In: New Signal Processing Research
Editor: Takumi Maeda, pp. 1-32
ISBN 978-1-60456-479-2

Chapter 1

PROJECTION AND INTERPOLATION BASED TECHNIQUES FOR STRUCTURED AND IMPULSIVE NOISE FILTERING

Bishnu Lamichhane and Laura Rebollo-Neira
Aston University
Birmingham B4 7ET, United Kingdom

Abstract

In this chapter we present the relevant mathematical background to address two well defined signal and image processing problems. Namely, the problem of structured noise filtering and the problem of interpolation of missing data. The former is addressed by recourse to oblique projection based techniques whilst the latter, which can be considered equivalent to impulsive noise filtering, is tackled by appropriate interpolation methods.

1 Introduction

Structured noise filtering is a particular problem of signal separation, in which the subspaces hosting the signal components are assumed to be known and complementary. Thus, the filtering can be realized in a straightforward manner by recourse to an oblique projection onto the subspace where one of the signals belongs, and along the subspace hosting the other components. A number of signal processing applications in which this procedure is of assistance are discussed in [1].

Oblique projectors were introduced early [2, 3]. Recently there has been a renewed interest in their properties [4–6]. As a very small sample of the publications concerning signal processing application of oblique projections we could mention [7–10]. Unfortunately, given two complementary subspaces, it is not always possible to construct a numerically stable oblique projector onto one of the subspaces and along the other. If the angle between such subspaces is small, the numerical errors in the calculations are magnified and yield, thereby, an oblique projector of poor quality. In relation to the problem of structured noise filtering this may cause the failure to correctly filter the noise. Nevertheless, if the signal component one wishes to discriminate belongs to an 'unknown' subspace of the given

one, and the construction of the oblique projector onto such smaller subspace is numerically stable, the problem of discriminating the signal component can be transformed into the one of finding the right subspace. This has motivated the adaptive Oblique Matching Pursuit approach for signal separation [11], extended in [12]. The approach is effectively implemented by recursive equations for adapting oblique projectors given in [13].

Impulsive noise is characterised by a linear combination of very sharp spikes. This type of noise can be regarded as a particular case of structured noise and in some situations its filtering could be accomplished by techniques used for structured noise. However, especially for images, impulsive noise can be effectively handled as a problem of interpolation of missing data or 'image inpainting' [14, 15]. This problem has been addressed from different points of view [14, 16, 17]. Here we focus on a method based on scattered data interpolation [18,19]. The interpolation is realised by using Delaunay triangulation [20,21]. Interpolation methods based on Delaunay triangulation have been previously applied to image processing in [22, 23].

The chapter is organised as follows: For the convenience of the readers all the elementary mathematical terms used throughout the chapter are defined in Section 2. More advanced technical terms are defined in the sections where they are introduced. Section 3 discusses oblique projectors in the context of structured noise filtering. Section 4 provides the basic mathematical background relevant to polynomial and piecewise polynomial interpolation in one and two dimensions. Delaunay triangulation technique for scattered data interpolation is discussed in the same section. Such a technique is applied to the problem of filtering salt and pepper impulsive noise from an image.

List of Symbols

We shall use standard set-theoretic notation

$$\cup, \quad \cap, \quad \subseteq, \quad \subset, \quad \in$$

to denote 'union', 'intersection', 'subset of', 'proper subset of', 'belong(s) to', respectively. For the sets V_1 and V_2, the set $\{v \in V_1 : v \notin V_2\}$ is denoted by $V_1 \setminus V_2$.

The following standard notations and symbols will be used without defining them explicitly:

$\mathbb{N}$	:	set of all positive integers
$\mathbb{Z}$	:	set of all integers
$\mathbb{R}$	:	field of all real numbers
$\mathbb{C}$	:	field of all complex numbers
$\mathbb{F}$	:	field of real or complex numbers
$\Longrightarrow$	:	imply (implies)
$\Longleftrightarrow$	:	if and only if
$\rightarrow$	:	maps to

The Kronecker symbol is given by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

The characteristic function χ_S of a set S is defined as

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise.} \end{cases}$$

The absolute value of a number $a \in \mathbb{F}$ is indicated as $|a|$.

If $a \in \mathbb{R}$

$$|a| = \begin{cases} a & \text{if } a \geq 0 \\ -a & \text{if } a < 0. \end{cases}$$

If $a \in \mathbb{C}$ its complex conjugate is denoted by $\overline{a}$ and $|a|^2 = a\overline{a}$.

Note: **Bold face** is used when a terminology is defined. *Italics* are used to emphasise a terminology or statement.

2 Elementary Definitions

A **vector space** over a field $\mathbb{F}$ is a set $\mathcal{V}$ together with two operations vector addition, denoted $v+w \in \mathcal{V}$ for $v,w \in \mathcal{V}$ and scalar multiplication, denoted $av \in \mathcal{V}$ for $a \in \mathbb{F}$ and $v \in \mathcal{V}$, such that the following axioms are satisfied:

1. $v+w = w+v,\ v,w \in \mathcal{V}$.
2. $u+(v+w) = (u+v)+w,\ u,v,w \in \mathcal{V}$.
3. There exists an element $0 \in \mathcal{V}$, called the zero vector, such that $v+0 = v,\ v \in \mathcal{V}$.
4. There exists an element $\tilde{v} \in \mathcal{V}$, called the additive inverse of v, such that $v+\tilde{v} = 0,\ v \in \mathcal{V}$.
5. $a(bv) = (ab)v,\ a,b \in \mathbb{F}$ and $v \in \mathcal{V}$.
6. $a(v+w) = av+aw,\ a \in \mathbb{F}$ and $v,w \in \mathcal{V}$.
7. $(a+b)v = av+bv,\ a,b \in \mathbb{F}$ and $v \in \mathcal{V}$.
8. $1v = v,\ v \in \mathcal{V}$, where 1 denotes the multiplicative identity in $\mathbb{F}$.

The elements of a vector space are called **vectors**. A subset $\mathcal{S}$ of a vector space $\mathcal{V}$ is a **subspace** of $\mathcal{V}$ if it is a vector space with respect to the vector space operations on $\mathcal{V}$. A subspace which is a proper subset of the whole space is called a **proper subspace**. Two subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are **complementary** or **disjoint** if $\mathcal{V}_1 \cap \mathcal{V}_2 = \{0\}$.

The sum of two subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ is the subspace $\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2$ of elements $v = v_1 + v_2, v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2$. If the subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are complementary, $\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2$ is

called **direct sum** and each element $v \in \mathcal{V}$ has a unique decomposition $v = v_1 + v_2$, $v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$.

Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be vectors spaces. A mapping $\hat{A} : \mathcal{V}_1 \to \mathcal{V}_2$ is a **linear operator** if

$$\hat{A}(v+w) = \hat{A}v + \hat{A}w, \quad \hat{A}(av) = a\hat{A}v,$$

for all $v, w \in \mathcal{V}_1$ and $a \in F$. $\mathcal{V}_1$ is called the **domain** of $\hat{A}$ and $\mathcal{V}_2$ its **codomain** or **image**. If the codomain of a linear operator is a scalar field, the operator is called a **linear functional** on $\mathcal{V}_1$. The set of all linear functionals on $\mathcal{V}_1$ is called the **dual space** of $\mathcal{V}_1$.

The **adjoint** of an operator $\hat{A} : \mathcal{V}_1 \to \mathcal{V}_2$ is the unique operator $\hat{A}^*$ satisfying that

$$\langle \hat{A}g_1, g_2 \rangle = \langle g_1, \hat{A}^* g_2 \rangle.$$

If $\hat{A}^* = \hat{A}$ the operator is **self-adjoint**.

An operator $\hat{A} : \mathcal{V}_1 \to \mathcal{V}_2$ has an **inverse** if there exists $\hat{A}^{-1} : \mathcal{V}_2 \to \mathcal{V}_1$ such that

$$\hat{A}^{-1}\hat{A} = \hat{I}_{\mathcal{V}_1} \quad \text{and} \quad \hat{A}\hat{A}^{-1} = \hat{I}_{\mathcal{V}_2},$$

where $\hat{I}_{\mathcal{V}_1}$ and $\hat{I}_{\mathcal{V}_2}$ denote the identity operators in $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. By a **generalised inverse** we shall mean an operator $\hat{A}^\dagger$ satisfying the following conditions

$$\begin{aligned} \hat{A}\hat{A}^\dagger\hat{A} &= \hat{A} \\ \hat{A}^\dagger\hat{A}\hat{A}^\dagger &= \hat{A}^\dagger. \end{aligned}$$

If $v_1, \ldots, v_n$ are some elements of a vector space $\mathcal{V}$, by a **linear combination** of $v_1, \ldots, v_n$ we mean an element in $\mathcal{V}$ of the form $a_1 v_1 + \cdots + a_n v_n$, with $a_i \in \mathbb{F}, i = 1, \ldots, n$.

Let S be a subset of element of $\mathcal{V}$. The set of all *linear combinations* of elements of S is called the **span** of S and is denoted by $\operatorname{span} S$.

A subset $S = \{v_i\}_{i=1}^n$ of $\mathcal{V}$ is said to be **linearly independent** if and only if

$$a_1 v_1 + \cdots + a_n v_n = 0, \quad \Longrightarrow \quad a_i = 0, i = 1, \ldots, n.$$

A subset is said to be **linearly dependent** if it is not linearly independent.

$\mathcal{S}$ is said to be a **basis** of $\mathcal{V}$ if it is linearly independent and $\operatorname{span} S = \mathcal{V}$. The **dimension** of a finite dimensional vector space $\mathcal{V}$ is the number of elements in a basis for $\mathcal{V}$. The number of elements in a set is termed the **cardinality** of the set.

Let $\{v_i\}_{i=1}^n$ be a basis for $\mathcal{V}$. For $v = a_1 v_1 + \cdots, a_n v_n$ let $f_i(v) : \mathcal{V} \to \mathbb{F}$ be defined by

$$f_i(v) = a_i, \quad i = 1, \ldots, n.$$

Then f_i is a linear functional for each i. The linear functionals $f_1, \ldots, f_n$ are called **coordinate functionals** on $\mathcal{V}$ with respect to the basis $\{v_i\}_{i=1}^n$.

We denote the **space of polynomials** of degree $m \in \mathbb{N}$ on $\mathbb{R}$ by

$$\mathcal{P}_m(\mathbb{R}) = \Big\{ p : \, p(x) = \sum_{i=0}^{m} a_i x^i, \; x \in \mathbb{R} \Big\}.$$

Such a space is a vector space.

Given a compact interval $[a,b]$ we define a **partition** of $[a,b]$ as the finite set of points

$$\Delta = \{x_i\}_{i=0}^{n}, n \in \mathbb{N}, \quad \text{such that} \quad a = x_0 < x_1 < \cdots < x_n = b.$$

An **inner product** on a vector space $\mathcal{V}$ is a map from $\mathcal{V}$ to $\mathbb{F}$ which satisfies the following axioms

1. $\langle v,v\rangle \geq 0$, $v \in \mathcal{V}$, and $\langle v,v\rangle = 0 \Longleftrightarrow v = 0$.
2. $\langle v+w,z\rangle = \langle v,z\rangle + \langle w,z\rangle$, $v,w,z \in \mathcal{V}$.
3. $\langle v,az\rangle = a\langle v,z\rangle$, $v,z \in \mathcal{V}$ and $a \in \mathbb{F}$.
4. $\langle v,w\rangle = \overline{\langle w,v\rangle}$, $v,w \in \mathcal{V}$.

A vector space $\mathcal{V}$ together with an inner product $\langle\cdot,\cdot\rangle$ is called an **inner product space**.

Two vectors v and w in an inner product space are said to be **orthogonal** if $\langle v,w\rangle = 0$.

Two subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are orthogonal if $\langle v_1,v_2\rangle = 0$ for all $v_1 \in \mathcal{V}_1$ and $v_2 \in \mathcal{V}_1$. The sum of two orthogonal subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ is termed **orthogonal sum** and will be indicated as $\mathcal{V} = \mathcal{V}_1 \oplus \mathcal{V}_2$. The subspace $\mathcal{V}_2$ is called the **orthogonal complement** of $\mathcal{V}_1$ in $\mathcal{V}$. Equivalently, $\mathcal{V}_1$ is the orthogonal complement of $\mathcal{V}_2$ in $\mathcal{V}$.

A **norm** $\|\cdot\|$ on a vector space $\mathcal{V}$ is a function from $\mathcal{V}$ to $\mathbb{R}$ such that for every $v,w \in \mathcal{V}$ and $a \in \mathbb{F}$ the following three properties are fulfilled

1. $\|v\| \geq 0$, and $\|v\| = 0 \Longleftrightarrow v = 0$.
2. $\|av\| = |a|\|v\|$.
3. $\|v+w\| \leq \|v\| + \|w\|$.

