\mathcal{S}_i$ of 6 polynomials from the total set. One of them is given here:

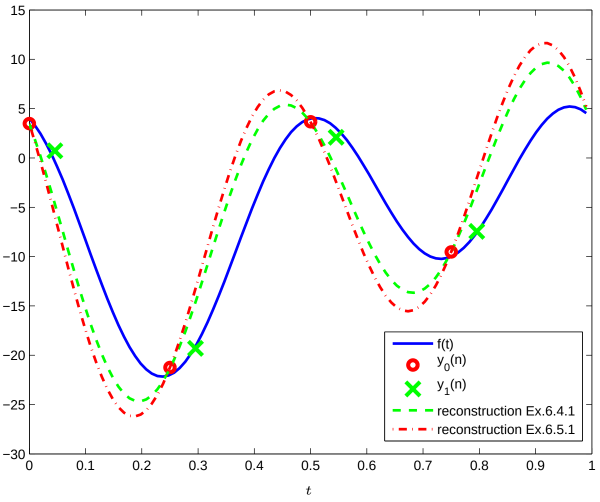


Figure 6.3: Fourier series signal used in Example 6.4.1. The signal with Fourier series coefficients $\mathbf{x} = (\,5-j \quad -3j \quad -6 \quad 3j \quad 5+j\,)^T$ is sampled with two sets of four noisy samples $\mathbf{y}_0 = (\,3.4845 \quad -21.2468 \quad 3.6672 \quad -9.5310\,)^T$ and $\mathbf{y}_1 = (\,2.0917 \quad -7.4480 \quad 0.7300 \quad -19.3078\,)^T$ with offset $t_1 = 24/11$. Its reconstruction both in Example 6.4.1 (– –) and Example 6.5.1 (– · –) are shown.

$$\begin{aligned}g_0 &= \alpha_2 - (11.5043 + 8.2663j)z_1 - (8.5363 + 13.2582j)z_1^2 - 9.4824, \\ g_1 &= 0.04567 - 2.9289j + \alpha_1, \\ g_2 &= 5.9065 + \alpha_0, \\ g_3 &= 0.04567 + 2.9289j + \alpha_{-1}, \\ g_4 &= \alpha_{-2} + (11.5043 + 8.2663j)z_1 + (8.5363 + 13.2582j)z_1^2, \\ g_5 &= z_1^3 + (1.1192 + 1.0848j)z_1^2 + (0.0312 + 0.9995j)z_1.\end{aligned} \tag{6.39}$$

We can then compute all the possible solutions for each of the Gröbner bases. We eliminate the invalid ones: those that do not correspond to valid offsets (values of z_1 that are not on the unit circle), as well as those that give a large error when evaluated on the two remaining equations. Typically, only a single solution remains for every Gröbner basis. From the remaining solutions, we compute the offsets t_1, and compute their average value:

$$t_{1,\text{avg}} = 2.0660. \tag{6.40}$$

This way of proceeding has the advantage that we keep a valid offset value. If we would just average the computed values for z_1, the result is typically not on the unit circle anymore, and does not represent a valid offset. Note that we performed a simple averaging operation here. A weighted average that takes the sensitivity of the results to the different sample values into account would probably improve the results further. We replace this average offset value in the original equations, and compute the least squares solution of this set of linear equations in the unknown signal parameters $\boldsymbol{\alpha}$:

$$\hat{\boldsymbol{\alpha}} = \begin{pmatrix} 4.7412 - 4.5812j \\ -0.0388 - 2.9566j \\ -5.9450 \\ -0.0388 + 2.9566j \\ 4.7412 + 4.5812j \end{pmatrix}. \tag{6.41}$$

The relative error is computed as the norm of the difference between the true coefficient vector $\boldsymbol{\alpha}$ and the estimated coefficient vector $\hat{\boldsymbol{\alpha}}$ divided by the norm of the coefficient vector: $\|\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}\|/\|\boldsymbol{\alpha}\|$. For this simulation, we obtain a relative error of 0.493. This error can be compared to the error that would be obtained from the noisy samples with the exact offset t_1, which is 0.080. Averaged over 250 such simulations with random signal coefficients and offsets, the estimated relative error is 0.340, compared to 0.095 in the ideal case using the exact offsets with the noisy samples. ☐

6.5 Complexity and optimizations

The main disadvantage of Gröbner bases for the multichannel sampling problem is the computational complexity of Buchberger's algorithm. As explained in Section 6.2, the set of polynomials p_i has to be expanded in the first part of the algorithm by adding the non-zero remainders of S-polynomials. Unlike in Gaussian elimination, we cannot simply replace a polynomial by a linear combination of that polynomial with another one. The linear combination has to be added to the existing set of polynomials. This expansion can become very large, and is one of the reasons for the high memory requirements of Buchberger's algorithm. The maximum total degree of the polynomials in a reduced Gröbner basis can be shown to be

$$E = 2\left(\frac{D^2}{2} + D\right)^{2^{S-1}}, \tag{6.42}$$

where D is the maximal total degree of the polynomials p_i, and S is the number of variables [23, 71]. In our setup, the maximal total degree $D = L$, and we would therefore typically obtain

$$E = 2\left(\frac{L^2}{2} + L\right)^{2^{L+M-2}}. \tag{6.43}$$

Fortunately, this double exponential function describes a worst-case scenario, while in practice the complexity is often much lower. In the examples from the previous sections, the degree was always much lower, with only linear terms remaining in Example 6.3.1, and a second degree polynomial in Example 6.3.2. The above upper bounds for those cases would be about 10^7 and 10^{40}, respectively.

Another reason for the high complexity of Buchberger's algorithm is given by the fact that the algorithm performs computations with infinite precision. If for example the input coefficients are (small) integers, quite complicated rational numbers are used in the computation of a Gröbner basis. Using Maple to solve a polynomial problem like the one in Example 6.1.1, with a 6-th degree polynomial, and 3 sets of 3 samples, the algorithm already requires more than 1 GB of memory. In the back substitution step of our solution method, we need to compute the zeros of a polynomial. The complexity of this operation will depend on the order of the specific polynomial that is obtained. Although theoretically, this order can only be bounded by (6.43), in practice, it is often much lower (as can also be seen from the examples). The roots of a polynomial with degree E can be computed using an algorithm with complexity

$$\left(E(\log E)^2 |\log \epsilon| + E^2(\log E)^2\right) \tag{6.44}$$

where ϵ is the precision of the computed roots [52].

Various optimizations of Buchberger's algorithm exist. For example, certain S-polynomials can already be excluded before examining them. Often, other orderings than the lexicographic ordering also result in lower complexity. Algorithms exist to convert a Gröbner basis using one ordering into a Gröbner basis for another ordering. It can therefore be computationally more efficient to compute a Gröbner basis first using another ordering, and convert it then into lexicographic ordering. Lexicographic ordering is required to apply the elimination theorem, which offers a simple way to compute the coefficients using back substitution. Various implementations of Gröbner basis algorithms including different optimizations exist (Gb [32], Macaulay2, Maple, Mathematica, Magma, Singular). We used Mathematica for our simulations. Even though this is probably not the optimal implementation [32], it allows us to implement and clearly show all the important concepts and ideas from this chapter.

