Nonequispaced discrete curvelet transform for seismic data reconstruction

by

Lloyd FÉNELON

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DIPLÔME D'INGÉNIEUR

awarded by

ÉCOLE NATIONALE SUPÉRIEURE DE PHYSIQUE
DE STRASBOURG

September, 2008

Résumé

L'acquisition des données sismiques est perturbée par les discontinuités du sous-sol ou les problèmes techniques encourus lors de l'acquisition. Les données sismiques receuillies sont alors incomplètes. Une méthode efficace de reconstruction des données sismiques s'appelle Curvelet Reconstruction by Sparsity promoting Inversion (CRSI). CRSI est une méthode d'interpolation bien adaptée aux données sismiques utilisant la transformée de curvelet. Le principal défaut de la méthode CRSI est de ne pas pouvoir interpoler de manière satisfaisante des données acquises à des positions irrégulières. Le problème se situe au niveau de l'implémentation de la transformée de curvelet qui ne gère pas ce genre de données. Dans ce rapport, l'implementation de la transformée de curvelet est modifiée afin offrir la possibilité à la méthode CRSI d'améliorer notablement la représentation des données sismiques.

Abstract

Physical constraints during seismic acquisitions lead to incomplete seismic datasets. Curvelet Reconstruction with Sparsity promoting Inversion (CRSI) is one of the most efficient interpolation method available to recover complete datasets from data with missing traces. The method uses in its definition the curvelet transform which is well suited to process seismic data. However, its main shortcoming is to not be able to provide an accurate result if the data are acquired at irregular positions. This come from the curvelet transform implementation which cannot handle this type of data. In this thesis the implementation of the curvelet transform is modified to offer the possibility to CRSI to give better representation of seismic data for high quality seismic imaging.

Table of Contents

List of Figures

Abbreviations

API	Application Programming Interface
CRSI	Curvelet Reconstruction with Sparsity promoting Inversion
DFT	Discrete Fourier Transform
FDCT	Fast Discrete Curvelet Transform
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform of the West
NCRSI	Nonequispaced Curvelet Reconstruction with Sparsity promoting Inversion
NDFT	Nonequispaced Discrete Fourier Transform
NFDCT	Nonequispaced Fast Discrete Curvelet Transform
NFFT	Nonequispaced Fast Fourier Transform
SINBAD	Seismic Imaging by Next-generation Basis-function Decomposition
SLIM	Seismic Laboratory for Imaging and Modeling

Preface

This thesis was prepared with Madagascar, a reproducible research software package available at `rsf.sf.net`. Most of the results shown in the thesis are directly linked to the code which generated them. In that sense, all figures are reproducible which facilitates the dissemination of the knowledge not only within the Seismic Laboratory for Imaging and Modeling (SLIM) but also between SLIM and its sponsors, and more generally, the entire research community.

The programs required to reproduce the reported results are Madagascar programs written in C/C++, Matlab®, or Python. The numerical algorithms and applications are mainly written in Python as part of SLIMpy (`slim.eos.ubc.ca/SLIMpy`) with a few exceptions written in Matlab® or Python.

Acknowledgments

My first and foremost thanks go to my supervisor, Gilles Hennenfent. To him goes my gratitude for his trust, his patience and his friendship with me during this internship. His reliable presence and help always motivated and guided me in this challenging experience.

Special thanks to Felix Herrmann for having given me the chance to be part of his team, for his trust and his kindness. My thanks also go to Henryk Modzelewski, Cody Brown, Sean Ross-Ross and the remainder of the SLIM team for all the help and friendship they had for me.

Many thanks to my family for their love and wholehearted support. To Simon for his friendship and the nice and pleasant time we spent together in Canada. Many thanks to Noémie who shares my adventures everyday and without whom my life would not be complete. I love you, and I always will.

To my grandfather, André Bet

Chapter 1

Introduction

This introduction gives a definition of seismic data which are processed all along this thesis. Seismic datasets are incomplete because of acquisition problems explained in the first section. Then, section 1.2 offers an overview of some of the existing methods to recover the seismic data. The outline of the thesis is presented section 1.3.

1.1 Seismic data

Seismic data are indirect measurements of the earth's subsurface structure. Figure 1.1 shows an acquisition of seismic data on land. A source, e.g. a vibroseis truck, is used to generate waves that travel through the earth. A part of these waves is reflected because of discontinuities occurring in the subsurface providing an information about the subsurface properties. The amplitudes of the reflected wavefronts are recorded as a function of time by receivers like a probe does in medical sonography. The final recording of one receiver is called a *trace*. All the traces gathered form a seismic data volume.

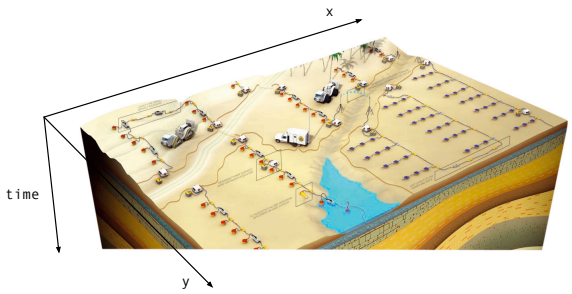
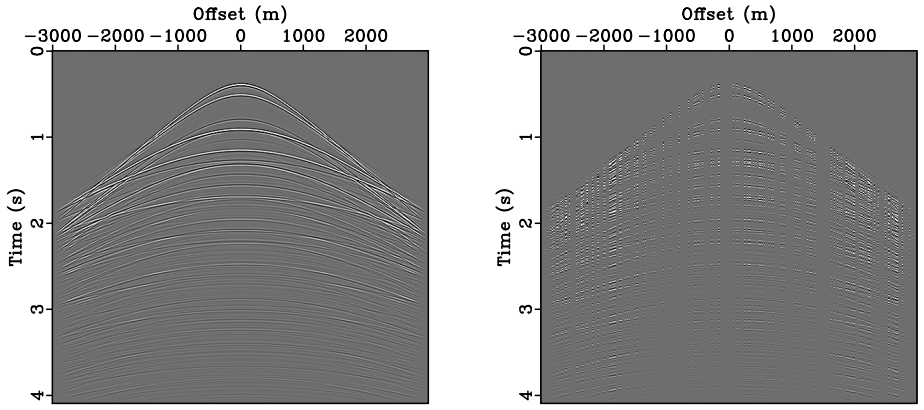


Figure 1.1: Seismic data acquisition on land. Thumper trucks generate seismic waves that travel through the subsurface. Geophones record the amplitudes of the reflected wavefields coming from the subsurface. (Image courtesy of ION Geophysical Company)

A number of fields in seismology, such as geophysics or subsurface exploration, use seismic data to produce images of the subsurface which are then analyzed. However, good images are difficult to obtain because important issues are encountered during seismic data acquisitions. Reasons for non ideal acquisitions come from two main categories which are the nature of the subsurface and the acquisition itself. Problems related to the subsurface are for instance multiple reflections from a single layer which appear as phantom layers on the final image. Complicated wave propagations or a shallow layer which absorbs too much energy and hides lower layers would also lead to bad images. Acquisition problems depend on the receivers placement on land. Obstacles, faulty equipment, errors in positioning and noise lead to improper acquisitions. Traces could be missing which results in a nonuniformly undersampled seismic dataset. Figure 2.4(a) represents a slice of a synthetic seismic data volume. The nonuniform undersampling can be simulated as shown in Figure 1.2(b) and may create aliasing in seismic images.