A vector space $\mathcal{V}$ together with a norm is called a **normed vector space**.

Two vectors v and w are said to be **orthonormal** if they are *orthogonal* and $||v|| = ||w|| = 1$.

The **Euclidean space** $\mathbb{R}^n$ is an inner product space with inner product defined by

$$\langle \mathbf{x},\mathbf{y}\rangle = x_i\overline{y}_i + \ldots x_n\overline{y}_n,$$

with $\mathbf{x} = (x_1,\ldots,x_n)$ and $\mathbf{y} = (y_1,\ldots,y_n)$. The norm $\|\mathbf{x}\|$ is induced by the inner product

$$\|\mathbf{x}\| = \langle \mathbf{x},\mathbf{x}\rangle^{\frac{1}{2}} = (x_i\overline{x}_i + \ldots x_n\overline{x}_n)^{\frac{1}{2}} = (|x_i|^2 + \ldots + |x_n|^2)^{\frac{1}{2}}.$$

The **space** $L^2[a,b]$ is an inner product space of functions on $[a,b]$ with inner product defined by

$$\langle x,y\rangle = \int_a^b x(t)\overline{y}(t)\,dt$$

and norm

$$\|x\| = \langle x,x\rangle^{\frac{1}{2}} = \left(\int_a^b |x(t)|^2\,dt\right)^{\frac{1}{2}}.$$

The **space** $C^k[a,b]$ is the space of functions on $[a,b]$ having continuous derivatives up to order $k \in \mathbb{N}$. The space of continuous functions on $[a,b]$ is denoted as $C^0[a,b]$.

3 Signal Representation, Reconstruction, and Projection

Throughout this chapter a **signal** is considered to be an element of an inner product space $\mathcal{H}$ with norm induced by the inner product, $\|\cdot\| = \langle\cdot,\cdot\rangle^{\frac{1}{2}}$. Moreover, for the problems to be addressed we assume that all the signals of interest belong to some finite dimensional subspace $\mathcal{V}$ of $\mathcal{H}$ spanned by the set $\{v_i \in \mathcal{H}\}_{i=1}^{M}$. Hence, a signal f can be expressed by a finite linear superposition

$$f = \sum_{i=1}^{M} c_i v_i,$$

where the coefficients c_i, $i = 1,\ldots,M$, are in $\mathbb{F}$.

We call **measurement** or **sampling** to the process of transforming a signal into a number. Hence a **measure** or **sample** is a *functional*. Because we restrict considerations to linear measures the associated functional is linear. Accordingly, making use of *Riesz theorem* [24] we can express a linear measure as

$$m = \langle w, f\rangle \quad \text{for some} \quad w \in \mathcal{H}.$$

We refer the vector w to as **measurement vector**.

Considering M measurements m_i, $i = 1,\ldots,M$, each of which is obtained by a measurement vector w_i, we have a numerical representation of f as given by

$$m_i = \langle w_i, f\rangle, \quad i = 1,\ldots,M.$$

Now we want to answer the question as to whether it is possible to reconstruct $f \in \mathcal{V}$ from these measurements. More precisely, we wish to find the requirements we need to impose upon the measurement vectors w_i, $i = 1,\ldots,M$, so as to use the concomitant measures $\langle w_i, f\rangle$, $i = 1,\ldots,M$, as coefficients for the signal reconstruction, i.e., we wish to have

$$f = \sum_{i=1}^{M} c_i v_i = \sum_{i=1}^{M} \langle w_i, f\rangle v_i. \tag{1}$$

By denoting

$$\hat{E} = \sum_{i=1}^{M} v_i \langle w_i, \cdot\rangle, \tag{2}$$

where the operation $\langle w_i,\cdot\rangle$ indicates that $\hat{E}$ acts by taking inner products, (1) is recast

$$f = \hat{E} f.$$

As will be discussed next, the above equation tells us that the measurement vectors w_i, $i = 1,\ldots,M$, should be such that the operator $\hat{E}$ is a projector onto $\mathcal{V}$.

An operator $\hat{E} : \mathcal{H} \to \mathcal{V}$ is a projector if it is *idempotent*, i.e.,

$$\hat{E}^2 = \hat{E}.$$

As a consequence, the projection is onto $\mathcal{R}(\hat{E})$, the range of the operator, and along $\mathcal{N}(\hat{E})$, the null space of the operator. Let us recall that

$$\mathcal{R}(\hat{E}) = \{f, \text{ such that } f = \hat{E} g, \ g \in \mathcal{H}\}.$$

Thus, if $f \in \mathcal{R}(\hat{E})$,

$$\hat{E}f = \hat{E}^2 g = \hat{E}g = f.$$

This implies that $\hat{E}$ behaves like the identity operator for all $f \in \mathcal{R}(\hat{E})$, regardless of $\mathcal{N}(E)$, which is the subspace of $\mathcal{H}$ defined as

$$\mathcal{N}(E) = \{g, \text{ such that } \hat{E}g = 0, \ g \in \mathcal{H}\}.$$

It is clear then that to reconstruct a signal $f \in \mathcal{V}$ by means of (1) the involved measurement vectors $w_i, i = 1, \ldots, M$, should give rise to an operator of the form (2), which must be a projector onto $\mathcal{V}$. Notice that the required operator is not unique, because there exist many projectors onto $\mathcal{V}$ having different $\mathcal{N}(\hat{E})$. Thus, for reconstructing signals in the range of the projector its null space can be chosen arbitrarily. However, the null space becomes extremely important when the projector acts on signals outside its range. A popular projector, to be discussed below, is the orthogonal one. When $\mathcal{N}(\hat{E})$ happens to be equal to $\mathcal{R}(\hat{E})^{\perp}$, which indicates the orthogonal complement of $\mathcal{R}(\hat{E})$, the projector is called **orthogonal projector** onto $\mathcal{R}(\hat{E})$. This is the case if and only if the projector is *self adjoint*.

A projector which is not orthogonal is called an **oblique projector** and we need two subscripts to represent it. One subscript to indicate the range of the projector and another to represent the subspace along which the projection is performed. Hence the projector onto $\mathcal{V}$ along $\mathcal{W}^{\perp}$ is indicated as $\hat{E}_{\mathcal{V}\mathcal{W}^{\perp}}$.

The particular case $\hat{E}_{\mathcal{V}\mathcal{V}^{\perp}}$ corresponds to an *orthogonal projector* and we use the special notation $\hat{P}_{\mathcal{V}}$ to indicate such a projector. When a projector onto $\mathcal{V}$ is used for signal processing, $\mathcal{W}^{\perp}$ can be chosen according to the processing task. The examples below illustrate two different situations.

Example 1. Let us assume that the signal processing task is to approximate a signal $f \in \mathcal{H}$ by a signal $f_{\mathcal{V}} \in \mathcal{V}$. In this case, one normally would choose $f_{\mathcal{V}} = \hat{P}_{\mathcal{V}} f$ because this is the unique signal in $\mathcal{V}$ minimising the distance $\|f - f_{\mathcal{V}}\|$. Indeed, let us take another signal g in $\mathcal{V}$ and write it as $g = g + \hat{P}_{\mathcal{V}} f - \hat{P}_{\mathcal{V}} f$. Since $f - \hat{P}_{\mathcal{V}} f$ is orthogonal to every signal in $\mathcal{V}$ we have

$$\|f - g\|^2 = \|f - g + \hat{P}_{\mathcal{V}} f - \hat{P}_{\mathcal{V}} f\|^2 = \|f - \hat{P}_{\mathcal{V}} f\|^2 + \|\hat{P}_{\mathcal{V}} f - g\|^2.$$

Hence $\|f - g\|$ is minimised if $g = \hat{P}_{\mathcal{V}} f$.

Example 2. Assume that the signal f to be analysed here is the superposition of two signals, $f = f_1 + f_2$, each component being produced by a different phenomenon we want to discriminate. Let us assume further that $f_1 \in \mathcal{V}$ and $f_2 \in \mathcal{W}^{\perp}$ with $\mathcal{V}$ and $\mathcal{W}^{\perp}$ disjoint subspaces. Thus, we can obtain, f_1 say, from f, by an oblique projector onto $\mathcal{V}$ and along $\mathcal{W}^{\perp}$. The projector will map to zero the component f_2 to produce

$$f_1 = \hat{E}_{\mathcal{V}\mathcal{W}^{\perp}} f.$$

3.1 Constructing Oblique Projectors for Structured Noise Filtering

Example 2 discusses the fact that a signal component can be discriminated from another components by an oblique projection. In this section we will focus on the actual construction of such a projector.

Given $\mathcal{V}$ and $\mathcal{W}^\perp$ disjoint, i.e., such that

$$\mathcal{V} \cap \mathcal{W}^\perp = \{0\},$$

in order to provide a prescription for constructing $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ we proceed as follows. Firstly we define $\mathcal{S}$ as the direct sum of $\mathcal{V}$ and $\mathcal{W}^\perp$, which we express as

$$\mathcal{S} = \mathcal{V} + \mathcal{W}^\perp.$$

Let $\mathcal{W} = (\mathcal{W}^\perp)^\perp$ be the orthogonal complement of $\mathcal{W}^\perp$ in $\mathcal{S}$. Thus we have

$$\mathcal{S} = \mathcal{V} + \mathcal{W}^\perp = \mathcal{W} \oplus \mathcal{W}^\perp,$$

where the operation $\oplus$ indicates the orthogonal sum.

Considering that $\{v_i\}_{i=1}^M$ is a spanning set for $\mathcal{V}$ a spanning set for $\mathcal{W}$ is obtained as

$$u_i = v_i - \hat{P}_{\mathcal{W}^\perp} v_i = \hat{P}_{\mathcal{W}} v_i, \quad i = 1, \ldots, M.$$

Denoting as $\{\mathbf{e}_i\}_{i=1}^M$ the standard orthonormal basis in $\mathbb{F}^M$, i.e., the Euclidean inner product $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \delta_{ij}$, we define the operators $\hat{V} : \mathbb{F}^M \to \mathcal{V}$ and $\hat{U} : \mathbb{F}^M \to \mathcal{W}$ as

$$\hat{V} = \sum_{i=1}^M v_i \langle \mathbf{e}_i, \cdot \rangle, \quad \hat{U} = \sum_{i=1}^M u_i \langle \mathbf{e}_i, \cdot \rangle.$$

Thus the adjoint operators $\hat{U}^*$ and $\hat{V}^*$ are

$$\hat{V}^* = \sum_{i=1}^M \mathbf{e}_i \langle v_i, \cdot \rangle, \quad \hat{U}^* = \sum_{i=1}^M \mathbf{e}_i \langle u_i, \cdot \rangle.$$

It follows that $\hat{P}_{\mathcal{W}} \hat{V} = \hat{U}$ and $\hat{U}^* \hat{P}_{\mathcal{W}} = \hat{U}^*$ hence, $\hat{G} : \mathbb{C}^M \to \mathbb{C}^M$ defined as:

$$\hat{G} = \hat{U}^* \hat{V} = \hat{U}^* \hat{U}$$

is self-adjoint and its matrix representation, G, has elements

$$g_{i,j} = \langle u_i, v_j \rangle = \langle \hat{P}_{\mathcal{W}} u_i, v_j \rangle = \langle u_i, \hat{P}_{\mathcal{W}} u_j \rangle = \langle u_i, u_j \rangle, \quad i, j = 1, \ldots, M.$$

From now on we restrict our signal space to be $\mathcal{S}$, since we would like to build the oblique projector $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ onto $\mathcal{V}$ and along $\mathcal{W}^\perp$ having the form

$$\hat{E}_{\mathcal{V}\mathcal{W}^\perp} = \sum_{i=1}^M v_i \langle w_i, \cdot \rangle. \tag{3}$$

Clearly for the operator to map to zero every vector in $\mathcal{W}^\perp$ vectors $\{w_i\}_{i=1}^M$ must span $\mathcal{W} = (\mathcal{W}^\perp)^\perp = \text{span}\{u_i\}_{i=1}^M$. This entails that for each w_i there exists a set of coefficients $\{b_{i,j}\}_{j=1}^M$ such that

$$w_i = \sum_{i=1}^M b_{i,j} u_i, \tag{4}$$

which guarantees that every w_i is orthogonal to all vectors in $\mathcal{W}^\perp$ and therefore $\mathcal{W}^\perp$ is included in the null space of $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$. Moreover, since every signal, g say, in $\mathcal{S}$ can be written as $g = g_{\mathcal{V}} + g_{\mathcal{W}^\perp}$ with $g_{\mathcal{V}} \in \mathcal{V}$ and $g_{\mathcal{W}^\perp} \in \mathcal{W}^\perp$, the fact that $\hat{E}_{\mathcal{V}\mathcal{W}^\perp} g = 0$ implies $g_{\mathcal{V}} = 0$. Hence, $g = g_{\mathcal{W}^\perp}$, which implies that the null space of $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ restricted to $\mathcal{S}$ is $\mathcal{W}^\perp$. In order for $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ to be a projector it is necessary that $\hat{E}^2_{\mathcal{V}\mathcal{W}^\perp} = \hat{E}_{\mathcal{V}\mathcal{W}^\perp}$. As will be shown in the next proposition, if the coefficients $b_{i,j}$ are the matrix elements of a *generalised inverse* of the matrix G defined above, this condition is fulfilled.