It is important to note that the multichannel sampling problem has a particular structure. From (6.1), which describes the problem for any kind of basis, and from the different examples in previous sections, we can see that the equations are linear in the signal parameters. They only have higher polynomial orders in the offset parameters $\mathbf{t}$. Typically, there are many (L) signal parameters, while only a small number (M) of different sets of samples is used. The (linear) signal parameters can be eliminated from the set of equations using

Gaussian elimination on the first L equations. This can be performed in $O(L^3)$ operations, and for our particular structure of the problem and with two sets of samples ($M = 2$), it does not increase the degree of the polynomial coefficients in $\mathbf{t}$. This can be seen from the examples in the previous sections. With a Fourier basis, each term in a signal parameter α_i has the same power of the offset variable z_1, and a varying complex coefficient (or has no offset variable at all, for the first set of N equations). The signal parameters can therefore be eliminated by multiplying equations with complex numbers and adding them together. We never need to multiply any of the equations from the second set by the offset variable z_1, and therefore do not increase its degree. For polynomial signals, we can perform a similar elimination. By ordering the signal parameters as α_{L-1}, α_{L-2}, ..., α_0, we can eliminate each of the parameters without needing to multiply equations by the offset variable t_1. If more than $M = 2$ sets are considered, the different offset variables have to be multiplied in the Gaussian elimination, and the results are more complex.

After this Gaussian elimination step, the computed values for the signal parameters (as a function of $\mathbf{t}$) can be replaced in the $MN - L$ remaining equations. We obtain a (much smaller) set of $MN - L$ polynomial equations in the unknown offsets $\mathbf{t}$. It is now sufficient to compute a Gröbner basis for this smaller set in much fewer unknowns ($M \ll L$). With noisy samples, we can now compute Gröbner bases for the $\binom{MN-L}{M-1}$ subsets of $M - 1$ equations instead of the $\binom{MN}{L+M-1}$ sets of $L + M - 1$ equations previously. Typically this results in much fewer subsets of smaller size. However, the precision with which the parameters are computed is also (slightly) lower. Compared to all the possible subsets of $L + M - 1$ equations in Algorithm 6.3, now only the subsets containing the first L equations and all possible combinations of $M - 1$ equations from the remaining set are considered. The maximum total degree of a Gröbner basis for such a subset is reduced to

$$2\left(\frac{L^2}{2} + L\right)^{2^{M-2}}, \tag{6.45}$$

where the number of sets of samples M is much smaller than the number of coefficients L. As the first L equations are linear in the signal parameters and have lower degrees than (6.45) in the offsets, this bound also replaces the previous bound (6.43) for the general Gröbner basis computation. For Examples 6.3.1 and 6.3.2, these bounds are 15 and 35, respectively. While this is still far beyond the actual degrees of the Gröbner bases, it is already a much closer bound than the one given in (6.43). Once this (smaller) Gröbner basis is computed, the offset values can be obtained using back substitution and a method to compute the zeros of a polynomial. We compute the signal parameters by substituting the offset values in the first L equations. Note however, that with most Gröbner basis algorithms, this procedure is also followed (implicitly), as the signal parameters are eliminated first. The algorithm with explicit Gaussian elimination of the signal parameters is given in Algorithm 6.4. We will now illustrate this method for the setup used in Example 6.4.1.

Example 6.5.1 (Fourier series using Gaussian elimination). We use the same signal and sample values as in Example 6.4.1. Instead of calculating a Gröbner basis for the 28 subsets of 6 equations, we now eliminate the signal

Algorithm 6.4: Algorithm for multichannel sampling from noisy samples using Gaussian elimination for the linear part.

1. Write out the equations from (6.1) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of polynomial equations. These are linear in the signal coefficients $\boldsymbol{\alpha}$, and higher order polynomials in the offsets $\mathbf{t}$.
3. Apply Gaussian elimination on the first L equations to compute the signal coefficients $\boldsymbol{\alpha}$ as a function of the offsets $\mathbf{t}$.
4. Replace these values of $\boldsymbol{\alpha}$ in the remaining $MN - L$ equations and multiply each equation by its common denominator to obtain a set of $MN - L$ polynomial equations in the offsets $\mathbf{t}$.
5. Divide these equations into at most $\binom{MN-L}{M-1}$ critical subsets of equations $\mathcal{S}_i$.
6. Compute a Gröbner basis for each set $\mathcal{S}_i$.
7. Calculate the possible offset values using back substitution and by computing the zeros of polynomial equations.
8. Eliminate offset values that do not give a valid solution (e.g. values not on the unit circle in the Fourier case).
9. Compute the weighted average of the offsets corresponding to the remaining solutions (typically one per set $\mathcal{S}_i$).
10. Replace this value in the equations obtained for the signal parameters $\boldsymbol{\alpha}$.

parameters first from the first $L = 5$ polynomials using Gaussian elimination. This gives us the signal parameters as a function of the offset:

$$\begin{aligned}
\alpha_{-2} &= 9.4824 + \tfrac{9.4824-(0.0457+2.9289j)z_1-7.9982z_1^2-(0.0457-2.9289j)z_1^3}{-1+z_1^4}, \\
\alpha_{-1} &= -0.0457 - 2.9289j, \\
\alpha_0 &= -5.90654, \\
\alpha_1 &= -0.0457 + 2.9289j, \\
\alpha_2 &= \tfrac{-9.48237+(0.0457+2.9289j)z_1+7.9982z_1^2+(0.0457-2.9289j)z_1^3}{-1+z_1^4},
\end{aligned} \tag{6.46}$$

where we assume that $z_1^4 \neq 1$. We can then replace these values in the remaining three equations, and multiply them by their common denominators. This results in three polynomial equations in the unknown offset z_1:

$$\begin{aligned}
&(2.9746 + 2.8833j)z_1 + 6.4568z_1^2 + (2.9746 - 2.8833j)z_1^3 \\
&\qquad -(2.9746 + 2.8833j)z_1^5 - 6.4568z_1^6 - (2.9746 - 2.8833j)z_1^7 = 0, \\
&(-0.0913 - 5.8579j)z_1 - 1.3617z_1^2 - (0.0913 - 5.8579j)z_1^3 \\
&\qquad +(0.0913 + 5.8579j)z_1^5 + 1.3617z_1^6 + (0.0913 - 5.8579j)z_1^7 = 0, \\
&(-2.8833 + 2.9746j)z_1 - 5.4031z_1^2 - (2.8833 + 2.9746j)z_1^3 \\
&\qquad +(2.8833 - 2.9746j)z_1^5 + 5.4031z_1^6 + (2.8833 + 2.9746j)z_1^7 = 0.
\end{aligned} \tag{6.47}$$

As there is only a single unknown offset, the three possible critical subsets of equations that can be formed are the three separate equations. We do not need to compute a Gröbner basis for these subsets and can therefore directly compute the zeros for each of the polynomials separately. After elimination of the zeros that are not valid solutions (additional zeros were added by multiplying with the common denominators, the zeros have to be on the unit circle, etc.), we have the following zeros remaining for the three polynomials:

$$\begin{aligned}
z_1^{(1)} &= -0.9957 - 0.0924j, \\
z_1^{(2)} &= -0.9949 - 0.1007j, \\
z_1^{(3)} &= -0.9982 - 0.0594j.
\end{aligned} \tag{6.48}$$