(a) (b)

Figure 1.2: Traces representing a slice of a synthetic seismic data volume. Amplitudes of seismic wavefronts are recorded as a function of time. Offset 0 corresponds to the source of the seismic wave firstly generated and the fastest recorded reflection. (a) Synthetic traces; (b) Corresponding simulated incomplete data with 70 % randomly missing traces.

Scientists are trying to deal with these artifacts to have better images of the subsurface. Pre-processing of seismic data like filtering helps to get rid of some acquisition problems. But to have good images an important step

is to recover the seismic data from incomplete datasets by interpolation.

1.2 Data reconstruction methods

Seismic data reconstruction methods can be classified into three categories which are filter based reconstruction, wavefield operator based reconstruction and transformation based reconstruction. The methods given below are only some examples among available methods.

1.2.1 Filter based reconstruction

The interpolation of data via filter based reconstruction is done by convolution with an interpolating filter such as the sinc function. Reconstruction of seismic data is achieved for example by the method of prediction-errors filters (Spitz, 1991; Clearbout and Nichols, 1991; Clearbout and Fomel, 2004). But filter based techniques cannot handle randomly sampled data. It is yet possible to consider nonuniformly sampled data as regular data and interpolate with a Gaussian window. But it leads to uncertain results and artifacts.

1.2.2 Wavefield operator based reconstruction

Wavefield operator based reconstruction use the Kirchhoff integral. This type of interpolation can handle nonuniformly sampled data but not necessarily coarse sampling. The technique which consists to use Kirchhoff operators via least-squares migration (Nemeth et al., 1999; Kuehl, 2002) eliminates some of the aliasing. However the computation of this method is very expensive.

1.2.3 Transformation based reconstruction

In transformation based reconstruction, the Fourier transform, Radon transform, wavelet transform or the curvelet transform is used to interpolate the data. The computation of these methods is more efficient than for the other type of reconstruction. They are faster and provide better results even though they are not perfect. Several methods have been developed for the reconstruction of wavefields in the Fourier domain (Sacchi et al., 1998; Xu et al., 2005; Zwartjes and Sacchi, 2007) and Radon domain (Trad et al., 2003). This thesis focuses on CRSI (Herrmann and Hennenfent, 2008; Hennenfent and Herrmann, 2005), a method which uses the curvelet transform.

1.3 Thesis outline

The aim of this thesis is to improve the CRSI interpolation method providing tools to handle nonuniformly sampled data without creating additional artifacts. An analysis of the CRSI method is done in chapter 2. In chapter 3, algorithms are introduced to overcome shortcomings of the CRSI computation which prevent it from ideal seismic data reconstruction. These algorithms leads to the implementation of a new curvelet transform for CRSI given in Chapter 4. The environment in which held the internship is presented chapter 5 and finally, chapter 6 is a conclusion on the improved CRSI method. It also gives an outlook about seismic data recovery and a personal opinion on the benefits of the internship.

Chapter 2

Curvelet reconstruction with sparsity-promoting inversion

This chapter introduces first the CRSI interpolation method. Then section 2.2 gives more details about the curvelet transform which is well adapted to process seismic data. However, CRSI is not a satisfying method to recover nonuniformly undersampled data. The chapter ends by giving the weaknesses of CRSI in section 2.3.

2.1 CRSI reconstruction method

The reconstruction from incomplete seismic datasets follows the forward model

$$\mathbf{y} = \mathbf{R}\mathbf{m} \tag{2.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ represents the acquired data, $\mathbf{m} \in \mathbb{R}^m$ is the model, i.e. the adequately sampled data, $\mathbf{R} \in \mathbb{R}^{n \times m}$ is the restriction operator that collects the acquired samples from $\mathbf{m}$ and $m \gg n$. The solutions of 2.1 are not unique or are acutely sensitive to changes in the data— this is an ill-posed inverse problem. Herrmann and Hennenfent (2008) suggested to reformulate as follows

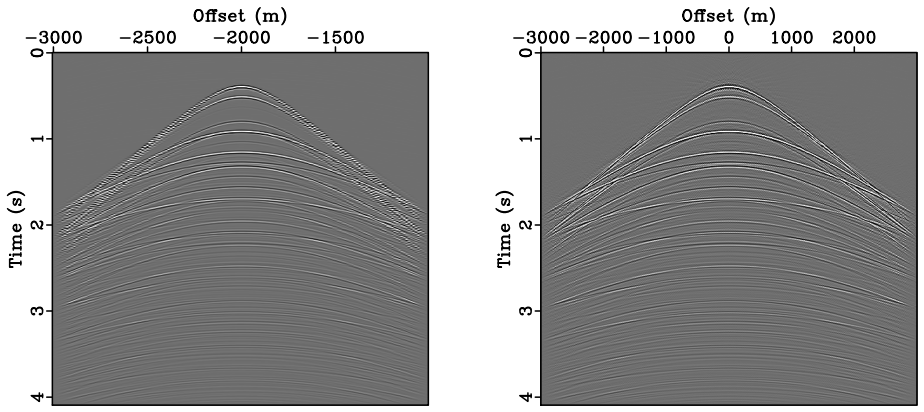
$$\mathbf{y} = \mathbf{R}\mathbf{C}^H\mathbf{x} \tag{2.2}$$

where $\mathbf{C}$ is the curvelet transform and $\mathbf{C}^H$ is its adjoint— i.e., its conjugate transpose, and $\mathbf{x} \in \mathbb{R}^N$ with $N \gg n$ is the representation of $\mathbf{m}$ in the curvelet domain. The curvelet transform gives a sparse representation of $\mathbf{m}$ which means that the vector $\mathbf{x}$ has few non zero coefficients. These properties makes it possible to successfully recover $\mathbf{m}$ according to the *compressive sampling* theory (Candès et al., 2006b; Candès, 2006; Donoho et al., 2006). But to solve an ill-posed problem additional information has to be provided. This technique is called *regularization*. The CRSI method promotes sparsity

as a regularization term and gives a solution to problem 2.2 by

$$\mathbf{P}_\sigma : \quad \begin{cases} \widetilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{R}\mathbf{C}^H\mathbf{x} - \mathbf{y}\|_2 \leq \sigma \\ \widetilde{\mathbf{m}} = \mathbf{C}^H\widetilde{\mathbf{x}} \end{cases} \tag{2.3}$$

where $\|\mathbf{x}\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^{N} |x_i|$ is the ℓ_1 norm. The recovered vector that solves $\mathbf{P}_\sigma$ is $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{m}} \in \mathbb{R}^m$ is the estimate of the recovered data obtained by applying $\mathbf{C}^H$. The data misfit is conditioned by σ which is linked to the noise variance. In our case there is no noise, then $\sigma = 0$. Figure 2.1 gives an example of synthetic seismic data interpolation using CRSI.