Proposition 1. *If the coefficients $b_{i,j}$ in* (4) *are the matrix elements of a generalised inverse of the matrix G, which has elements $g_{i,j} = \langle v_i, u_j \rangle$, $i, j = 1, \ldots, M$, the operator in* (3) *is a projector.*

Proof. For the measurement vectors in (4) to yield a projector of the form (3), the corresponding operator should be idempotent, i.e.,

$$\sum_{n=1}^{M}\sum_{m=1}^{M}\sum_{i=1}^{M}\sum_{j=1}^{M} v_i b_{i,j} \langle u_j, v_n \rangle \overline{b_{n,m}} \langle u_m, \cdot \rangle = \sum_{i=1}^{M}\sum_{j=1}^{M} v_i \overline{b_{i,j}} \langle u_j, \cdot \rangle. \tag{5}$$

Defining

$$\hat{B} = \sum_{i=1}^{M}\sum_{j=1}^{M} \mathbf{e}_i \overline{b_{i,j}} \langle \mathbf{e}_j, \cdot \rangle \tag{6}$$

and using the operators $\hat{V}$ and $\hat{U}^*$, as given above, the left hand side in (5) can be expressed as

$$\hat{V}\hat{B}^*\hat{U}^*\hat{V}\hat{B}^*\hat{U}^* \tag{7}$$

and the right hand side as

$$\hat{V}\hat{B}^*\hat{U}^*. \tag{8}$$

Assuming that $\hat{B}^*$ is a generalised inverse of $(\hat{U}^*\hat{V})$ indicated as $\hat{B}^* = (\hat{U}\hat{V})^\dagger$ it satisfies (c.f. Section 2)

$$\hat{B}^*(\hat{U}^*\hat{V})\hat{B}^* = \hat{B}^*, \tag{9}$$

and therefore, from (7), the right hand side of (5) follows. Since $\hat{B}^* = (\hat{U}^*\hat{V})^\dagger = \hat{G}^\dagger$ and $\hat{G}^* = \hat{G}$, we have $\hat{B} = \hat{G}^\dagger$. Hence, if the elements $b_{i,j}$ determining $\hat{B}$ in (6) are the matrix elements of a generalised inverse on the matrix representation of $\hat{G}$, the corresponding vectors $\{w_i\}_{i=1}^n$ obtained by (4) yield an operator of the form (3), which is an oblique projector. □

Property 1. *Let $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ be the oblique projector onto $\mathcal{V}$ and along $\mathcal{W}^\perp$ and $\hat{P}_{\mathcal{W}}$ the orthogonal projector onto $\mathcal{W} = (\mathcal{W}^\perp)^\perp$. Then the following relation holds*

$$\hat{P}_{\mathcal{W}}\hat{E}_{\mathcal{V}\mathcal{W}^\perp} = \hat{P}_{\mathcal{W}}.$$

Proof. $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ given in (3) can be recast, in terms of operator $\hat{V}$ and $\hat{U}^*$, as:

$$\hat{E}_{\mathcal{V}\mathcal{W}^\perp} = \hat{V}(\hat{U}^*\hat{V})^\dagger\hat{U}^*.$$

Applying $\hat{P}_{\mathcal{W}}$ both sides of the equation we obtain:

$$\hat{P}_{\mathcal{W}}\hat{E}_{\mathcal{V}\mathcal{W}^\perp} = \hat{P}_{\mathcal{W}}\hat{V}(\hat{U}^*\hat{V})^\dagger\hat{U}^* = \hat{U}(\hat{U}^*\hat{V})^\dagger\hat{U}^* = \hat{U}(\hat{U}^*\hat{U})^\dagger\hat{U}^*,$$

which is a well known form for the orthogonal projector onto $\mathcal{R}(\hat{U}) = \mathcal{W}$. □

Notice that the operative steps for constructing an oblique projector are equivalent to those for constructing an orthogonal one. The difference being that in general the spaces $\text{span}\{v_i\}_{i=1}^M = \mathcal{V}$ and $\text{span}\{w_i\}_{i=1}^M = \mathcal{W}$ are different. For the special case $u_i = v_i$, $i = 1,\ldots,M$, both sets of vectors span $\mathcal{V}$ and we have an orthogonal projector onto $\mathcal{V}$ along $\mathcal{V}^\perp$.

Example 3. Suppose that the chirp signal in the first graph of Figure 1 is corrupted by impulsive noise belonging to the subspace

$$\mathcal{W}^\perp = \text{span}\{y_j(t) = e^{-100000(t-0.05j)^2},\ \ t \in [0,1]\}_{j=1}^{200}.$$

The chirp after being corrupted by a realization of the noise consisting of 50 pulses taken randomly from elements of $\mathcal{W}^\perp$ is plotted in the second graph of Figure 1.

Consider that the signal subspace is well represented by $\mathcal{V}$ given by

$$\mathcal{V} = \text{span}\{v_{i+1}(t) = \cos \pi i t,\ \ t \in [0,1]\}_{i=0}^{M=99}.$$

In order to eliminate the impulsive noise from the chirp we have to compute the measurement vectors $\{w_i\}_{i=1}^{100}$, here functions of $t \in [0,1]$, determining the appropriate projector. For this we first need a representation of $\hat{P}_{\mathcal{W}^\perp}$, which is obtained simply by transforming the set $\{y_j\}_{j=1}^{200}$ into an orthonormal set. Since the given set $\{y_j\}_{j=1}^{200}$ is linear independent its orthogonalisation can be achieved by a *Gram Schmidt*, or *Generalised Gram Schmidt*, procedure. For the extension to include the non-linear independent case, see [25]. The corresponding codes can be found at [26]. Using the orthonormal basis $\{o_j\}_{j=1}^{200}$ for $\mathcal{W}^\perp$ the orthogonal projector onto this subspace is the operator

$$\hat{P}_{\mathcal{W}^\perp} = \sum_{j=1}^{200} o_j \langle o_j, \cdot \rangle.$$

With this projector we construct vectors

$$u_{i+1}(t) = \cos \pi i t - \sum_{j=1}^{200} o_j(t) \langle o_j(t), \cos \pi i t \rangle, \quad i = 0,\ldots,99,\ \ t \in [0,1]$$

using a quadrature formula [27] for computing the inner products involved in the above equations and in the elements, $g_{i,j}$, of matrix G

$$g_{i+1,j+1} = \int_0^1 u_{i+1}(t) \cos \pi j t \, dt, \quad i = 0,\ldots,99,\ \ j = 0,\ldots,99.$$

This matrix has an inverse, which is used to obtain functions $\{w_i(t), t \in [0,1]\}_{i=1}^{100}$ giving rise to the required oblique projector. The chirp filtered by such a projector is depicted in the last graph of Figure 1.

Example 4. Here we deal with the image of Lena shown in the first picture of Figure 4. This image is an array of 512×512 pixels that we process row by row. Each row is a vector $\mathbf{I}_i \in \mathbb{R}^{512}$, $i = 1,\ldots,512$. The image is affected by structured noise produced when random

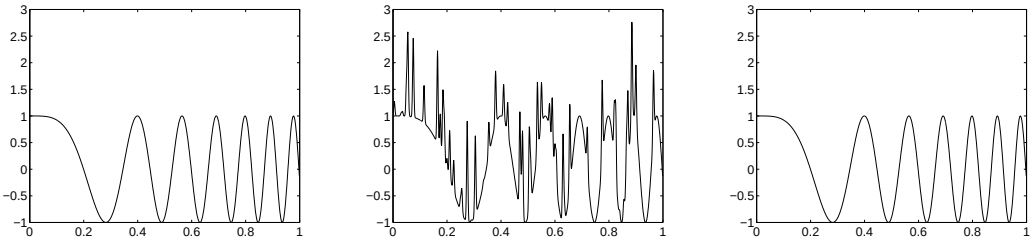

Figure 1: Chirp signal (first graph). Chirp corrupted by 50 randomly taken pulses (middle graph). Chirp denoised by oblique projection (last graph).

noise passes through a channel characterised by a given matrix A having 100 columns and 512 rows. The model for each row $\mathbf{I}_i^o \in \mathbb{R}^{512}$ of the noisy image is

$$\mathbf{I}_i^o = \mathbf{I}_i + A\mathbf{h}_i, \quad i = 1,\ldots,512,$$

where each $\mathbf{h}_i$ is a random vector in $\mathbb{R}^{100}$. The image plus noise is represented in the middle graph of Figure 4. In order to denoise the image we consider that every row $\mathbf{I}_i \in \mathbb{R}^{512}$ is well represented in a subspace $\mathcal{V}$ spanned by discrete cosines [28]. More precisely, we consider $\mathbf{I}_i \in \mathcal{V}$ for $i = 1,\ldots,512$, where

$$\mathcal{V} = \text{span}\left\{\mathbf{x}_i = \cos\left(\frac{\pi(2j-1)(i-1)}{2L}\right), \quad j = 1,\ldots,512\right\}_{i=1}^{300}.$$

The space of the noise is spanned by the 100 vectors in $\mathbb{R}^{512}$ corresponding to the columns of the given matrix A. The image, after being filtered row by row by the oblique projector onto $\mathcal{V}$ and along the space of the noise, is depicted in the last graph of Figure 4.

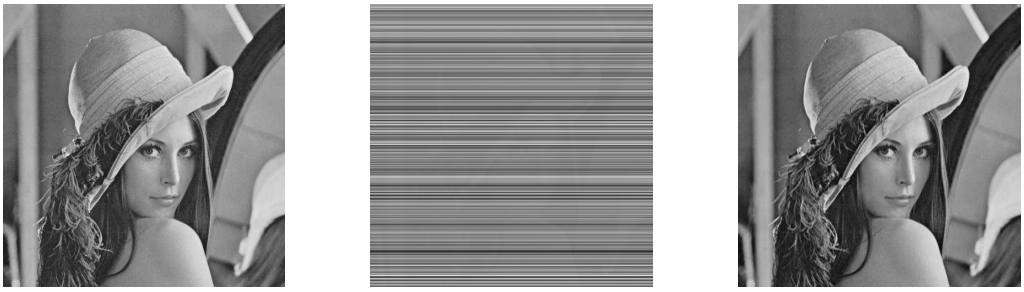

Figure 2: Image of Lena (first picture). Image plus structured noise (middle picture). The image obtained from the middle picture by an oblique projection (last picture).

Notice that the oblique projector onto $\mathcal{V}$ is independent of the selection of the spanning set for $\mathcal{W}$. Thus, one has a number of possibilities for expressing the oblique projector. Although all the forms are theoretically equivalent, they may not always be 'numerically' equivalent. This feature is illustrated in the next example.

Example 5. Consider that $\mathcal{W}^{\perp}$ is as in Example 3 and $\mathcal{V}$ is also as in that example but with $M = 250$ which, as illustrated by the experiment below, makes the construction of the corresponding oblique projector numerically unstable.

We consider four theoretically equivalent ways of computing vectors $\{w_i\}_{i=1}^{250}$.

i) $w_i = \sum_{j=1}^{250} \tilde{g}_{i,j} u_j$, where $\tilde{g}_{i,j}$ is the (i,j)-th element of the inverse of the matrix G having elements $g_{i,j} = \langle u_i, v_j \rangle$, $i,j = 1,\ldots,250$.

ii) Vectors $\{w_i\}_{i=1}^{250}$ are as in i) but the matrix elements of G are computed as $g_{i,j} = \langle u_i, u_j \rangle$, $i,j = 1,\ldots,250$.

iii) Orthonormalising $\{u_i\}_{i=1}^{250}$ to obtain $\{q_i\}_{i=1}^{250}$, vectors $\{w_i\}_{i=1}^{250}$ are then computed as

$$w_i = \sum_{j=1}^{250} \tilde{g}_{i,j} q_j,$$

with $g_{i,j} = \langle q_i, v_j \rangle$, $i,j = 1,\ldots,250$.

iv) Same as in iii) but $g_{i,j} = \langle q_i, u_j \rangle$, $i,j = 1,\ldots,250$.