From these values, we can compute the offsets $t_1^{(1)} = 2.0589$, $t_1^{(2)} = 2.0642$, and $t_1^{(3)} = 2.0378$. We take the average of these solutions ($t_{1,\mathrm{avg}} = 2.0537$), and replace the corresponding value of z_1 in the signal parameter equations (6.46). We obtain the coefficient vector

$$\hat{\boldsymbol{\alpha}} = \begin{pmatrix} 4.7412 - 5.7629j \\ -0.0457 - 2.9289j \\ -5.9065 \\ -0.0457 + 2.9289j \\ 4.7412 + 5.7629j \end{pmatrix}. \tag{6.49}$$

The relative error for our estimation, $\|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|/\|\boldsymbol{\alpha}\|$, is 0.655. This error can be compared to the error that would be obtained by applying a least squares estimation on the noisy samples with the exact offset t_1, which is 0.080. Averaged over 250 simulations with random coefficients and offsets, the estimated error norm is 0.800, compared to 0.095 with the exact offsets. We can see that our estimation has a larger error than in Example 6.4.1, but the computational complexity is also highly reduced. Instead of 28 Gröbner bases for sets of 8 equations, only 3 sets of a single equation remained, which could be directly solved. □

The above simulations show that these Gröbner basis methods are not very robust to noise. At the same time, it is important to note however that the offsets between the sets of samples are unknown. This allows much more variation for possible solutions than when all the sample positions are known.

Remark also that the computation of a Gröbner basis does not depend on the specific values of the samples, except in some degenerate cases. Once the size of the problem (L, M, N) is fixed, we could therefore compute the generic Gröbner basis for this setup. The first six steps from Algorithm 6.4 can then be precomputed. The online computations are reduced to steps 7-10: computing the zeros of a polynomial and replacing the solution(s) in the set of equations for the signal parameters. The zeros of a polynomial can be computed with the complexity given in (6.44). The other operations are negligible compared to this. Buchberger's algorithm is not needed anymore in the actual solution of the specific problem, which can be computed very efficiently.

6.6 Conclusions

In this chapter, we have presented a method to reconstruct a signal from multiple sets of unregistered, aliased samples using Gröbner bases. First, we have shown how multichannel sampling with unknown offsets can be written as a set of polynomial equations. This was shown both for a polynomial signal and for a signal described by its Fourier series. Next, we applied Buchberger's algorithm to compute a Gröbner basis for the ideal corresponding to this set of equations. From a Gröbner basis, we can easily derive the unknown signal parameters. We presented an adaptation to our algorithm in the case of noisy measurements. Gröbner bases are then computed for critical subsets of the offset polynomials. Finally, some complexity issues were discussed, and a more efficient method was presented that computes the linear signal parameters first, such that a Gröbner basis has to be computed only for a much smaller set of equations in the unknown offsets. Even after this optimization, such methods have high memory requirements. Therefore, we only applied them to one-dimensional signals in our simulations.

Chapter 7

Conclusions

7.1 Thesis summary

We have presented super-resolution methods for unregistered aliased images. In such algorithms, an accurate registration is a necessary prerequisite for a good high resolution image reconstruction. We have formulated the reconstruction from multiple unregistered sets of samples (images) as a multichannel sampling problem with unknown offsets. This results in a set of nonlinear equations in the unknown signal coefficients and the offsets. Using this formulation, we have shown that the solution is generally unique if the total number of sample values is larger than or equal to the total number of unknowns (signal parameters and offsets).

Table 7.1 gives a summary and comparison of the applications, advantages and disadvantages of the different methods presented in this thesis. If the images are aliased, but the sampling frequency is above the maximum signal frequency present in the images, the low frequency part of the images is free of aliasing. It is then possible to register each of the images pairwise to the first image using only these low frequencies. A registration method to compute the planar shift and rotation between two images has been developed. Using a frequency domain method, shift and rotation parameters can be estimated separately. First, the rotation parameters are estimated from a radial projection of the absolute values of the Fourier transform image. A simple one-dimensional correlation can be performed to compute the rotation angle from the projections for two images. Shifts can then be estimated from the linear phase difference between the rotation corrected images. After registration, a high resolution image is reconstructed using Delaunay triangulation and a bicubic interpolation method. The performance of this method is shown and compared to other image registration methods in both simulations and practical experiments on real images.

Pairwise alignment of the images is not possible anymore if the sampling frequency is lower than the maximum signal frequency. All the images have to be aligned jointly using the information present in the aliased frequencies. Using the multichannel sampling description, several methods for this registration are designed. The motion model in this case is limited to horizontal and vertical shifts in a plane parallel to the image plane.

Table 7.1: Comparison of the different methods presented in this thesis.

	partially aliased (Chapter 3)	rank-based (Section 5.1)	projection-based (Section 5.2)	Gröbner basis (Chapter 6)
subsampling	< 2	any	any	any
signal model	Fourier	Fourier	Hilbert space	polynomial/Fourier
motion	shift & rotation	shift	shift	1D shift
complexity	low	high	high	very high
applications	real images	small images	small images	1D signals

The first method uses the Fourier domain description of a signal as a sum of the different aliased components of the spectrum. An objective function is obtained in which the smallest nonzero singular value of a matrix has to be computed. In the second method, the samples are written as a linear combination of sampled basis functions. In other words, the sample vector should be in the subspace spanned by the sampled basis functions. The objective function therefore computes the difference between the sample vector and its projection onto the basis function subspace. This subspace varies as a function of the offsets. This method can be applied more generally to any type of signals in a finite-dimensional Hilbert space. The objective functions obtained in both methods are very flat functions with many local minima. This makes their minimization a non-trivial operation, for which different heuristics are presented. Both methods are tested in simulations on one and two-dimensional signals, and show good results.

In many practical cases, like for example with bandlimited signals, the multichannel sampling problem can be written as a set of polynomial equations. These equations are linear in the signal coefficients, and higher order polynomials in the offsets. An elegant solution for such a set of equations can be computed using Buchberger's algorithm, which is the equivalent of Gaussian elimination for a set of polynomial equations. This results in a Gröbner basis for the polynomial ideal. The solutions can then be easily derived from this Gröbner basis using back substitution. A version for noisy measurements, and therefore only approximate equations, is obtained by computing Gröbner bases for a number of critical subsets of the total set of equations. The offsets can then be obtained as a weighted average of the results for the different subsets, and the signal parameters can be computed using back substitution. Due to the high computational complexity of algorithms to compute Gröbner bases, this method is only applied to (small) one-dimensional signals.

All the journal papers written during this PhD research, as well as this thesis itself is entirely reproducible. All the results and figures can be reproduced using the Matlab/Mathematica code that is available online. We argue that it is very important for scientific research to be performed in a reproducible manner. This gives an easy way of checking results, and gives everyone access to existing algorithms and results.

7.2 Future research

The limiting factor in the super-resolution methods for totally aliased images is their computational complexity. As all the images need to be registered jointly, they are inherently more complex than the methods that allow pairwise registration. However, the presented methods are also computationally intensive due to the complex minimization function and their large matrix sizes. Other minimization procedures should therefore be examined, like for example coupled simulated annealing [130], or more efficient line search algorithms to minimize the objective functions. Similarly, both the decomposition in orthogonal subspaces presented for the projection-based method, and a decomposition of an image in smaller blocks could reduce the sizes of the matrices. As the main part of these algorithms consists of evaluating the objective function for a large number of offset values, they can be also solved using parallel computing methods (such as the coupled simulated annealing method presented above).