(a) (b)

Figure 2.1: Recovery of synthetic seismic data using the CRSI method. (a) Synthetic data randomly three-fold undersampled and (b) interpolated data using CRSI.

2.2 The Curvelet transform

The curvelet transform makes the CRSI method a successful method to recover seismic data. In this section, the curvelet properties are examined to understand why curvelets are well suited to process seismic data. In addition, we examine the curvelet transform implementation and highlight the different steps of the transform computation.

2.2.1 Curvelets properties

Curvelets were first introduced by Candès and Donoho (2000). A representation of curvelets in the physical domain and in the frequency domain is given in figure 2.3. The three main properties of curvelets are:

1. *multiscale*, with frequency support on dyadic coronae in the 2-D Fourier plane (see figure 2.2 for an example of the 2-D tilling in the Fourier domain).

2. *multidirectional*, with angles that correspond to the centers of the wedges (for every other resolution doubling, the number of angles doubles).

3. *anisotropic*, obeying the scaling law width $\propto$ length^2.

Given these properties, two main aspects of curvelets are interesting for the reconstruction of seismic data. Curvelets have a large correlation with wavefronts of seismic data—they locally have the same direction and frequency content. Hence, they can catch most of the energy of these wavefronts. Besides, curvelets provide also the sparsest representation of complex seismic data according to Candès et al. (2006a); Hennenfent and Herrmannn (2006). One can also refer to Herrmann and Hennenfent (2008) for a comparison of the sparsity on seismic data using the Fourier transform, the wavelet transform and the curvelet transform. While CRSI requests a sparse representation in its solution, the curvelet transform was a natural choice to process seismic data.

2.2.2 Implementation of the discrete curvelet transform

One method to compute the curvelet transform consists of the wrapping of specially selected Fourier samples. This algorithm is called the Fast Discrete Curvelet Transform (FDCT). The FDCT algorithm is divided into three steps. First, it computes a Fourier transform $\mathbf{F}_t$ of the data along time. Secondly, the algorithm computes the Fourier transform $\mathbf{F}_s$ along the space. In fact, these two Fourier transforms are the representation of a single 2-D Fourier transform. The third step, called the tilling transform $\mathbf{T}$, enables to go from the Fourier coefficients to the curvelet coefficients, see Candès et al. (2006a) for more details. In matrix notations FDCT in CRSI is defined as

$$\mathbf{C} \stackrel{\text{def}}{=} \mathbf{T}\mathbf{F}_t\mathbf{F}_s \tag{2.4}$$

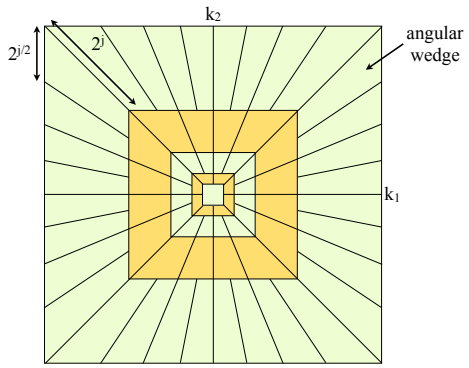


Figure 2.2: Discrete curvelet partitioning of the 2-D Fourier plane into second dyadic coronae and sub-partitioning of the coronae into angular wedges. The variable j is the curvelet scale. Each scale is represented at a number of angles that double at every other scale. (Image courtesy of Herrmann and Hennenfent (2008))

The Fourier transform algorithm used to compute the discrete curvelet transform is the Fast Fourier Transform (FFT). For a finite number of Fourier coefficients $\hat{f}_{\mathbf{k}} \in \mathbb{C}, \mathbf{k} \in \mathbb{Z}^d$, FFT estimates the Fourier transform f at regular time or space nodes $\mathbf{x}$

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{\mathbf{k}} \hat{f}_{\mathbf{k}} e^{-2\pi \mathbf{k}\mathbf{x}} \tag{2.5}$$

As explained in the next section, the implementation of the curvelet transform is one of the limitations of CRSI as it requires regular data.

2.3 Shortcomings of the CRSI method

Shortcomings CRSI are directly linked to the sampling of the seismic data. Figure 2.4 gives the three possible sampling scenarios in 1-D with spectra in the Fourier domain (the calculation of the sampling functions can be found in appendix A). In figure 2.4(a), the data are regularly sampled and the spectrum figure 2.4(b) has no artifacts. Now if the data are regularly undersampled like in figure 2.4(c), aliases are created, figure 2.4(d)). These aliases are also created by the curvelet transform. Thus, the ℓ_1 norm cannot efficiently discriminate the original spectrum from its aliases. On a slice of a seismic dataset aliases look like events with the wrong slope representing the wrong dip of the subsurface. Aliases are difficult to eliminate with a

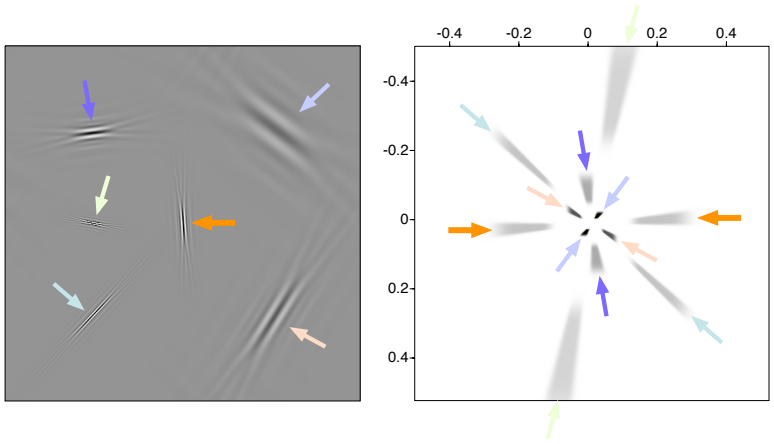
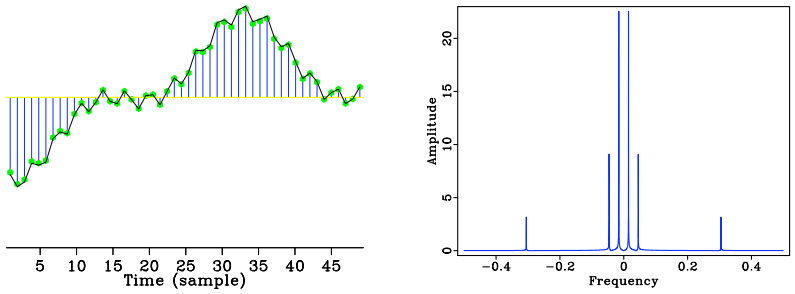


Figure 2.3: Spatial and frequency representation of curvelets. (a) Six different curvelets in the spatial domain at five different scales. (b) Dyadic partitioning in the frequency domain, where each wedge corresponds to the frequency support of a curvelet in the spatial domain. Each pair of opposing wedges represents a real curvelet. This figure illustrates the correspondence between curvelets in the physical and Fourier domain. Curvelets are characterized by rapid decay in the physical space and of compact support in the Fourier space. Notice the correspondence between the orientation of curvelets in the two domains. The 90° rotation is a property of the Fourier transform. (Image courtesy of Herrmann and Hennenfent (2008))

reconstruction method like CRSI. Other methods have been developed for this kind of problem. For example, a solution has been proposed by Xu et al. (2005) with an anti-leakage Fourier transform. The regular undersampling still is the most challenging case of seismic data recovery.