We test the numerical quality of the corresponding projectors by comparing the following quantities

a) $\max_{g \in \mathcal{V}+\mathcal{W}^\perp} \frac{\|(\hat{E}_{\mathcal{V}\mathcal{W}^\perp} - \hat{E}^2_{\mathcal{V}\mathcal{W}^\perp})g\|}{\|g\|}$, $\|g\| \neq 0$,

b) $\max_{v_i \in \mathcal{V}} \frac{\|\hat{E}_{\mathcal{V}\mathcal{W}^\perp} v_i - v_i\|}{\|v_i\|}$, $\|v_i\| \neq 0$,

c) $\max_{s_i \in \mathcal{W}^\perp} \frac{\|\hat{E}_{\mathcal{V}\mathcal{W}^\perp} s_i\|}{\|s_i\|}$, $\|s_i\| \neq 0$.

The quality of the projector is indicated by the quantities a), b), and c), which should be small if the numerical representation of the projector is accurate. All the projectors produce equivalent results for quantity c). Nevertheless, there is a significant difference between case iii) and the others with respect to the quantities a) and b). Case iii) produces a value of a) and a value of b) which are 10^{-5} times smaller than the corresponding values produced by cases i), ii), and iv).

Unfortunately, even when the subspaces $\mathcal{V}$ and $\mathcal{W}^\perp$ are 'theoretically' complementary, in practice, due to the calculations being performed in finite precision arithmetics, the inaccuracy in the computation of the corresponding projector may cause the failure to correctly filter structured noise. This is discussed in [11, 12], where examples illustrating such situations are provided. As pointed out in those efforts, the problem can be overcome if the signal, $f_{\mathcal{V}}$, one is trying to discriminate from the noise, admits a sparse representation in a spanning set for $\mathcal{V}$. This implies that given $\{v_i\}_{i=1}^{M}$ there exists a subset of elements characterised by the set of indices J, of cardinality $K < M$, spanning the subspace $\mathcal{V}_K = \text{span}\{v_\ell\}_{\ell \in J}$ and such that $f_{\mathcal{V}} = \hat{E}_{\mathcal{V}_K \mathcal{W}^\perp} f$. Under the hypothesis that the numerical construction of $\hat{E}_{\mathcal{V}_K \mathcal{W}^\perp}$ is well posed, this projector will produce the correct signal splitting. The problem one has to address then is the one of finding the 'right' subspace $\mathcal{V}_K$. Assuming that $\{v_i\}_{i=1}^{M}$ is a basis for $\mathcal{V}$, out of it we have $\binom{M}{K}$ possible basis of dimension K. Hence the problem of finding that of the right subspace $\mathcal{V}_K$ is in general intractable. In order to reduce complexity one can make the search for the right subspace signal dependent. An adaptive approach is considered in the next section.

3.2 Adaptive Pursuit Strategy and Structured Noise Filtering

Given a signal f, and assuming that the subspaces $\mathcal{W}^\perp$ and $\mathcal{V}$, are known, the goal is to find a subset $\tilde{\mathcal{V}} \in \mathcal{V}$ such that $\hat{E}_{\tilde{\mathcal{V}}\mathcal{W}^\perp} f = \hat{E}_{\mathcal{V}\mathcal{W}^\perp} f$. Moreover, it is assumed that $\mathcal{V} = \text{span}\{v_i\}_{i=1}^M$ and $\tilde{\mathcal{V}} = \text{span}\{v_\ell\}_{\ell\in J}$, where J is a unknown subset of indices, out of the original set $i = 1,\ldots,M$. The cardinality of J is such that the construction of $\hat{E}_{\tilde{\mathcal{V}}\mathcal{W}^\perp}$ is well posed. This assumption restricts the class of signals that can be handled by the approach.

Under the above hypothesis, if the subspace $\tilde{\mathcal{V}}$ were known, one would have

$$\hat{E}_{\mathcal{V}\mathcal{W}} f = \hat{E}_{\tilde{\mathcal{V}}\mathcal{W}^\perp} f = \sum_{\ell\in J} v_\ell \langle w_\ell, f\rangle. \tag{10}$$

However, if the computation of $\hat{E}_{\mathcal{V}\mathcal{W}^\perp}$ is an ill posed problem, which is the situation we are considering, $\hat{E}_{\mathcal{V}\mathcal{W}^\perp} f$ is not available. In order to look for the subset of indices J yielding $\tilde{\mathcal{V}}$ one may proceed as follows: Applying $\hat{P}_{\mathcal{W}}$ on every term of (10) and using the properties $\hat{P}_{\mathcal{W}}\hat{E}_{\mathcal{V}\mathcal{W}^\perp} = \hat{P}_{\mathcal{W}}$ and $\hat{P}_{\mathcal{W}}\hat{E}_{\tilde{\mathcal{V}}\mathcal{W}^\perp} = \hat{P}_{\tilde{\mathcal{W}}}$, where $\tilde{\mathcal{W}} = \text{span}\{u_\ell\}_{\ell\in J}$, (10) becomes

$$\hat{P}_{\mathcal{W}} f = \hat{P}_{\tilde{\mathcal{W}}} f = \sum_{\ell\in J} u_\ell \langle w_\ell, f\rangle. \tag{11}$$

Since $\mathcal{W}^\perp$ is given and $\hat{P}_{\mathcal{W}} f = f - \hat{P}_{\mathcal{W}^\perp} f$, the left hand side of (10) is available. We are then in a position to look for the set $\{u_\ell\}_{\ell\in J}$, out of the set $\{u_i\}_{i=1}^M$, in a stepwise manner. This can be achieved by an adaptive pursuit approach, termed Optimised Orthogonal Matching Pursuit approach [29], which at each step, say the step $k+1$, selects the element $u_{\ell_{k+1}}$ minimising the norm of the residual error $\|\hat{P}_{\mathcal{W}} f - \hat{P}_{\mathcal{W}_{k+1}}\hat{P}_{\mathcal{W}} f\|^2$, where $\mathcal{W}_{k+1} = \text{span}\{u_{\ell_i}\}_{i=1}^k + u_{\ell_{k+1}}$ and $\{u_{\ell_i}\}_{i=1}^k$ is the set of elements that have been selected in the previous steps. Minimisation of $\|\hat{P}_{\mathcal{W}} f - \hat{P}_{\mathcal{W}_{k+1}}\hat{P}_{\mathcal{W}} f\|^2 = \|\hat{P}_{\mathcal{W}} f - \hat{P}_{\mathcal{W}_{k+1}} f\|^2$ is equivalent to choosing the index ℓ_{k+1} such that

$$\ell_{k+1} = \arg\max_{n\in J\setminus J_k} \frac{|\langle \gamma_n, \hat{P}_{\mathcal{W}} f\rangle|}{\|\gamma_n\|} = \arg\max_{n\in J\setminus J_k} \frac{|\langle \gamma_n, f\rangle|}{\|\gamma_n\|}, \;\; \|\gamma_n\| \neq 0,$$

with $\gamma_n = u_n - \hat{P}_{\mathcal{W}_k} u_n$ and $J \setminus J_k$ the set of indices not selected in previous steps.

Details on the implementation of this approach and extended versions of it are presented in [30]. Further discussions in the context of structured noise filtering are given in [12] and relevant MATLAB codes are available at [31].

4 Signal Interpolation

As already mentioned, we wish to consider the problem of impulsive noise filtering, specially in the case of images, as a problem of 'missing data' to be addressed here by means of interpolation techniques. The interpolation methods we propose are based on particular piecewise polynomial interpolation. However, in order to assist the readers unfamiliar with approximation theory, we introduce in the next two sections some basic elements of polynomial and piecewise polynomial interpolation.

4.1 Polynomial Interpolation

Suppose that we are given data as a set of $n+1$ points $\{x_i\}_{i=0}^n$ and a function f having values $\{y_i = f(x_i)\}_{i=0}^n$. Suppose further that we want to approximate f by a simple function p with the properties

$$p(x_i) = y_i, \quad i = 0, \ldots, n. \tag{12}$$

A function p satisfying the properties given in (12) is said to interpolate the function f at the points $\{x_i\}_{i=0}^n$ and is called an **interpolant**. In practice, the simple function p is *a polynomial, a piecewise polynomial or a rational function*. Different interpolation methods arise from the choice of the interpolating function. If the interpolating function is chosen to be a polynomial, the interpolation is called **polynomial interpolation**.

A polynomial interpolation problem can be stated as: Given a set of $n+1$ pairs of real numbers $\{(x_i, y_i)\}_{i=0}^n$, find a polynomial $p_m \in \mathcal{P}_m(\mathbb{R})$ such that $y_i = p_m(x_i)$, $i = 0, \ldots, n$. If $n \neq m$, the problem is over or under-determined. The following theorem holds if $n = m$. We refer to [32, 33] for a proof.

Theorem 1. *If $\{x_i\}_{i=0}^n$ is a set of distinct points, for an arbitrary set $\{y_i\}_{i=0}^n$ of $n+1$ numbers there exists a unique polynomial $p_n \in \mathcal{P}_n(\mathbb{R})$ such that $p_n(x_i) = y_i$, $i = 0, \ldots, n$.*

Assume that we are given a function f in $[a,b]$, and we have an interpolating polynomial p_n of degree n on a partition $\{x_i\}_{i=0}^n$ of $[a,b]$. Clearly, the function f and the polynomial p_n have exactly the same values at the interpolation points $\{x_i\}_{i=0}^n$. However, if we pick some arbitrary point $x \in [a,b]$ which is not an interpolating point, the function value $f(x)$ may be quite different from $p_n(x)$. Under the assumption that the function f is sufficiently smooth, the interpolation error is estimated in the following theorem, the proof of which can be found in many numerical analysis textbooks, e.g., [27, 33].

Theorem 2. *Suppose that f is an $(n+1)$-times continuously differentiable real-valued function on $[a,b]$. Suppose further that the interpolating points $\{x_i\}_{i=0}^n$ are distinct. Then, for $x \in [a,b]$,*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x),$$

where $\xi \in [a,b]$ is a function of x, and $\pi_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$. Denoting by M_{n+1} the maximum value of $|f^{(n+1)}(\xi)|$ in $[a,b]$, we can bound the above error as

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!}|\pi_{n+1}(x)|. \tag{13}$$

Lagrange Form of the Interpolation Polynomial

The Lagrange form of the interpolation polynomial is obtained by using the Lagrange basis for the vector space $\mathcal{P}_n(\mathbb{R})$.

Definition 1. *A set of functions $\{l_i\}_{i=0}^n$ is said to be a* **Lagrange basis** *for the space of polynomials $\mathcal{P}_n(\mathbb{R})$ with respect to the set of distinct points $\{x_i\}_{i=0}^n$ if $l_i(x_j) = \delta_{ij}$.*

Explicitly, the Lagrange basis functions with respect to the set of distinct points $\{x_i\}_{i=0}^n$ is

$$\left\{ l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j} \right\}_{i=0}^{n}.$$

In this case the interpolant p_n is given by

$$p_n(x) = \sum_{i=0}^{n} y_i l_i(x),$$

and is said to have the Lagrange form of polynomial interpolation.

Newton Form of the Interpolation Polynomial

Although the Lagrange form of the interpolation polynomial is suitable from the theoretical point of view, practically it is not the most convenient form. It is sometimes useful to start from an interpolation polynomial of lower degree and construct higher degree interpolation polynomials. In the case of the Lagrange polynomial there is no constructive relation between p_{n-1} and p_n. The Newton form of polynomial interpolation is designed to do that.

Let $\{x_i\}_{i=0}^n$ be a partition of $[a,b]$ and assume that we are given an interpolation polynomial p_{n-1} of degree $n-1$ for n pairs of numbers $\{(x_i,y_i)\}_{i=0}^{n-1}$ with $p_{n-1}(x_i) = y_i$, $i = 0,\ldots,n-1$. We want to represent the interpolation polynomial p_n of degree n for a set of $n+1$ pairs of numbers $\{(x_i,y_i)\}_{i=0}^n$ as a sum of p_{n-1} and a polynomial of degree n with only one unknown coefficient depending on the set of points $\{x_i\}_{i=0}^n$. Thus, taking $q_n \in \mathcal{P}_n(\mathbb{R})$, we set

$$p_n(x) = p_{n-1}(x) + q_n(x).$$

Using the fact that $p_{n-1}(x_i) = y_i$, we have $q_n(x_i) = p_n(x_i) - p_{n-1}(x_i) = 0$, $i = 0,\ldots,n-1$. Hence, q_n can be written as $q_n(x) = b_n \Pi_{i=0}^{n-1}(x-x_i)$, where b_n is to be determined. Assuming that $y_i = f(x_i)$, $i = 0,\ldots,n$, the coefficient b_n can be found by setting $p_n(x_n) = f(x_n)$. Thereby

$$b_n = \frac{f(x_n) - p_{n-1}(x_n)}{\pi_n(x_n)}, \tag{14}$$

where $\pi_n(x) = \Pi_{i=0}^{n-1}(x-x_i)$. The coefficient b_n is called the n-th Newton divided difference and is denoted by

$$b_n = f[x_0,x_1,\ldots,x_n], \;\; n \geq 1. \tag{15}$$

The interpolation polynomial p_n can then be written as

$$p_n(x) = p_{n-1}(x) + \pi_n(x) f[x_0,\ldots,x_n].$$

Using recursion on n, we obtain the formula for the interpolation polynomial in Newton form

$$p_n(x) = \sum_{i=0}^{n} \pi_i(x) f[x_0,\ldots,x_i], \tag{16}$$

where $p_0(x) = f(x_0) = f[x_0] = y_0$ and $\pi_0 = 1$.