Next to standard digital cameras, it would also be interesting to apply our algorithms to other types of capturing devices. For example in aerial or satellite imaging, the motion between images can often be considered to be planar [3, 97]. Super-resolution would be very interesting in satellite imaging, as it is typically impossible to replace a sensor by a higher resolution sensor, while it is relatively easy to capture multiple images in multiple passes over a region. Super-resolution methods could also be used to measure small objects with higher precision than what can be obtained using a single image. This can be applied to microscopy applications [131] and visual inspection methods.

Super-resolution imaging methods could also be combined with other techniques to improve the image quality further. As most images taken with a digital camera are mosaiced, we could perform our super-resolution algorithms prior to the demosaicing phase (which might introduce interpolation artifacts), or in combination with a demosaicing algorithm [29]. Similarly, we could combine super-resolution and high dynamic range imaging methods to increase both resolution and dynamic range from a set of images taken with different exposure times [72].

Finally, if images are captured from different points of view, they can be used to reconstruct the plenoptic function [1], a seven-dimensional function representing all the images taken from any arbitrary point of view. It would be interesting to look into the sampling and interpolation of the plenoptic function using the intuition from super-resolution imaging.

Appendix A

Reproducible Research

In this appendix, we present the main ideas of reproducible research, and we discuss its importance. The motivation for reproducible research can be well described using a quote from Buckheit and Donoho [14]:

> *An article about computational science in a scientific publication is* not *the scholarship itself, it is merely* advertising *of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.*

A reproducible publication is defined as a publication where all the results (figures, tables, etc.) can be reproduced using the data and code that is provided with the publication. The code should come with sufficient documentation and some simple commands such that a reader can easily reproduce the results.

This may all sound very trivial, and in discussions with colleagues, there is a general agreement that this is how research should be performed [7]. Wikipedia also states that "reproducibility is one of the main principles of the scientific method, and refers to the ability of a test or experiment to be accurately reproduced, or replicated" [128]. However, in practice, only few examples are available today. Vinod has performed a study in a prestigious econometrics journal, which showed that 70 % of the articles were not reproducible [122]. Such studies do not exist for signal processing, but we believe that the results would be similar. Making articles reproducible indeed requires a certain investment in time. However, we think that it is worth the investment.

Unfortunately, much of today's research could still be described as "single person, single image, at a single moment". This means that often, due to time pressure, the results can only be made by a single person at a specific moment in time, and are tested on a single image. We illustrate this with some examples that are very familiar to most of us. Some examples did really happen, while others are fictional, but close to reality.

Single person Most often, papers are written with a strict deadline. There is typically no time (or interest?) to document code carefully. As a consequence, only the author of the code (and the paper) knows how to run the code, which parameters to apply to get a good result, etc. Once this person leaves the lab, this knowledge is gone, and even if a copy of the code is still available, no one really knows how to use it. Or similarly, as not all the details of the implementation can be described in the corresponding paper due to space limitations, a colleague at another institution is not able to implement the algorithm himself and test the results.

Single image Many image processing papers show the results of the described algorithms only on a single image, typically the “Lena” image. While one can argue that it is good for comparison to use a standard set of images, results on a single image are not very convincing. Does the algorithm work well only on this image? Is the selected image a good choice to compare such and such features?

Single moment In many cases, the situation is even much worse than the first example. When a figure has to be redrawn for a thesis, or for a revision of the paper, even the author himself can not find back the parameters that produced such a nice result. The tedious work of tuning parameters then starts again, or, when the deadline is too close, the idea is abandoned, and the same figure is reused.

Some further motivations for reproducible research are given in Section A.1, and an overview of reproducible research initiatives in different fields is given in Section A.2. The procedure we follow to make our research reproducible is described in Section A.3, and some results are discussed in Section A.4. All the research presented in this thesis is reproducible and follows this procedure.

A.1 Motivation

Our motivation to make research reproducible is twofold. First of all, we want to make our research more accessible to other people. With a reproducible paper, it is possible for a reader to analyze the program code, make changes, and experiment himself with the code, without having to do the entire implementation. It also allows colleagues to take this work up later and start from the state of the art, or to compare their methods with existing methods. Second, this also helps ourselves by greatly simplifying the work when we need to take something up again after a few months/years, for a revision of a paper or when writing a thesis.

As a side effect, this also allows a more thorough review of publications. Reviewers or journal editors could check that the figures and other results shown in the paper are obtained using the included code. They could even check the code to see that the implementations correspond to what is described in the paper. This would avoid most problems with unintended errors or even falsified results. A lot of media attention was recently drawn to the verification of research results published in several journals [126, 129]. The Journal of Cell Biology started in 2002 to conduct tests on the images in their publications to verify if they are manipulated [81, 91, 123]. They say that about 25% of the papers have some problems with the images, with 1% being really fraudulent. If a standardized method of making research reproducible is available, publishers could reproduce all the results in a publication to easily verify their correctness.

A.2 Related work

We got most of our ideas about reproducible research from Jon Claerbout and his colleagues at the Stanford Exploration Project [95]. They initiated (to our knowledge) the discussion about reproducible research, and apply this to their geophysics research since 1991. They have developed makefiles to build, destroy and clean all their results. Using such a makefile, one can easily recompute all the figures with the command `make build`. A disadvantage is that such makefiles only work under Unix/linux. Claerbout makes a distinction between easily reproducible results, which are reproducible in less than ten minutes, conditionally reproducible results, requiring more than ten minutes or some specific data or software, and non-reproducible results, which are typically hand-drawn illustrations or scanned figures.

Reproducible research is applied to wavelets by Dave Donoho and his colleagues at the Stanford statistics department [14]. They developed a Matlab library of routines for wavelet analysis, wavelet packets, cosine packets and matching pursuit. Using this library, it is possible to reproduce the results from several of their publications.

It is interesting to see how a similar concept was presented by Donald Knuth as literate programming [55, 62] in 1984. Instead of starting from the research results and publications, and adding software to it to make things reproducible, he presented literate programming as a way of improving significantly the documentation of programs:

> *Instead of imagining that our main task is to instruct a* computer *what to do, let us concentrate rather on explaining to* human beings *what we want a computer to do.*

Knuth developed the WEB system for literate programming. A piece of software is then a WEB document containing both LaTeX and Pascal code. From such a document, both a LaTeX file and a Pascal program can be generated, which can be compiled separately.

This WEB system has later been adapted to different other languages than Pascal. It seems to have drawn most attention in combination with the statistical software R, for which a special package Sweave was developed [37, 59]. Instead of inserting a figure in a Sweave article, the author simply inserts the R code necessary to produce the figure. The figure itself is then automatically created and inserted in the LaTeX document when the Sweave document is processed. An example of the use of R and Sweave in bioinformatics is given by Gentleman [36].

Images are typically rather easy to obtain, and therefore it does not pose a big problem to make them freely available online. However, with other datasets such as results from large medical studies or sensor network measurements, this is less trivial, and other modalities might be needed. Peng et al. discuss some other types of making such datasets available in epidemiologic research [82].

Although reproducible research is not widely used yet, its ideas have appeared in different domains, mainly centered around statistics. It is applied to neurophysiological data analysis by Pouzat [85], to econometrics by Koenker [56] and Vinod [122] and to epidemiology by Peng et al. [82]. This list of application domains is by no means exhaustive, but gives a good indication of the wide range of fields where reproducible research is applied and/or discussed.