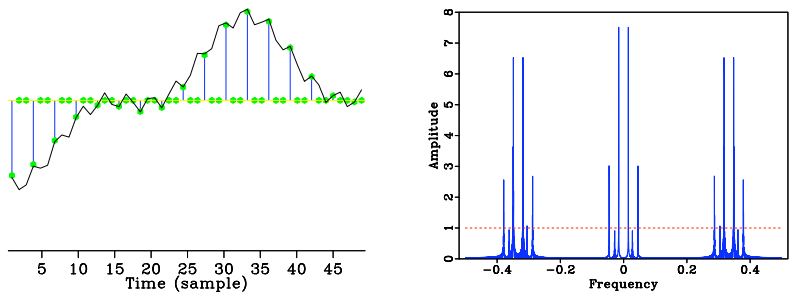
However, in practice seismic data are most of the time undersampled and acquired at irregular positions like in figure 2.4(e). It was shown in Donoho et al. (2006) that the spectral leakage can be approximated by Gaussian white noise. In some case, the noise in the spectrum figure 2.4(f) can be removed by thresholding. Then, one can retrieved the data with the inverse transform. CRSI, which is based on this principle, is trying to estimates the good curvelet coefficients from incomplete data. But if the data are recovered without taking into account their exact positions, discontinuities appear as depicted in figure 2.5. Actually, the FFTs used to compute the curvelet coefficients cannot handle irregular nodes. Thus, the curvelet coefficients given by CRSI does not explain the seismic data. The data are recovered with the wrong curvelet coefficients. To achieve better images with CRSI in case of irregular undersampling, one needs to use a Fourier transform which handles irregular positions. That what does the Nonequispaced Fast Fourier Transform (NFFT - J. Keiner, 2006).

The next chapter presents algorithms to build a new curvelet transform for CRSI. The objective is to replace the 2-D FFT in the curvelet transform with a 1-D NFFT along space and a 1-D FFT along time with the Fastest Fourier Transform of the West (FFTW - Frigo and Johnson, 1997, 1998; Frigo, 1999).



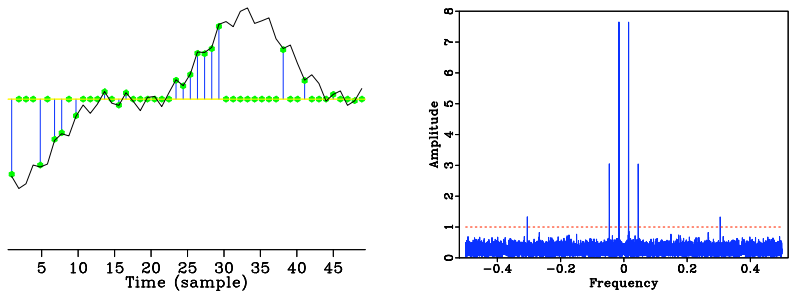
(a)

(b)



(c)

(d)



(e)

(f)

Figure 2.4: Different (under)sampling schemes and their imprint in the Fourier domain for a signal that is the superposition of three cosine functions. Signal (a) regularly sampled above Nyquist rate, (c) regularly three-fold undersampled, and (e) randomly three-fold undersampled according to a discrete uniform distribution. The respective amplitude spectra are plotted in (b), (d) and (f). Unlike aliases, the undersampling artifacts due to random undersampling can be removed using a denoising technique promoting sparsity, e.g., nonlinear thresholding (dashed line), effectively recovering the original signal. (Adapted from Hennenfent and Herrmann, 2008)

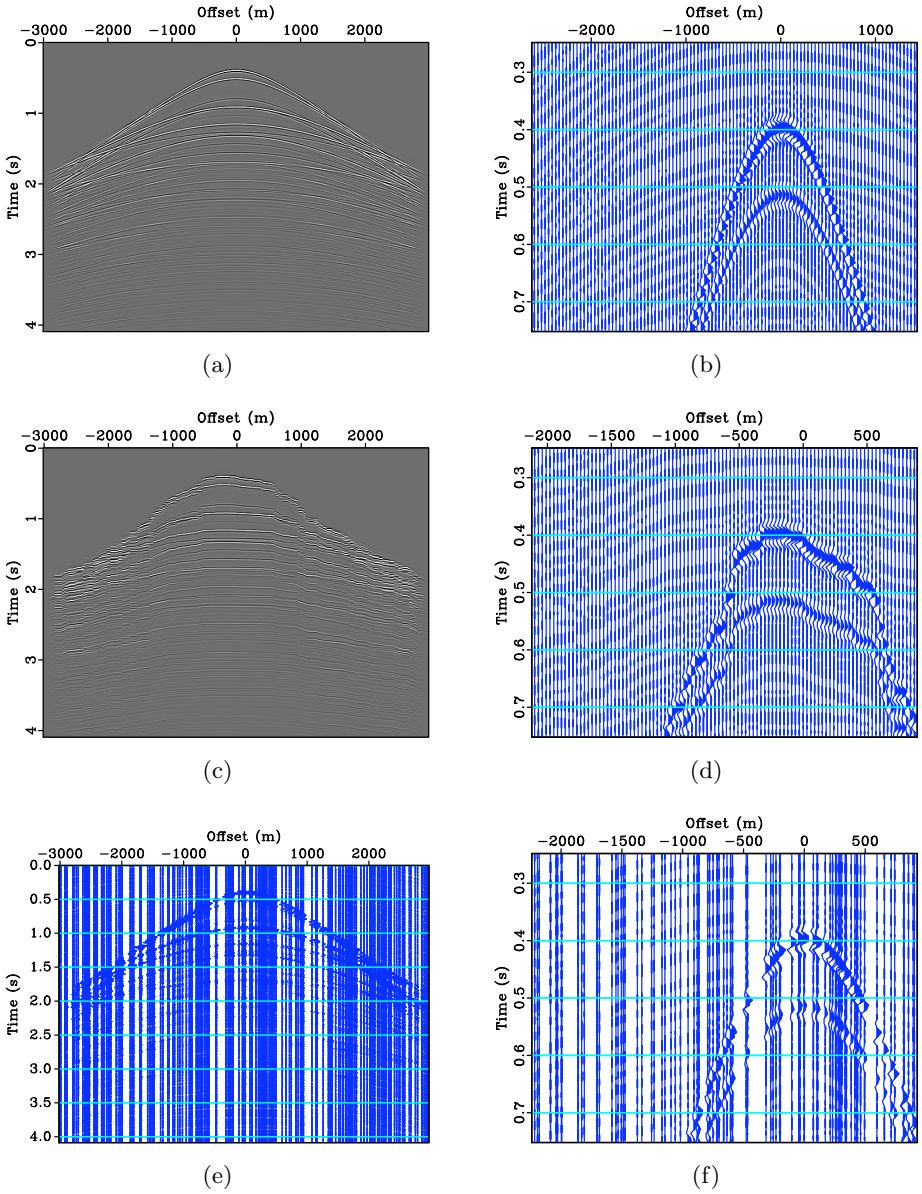


Figure 2.5: Synthetic seismic data. (a) Data sampled at regular positions; (b) regular data windowed; (c) irregularly sampled data cast to a regular grid; (d) irregular data windowed; (e) irregularly sampled data cast to an irregular grid; (f) irregular data on an irregular grid windowed. Notice the continuity along the arriving wavefront in (e) and (f). Recasting irregular data onto a regular grid destroys the continuity.