The Lagrange and Newton forms yield the same interpolation polynomial due to the uniqueness of the interpolation polynomial. The interpolation polynomial in the form (16) is called the **Newton divided difference** formula. There are many properties of the Newton divided differences which make them computationally efficient [27, 32].

It is interesting to compare the two forms of interpolation polynomial in terms of the basis of the polynomial space $\mathcal{P}_n(\mathbb{R})$. The interpolation polynomial of the Lagrange form is obtained by using the Lagrange basis $\{l_i\}_{i=0}^n$, whereas the Newton form is obtained by taking the basis $\{\pi_i\}_{i=0}^n$. In the case of Lagrange form, the coordinate functionals are simply the function values at the given set of points whereas the coordinate functionals in the Newton form are given by Newton divided differences.

Hermite Interpolation

If the polynomial interpolation consists of finding an interpolant which is required to take not only the function values but also the derivatives of the function at the prescribed points, it is called **Hermite interpolation**.

Assume that we require the polynomial of lowest degree which interpolates a function f and its derivatives at two distinct points, say, x_0 and x_1 with $x_0 \neq x_1$. Thus, we have to find a polynomial p which fulfils the properties:

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i), \quad i = 0, 1.$$

Since there are four conditions, we look for a solution in $\mathcal{P}_3(\mathbb{R})$. Let

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad \text{and} \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}$$

be the two Lagrange basis functions with respect to the partition $\{x_0, x_1\}$. We consider the following basis of $\mathcal{P}_3(\mathbb{R})$: $H_0(x) = (x - x_1)^2(1 + 2l_1(x))$, $H_1(x) = (x - x_0)^2(1 + 2l_0(x))$, $K_0(x) = (x - x_1)^2(x - x_0)$, $K_1(x) = (x - x_0)^2(x - x_1)$. It is immediate to verify that

$$H_i(x_j) = c\delta_{ij}, \; H_i'(x_j) = 0, \; K_i(x_j) = 0, \; K_i'(x_j) = c\delta_{ij} \quad \text{with} \quad c = (x_1 - x_0)^2.$$

Hence, the required polynomial of lowest degree is simply

$$p(x) = \frac{1}{c}\left(\sum_{i=0}^{1} f(x_i)H_i(x) + f'(x_i)K_i(x)\right).$$

The general Hermite interpolation problem can be posed as follows: Given a partition $\Delta = \{x_i\}_{i=0}^n$ of $[a, b]$, find a polynomial p satisfying the conditions

$$\frac{d^j p}{dx^j}(x_i) = c_{ij}, \quad j = 0, \ldots, k_i - 1, \; i = 0, \ldots, n, \tag{17}$$

where $\frac{d^j p}{dx^j} = p$ for $j = 0$. At node x_i, k_i interpolatory conditions are imposed, where k_i might change with respect to i. There are in total $m = \sum_{i=0}^n k_i$ conditions. The proof of the following theorem can be found in [32].

Theorem 3. *There exists a unique polynomial* $p \in \mathcal{P}_{m-1}(\mathbb{R})$ *fulfilling the Hermite conditions in* (17).

Example 6. We give a construction of Hermite interpolation with $k_i = 2$ for all i. Let $\Delta = \{x_i\}_{i=0}^n$ be a partition of $[a,b]$ with $n+1$ points, and $\{y_i\}_{i=0}^n$ and $\{z_i\}_{i=0}^n$ two sets of real numbers. We wish to find a polynomial $p \in \mathcal{P}_{2n+1}(\mathbb{R})$ so that $p(x_i) = y_i$ and $p'(x_i) = z_i$ for $i = 0,\ldots,n$. Our construction is based on a polynomial basis

$$B = \{H_0,\ldots,H_n,K_0,\ldots,K_n\}$$

for $\mathcal{P}_{2n+1}(\mathbb{R})$ with the properties

$$H_j(x_i) = \delta_{ij},\ H_j'(x_i) = 0,\ K_j(x_i) = 0,\ K_j'(x_i) = \delta_{ij},\ \ i,j = 0,\ldots,n. \tag{18}$$

Assuming that $\{l_i\}_{i=0}^n$ is the Lagrange basis with respect to the partition Δ, if we define

$$H_i(x) = [l_i(x)]^2(1-2l_i'(x_i)(x-x_i)),\ K_i(x) = [l_i(x)]^2(x-x_i)$$

for $i = 0,\ldots,n$, the basis B satisfies the properties (18). Hence, the interpolant

$$p(x) = \sum_{i=0}^{n} a_i H_i(x) + b_i K_i(x)$$

fulfils the required conditions for $a_i = y_i$ and $b_i = z_i$, $i = 0,\ldots,n$.

Extension to Two Dimensions

The two-dimensional interpolation is concerned with interpolating a function of two variables. Assume that $\mathcal{G} = \{(x_i,y_i)\}_{i=0}^N$ is a set of points in $\mathbb{R}^2$, and a function f is given on $\mathcal{G}$ with $\{z_i = f(x_i,y_i)\}_{i=0}^N$. We are interested in an interpolation problem posed in a polygonal domain $\Omega \subset \mathbb{R}^2$.

Definition 2. *A* **domain or subdomain** *in two dimensions is an open bounded region. A* **polygonal domain or subdomain** *is an open and bounded region whose boundary consists of pieces of lines.*

In order to state the two-dimensional interpolation problem, we need definitions of a convex set and a convex hull.

Definition 3. *A set* $S \subset \mathbb{R}^k$, $k \in \mathbb{N}$, *is* **convex** *if for all* $\mathbf{x},\mathbf{y} \in S$ *and all* $t \in [0,1]$, *the point* $(1-t)\mathbf{x} + t\mathbf{y} \in S$.

Definition 4. *The* **convex hull** *for a set of points* $\mathcal{G}$ *is the minimal open convex set containing* $\mathcal{G}$.

Definition 5. *The* **closure** *of a domain* Ω *is defined by* $cl(\Omega) = \Omega \cup \partial\Omega$, *where* $\partial\Omega$ *denotes the boundary of the domain* Ω.

A two-dimensional interpolation problem is then posed as follows: Given a set of points $\mathcal{G}$, the convex hull, Ω, of $\mathcal{G}$ and given a function f defined on $\mathcal{G}$, find a function $p : \mathrm{cl}(\Omega) \to \mathbb{R}$ with $\{z_i = p(x_i, y_i)\}_{i=0}^{N}$.

We only consider the situation for which p is a polynomial or a piecewise polynomial. If the set of points $\mathcal{G}$ has a tensor product structure, it is easy to extend the idea of one-dimensional construction to the multi-dimensional case. A tensor product partition is defined as follows:

Definition 6. *Assume that $\Delta_x = \{x_i\}_{i=0}^{n}$ is a partition of the closed interval $[a,b]$ and $\Delta_y = \{y_j\}_{j=0}^{m}$ is that of $[c,d]$. Then the set of points $\Delta_{xy} = \{(x_i, y_j)\}_{i=0,j=0}^{n,m}$ is called a* **tensor product partition** *of the rectangular region $[a,b] \times [c,d]$. In short, we write $\Delta_{xy} = \Delta_x \otimes \Delta_y$.*

Let $\Delta_{xy} = \Delta_x \otimes \Delta_y$ be a tensor product partition of $[a,b] \times [c,d]$. Assume that $\{l_i^{\Delta_x}\}_{i=0}^{n}$ is the Lagrange basis of $\mathcal{P}_n(\mathbb{R})$ with respect to the partition Δ_x, and $\{l_j^{\Delta_y}\}_{j=0}^{m}$ is that of $\mathcal{P}_m(\mathbb{R})$ with respect to the partition Δ_y. Then, given the values of the function $f(x,y)$ at the partition Δ_{xy}, the Lagrange interpolation polynomial of $f(x,y)$ with respect to the partition Δ_{xy} is of degree n in x and degree m in y, and is given by

$$\sum_{i=0}^{n} \sum_{j=0}^{m} f(x_i, y_j) l^{\Delta_x}(x) l^{\Delta_y}(y).$$

If $\mathcal{G}$ does not have a tensor product structure, we have to solve a global polynomial interpolation problem in a non-rectangular domain or non-tensor product partition, which is difficult and often ill-posed [27]. Furthermore, the following remark points out another limitation of the global polynomial interpolation.

Remark 1. *If the function f to be interpolated is not a polynomial, the quantity $M_{n+1}|\pi_{n+1}(x)|$ in* (13) *can be very large when n is large, leading to a severe limitation of the higher order polynomial interpolation. This problem is typically known as Runge's phenomenon and is explained by Runge's example* [27,32]*. If one has the freedom to choose the interpolating points, the expression $|\pi_{n+1}(x)|$ can be made small by choosing the set of interpolation points as the zeros or the maxima of a Chebyshev polynomial* [32,33]*. However, in many interpolation problems, the set of points is already given and one cannot use a different set of points.*

Piecewise polynomial interpolation to be discussed in the next section provides a flexible and efficient solution to the above discussed problems.

4.2 Piecewise Polynomial Interpolation

So far we have considered the approximation of a function by a global polynomial. This implies that if the definition of the function is to be modified at a point, the polynomial interpolant changes globally. On the contrary, the piecewise polynomial interpolant does not change globally if the definition of the function changes locally.

Assume that the interpolation problem is posed in a domain $\Omega \subset \mathbb{R}^k$ with $k = 1,2$. The central idea of piecewise interpolation is to decompose the domain Ω into non-overlapping subdomains yielding its decomposition and define polynomial basis functions in each subdomain.

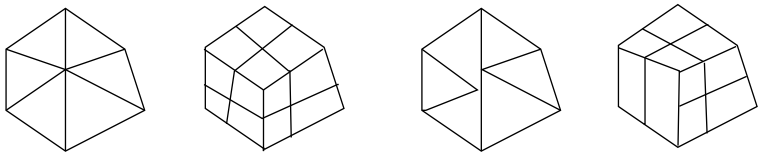

Figure 3: Four decompositions of the domain Ω: geometrically conforming (first two) and geometrically non-conforming (last two).

Definition 7. *Let $\Omega \subset \mathbb{R}^k$ be a domain with $k = 1,2$. The collection of disjoint subdomains $\mathcal{T}$ with $cl(\Omega) = \cup_{T\in\mathcal{T}} cl(T)$ is called a* **decomposition** *of Ω.*

In the two-dimensional case, if the interpolant is to be at least continuous, the decomposition should be geometrically conforming.

Definition 8. *A decomposition $\mathcal{T}$ of $\Omega \subset \mathbb{R}^2$ is called* **geometrically conforming** *if the intersection between the boundaries of any two different subdomains $\partial T_l \cap \partial T_k$, $k \neq l$, $T_k, T_l \in \mathcal{T}$ is either empty, a vertex or a common edge.*

Four decompositions of the domain Ω are shown in Figure 3. The two on the left are geometrically conforming and the two on the right are geometrically non-conforming. In the one-dimensional case, the subdomains are always intervals. In the two-dimensional case, only quadrilaterals or triangles are allowed.

The polynomial space $\mathcal{P}_m(T)$ will denote three different spaces depending on T. If T is an interval,

$$\mathcal{P}_m(T) = \Big\{ p : \; p(x) = \sum_{i=0}^{m} a_i x^i \Big\},$$

if T is a triangle,

$$\mathcal{P}_m(T) = \Big\{ p : \; p(x,y) = \sum_{\substack{i,j=0 \\ i+j\leq m}}^{m} a_{ij} x^i y^j \Big\},$$

and finally, if T is a quadrilateral,

$$\mathcal{P}_m(T) = \Big\{ p : \; p(x,y) = \sum_{i,j=0}^{m} a_{ij} x^i y^j \Big\}.$$

The smoothness of the interpolation polynomial is determined by the smoothness of the piecewise polynomial on the boundary of two subdomains.