A.3 Practical method

In practice, all our software is developed in Matlab or Mathematica. Although software packages such as Matlab and Mathematica are not freely available, and thus restrict the reproducibility of the results, we consider them as quite standard packages to which most researchers have access. Matlab/Mathematica files were created such that all the figures and tables can be reproduced by running the *figure_x* or *table_y* files. Of course, all the data necessary to reproduce the results are also added. In most cases the data themselves are not reproducible, so those should be added as complete and well documented as possible. Next to the documentation per function in each file, a Readme file and a Contents.m file are also added. These files give an overview of all the files and their behavior. The code is distributed under the GNU Public License [38].

A web page is created containing the title, authors, abstract and a PDF version of the publication. Next, the code and data are added in a compressed archive. They are typically put in two separate files, as the data can become quite large. The computer configurations on which the code was tested are also added for reference, and an e-mail address is given for comments and remarks. Finally, some optional information can be given: full resolution images, reference list, etc. A list of these items is given in Table A.1. The reproducible papers published at LCAV are available online at `http://lcavwww.epfl.ch/reproducible_research`.

Table A.1: Required items for a reproducible publication.

1. Title of the publication
2. Authors
3. Abstract
4. PDF version
5. Full reference to latest publication status
6. Zipped archive with code and data
   a) Matlab/Mathematica/... code
   b) Data necessary to reproduce the results
   c) Readme file (and Contents.m file for Matlab code)
   d) GPL file with the GNU Public License
7. List of tested configurations or system requirements
8. E-mail address for comments and remarks
9. Optional information:
   a) Full resolution figures if certain details might be lost when printing
   b) List of references (if possible with url pointing to the paper online)
   c) . . .

A.4 Results

It is very difficult to quantify the advantage of making papers reproducible. The merit of having publications available online has been studied by Lawrence [58]. He showed that papers that are available online are cited about three times more often than papers that are not online. We believe that a similar argument holds for reproducible papers as compared to non-reproducible papers. This is difficult to study, as the number of reproducible papers available online is still quite small.

However, we can already say qualitatively that reproducible research has certainly been worth the effort for us. First of all, it has largely simplified work for ourselves in picking up previous work again. Readers downloading our software have also sent us very grateful e-mails. Moreover, these comments have already allowed us to remove some errors in the Matlab implementation of one of the methods. The online availability of our code has also facilitated some interesting collaborations.

Note that we are only talking about making journal publications reproducible. We consider that for conference publications, this is not worth the effort. Conference papers typically describe research that is still ongoing, and in that sense it might be premature to develop software environments for those. Also, conference papers often have sharper deadlines than journal papers, making it more difficult to add some time for nicely putting the code together.

A.5 Conclusions and future work

All the journal publications written during this PhD thesis work [114, 119], as well as this thesis are reproducible, and follow the guidelines described above. Making publications reproducible requires some effort, but we believe the benefits largely compensate the effort.

In future work, it would be interesting to study if we can use open source alternatives such as Octave [25] or Scilab [44] instead of (expensive) commercial programs like Matlab. Also, as Buckheit and Donoho describe, in an ideal world one could click on figures or tables to connect to the source code and the computational environment that produced the results [14]. A reader could then modify the source code to see what happens with different input parameters, and the figure would be displayed for comparison. We will study how close we can get to such an ideal world using existing applications like LaTeX, literate programming, hyperlinks, etc.

Bibliography

[1] E. H. Adelson and J. R. Bergen, *The Plenoptic Function and the Elements of Early Vision.* MIT Press, 1991, pp. 3–20.

[2] S. Alliney, "Digital Analysis of Rotated Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 499–504, May 1993.

[3] A. Almansa, "Echantillonnage, interpolation et détection. Applications en imagerie satellitaire." Ph.D. dissertation, Ecole Normale Supérieure de Cachan, Dec. 2002.

[4] P. E. Anuta, "Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform techniques," *IEEE Transactions on Geoscience Electronics*, vol. 8, no. 4, pp. 353–368, Oct. 1970.

[5] 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, Sept. 2002.

[6] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, Dec. 1996.

[7] M. Barni and F. Perez-Gonzalez, "Pushing Science into Signal Processing," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 119–120, July 2005.

[8] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," *Proc. Second European Conference on Computer Vision, Lecture Notes in Computer Science*, pp. 237–252, May 1992.

[9] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences - a comprehensive review with directions for future research," University of Notre Dame, Tech. Rep., 1998.

[10] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, 1992.

[11] B. Buchberger, "An Algorithm for Finding the Bases Elements of the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal (German)," Ph.D. dissertation, University of Innsbruck (Austria), 1965.

[12] B. Buchberger, “Gröbner Bases: A Short Introduction for Systems Theorists,” *Lecture Notes in Computer Science*, vol. 2178, pp. 1–19, Feb. 2001.

[13] B. Buchberger, “Groebner Bases and Systems Theory,” *Multidimensional Systems and Signal Processing*, no. 12, pp. 223–251, 2001.

[14] J. B. Buckheit and D. L. Donoho, “WaveLab and Reproducible Research,” Dept. of Statistics, Stanford University, Tech. Rep. 474, 1995. [Online]. Available: http://www-stat.stanford.edu/~donoho/Reports/1995/wavelab.pdf

[15] S. C. Cain, M. M. Hayat, and E. E. Armstrong, “Projection-Based Image Registration in the Presence of Fixed-Pattern Noise,” *IEEE Transactions on Image Processing*, vol. 10, no. 12, pp. 1860–1872, Dec. 2001.

[16] D. Capel, *Image Mosaicing and Super-resolution*, ser. Distinguished Dissertations. Springer, 2004.

[17] D. Capel and A. Zisserman, “Computer vision applied to super resolution,” *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 75–86, May 2003.

[18] C. Charoenlarpnopparut, “Gröbner Bases in Multidimensional Systems and Signal Processing,” Ph.D. dissertation, Pennsylvania State University, May 2000.

[19] S. Chaudhuri, Ed., *Super-resolution Imaging.* Kluwer Academic Publishers, 2001.

[20] S. Chaudhuri and J. Manjunath, *Motion-Free Super-Resolution.* Springer, 2005.

[21] D. A. Cox, J. B. Little, and D. O’Shea, *Ideals, Varieties and Algorithms*, 2nd ed. Springer-Verlag, 1996.

[22] V. Divi and G. Wornell, “Signal recovery in time-interleaved analog-to-digital converters,” in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2004, pp. 593–596.

[23] T. W. Dubé, “The Structure of Polynomial Ideals and Gröbner Bases,” *SIAM Journal on Computing*, vol. 19, no. 4, pp. 750–773, Aug. 1990.

[24] D. S. Early and D. G. Long, “Image reconstruction and enhanced resolution imaging from irregular samples,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 291–302, Feb. 2001.

[25] J. W. Eaton. (1994) Gnu octave. [Online]. Available: http://www.octave.org

[26] 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.

[27] M. Elad and A. Feuer, “Superresolution Restoration of an Image Sequence: Adaptive Filtering Approach,” *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 387–395, Mar. 1999.

[28] M. Elad and Y. Hel-Or, “A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space-Invariant Blur,” *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1187–1193, Aug. 2001.

[29] S. Farsiu, M. Elad, and P. Milanfar, “Multi-Frame Demosaicing and Super-Resolution of Color Images,” *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 141–159, Jan. 2006.