Chapter 3

New algorithms for the curvelet transform

The chapter details FFTW, NFFT and their implementation. Algorithms and basic tests of the new functions are described in the two first sections. The transforms substitute the existing FFTs in FDCT to lead to a new curvelet transform algorithm, the Nonequispaced Fast Discrete Curvelet Transform (NFDCT). NFDCT is defined in the new solution of CRSI section 3.3.

3.1 FFTW - fastest Fourier transform of the west

FFTW computes a fast Discrete Fourier Transform (DFT). It optimizes the DFT according to the underlying hardware. Hence, the computation of the transform is divided into two phases. First, FFTW learns the fastest way to compute the transform. This information is stored into a *plan*. Then, data are given to the plan which returns the Fourier coefficients. An other advantage to use FFTW is its capability to computes complex or real data of arbitrary length which property is interesting for the significant amount of seismic data (several terabytes). More details about the architecture of FFTW can be found in Frigo and Johnson (1997, 1998); Frigo (1999).

3.1.1 The discrete Fourier transform

FFTW computes an unnormalized DFT. The transform of a 1-D complex array $\mathbf{X}$ of size N computes an array $\mathbf{Y}$, where

$$\mathbf{Y}_k \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} \mathbf{X}_n e^{-2\pi i n \frac{k}{N}} \tag{3.1}$$

And the backward DFT is computed with

$$\mathbf{Y}_k \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} \mathbf{X}_n e^{2\pi i n \frac{k}{N}} \tag{3.2}$$

The forward transform is characterized by a sign of -1 in the exponent of the DFT. Since the transform is unnormalized, the forward transform of a vector followed by a backward transform will multiply the input by N. Finally, the k-th output corresponds to the frequency $\frac{k}{N}$.

3.1.2 FFTW implementation

It is described here how the FFTW function is wrapped in order to use it. The FFTW C library provides the plan for fast computation of the DFT. The code to wrap the FFTW function looks like

```
for (one trace of seismic data){

    for (each sample of the trace){
        if (inverse transform){
           shift frequencies over FNyquist/2;
        }
        create input for FFTW plan;
    }

    if (inverse transform){
       process backward FFTW plan};
    else {
       process forward FFTW plan};

    for (each sample of the trace){
        if (inverse transform){
           normalize samples from FFTW plan;
        }
        else {
           shift Fourier coefficients over FNyquist/2;
           normalize Fourier coefficients;
        }
    }
}
```

Figure 3.1: Wrapper of the FFTW function.

The function takes a real or complex data file with *rsf* format which contains the seismic data traces. The rsf format stores data values in a binary format and have the advantage to provide a header which gives all the information about the data in text format (number of traces, sampling rate, size of the file, etc...). The FFTW plan is then processed. The function allows to compute either a forward or backward normalized Fourier transform. In order to visualize the Fourier transform with frequencies centered around 0, the output array needs to be shifted by half that of the Nyquist

frequency. The inverse shift is done for the inverse transform. The output of the function is a complex data file with rsf format. The function is able to compute a 1-D Fourier transform as well as a two times 1-D transform as the code loop over traces of the data.

3.1.3 FFTW properties

Some properties of the transform are easy to verify to make sure that FFTW behaves as expected. The spectrum of a real signal $s(t)$ verify the symmetry property

$$S(f) = \overline{S(-f)} \tag{3.3}$$

The symmetry can be observed in figure 3.2(b) which is the spectrum of the superposition of three sinusoid functions. In 2-D, main directions of an image look like straight lines in the spectrum. Vertical and diagonal directions of the image figure 3.2(c) are visible in the spectrum figure 3.2(d) (revolved by 90 degrees).

The operator FFTW also satisfy the adjoint test. The adjoint of an operator generalizes the conjugate transpose of square matrices to (possibly) infinite-dimensional situations. The test consists on verifying the following property for all vector $\mathbf{x}$ and $\mathbf{y}$

$$\mathbf{x}^H \mathbf{F}_w^H \mathbf{y} = \mathbf{y}^H \mathbf{F}_w \mathbf{x} \tag{3.4}$$

where the linear operator $\mathbf{F}_w$ is the FFTW and $\mathbf{F}_w^H$ is the adjoint of $\mathbf{F}_w$. Vectors $\mathbf{x}^H$ and $\mathbf{y}^H$ denote the complex conjugate of vectors $\mathbf{x}$ and $\mathbf{y}$. Finally, the FFTW wrapper satisfies the following formula

$$\mathbf{F}_w^H \mathbf{F}_w = \mathbf{F}_w \mathbf{F}_w^H = \mathbf{I} \tag{3.5}$$

which means that the adjoint of FFTW is its inverse transform.

3.2 NFFT - nonequispaced fast Fourier transform

Ordinary, FFTs compute the Fourier transform of signals regularly sampled which satisfy the Nyquist-Shannon theory: the sampling frequency must be at least twice equal that of the maximum frequency of the signal. The non respect of this condition leads to artifacts. Hence, irregularly (under)sampled signals are miss-processed. NFFT overcomes this shortcoming by computing the Nonequispaced Discrete Fourier Transform (NDFT). The NDFT uses the information on the positions of samples to compute the