Definition 9. *Let $m,k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$. The* **piecewise polynomial space** *of degree m and smoothness k with respect to the decomposition $\mathcal{T}$ is defined as*

$$S_{m,k}(\mathcal{T}) = \big\{ f \in C^k(\Omega) : \; f|_T \in \mathcal{P}_m(T), \; T \in \mathcal{T} \big\},$$

where $f|_T$ represents the restriction of the function f to the element T. The space of piecewise constant function with respect to the decomposition $\mathcal{T}$ is denoted by $S_0(\mathcal{T})$.

A generalisation of a Lagrange basis to the piecewise polynomial space is a nodal basis.

Definition 10. *Let $\mathcal{G} = \{(x_i, y_i)\}_{i=0}^{N}$ be a set of points in $\mathbb{R}^2$, and $\mathcal{T}$ be a decomposition of $\Omega =$ convex hull of $\mathcal{G}$. Then, a basis $\{\phi_i\}_{i=0}^{N}$ of $S_{m,k}(\mathcal{T})$ is called a* **nodal basis** *of $S_{m,k}(\mathcal{T})$ with respect to $\mathcal{G}$ if and only if $\phi_j(x_i, y_i) = \delta_{ij}$ for $i, j = 0, \ldots, N$.*

One-Dimensional Case

Before constructing some examples, we introduce a decomposition induced by a partition.

Definition 11. *Let $\Delta = \{x_i\}_{i=0}^{n}$ be a partition of the closed interval $[a,b]$, and $I_i = (x_i, x_{i+1})$ an interval. The decomposition $\mathcal{T} = \{I_i\}_{i=0}^{n-1}$ of the open interval (a,b) is called* **the decomposition** *induced by the partition Δ of $[a,b]$.*

Example 7. Assume that $\mathcal{T}$ is the decomposition of (a,b) induced by a partition $\Delta = \{x_i\}_{i=0}^{n}$ of $[a,b]$. Then, $S_{1,0}(\mathcal{T})$ is the space of linear splines on the decomposition $\mathcal{T}$. Let

$$\phi_0(x) = \begin{cases} \dfrac{x - x_1}{x_0 - x_1} & \text{if } x \in I_0 \\ 0 & \text{otherwise} \end{cases}, \quad \phi_n(x) = \begin{cases} \dfrac{x - x_{n-1}}{x_n - x_{n-1}} & \text{if } x \in I_{n-1} \\ 0 & \text{otherwise} \end{cases} \quad \text{and}$$

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x \in I_{i-1} \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} & \text{if } x \in I_i \\ 0 & \text{otherwise} \end{cases}, \quad \text{for} \quad i = 1, \ldots, n-1.$$

The set $\{\phi_i\}_{i=0}^{n}$ forms a basis for the space $S_{1,0}(\mathcal{T})$. Thus, a function $s_l \in S_{1,0}(\mathcal{T})$ can be written as

$$s_l(x) = \sum_{i=0}^{n} c_i \phi_i(x),$$

where $c_0, \ldots, c_n$ are arbitrary constants. As $\phi_i(x_j) = \delta_{ij}$, the basis $\{\phi_i\}_{i=0}^{n}$ is a nodal basis of $S_{1,0}(\mathcal{T})$ with respect to the partition Δ. Therefore, the piecewise linear interpolation of a continuous function $f : [a,b] \to \mathbb{R}$ on the decomposition $\mathcal{T}$ is obtained by setting $c_i = f(x_i)$, $i = 0, \ldots, n$.

Example 8. *Nearest Neighbour Interpolation in $[a,b]$:* Assume that the values of a function f at the partition $\Delta = \{x_i\}_{i=0}^{n}$ are given. Associated with the partition Δ, we form a dual partition $\tilde{\Delta} = \{z_i\}_{i=0}^{n+1}$ with $z_0 = x_0$, $z_i = \frac{x_{i-1} + x_i}{2}$, $i = 1, \ldots, n$, and $z_{n+1} = x_n$. Let χ_{I_i} be a characteristic function of the interval $I_i = [z_i, z_{i+1})$, $i = 0, \ldots, n$. If $\mathcal{T}$ is the decomposition induced by the partition $\tilde{\Delta}$, $S_0(\mathcal{T})$ is spanned by the basis $\{\chi_{I_i}\}_{i=0}^{n}$. Then, the nearest neighbour interpolation of the function f at the partition $\tilde{\Delta}$ is given by

$$N(x) = \sum_{i=0}^{n} f(x_i) \chi_{I_i}(x).$$

Example 9. *Piecewise Cubic Hermite Interpolation:* Given a partition $\Delta = \{x_i\}_{i=0}^{n}$ of the interval $[a,b]$ having $n+1$ number of points, we want to find a piecewise polynomial p with $p(x_i) = y_i$, and $p'(x_i) = z_i$ for $i = 0, \ldots, n$. Using the construction of Example 6, we define a piecewise polynomial p for $i = 0, \ldots, n-1$ as

$$p(x) = H_i(x) y_i + H_{i+1}(x) y_{i+1} + K_i(x) z_i + K_{i+1}(x) z_{i+1}, \quad x \in [x_i, x_{i+1})$$

with H_i, H_{i+1}, K_i, K_{i+1} as in Example 6, but l_i and l_{i+1} are now piecewise linear polynomials given by

$$l_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}}, \quad l_{i+1}(x) = \frac{x - x_i}{x_{i+1} - x_i}.$$

Assume that no derivatives are provided at the partition, but only function values. A piecewise cubic polynomial can be constructed also in this case assigning some suitable derivatives of the function at the partition. The derivatives are assigned in such a way that the resulting piecewise curve is continuously differentiable. One such example can be found in [34] and is used in piecewise cubic interpolation of MATLAB. The derivatives z_i are assigned in such a way that the function is continuously differentiable and function values do not locally overshoot the data values. Let d_i be defined as

$$d_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \quad i = 0, \ldots, n-1.$$

For an inner point x_i, if d_i and d_{i-1} are of opposite signs or if either of them is zero, x_i is the local extremum. Thus z_i is set to be zero. If d_i and d_{i-1} have the same sign, z_i is set to be a weighted harmonic mean of the two discrete slopes

$$\frac{1}{z_i} = \frac{1}{w_1 + w_2} \left(\frac{w_1}{d_{i-1}} + \frac{w_2}{d_i} \right)$$

with $w_1 = 2h_i + h_{i-1}$ and $w_2 = h_i + 2h_{i-1}$. For more details see [34].

Although in the one-dimensional case a decomposition can always be induced by a partition, this is not the case in two dimensions. Therefore, for simplicity, we divide the two-dimensional case into two parts depending on whether the decomposition has a tensor product structure or not.

Two-Dimensional Tensor Product Case

We start by defining a decomposition of a rectangular domain induced by a tensor product partition.

Definition 12. *Let $\Delta_{xy} = \Delta_x \otimes \Delta_y$ be a tensor product partition of the rectangular region $[a,b] \times [c,d]$, where $\Delta_x = \{x_i\}_{i=0}^n$ is a partition of $[a,b]$ and $\Delta_y = \{y_i\}_{i=0}^n$ is that of $[c,d]$. Let $I_{ij} = (x_i, x_{i+1}) \times (y_j, y_{j+1})$, $i = 0, \ldots, n-1$, $j = 0, \ldots, m-1$. Then $\mathcal{T} = \{I_{ij}\}_{i=0,j=0}^{n-1,m-1}$ is called* **the decomposition of the domain** $(a,b) \times (c,d)$ *induced by the partition Δ_{xy}.*

Example 10. Let $\mathcal{T}_x$ be the decomposition of (a,b) induced by a partition $\Delta_x = \{x_i\}_{i=0}^n$ and $\mathcal{T}_y$ that of (c,d) induced by a partition $\Delta_y = \{y_j\}_{j=0}^m$. Let $\{\phi_i\}_{i=0}^n$ and $\{\varphi_j\}_{j=0}^m$ be bases of $S_{1,0}(\mathcal{T}_x)$ and $S_{1,0}(\mathcal{T}_y)$, respectively, as constructed in Example 7. If $\mathcal{T}_{xy}$ is the decomposition of the domain $(a,b) \times (c,d)$ induced by the partition $\Delta_{xy} = \Delta_x \otimes \Delta_y$, then $\{\psi_{ij}\}_{i=0,j=0}^{n,m}$ with $\psi_{ij}(x,y) = \phi_i(x)\varphi_j(y)$ forms a nodal basis for $S_{1,0}(\mathcal{T}_{xy})$ with respect to the set of points Δ_{xy}. A piecewise linear interpolation of a continuous function $f : \mathbb{R}^2 \to \mathbb{R}$ can be obtained as

$$\sum_{i=0}^{n} \sum_{j=0}^{m} f(x_i, y_j) \psi_{i,j}(x,y).$$

Image Transformation through Interpolation

A digital image can be resized, rotated or distorted by using piecewise polynomial interpolation. Although the interpolation technique can be applied to any digital image, we only consider intensity images. An intensity image is an array of size $m \times n$, where each element of the array represents the intensity or gray scale of the pixel.

Associated with an image I of size $m \times n$ we define a tensor product partition Δ^0_{xy} of the square $[0,1] \times [0,1]$ as

$$\Delta^0_{xy} = \big\{(a_i, b_j)\big\}_{i=0,j=0}^{n-1,m-1}, \quad \text{where} \quad a_i = \frac{i}{n-1} \text{ and } b_j = \frac{j}{m-1}.$$

We also define an **image function** $I_f : \Delta^0_{xy} \to \mathbb{R}$ in such a way that $I_f(a_i, b_j)$ gives the intensity of the pixel at the point (a_i, b_j).

Suppose that we are given an intensity image I of size $m \times n$, and we want to convert this image into another image $\tilde{I}$ of size $m_1 \times n_1$. The image function $\tilde{I}_f$ for the new image is to be defined on the new tensor product partition

$$\Delta^1_{xy} = \big\{(c_i, d_j)\big\}_{i=0,j=0}^{n_1-1,m_1-1} \text{ with } c_i = \frac{i}{n_1-1} \text{ and } d_j = \frac{j}{m_1-1}.$$

The central idea of image resizing is to define a function on the square $[0,1] \times [0,1]$ and to sample it at the tensor product partition Δ^1_{xy}. This procedure leads to the image function $\tilde{I}_f : \Delta^1_{xy} \to \mathbb{R}$.

In general, image resizing, rotation or distortion can be seen as a transformation of an image. Suppose that we are given an invertible transformation $\hat{T}$ to be applied to an intensity image of size $m \times n$. At the first step, we apply the transformation to the underlying tensor product partition Δ^0_{xy} to obtain a new set of points Δ^2_{xy}, and find a suitable rectangular region $[a,b] \times [c,d]$ which contains all the points in Δ^2_{xy}.

Let

$$\Delta^1_{xy} = \big\{(c_i, d_j)\big\}_{i=0,j=0}^{n_1-1,m_1-1}$$

be a tensor product partition of the region $[a,b] \times [c,d]$, where we want to define the new image function $\tilde{I}_f : \Delta^1_{xy} \to \mathbb{R}$. Notice that for image resizing $\hat{T}$ is an identity transformation and therefore Δ^0_{xy} and Δ^2_{xy} are the same. Since Δ^0_{xy} has a tensor product structure, it is easy to find a piecewise polynomial p on $[0,1] \times [0,1]$ which interpolates the image function I_f at the tensor product partition Δ^0_{xy} (see Example 10).

Let (c_i, d_j) be a point in the tensor product partition Δ^1_{xy} with

$$\begin{bmatrix} x_i \\ y_j \end{bmatrix} = \hat{T}^{-1} \begin{bmatrix} c_i \\ d_j \end{bmatrix}.$$

At the second step, the new image function $\tilde{I}_f$ is obtained by setting $\tilde{I}_f(c_i, d_j) = p(x_i, y_j)$ if (x_i, y_j) is inside the closure of the convex hull of the set of points Δ^0_{xy}, otherwise $\tilde{I}_f(c_i, d_j)$ is set to zero. The algorithm for image transformation is given in Algorithm 1. MATLAB codes are available at [31].