[30] S. Farsiu, D. Robinson, and P. Milanfar. (2004) MDSP resolution enhancement software. [Online]. Available: http://www.soe.ucsc.edu/~milanfar/SR-Software.htm

[31] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[32] J.-C. Faugère, F. Moreau de Saint-Martin, and F. Rouillier, “Design of Regular Nonseparable Bidimensional Wavelets Using Gröbner Basis Techniques,” *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 845–856, Apr. 1998.

[33] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[34] H. Foroosh, J. B. Zerubia, and M. Berthod, “Extension of Phase Correlation to Subpixel Registration,” *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 188–200, Mar. 2002.

[35] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, “Recovering motion fields: An evaluation of eight optical flow algorithms,” in *Proceedings of the British Machine Vision Conference*, Sept. 1998, pp. 454–460.

[36] R. Gentleman, “Reproducible Research: a Bioinformatics Case Study,” *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, pp. 1–23, 2005.

[37] C. Geyer. (2006) An sweave demo. [Online]. Available: http://www.stat.umn.edu/~charlie/Sweave

[38] GNU. (1991) GNU General Public License. [Online]. Available: http://www.gnu.org/copyleft/gpl.html

[39] N. Goldberg, A. Feuer, and G. C. Goodwin, “Super-resolution reconstruction using spatio-temporal filtering,” *Journal of Visual Communication and Image Representation*, vol. 14, no. 4, pp. 508–525, Dec. 2003.

[40] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.

[41] T. C. Hofner, “Boost your sampling rate with time-interleaved data converters,” *Sensors Magazine*, February 2001.

[42] W. S. Hoge, “A subspace identification extension to the phase correlation method,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 277–280, Feb. 2003.

[43] R. J. Holt, T. S. Huang, and A. N. Netravali, “Algebraic Methods for Image Processing and Computer Vision,” *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 976–986, June 1996.

[44] INRIA. (2003) Scilab, a free scientific software package. [Online]. Available: http://www.scilab.org

[45] International Organization for Standardization, “ISO 12233:2000 - Photography - Electronic still picture cameras - Resolution measurements,” 2000.

[46] 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.

[47] M. Irani, B. Rousso, and S. Peleg, “Computing occluding and transparent motions,” *International Journal of Computer Vision*, vol. 12, no. 1, pp. 5–16, Feb. 1994.

[48] M. V. Joshi, S. Chaudhuri, and R. Panuganti, “Super-resolution imaging: use of zoom as a cue,” *Image and Vision Computing*, vol. 22, no. 14, pp. 1185–1196, Dec. 2004.

[49] T. Kalker, H. Park, and M. Vetterli, “Groebner Basis Techniques in Multidimensional Multirate Systems,” in *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 4, 1995, pp. 2121–2124.

[50] M. G. Kang and S. Chaudhuri, Eds., *IEEE Signal Processing Magazine, special issue on super-resolution*, vol. 20, no. 3, May 2003.

[51] D. Keren, S. Peleg, and R. Brada, “Image sequence enhancement using sub-pixel displacement,” in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, June 1988, pp. 742–746.

[52] M.-H. Kim and S. Sutherland, “Polynomial Root-Finding Algorithms and Branched Covers,” *SIAM Journal of Computing*, vol. 23, pp. 415–436, 1994.

[53] 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.

[54] S. P. Kim and W.-Y. Su, “Subpixel accuracy image registration by spectrum cancellation,” in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, Apr. 1993, pp. 153–156.

[55] D. E. Knuth, “Literate Programming,” *The Computer Journal*, vol. 27, no. 2, pp. 97–111, May 1984.

[56] R. Koenker, “Reproducible Econometric Research,” Department of Econometrics, University of Illinois, Urbana-Champaign, IL, Tech. Rep., 1996.

[57] V. Kotel’nikov, “On the transmission capacity of ”ether” and wire in electrocommunications,” *Izd. Red. Upr. Zvyazzi RKKA (Moscow)*, 1933.

[58] S. Lawrence, “Free online availability substantially increases a paper’s impact,” *Nature*, vol. 411, no. 6837, p. 521, 2001.

[59] F. Leisch, “Sweave: Dynamic generation of statistical reports using literate data analysis,” in *Compstat 2002 — Proceedings in Computational Statistics*, W. Härdle and B. Rönz, Eds. Physica Verlag, Heidelberg, 2002, pp. 575–580, iSBN 3-7908-1517-9. [Online]. Available: http://www.ci.tuwien.ac.at/~leisch/Sweave

[60] W. Li, “A bibliography on 1/f noise,” 2006. [Online]. Available: http://www.nslij-genetics.org/wli/1fnoise/

[61] Z. Lin and H.-Y. Sum, “Fundamental Limits of Reconstruction-Based Superresolution Algorithms under Local Translation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, Jan. 2004.

[62] literateprogramming.com. (2006) Literate programming. [Online]. Available: http://www.literateprogramming.com

[63] L. Lucchese and G. M. Cortelazzo, “A noise-robust frequency domain technique for estimating planar roto-translations,” *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1769–1786, June 2000.

[64] D. G. Luenberger, *Optimization by Vector Space Methods.* John Wiley & Sons, Inc., 1969.

[65] H. D. Lüke, “The origins of the sampling theorem,” *IEEE Communications Magazine*, pp. 106–108, Apr. 1999.

[66] B. Marcel, M. Briot, and R. Murrieta, “Calcul de translation et rotation par la transformation de Fourier,” *Traitement du Signal*, vol. 14, no. 2, pp. 135–149, 1997.

[67] F. Marvasti, Ed., *Nonuniform Sampling: Theory and Practice.* Springer, 2001.

[68] P. Marziliano, “Sampling innovations,” Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2001.

[69] P. Marziliano and M. Vetterli, “Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations,” *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3462–3471, Dec. 2000.

[70] Mathworks (The), “Matlab function reference: griddata,” 2006. [Online]. Available: http://www.mathworks.com/access/helpdesk/help/techdoc/ref/griddata.html

[71] E. W. Mayr, “Some Complexity Results for Polynomial Ideals,” *Journal of Complexity*, vol. 13, pp. 303–325, 1997.

[72] S. G. Narasimhan and S. K. Nayar, “Enhancing resolution along multiple imaging dimensions using assorted pixels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 518–530, April 2005.

[73] M. Ng, T. Chan, M. G. Kang, and P. Milanfar, Eds., *EURASIP journal on applied signal processing, special issue on super-resolution*, 2006.

[74] N. Nguyen and P. Milanfar, “A Wavelet-Based Interpolation-Restoration Method for Superresolution,” *Circuits, Systems, and Signal Processing, Special Issue on Advanced Signal and Image Reconstruction*, vol. 19, no. 4, pp. 321–338, Aug. 2000.

[75] H. Nyquist, “Certain topics in telegraph transmission theory,” *Trans. Amer. Inst. Elect. Eng.*, vol. 47, pp. 617–644, 1928.

[76] A. Papoulis, *Systems and Transforms with Applications in Optics*. McGraw-Hill Book Company, 1968.

[77] A. Papoulis, “Generalized sampling expansion,” *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, Nov. 1977.

[78] H. Park, T. Kalker, and M. Vetterli, “Groebner bases and multidimensional FIR multirate systems,” *Multidimensional Systems and Signal Processing*, vol. 8, pp. 11–30, 1997. [Online]. Available: http://infoscience.epfl.ch/getfile.py?mode=best&recid=33876

[79] 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.

[80] A. J. Patti, M. I. Sezan, and A. M. Tekalp, “Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time,” *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1064–1076, Aug. 1997.