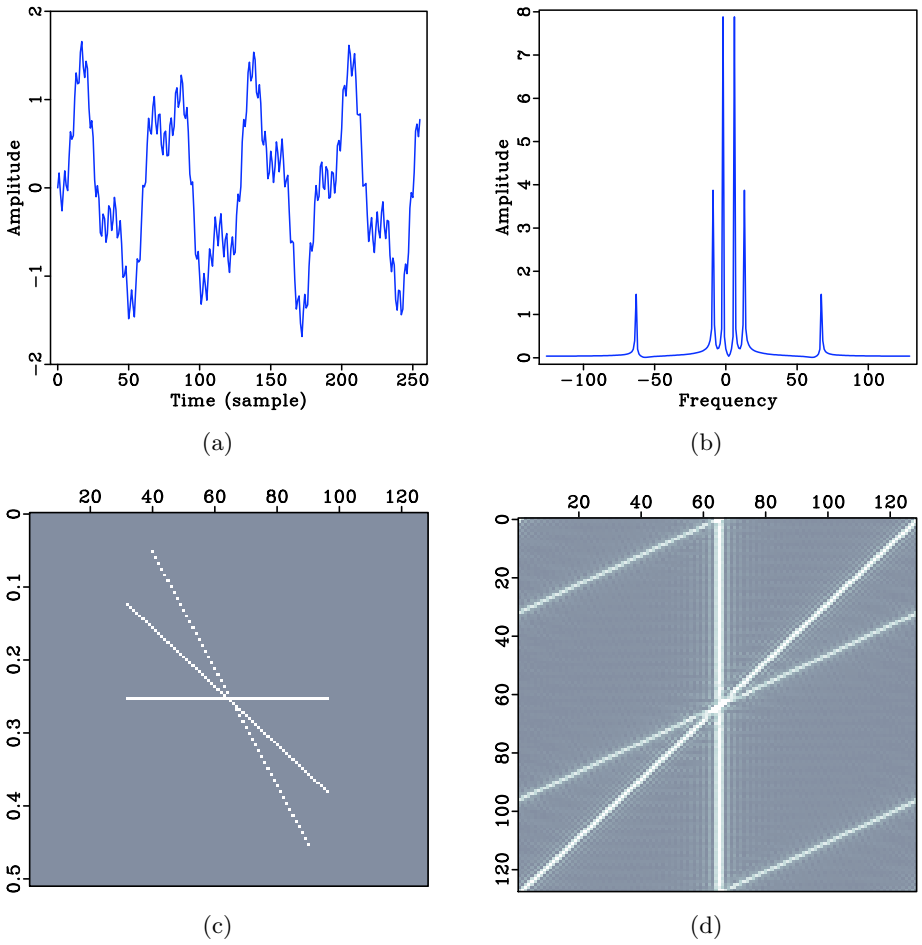


Figure 3.2: Fourier properties with FFTW. (a) 1-D signal made of the superposition of three sinusoids with different frequencies and amplitudes; (b) spectrum of the 1-D signal with symmetry around zero; (c) 2-D image whose main directions are horizontal and diagonal. The directions are visible in the spectrum (d) but revolved by 90 degrees.

Fourier transform. Nodes of samples are first stored into a *plan*. Then, samples are given to the plan which estimates the good Fourier coefficients. More details about the procedure for computing the NFFT can be found in J. Keiner (2006) for example.

3.2.1 The ansatz

The NDFT reads the following sum

$$f(\mathbf{x}_j) \overset{def}{=} \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k}\mathbf{x}_j} \tag{3.6}$$

where $\{\mathbf{x}_j\}_{j=0,\dots,M-1}$ are the nonequispaced nodes of the interpolated data, $\hat{f}_{\mathbf{k}} \in \mathbb{C}$ are a finite number of given Fourier coefficients and $\mathbf{k}$ are the possible frequencies. In matrix notation the NDFT becomes

$$\boldsymbol{f} = \mathbf{N}\hat{\boldsymbol{f}} \tag{3.7}$$

Where

$$\boldsymbol{f} \overset{\text{def}}{=} (f_j)_{j=0,\dots,M-1}, \quad \mathbf{N} \overset{\text{def}}{=} (e^{2\pi i \mathbf{k}\mathbf{x}_j}), \quad \hat{\boldsymbol{f}} \overset{\text{def}}{=} (\hat{f}_{\mathbf{k}})_{\mathbf{k}\in\mathbb{Z}^d}$$

Estimated frequencies are computed thanks to the adjoint $\mathbf{N}^H$ of the nonequispaced Fourier matrix $\mathbf{N}$. Frequencies are given by

$$\hat{\boldsymbol{h}} = \mathbf{N}^H \boldsymbol{f}, \quad \hat{h}_{\mathbf{k}} = \sum_{j=0}^{M-1} f_j e^{-2\pi i \mathbf{k}\mathbf{x}_j} \tag{3.8}$$

The NFFT satisfies the adjoint test equation 3.4. But for nonequispaced nodes the adjoint NFFT is not the inverse transform like the adjoint of FFTW. The NFFT operator does **not** verified the inversion formula $\mathbf{N}^H\mathbf{N} = \mathbf{I}$.

3.2.2 NFFT implementation

The NFFT C library is wrapped in the NFFT function. The wrapper is summarized in figure 3.3 and has three steps. First, it requires some information. One need to specify the number of Fourier coefficients, the number of nonequispaced nodes and the value of each node. Nodes come

from an external file to provide to the function and values must be in [-0.5,0.5[. The second step is to give the data to the plan which will compute the transform. Data are finally normalized as the NDFT is unnormalized and would multiply the coefficients by $\sqrt{N}$. The input is either real or complex with rsf format and the output is an rsf complex file of normalized data. The NFFT function allows to compute a 1-D nonequispaced Fourier transform or a two times 1-D transform as the code loops over the traces of the data.

```
set NFFT plan(
   # of Fourier coefficients
   # of nonequispaced nodes
   )

read positions(values in [-0.5,0.5])
set NFFT plan(nodes)

for (one trace of seismic data){

   create input for NFFT plan

   if (inverse transform){
      Process backward NFFT plan
   }
   else {
      Process forward NFFT plan
   }
   normalize output of NFFT plan
}
```

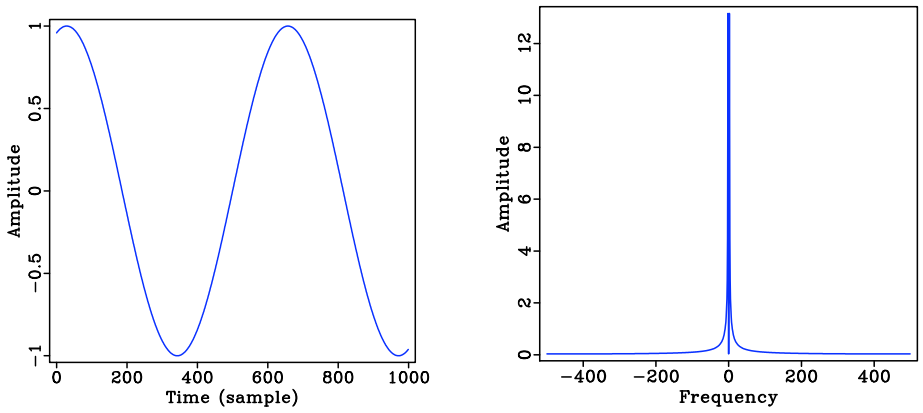
Figure 3.3: Wrapper of the NFFT function.

3.2.3 Behavior of NFFT with nonequispaced sampling

In case of regular sampling, the NFFT behaves like a regular FFT. The spectrum and the inverse transform will give the same results as FFTW (example figure 3.4).

For irregular sampling, NFFT is able to compute the number of frequencies desired or samples at irregular nodes provided that the positions are given. In figure 3.5(a), irregular samples are displayed at their irregular positions. This is the good signal to process. In figure 3.5(b), samples are casted on a regular grid, which lead to a wrong signal. A regular FFT will consider the samples at regular positions (wrong signal) and thus the frequencies computed will be wrong too. NFFT takes samples with their

irregular positions (good signal) and estimates the transform. Because the frequencies are computed from irregular positions a little error is induced. This comes from the fact that the columns of the matrix $\mathbf{N}^H$ are not orthogonal. This error is also generated if an undersampled signal is processed with NFFT. In the spectrum, the error looks like noise, see figure 3.6. This noise is increasing as the signal gets more undersampled. Nevertheless, one can easily identify the picks in the spectrum. With an optimization problem, it becomes possible to compute the good coefficients of such a signal which will be recovered by applying the inverse Fourier transform. For seismic data recovery with CRSI, it is proposed to wrap the NFFT in the curvelet transform as explained in the following section.