In Figure 4, we show three examples of image transformation by using linear interpolation of the test image. The top left picture of Figure 4 shows the original image of size

Require: A digital image I and a transformation $\hat{T}$
Ensure: A transformed digital image $\tilde{I}$

1: Associate a tensor product partition Δ^0_{xy} and an image function $I_f : \Delta^0_{xy} \to \mathbb{R}$
2: Compute

$$\Delta^2_{xy} = \{\hat{T} \begin{bmatrix} a_i \\ b_j \end{bmatrix}, (a_i, b_j) \in \Delta^0_{xy}\}$$

3: Find a rectangle $[a,b] \times [c,d]$ with $\Delta^2_{xy} \subset [a,b] \times [c,d]$
4: Define a tensor product partition $\Delta^1_{xy} = \{(c_i, d_j)\}_{i=0,j=0}^{n_1-1,m_1-1}$ of $[a,b] \times [c,d]$, $n_1 \times m_1$ is the size of the new image
5: For each $(c_i, d_j) \in \Delta^1_{xy}$, find (x_i, y_j) with

$$\begin{bmatrix} x_i \\ y_j \end{bmatrix} = \hat{T}^{-1} \begin{bmatrix} c_i \\ d_j \end{bmatrix}$$

6: Compute an interpolation polynomial p of I_f on the unit square
7: Define an image function $\tilde{I}_f : \Delta^1_{xy} \to \mathbb{R}$ as

$$\tilde{I}_f(c_i, d_j) = \begin{cases} p(x_i, y_j) & \text{if } (x_i, y_j) \in \text{convex hull of } \Delta^0_{xy} \\ 0 & \text{otherwise} \end{cases}$$

8: The new digital image $\tilde{I}$ is then given by $\tilde{I}(i,j) = \tilde{I}_f(c_i, d_j)$

Algorithm 1: Image Transformation

512×512. The top right one is the original image rotated 40° anti-clockwise. The pictures at the bottom correspond to the original image resized to 200×500 (left) and to 200×200 (right).

Two-Dimensional Non-Tensor Product Case

The two-dimensional interpolation problem is more difficult if the interpolation points do not have a tensor product structure. In such a situation, the two-dimensional interpolation is called **scattered data interpolation**. There is a vast amount of literature devoted to scattered data interpolation. We refer to [18, 19] for extensive surveys on this subject. Here, we consider an approach based on decomposing the convex hull of $\mathcal{G}$ into triangles with vertices in $\mathcal{G}$ and piecewise polynomial interpolation. Therefore, in what follows we restrict ourselves to the case of piecewise interpolation on triangles. The most efficient and popular way of decomposing the polygonal domain Ω into triangles when Ω is the convex hull of the scattered points is the Delaunay triangulation.

Definition 13. *Given a set $\mathcal{G}$ of points in $\mathbb{R}^2$, a* **Delaunay triangulation** *for $\mathcal{G}$ is a decomposition $\mathcal{T}$ of convex hull of $\mathcal{G}$ into triangles such that no point in $\mathcal{G}$ is inside the circumcircle of any triangle in $\mathcal{T}$.*

A Delaunay triangulation of a finite set of points in the plane is a triangulation that minimises the standard deviations of the angles of the triangles. In this sense, the Delau-

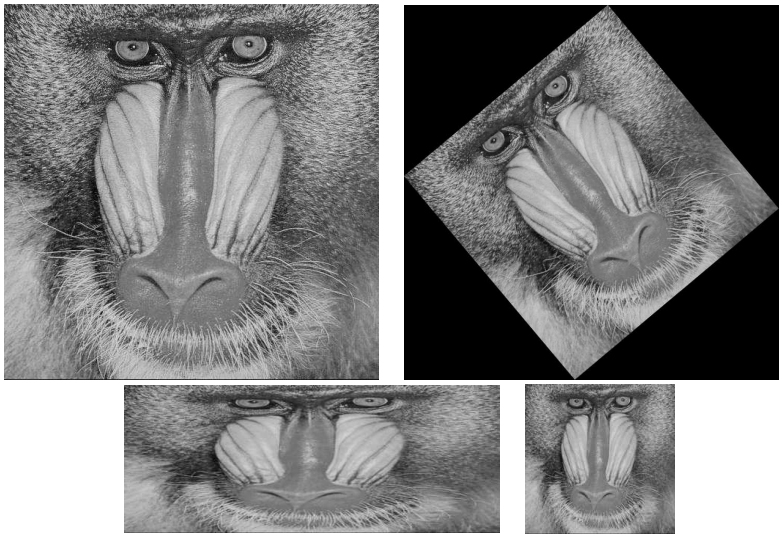

Figure 4: The original image of size 512×512 and image rotated 40°(top left and right). Resized image of size 200×500 and of size 200×200 (bottom left and right).

nay triangulation is the most equi-angular triangulation. The dual graph of the Delaunay triangulation is a Voronoi diagram for the same set of points.

Definition 14. *For a set of points $\mathcal{G} \subset \mathbb{R}^2$, the* **Voronoi diagram** *is the decomposition of the plane into convex polygons such that each polygon contains exactly one generating point from $\mathcal{G}$ and every point in a given polygon is closer to its generating point than to any other point in $\mathcal{G}$. A convex polygon $V_{\mathbf{x}}$ associated with the generating point $\mathbf{x} \in \mathcal{G}$ is called the* **Voronoi cell** *for the point $\mathbf{x} \in \mathcal{G}$.*

In other words, the Voronoi cell $V_{\mathbf{x}}$ for the point $\mathbf{x} \in \mathcal{G}$ has the property that the distance from every $\mathbf{y} \in V_{\mathbf{x}}$ to $\mathbf{x}$ is less than or equal to the distance from $\mathbf{y}$ to any other point in $\mathcal{G}$. The circle circumscribed about a Delaunay triangle has its centre at the vertex of a Voronoi cell, see the right graph of Figure 5. The idea of Delaunay triangulation and Voronoi diagram is also extended to higher dimension. An efficient algorithm for computing Delaunay triangulations and Voronoi diagrams are presented in [35], see also [20, 36]. A lot of interesting materials can be found in websites [37, 38].

As an example of a Delaunay triangulation and Voronoi diagram, we define a set $\mathcal{G}_1 = \{(0.1,0.4),(0.5,0.1),(0.45,0.5),(0.3,0.6),(0.3,0.3),(0.1,0.4),(0.9,0.8),(0.3,0.9),$ $(0.2,0.1),(0.8,0.9)\}$, and generate the Delaunay triangulation and the Voronoi diagram of $\mathcal{G}_1$. The Delaunay triangulation and the Voronoi diagram of $\mathcal{G}_1$ are shown in the left and middle graphs of Figure 5, respectively. The right graph depicts the circumcircle of a triangle with its centre at a vertex of the Voronoi diagram (filled circle).

Example 11. Let $\mathcal{T}$ be a Delaunay triangulation of the set of points $\mathcal{G}$. We construct a nodal basis $\{\phi_i\}_{i=0}^N$ of $S_{1,0}(\mathcal{T})$ with each basis function ϕ_i corresponding to a point $(x_i,y_i) \in \mathcal{G}$ and satisfying $\phi_i(x_j,y_j) = \delta_{ij}$, $i,j = 0,\ldots,N$. A basis function ϕ_i is shown in the right graph of Figure 6. The basis function ϕ_i vanishes outside $\cup_{j=1}^5 \mathrm{cl}(T_j)$, where $T_j \in \mathcal{T}$, $j = 1,\ldots,5$,

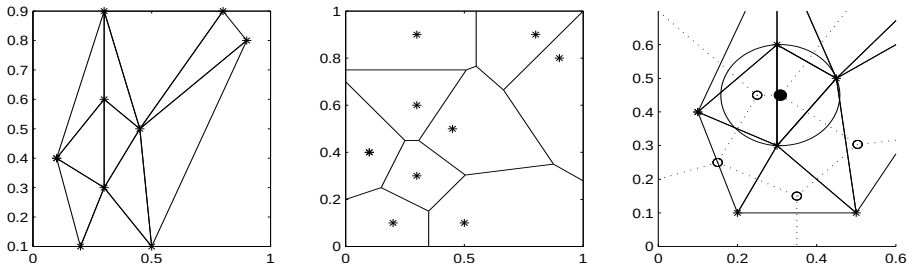

Figure 5: The Delaunay triangulation of the set $\mathcal{G}_1$ (left graph). The corresponding Voronoi diagram (middle graph). The circumcircle of a triangle with Delaunay triangulation and Voronoi diagram (last graph).

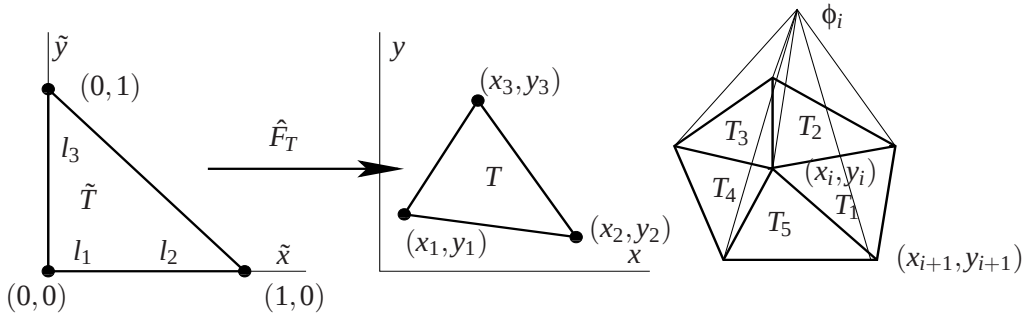

Figure 6: The reference triangle and an affine map $\hat{F}_T$ from the reference triangle to a generic triangle (first two graphs). A nodal basis function ϕ_i (last graph).

are the triangles having the common vertex (x_i,y_i) as shown in the right graph of Figure 6. The key point for the construction is to define a reference triangle, where it is easy to compute polynomial basis functions, and map the triangle to the actual element by using an affine transformation. It is convenient to choose the right-angled triangle $\tilde{T} = \{(\tilde{x},\tilde{y}) : \tilde{x} > 0, \tilde{y} > 0, \tilde{x}+\tilde{y} < 1\}$ as the reference triangle.

The local Lagrange basis functions associated with three vertices of $\tilde{T}$ for the linear interpolation are given by $l_1(\tilde{x},\tilde{y}) = (1-\tilde{x}-\tilde{y})$, $l_2(\tilde{x},\tilde{y}) = \tilde{x}$, and $l_3(\tilde{x},\tilde{y}) = \tilde{y}$. See the left graph of Figure 6. If $(\tilde{x}_j,\tilde{y}_j)$, $j = 1,\ldots,3$ are the three vertices of the reference triangle, the basis functions satisfy $l_i(\tilde{x}_j,\tilde{y}_j) = \delta_{ij}$, $i,j = 1,\ldots,3$. Let $T \in \mathcal{T}$ have three vertices (x_1,y_1), (x_2,y_2) and (x_3,y_3) as in the middle graph of Figure 6. The mapping $\hat{F}_T$ transforms a point in the reference triangle $\tilde{T}$ to a point in the actual triangle T as follows

$$\begin{bmatrix} x \\ y \end{bmatrix} = \hat{F}_T \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} \quad \text{with} \quad \hat{F}_T \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}.$$

The first two graphs of Figure 6 show a reference triangle $\tilde{T}$ and the triangle T.

If the three vertices of the triangle T are not collinear, the determinant of the matrix

$$\begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}$$

does not vanish and hence $\hat{F}_T$ is invertible. The three global Lagrange basis functions on

the triangle T are then given by $g_i(x,y) = l_i(\tilde{x},\tilde{y})$, $i = 1,\ldots,3$, with

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\end{bmatrix} = \hat{F}_T^{-1}\begin{bmatrix}x\\ y\end{bmatrix}.$$

By construction the global Lagrange basis functions g_i satisfy $g_i(x_j,y_j) = \delta_{ij}$, $i,j = 1,\ldots,3$, and are linear in T.

Let us consider the construction of the basis function ϕ_i corresponding to the point (x_i,y_i) in more detail. The point (x_i,y_i) is a vertex of five triangles as shown in the right graph of Figure 6. The Lagrange basis function corresponding to the vertex (x_i,y_i) for each of these five triangles has value one at the point (x_i,y_i) and value zero on all edges of the triangle opposite to this vertex. Let $\tilde{\phi}_j$ be the Lagrange basis function for the triangle T_j, $j = 1,\ldots,5$, corresponding to the point (x_i,y_i). The global basis ϕ_i corresponding to the point (x_i,y_i) is then defined by

$$\phi_i(x,y) = \begin{cases}\tilde{\phi}_j(x,y) & \text{if} \quad (x,y) \in T_j, \;\; j = 1,\ldots,5\\ 0 & \text{otherwise.}\end{cases}$$

It is obvious that the basis function ϕ_i is piecewise linear and $\phi_i(x_j,y_j) = \delta_{ij}$. Furthermore, the following lemma shows that ϕ_i is continuous.