[81] H. Pearson, “Image manipulation: CSI: cell biology,” *Nature*, vol. 434, pp. 952–953, Apr. 2005.

[82] R. D. Peng, F. Dominici, and S. L. Zeger, “Reproducible Epidemiologic Research,” *American Journal of Epidemiology*, 2006.

[83] T. Q. Pham, M. Bezuijen, L. J. van Vliet, K. Schutte, and C. L. Luengo Hendriks, “Performance of Optimal Registration Estimators,” in *Proc. SPIE Visual Information Processing XIV*, vol. 5817, May 2005, pp. 133–144.

[84] 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*, vol. 2006, 2006.

[85] C. Pouzat. (2006) Reproducible neurophysiological data analysis? [Online]. Available: http://www.biomedicale.univ-paris5.fr/physcerv/C_Pouzat/Compendium.html

[86] D. Rajan, S. Chaudhuri, and M. V. Joshi, “Multi-objective super-resolution: Concepts and examples,” *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 49–61, May 2003.

[87] B. S. Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation and scale-invariant image registration,” *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, Aug. 1996.

[88] D. Robinson and P. Milanfar, “Fast Local and Global Projection-Based Methods for Affine Motion Estimation,” *Journal of Mathematical Imaging and Vision*, vol. 18, pp. 35–54, 2003.

[89] D. Robinson and P. Milanfar, “Fundamental performance limits in image registration,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1185–1199, Sept. 2004.

[90] D. Robinson and P. Milanfar, “Statistical performance analysis of super-resolution,” *accepted to IEEE Transactions on Image Processing*, 2005.

[91] M. Rossner and K. M. Yamada, “What’s in a picture? The temptation of image manipulation,” *Journal of Cell Biology*, vol. 166, no. 1, pp. 11–15, July 2004.

[92] A. Ruhe and P. A. Wedin, “Algorithms for Separable Nonlinear Least Squares Problems,” *SIAM Review*, vol. 22, no. 3, pp. 318–337, July 1980.

[93] L. Sbaiz, P. Vandewalle, and M. Vetterli, “Groebner Basis Methods for Multichannel Sampling with Unknown Offsets,” *submitted to Applied and Computational Harmonic Analysis*, 2006. [Reproducible]. Available: http://lcavwww.epfl.ch/reproducible_research/SbaizVV06

[94] R. R. Schultz, L. Meng, and R. L. Stevenson, “Subpixel motion estimation for super-resolution image sequence enhancement,” *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 38–50, Mar. 1998.

[95] M. Schwab, M. Karrenbach, and J. Claerbout, “Making scientific computations reproducible,” *Computing in Science & Engineering*, vol. 2, no. 6, pp. 61–67, Nov. 2000.

[96] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[97] H. Shekarforoush, M. Berthod, J. Zerubia, and M. Werman, “Sub-pixel Bayesian Estimation of Albedo and Height,” *International Journal of Computer Vision*, vol. 19, no. 3, pp. 289–330, 1996.

[98] K. Shirayanagi, “Floating point Gröbner bases,” *Mathematics and Computers in Simulation*, vol. 42, pp. 509–528, 1996.

[99] H. J. Stetter, “Stabilization of Polynomial Systems Solving with Groebner Bases,” in *Proceedings international symposium on Symbolic and algebraic computation (ISSAC)*, 1997, pp. 117–124.

[100] H. S. Stone, M. T. Orchard, E.-C. Chang, and S. A. Martucci, “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, October 2001.

[101] G. Strang, *Linear Algebra and Its Applications*, 3rd ed. Saunders College Publishing, 1988.

[102] T. Strohmer, “Computationally attractive reconstruction of bandlimited images from irregular samples,” *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 540–548, Apr. 1997.

[103] 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 *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, Mar. 1992, pp. 169–172.

[104] B. C. Tom and A. K. Katsaggelos, “Resolution enhancement of monochrome and color video using motion compensation,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 278–287, Feb. 2001.

[105] A. Torralba and A. Oliva, “Statistics of natural image categories,” *Network: Computation in Neural Systems*, vol. 14, no. 3, pp. 391–412, Aug. 2003.

[106] C. Traverso and A. Zanoni, “Numerical stability and stabilization of Groebner basis computation,” in *ISSAC ’02: Proceedings of the 2002 international symposium on Symbolic and algebraic computation.* New York, NY, USA: ACM Press, 2002, pp. 262–269.

[107] R. Y. Tsai and T. S. Huang, “Multiframe image restoration and registration,” in *Advances in Computer Vision and Image Processing*, T. S. Huang, Ed. JAI Press, 1984, vol. 1, pp. 317–339.

[108] M. Unser, “Sampling—50 Years after Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.

[109] M. Unser and J. Zerubia, “Generalized sampling: Stability and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 2941–2950, Dec. 1997.

[110] M. Unser and J. Zerubia, “Generalized sampling without bandlimiting constraints,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Apr. 1997, pp. 2113–2116.

[111] M. van Ginkel, G. van Kempen, C. Luengo, and L. J. van Vliet. (2006) Dipimage, a scientific image processing toolbox for matlab. [Online]. Available: http://www.ph.tn.tudelft.nl/DIPlib/

[112] P. Vandewalle, L. Sbaiz, S. Süsstrunk, and M. Vetterli, “Registration of Aliased Images for Super-Resolution Imaging,” in *SPIE/IS&T Visual Communications and Image Processing Conference*, vol. 6077, Jan. 2006, pp. 13–23, (invited paper).

[113] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli, “How to Take Advantage of Aliasing in Bandlimited Signals,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, May 2004, pp. 948–951.

[114] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli, “Super-Resolution from Unregistered and Totally Aliased Signals using Subspace Methods,” *accepted to IEEE Transactions on Signal Processing*, 2006. [Reproducible]. Available: http://lcavwww.epfl.ch/reproducible_research/VandewalleSVV06

[115] P. Vandewalle, L. Sbaiz, and M. Vetterli, “Signal Reconstruction from Multiple Unregistered Sets of Samples using Groebner Bases,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May 2006, pp. 604–607.

[116] P. Vandewalle, L. Sbaiz, M. Vetterli, and S. Süsstrunk, “Super-Resolution from Highly Undersampled Images,” in *IEEE International Conference on Image Processing*, vol. 1, Sept. 2005, pp. 889–892.

[117] P. Vandewalle, S. Süsstrunk, and M. Vetterli, “Superresolution images reconstructed from aliased images,” in *SPIE/IS&T Visual Communication and Image Processing Conference*, T. Ebrahimi and T. Sikora, Eds., vol. 5150, July 2003, pp. 1398–1405.

[118] P. Vandewalle, S. Süsstrunk, and M. Vetterli, “Double resolution from a set of aliased images,” in *Proc. SPIE/IS&T Electronic Imaging 2004: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V*, vol. 5301, Jan. 2004, pp. 374–382.

[119] P. Vandewalle, S. Süsstrunk, and M. Vetterli, “A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution,” *accepted to EURASIP Journal on Applied Signal Processing, Special Issue on Super-Resolution Imaging*, 2005. [Reproducible]. Available: http://lcavwww.epfl.ch/reproducible_research/VandewalleSV05

[120] P. Vandewalle, P. Zbinden, S. Süsstrunk, and M. Vetterli. (2005) Super-resolution software. [Online]. Available: http://lcavwww.epfl.ch/software/superresolution

[121] V. Velisavljevic, “Directionlets,” Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2005, ph.D. Thesis EPFL 3358 (2005), School of Computer and Communication Sciences. [Online]. Available: http://library.epfl.ch/theses/?nr=3358

[122] H. D. Vinod, "Care and feeding of reproducible econometrics," *Journal of Econometrics*, vol. 100, no. 1, pp. 87–88, Jan. 2001.