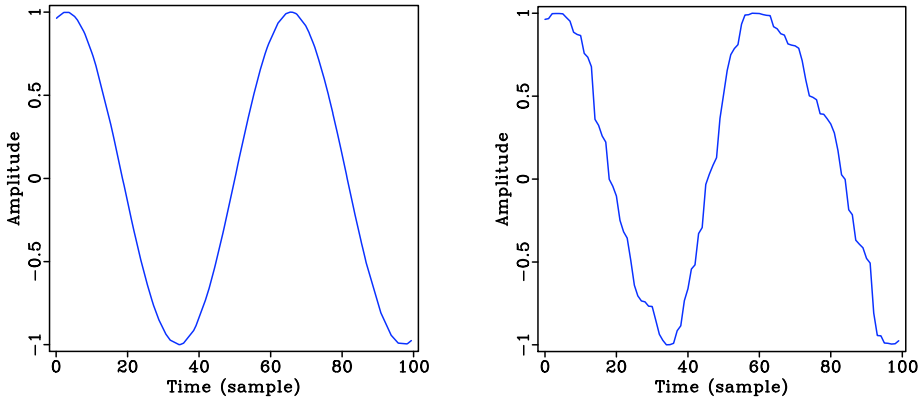


(a) (b)

Figure 3.4: Regular sampled signal and NFFT. (a) Signal and (b) its spectrum obtained by applying NFFT. The spectrum is the same as if a regular FFT would have computed it.

3.3 NFDCT - nonequispaced fast discrete curvelet transform

The existing curvelet transform used in CRSI is not able to give the good curvelet coefficients of a nonuniform signal. It is just used to process regularly sampled seismic dataset. Given the properties of NFFT, it is possible to modify the curvelet implementation to include NFFT in it to create the Nonequispaced Fast Discrete Curvelet Transform.



(a) (b)

Figure 3.5: Representation of a signal irregularly sampled. (a) Sinusoid with samples at irregular positions on a regular grid; (b) is the same sinusoid but plotted on a regular grid. A regular FFT will compute the coefficients with regular nodes which is not the signal that is expected to process. NFFT will process the good signal as the positions are given to it.

3.3.1 NFDCT implementation

As explained in subsection 2.2.2, the FDCT algorithm is composed of a 2-D Fourier transforms and what is called the tilling of the curvelet coefficients. As the time domain remains regularly sampled FFTW will $\mathbf{F}_w$ in NFDCT. In order to take into account positions in space, NFFT will play the second Fourier transform in NFDCT. The tilling of coefficients remains the same. Hence, the matrix definition of NFDCT is

$$\mathbf{C}_N \stackrel{\text{def}}{=} \mathbf{T}\mathbf{F}_w\mathbf{N} \tag{3.9}$$

Subsequently, the code of NFDCT to process 2-D signals is organized like in figure 3.7. The input is either an rsf file of curvelet coefficients or an image. All parameters of NFFT must be given without omitting the file of positions. Besides curvelets parameters must be specified too. They are the number of scale and the number of angles at second coarsest scale. The output is an rsf file of samples computed at given positions or an rsf file of curvelet coefficients.

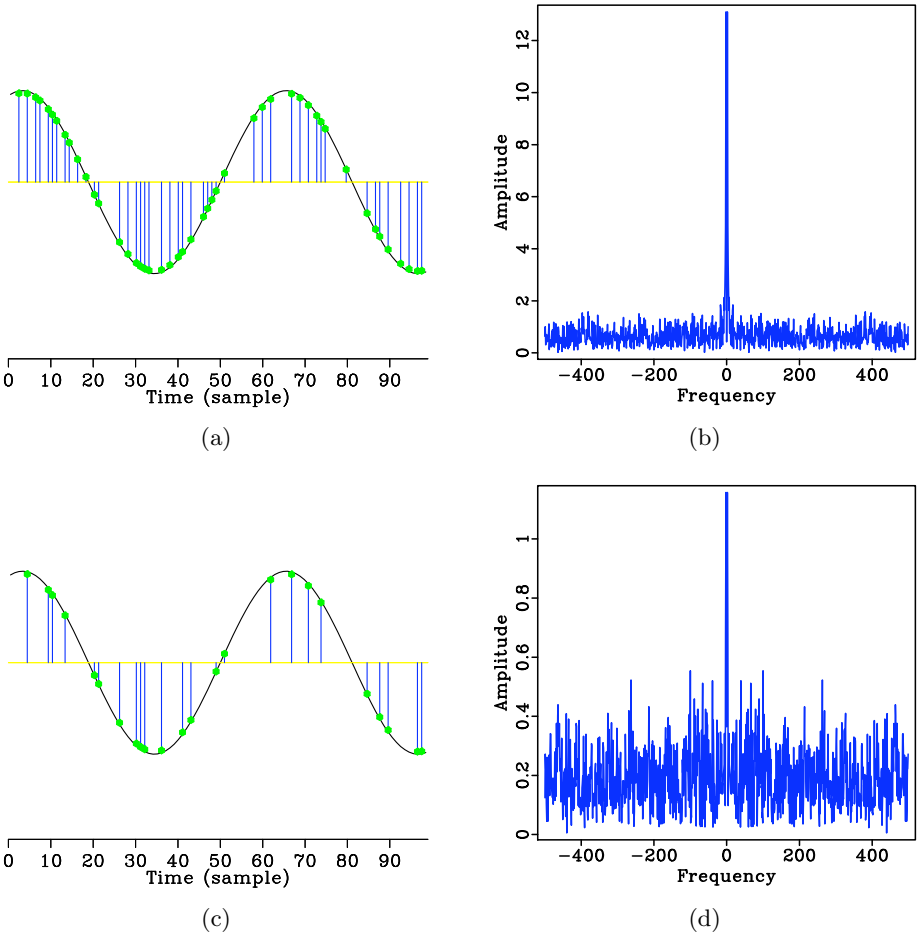


Figure 3.6: Spectrum of an undersampled signal with NFFT.(a) The computation of the frequencies with samples at irregular positions induces an error which looks like white noise. (b) The error increases if the input signal is even more undersampled.

```
if (inverse transform){
   input : vector of curvelet coefficients
           # of Fourier coeficients
           external file of positions
           curvelet parameters

    1: Tilling
    2: NFFT in space direction
    3: FFTW in time direction

   output : 2D data with samples at given positions
}
else {
   input : 2D data
           # of Fourier coeficients
           external file of positions
           curvelet parameters

   1: Adjoint FFTW in time direction
   2: Adjoint NFFT in space direction
   3: Inverse tilling

   output : vector of curvelet coefficients
}
```

Figure 3.7: Implementation of the NFDCT function.