Lemma 1. *The basis function ϕ_i defined as above is continuous on the boundary of two triangles.*

Proof. Let us analyse the value of the function ϕ_i on the boundary of T_1 and T_5, which is a line joining two points (x_i,y_i) and (x_{i+1},y_{i+1}), see the right graph of Figure 6. Here, both $\tilde{\phi}_1$ and $\tilde{\phi}_5$ take the values one at (x_i,y_i) and zero at (x_{i+1},y_{i+1}). In between are linear. Since there is a unique linear polynomial having this property, they have exactly the same value on the boundary of T_1 and T_5. □

Each vertex in $\mathcal{G}$ has an associated basis function which has value one at this vertex and zero at all other vertices. This leads to a nodal basis $\{\phi_i\}_{i=0}^N$ of the piecewise polynomial space $S_{1,0}(\mathcal{T})$.

Example 12. *Nearest Neighbour Interpolation:* The nearest neighbour interpolation of a set of scattered data $\mathcal{G} = \{(x_i,y_i)\}_{i=0}^N$ can be realised by generating the Voronoi diagram of the set $\mathcal{G}$. Assume that we are given the values $\{z_i\}_{i=0}^N$ of a function at $\mathcal{G}$, and χ_{V_i} is the characteristic function of the Voronoi cell V_i corresponding to the point (x_i,y_i), $i = 0,\ldots,N$. Then, the nearest neighbour interpolant p of the given data is obtained as

$$p(x) = \sum_{i=0}^{N} z_i \chi_{V_i}(x).$$

Image Denoising through Interpolation

As an application of the scattered data interpolation, we propose a very simple method of denoising an image in those cases in which the noisy and pure pixels can be identified. For example, we can identify the noisy and pure pixels in an image corrupted with salt and

Require: A noisy digital image I
Ensure: A filtered digital image $\tilde{I}$

1: Associate a tensor product partition Δ^0_{xy} and an image function $I_f : \Delta^0_{xy} \to \mathbb{R}$ with the image I
2: Let S^p and S^n denote the set of points with pure and noisy pixels with $\Delta^0_{xy} = S^p \cup S^n$, respectively, and $I^p_f = I_f|_{S^p}$
3: Compute the Delaunay triangulation $\mathcal{T}$ of the set of points S^p
4: Find a piecewise interpolation polynomial p of I^p_f on the convex hull of S^p
5: Let $\Delta^1_{xy} = \{(c_i, d_j)\}_{i=0,j=0}^{n_1-1,m_1-1}$ be a tensor product partition of the square $[0,1] \times [0,1]$ where we want to define the new image function $\tilde{I}_f$ for the filtered image. Define the new image function $\tilde{I}_f : \Delta^1_{xy} \to \mathbb{R}$ as

$$\tilde{I}_f(c_i, d_j) = \begin{cases} p(c_i, d_j) & \text{if} \quad (c_i, d_j) \in \text{convex hull of } S^p \\ I^p_f(\tilde{x}_i, \tilde{y}_j) & \text{otherwise,} \end{cases}$$

where $(\tilde{x}_i, \tilde{y}_j) \in S^p$ and is nearest to (c_i, d_j)
6: The filtered image $\tilde{I}$ is then given by $\tilde{I}(i, j) = \tilde{I}_f(c_i, d_j)$

Algorithm 2: Impulsive Noise Removal

pepper noise. The salt and pepper noise is of impulsive nature. In an intensity image the noisy pixels are randomly set either to white or black [15]. Here we apply the scattered data interpolation to filter this type of noise. An alternative approach for removing high density salt and pepper noise is proposed in [17]. However, the method of scattered data interpolation based on a Delaunay triangulation is easy and efficient.

Assume that a digital image I of size $m \times n$ is corrupted with salt and pepper noise. As in the previous case, we associate with the image a tensor product partition

$$\Delta^0_{xy} = \{(a_i, b_j)\}_{i=0,j=0}^{n-1,m-1}, \quad \text{where} \quad a_i = \frac{i}{n-1} \text{ and } b_j = \frac{j}{m-1}$$

and the image function $I_f : \Delta^0_{xy} \to \mathbb{R}$.

Let S^n be the set of points in Δ^0_{xy} having corrupted pixels. Denoting by S^p the set of points where only the pure pixels of the image are located we split the image function I_f into two functions I^n_f and I^p_f, which are defined as $I^n_f = I_f|_{S^n}$ and $I^p_f = I_f|_{S^p}$, respectively. As the positions of the noisy pixels are random, the points in S^n have no structure.

The key idea for getting rid of noisy pixels is to find an interpolant which interpolates the scattered data in S^p. To such an end we generate a Delaunay triangulation of the scattered points in S^p and compute an interpolant p on the convex hull of S^p. Hence $p(x,y) = I^p_f(x,y)$ for all $(x,y) \in S^p$. Choosing a tensor product partition $\Delta^1_{xy} = \{(c_i, d_j)\}_{i=0,j=0}^{n_1-1,m_1-1}$ we define a new image function $\tilde{I}_f : \Delta^1_{xy} \to \mathbb{R}$ so that $\tilde{I}_f(x,y) = p(x,y)$ for all $(x,y) \in \Delta^1_{xy}$. If the point $(x,y) \in \Delta^1_{xy}$ does not belong to the closure of the convex hull of S^p, we need to extrapolate for that point. For this, we choose one point $(\tilde{x}, \tilde{y})$ in S^p, out of the nearest to (x,y), and set $\tilde{I}_f(x,y) = I^p_f(\tilde{x}, \tilde{y})$. We have canned the complete denoising procedure in Algorithm 2.

Assuming that we know the image before being corrupted with noise, we use peak signal-to-noise ratio (PSNR) to compare our results with the results obtained by using the

standard median filter [16]. The peak signal-to-noise ratio (PSNR) is defined by

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

where MAX_I is the maximum pixel value of the image, and MSE is the mean square error, i.e.,

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \|IO(i,j) - IR(i,j)\|^2,$$

where IO is the original image before corruption and IR is the image recovered after removing the noise.

In Figure 7, we show an example of applying the linear interpolation to remove the salt and pepper noise. The first picture (from the left) of Figure 7 shows the original image. The second one shows the noisy image with noise density 50%. The third picture is the image reconstructed by using the linear interpolation, and the fourth one is the image denoised by the median filter. In this example, the nearest neighbour interpolation produces the image which is visually equivalent to that produced by the linear interpolation. We also observe that the image obtained by the median filter does not differ much from the previous ones. However, if the noise density is increased, the median filter does not recover even the main feature of the image. This is illustrated in Figure 8, where we deal with noise density 95%. The first picture of this figure shows the noisy image. The second picture shows the image reconstructed by using the linear interpolation. The third picture shows the image reconstructed by using the nearest neighbour interpolation, and the last one the reconstruction by using the median filter. We can see that here the median filter produces a meaningless image.

The peak signal-to-noise ratio (PSNR) for two test images are given in Table 1. The two images we have considered are Lena (first picture in Figure 7) and Baboon (top left picture of Figure 4). For both images, the linear interpolation produces the best PSNR values for all noise densities. The PSNR values from the nearest neighbour interpolation are not far from these values. The PSNR values from median filter are acceptable in the cases of low noise densities, whereas at high noise densities (above 50%) the PSNR values decrease considerably.

Figure 7: The original image. Noisy image (noise density 50%). Image denoised with linear interpolation and median filter.

Figure 8: Noisy image (noise density 95%). Image denoised with linear interpolation, nearest neighbour interpolation and median filter.

Table 1: PSNR of the test images with various noise densities

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denois.	30%	50%	70%	90%	30%	50%	70%	90%
Linear Interp.	38.57	35.20	32.12	27.73	26.60	23.91	21.76	19.34
Nearest Neighbour Interp.	34.43	31.53	29.18	25.61	23.79	21.74	20.10	17.89
Median Filter	26.11	23.68	19.34	9.47	19.01	18.14	17.23	9.36

As a final remark we would like to stress that the method of scattered data interpolation based on Delaunay triangulation provides a promising approach to remove high density impulsive noise. It is easy to implement and computationally efficient. Our numerical results show that linear and nearest-neighbour interpolation can outperform the standard median filter.

Acknowledgement

Support from EPSRC (EP/D062632/1) is gratefully acknowledged.

References

[1] R.T. Behrens and L.L. Scharf. Signal processing applications of oblique projection operators. *IEEE Transactions on Signal Processing*, 42:1413–1424, 1994.

[2] S. N. Afriat. Orthogonal and oblique projectors and the characteristics of pairs of vectors spaces. *Philos. Soc.*, 53:800–816, 1957.

[3] T. N. Greville. Solutions of the matrix equations $XAX = X$ and relations between oblique and orthogonal projectors. *SIAM Journal on Applied Mathematics*, 26:828–832, 1974.

[4] G. Gorach, A. Maestripieri, and D Stojanoff. Oblique projection and abstract splines. *Journal of Approximation Theory*, 117:189–206, 2002.

[5] A. Galantai. *Projectors and Projection Methods*. Kluwer Academic Publishers, London, 2004.

[6] G. Gorach, A. Maestripieri, and D Stojanoff. Projections in operators ranges. *Proc. Amer. Math. Soc*, 134(3):765–788, 2005.

[7] M. Unser and A. Aldroubi. A general sampling theory for nonideal acquisition devices. *IEEE Transactions Signal Processing*, 42:2915–2925, 1994.

[8] T. Blu and M. Unser. Quantitative fourier analysis of approximation techniques: part i–interpolations and projections. *IEEE Transactions Signal Processing*, 47:2783–2795, 1999.

[9] Y.C. Eldar. Sampling with arbitrary sampling and reconstruction spaces and oblique dual frame vectors. *Journal of Fourier Analysis and Applications*, 9:77–96, 2003.

[10] X. Yu and L. Tong. Joint channel and symbol estimation by oblique projections. *IEEE Transactions Signal Processing*, 49:3074–3083, 2001.

[11] L. Rebollo-Neira. Oblique matching pursuit. *IEEE Signal Processing Letters*, 14(10):703–706, 2007.

[12] B. Lamichhane and L. Rebollo-Neira. Sparse representations for structured noise filtering. *http://arxiv.org/abs/0709.2475v1*, 2007.

[13] L. Rebollo-Neira. Constructive updating/downdating of oblique projectors: a generalization of the GramSchmidt process. *Journal of Physics A: Mathematical and Theoritcal*, 40:6381–6394, 2007.

[14] K.N. Plataniotis and A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, first edition, 2000.

[15] T.F. Chan and J. Shen. *Image Processing And Analysis: Variational, PDE, Wavelet, And Stochastic Methods*. SIAM, Philadelphia, 2005.

[16] J.S. Lim. *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ, Prentice Hall, 1990.

[17] R.H. Chan, C. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, 14:1479–1485, 2005.

[18] R. Franke and G.M. Nielson. Scattered data interpolation and applications: A tutorial and survey. In H. Hagen and D. Roller, editors, *Geometric Modelling: Methods and Their Application*, pages 131–160. Springer-Verlag, 1991.

[19] I. Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11:157–176, 2002.

[20] F. R. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[21] N. Dyn, D. Levin, and S. Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10:137–154, 1990.

[22] E. Besdok. Impulsive noise suppression from images by using anfis interpolant and lillietest. *EURASIP Journal on Applied Signal Processing*, 2004:2423–2433, 2004.

[23] L. Demaret and A. Iske. Adaptive image approximation by linear splines over locally optimal Delaunay triangulations. *IEEE Signal Processing Letters*, 13:281–284, 2006.

[24] M. Reed and B. Simon. *Functional Analysis*. Academic Press, London, 1980.

[25] L. Rebollo-Neira. Dictionary redundancy elimination. *IEE Proceedings – Vision, Image and Signal Processing*, 151:31–34, 2004.

[26] Biorthogonal techniques for optimal signal representation. `http://www.ncrg.aston.ac.uk/Projects/BiOrthog/`.

[27] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, second edition, 2007.

[28] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, London, second edition, 1999.

[29] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9:137–140, 2002.

[30] M. Andrle and L. Rebollo-Neira. A swapping-based refinement of orthogonal matching pursuit strategies. *Signal Processing*, 86:480–495, 2006.

[31] Highly nonlinear approximations for sparse signal representation. `http://www.ncrg.aston.ac.uk/Projects/BiOrthog/`.

[32] D. Kincaid and W. Cheney. *Numerical Analysis*. Brooks/Cole Publishing Company, second edition, 2002.

[33] E. Süli and D. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, first edition, 2003.

[34] F.N. Fritsch and R.E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17:238–246, 1980.

[35] C. B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22:469–483, 1996.

[36] F. Aurenhammer and R. Klein. Voronoi diagrams. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290. North-Holland, Amsterdam, Netherlands, 2000.

[37] D. Eppstein. Nearest neighbors and voronoi diagrams. http://www.ics.uci.edu/~eppstein/junkyard/nn.html.

[38] Qhull computes the convex hull, Delaunay triangulation, Voronoi diagram and halfspace intersection about a point. http://www.qhull.org.