[123] N. Wade, "It May Look Authentic; Here's How to Tell It Isn't," *New York Times*, 24 Jan. 2006.

[124] E. T. Whittaker, "On the functions which are represented by the expansion of interpolating theory," *Proc. R. Soc. Edinburgh*, vol. 35, pp. 181–194, 1915.

[125] Wikipedia, "Projection operator — wikipedia, the free encyclopedia," 2005, [Online; accessed 13-March-2006]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Projection_operator&oldid=28924331

[126] Wikipedia, "Journals tackle scientific fraud — wikinews," 2006, [Online; accessed 19-April-2006]. [Online]. Available: http://en.wikinews.org/w/index.php?title=Journals_tackle_scientific_fraud\&oldid=206137

[127] Wikipedia, "Pink noise — wikipedia, the free encyclopedia," 2006, [Online; accessed 14-March-2006]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Pink_noise&oldid=43310475

[128] Wikipedia, "Reproducibility — wikipedia, the free encyclopedia," 2006, [Online; accessed 19-April-2006]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Reproducibility&oldid=48837772

[129] Wikipedia, "Scientific misconduct — wikipedia, the free encyclopedia," 2006, [Online; accessed 19-April-2006]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Scientific_misconduct&oldid=48972241

[130] S. Xavier-de-Souza, J. A. K. Suykens, J. Vandewalle, and D. Bollé, "Cooperative behavior in coupled simulated annealing processes with variance control," in *Accepted for publication in The 2006 International Symposium on Nonlinear Theory and its Applications (NOLTA2006)*, Bologna, Italy, Sept. 2006.

[131] X.-Y. Zhai, E. I. Christensen, and A. Andreasen, "Low Noise Electron Microscopy by Merging Multiple Images Digitized from Conventional Films with Reference to the Mouse Kidney," *Microscopy and Microanalysis*, vol. 12, no. 3, pp. 255–261, June 2006.

[132] J. Zhou and M. N. Do, "Multidimensional Multichannel FIR Deconvolution Using Gröbner Bases," *to be published in IEEE Transactions on Image Processing*, 2006.

[133] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[134] S. Zokai and G. Wolberg, "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1422–1434, Oct. 2005.

[135] A. Zomet, A. Rav-Acha, and S. Peleg, “Robust super-resolution,” in *Proceedings International Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2001.

Curriculum Vitae

Patrick Vandewalle

Chemin de Maillefer 3
1018 Lausanne
Switzerland

Audiovisual Communications Laboratory (LCAV)
School of Computer and Communication Sciences (I&C)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
BC318 - Station 14
1015 Lausanne, Switzerland

Email: `Patrick.Vandewalle@epfl.ch`
Web: `http://lcavwww.epfl.ch/~patrick`

Personal

Date of birth: June 30, 1978.
Nationality: Belgian and USA.
Civil status: Married to Els Coorevits.

Education

2002 - 2006 **Ph.D.** in Computer and Communication Sciences, Audiovisual Communications Laboratory (LCAV), School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

1996 - 2001 **M.Sc.** in Electrical Engineering, Katholieke Universiteit Leuven, Belgium.

Professional Experience

2002 - present **Research and Teaching Assistant**, Audiovisual Communications Laboratory (LCAV), I&C, EPFL, Switzerland.
Research in super-resolution imaging
Assisted students in undergraduate and graduate courses
Supervised students for semester, thesis and doctoral school projects

2004	**Internship at Foveon, Inc.**, Santa Clara, California, USA Research in image interpolation methods
2001 - 2002	**Research Assistant**, Medical Image Computing (MIC/PSI), ESAT, K. U. Leuven, Belgium Research in soft tissue modelling for maxillofacial surgery using finite element methods
2000	**Internship at LCAV, EPFL and Phonak**, Switzerland Stability and feedback compensation in hearing aids using all-pass filters
1999	**Internship at Barco Machine Vision**, Belgium Analysis of the difference between single and double source X-ray machines used for visual inspection

Publications

Journal papers

1. L. Sbaiz, P. Vandewalle and M. Vetterli, "Groebner Basis Methods for Multichannel Sampling with Unknown Offsets," submitted to *Applied and Computational Harmonic Analysis*, 2006.

2. P. Vandewalle, L. Sbaiz, J. Vandewalle and M. Vetterli, "Super-Resolution from Unregistered and Totally Aliased Signals using Subspace Methods," accepted to *IEEE Transaction on Signal Processing*, 2006.

3. P. Vandewalle, S. Süsstrunk and M. Vetterli, "A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution," accepted to *EURASIP Journal on Applied Signal Processing* (special issue on super-resolution), 2005.

Conference papers

1. F. Volken, J. Terrier and P. Vandewalle, "Automatic Red-Eye Removal based on Sclera and Skin Tone Detection," *Proc. IS&T Third European Conference on Colour in Graphics, Imaging and Vision (CGIV)*, pp. 359-364, 2006.

2. P. Vandewalle, L. Sbaiz and M. Vetterli, "Signal Reconstruction from Multiple Unregistered Sets of Samples using Groebner Bases," accepted to *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 3, pp. 604-607, 2006.

3. P. Vandewalle, L. Sbaiz, S. Süsstrunk and M. Vetterli, "Registration of Aliased Images for Super-Resolution Imaging," *Proc. SPIE/IS&T Visual Communications and Image Processing Conference*, Vol. 6077, pp. 13-23, 2006.

4. P. Vandewalle, L. Sbaiz, M. Vetterli and S. Süsstrunk, "Super-Resolution from Highly Undersampled Images," *Proc. IEEE International Conference on Image Processing*, Vol. 1, pp. 889-892, 2005.

5. P. Vandewalle, L. Sbaiz, J. Vandewalle and M. Vetterli, “How to Take Advantage of Aliasing in Bandlimited Signals,” *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 3, pp. 948-951, 2004.

6. P. Vandewalle, S. Süsstrunk and M. Vetterli, “Double Resolution from a Set of Aliased Images,” *Proc. SPIE/IS&T Electronic Imaging 2004: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V*, Vol. 5301, pp. 374-382, 2004.

7. P. Vandewalle, S. Süsstrunk and M. Vetterli, “Superresolution Images Reconstructed from Aliased Images,” *Proc. SPIE/IS&T Visual Communications and Image Processing Conference*, Vol. 5150, pp. 1398-1405, 2003.

8. P. Vandewalle, F. Schutyser, J. Van Cleynenbreugel and P. Suetens, “Modelling of Facial Soft Tissue Growth for Maxillofacial Surgery Planning Environments,” *Lecture Notes in Computer Science* (Proc. International Symposium on Surgery Simulation and Soft Tissue Modeling - IS4TM 2003), Vol. 2673, pp. 27-37, 2003.

Thesis

1. P. Vandewalle, “Dereverberatietechnieken voor de Performantieverbetering van Spraakgestuurde Systemen (Speech Dereverberation Techniques for Voice-Controlled Systems),” *M.Sc. Thesis*, ESAT, Katholieke Universiteit Leuven, Belgium, 2001.