3.3.2 Examples using NFDCT

Two examples are given here to understand how NFDCT behaves on seismic data. First, figure 3.8 shows that NFDCT is able to represent any uniform seismic data at any positions. NFDCT calculates samples at irregular positions by applying the inverse transform on a curvelet vector. The second example figure 3.9 illustrates that the same inverse transform does not work on nonuniform signals. Actually, NFDCT passes the adjoint test but the adjoint is not the inverse. This property comes from the NFFT wrapped in NFDCT which adjoint is not the inverse either for nonuniform signals. Hence, NFDCT does **not** verifies the formula $\mathbf{C_N}^H \mathbf{C_N} = \mathbf{I}$. Consequently, curvelet coefficients are slightly wrong and the good signal cannot be recovered. To recover complete data from nonuniform signals one need the CRSI optimization which will adjust the curvelet coefficients. Examples of reconstruction with CRSI and NFDCT are given in the next chapter.

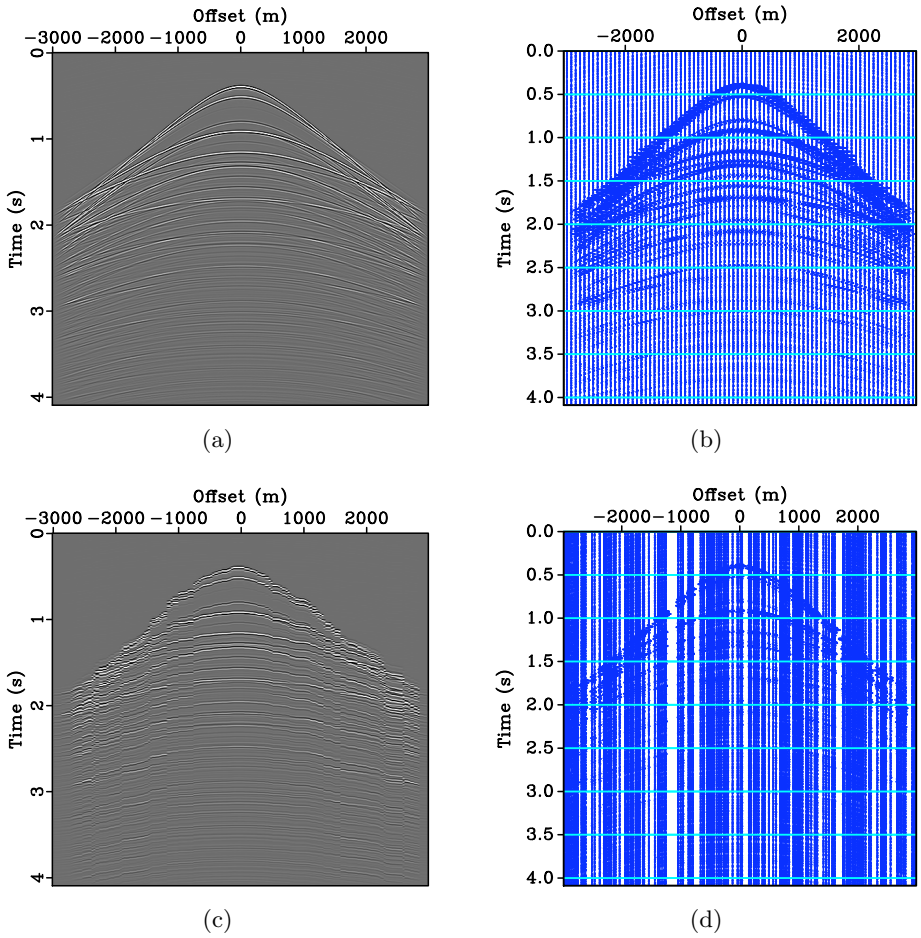
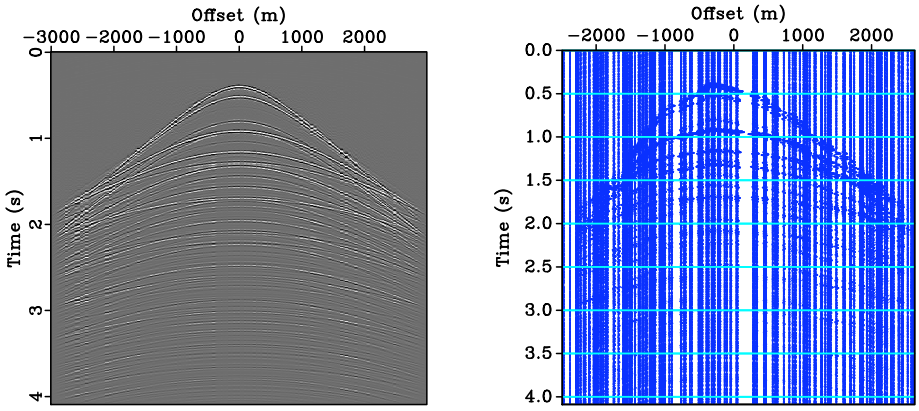


Figure 3.8: Reconstruction with NFDCT. (a) Synthetic seismic data; (b) same data with a wiggle plot. The grid is regular, so are the data. (c) The regular data after a forward regular curvelet transform followed by inverse curvelet transform with NFDCT. Samples of the recovered data are computes at irregular nodes. The figure shows samples casted on a regular grid; (d) is the recovered data displayed at their real irregular positions.



(a) (b)

Figure 3.9: Effect of the inverse NFDCT. (a) The recovery with NFDCT of the regular data figure 3.8(a) after applying the forward NFDCT transform with irregular positions. The inverse of NFDCT on NFDCT does not give the same initial data for irregular positions. The difference is more visible between wiggle plots (b) and figure 3.8(d).

Chapter 4

Reconstruction of seismic data with NFDCT

In this chapter the algorithm of the curvelet transform in CRSI is changed to NFDCT. The definition of the Nonequispaced Curvelet Transform by Sparsity promoting Inversion (NCRSI) is given in the first section. Results of interpolation on seismic data with NCRSI are given in section 4.2.

4.1 NCRSI definition

The data interpolation problem reviewed in chapter 2 is reminded. One want to recover seismic data $\mathbf{x}$ from observations $\mathbf{y}$ via a modeling matrix $\mathbf{A}$

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad \text{with} \quad \mathbf{A} = \mathbf{R}\mathbf{C}^{\mathbf{H}}$$

The CRSI method uses curvelets as it provides the sparsest representation of the data. This representation is required by the compressive sampling theory to successfully recover the data. The CRSI method gives a solution by resolving

$$\mathbf{P}_\sigma : \qquad \begin{cases} \widetilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \sigma \\ \widetilde{\mathbf{f}} = \mathbf{C}^H \widetilde{\mathbf{x}} \end{cases}$$

Where $\mathbf{C} \overset{\text{def}}{=} \mathbf{T}\mathbf{F}_t\mathbf{F}_s$ is the curvelet transform. The nonequispaced fast discrete curvelet transform $\mathbf{C}_N$ able to process nonuniform signals is now proposed

$$\mathbf{C}_N \overset{\text{def}}{=} \mathbf{T}\mathbf{F}_w\mathbf{N}$$

Consequently, the modeling matrix for CRSI becomes

$$\mathbf{A} \overset{\text{def}}{=} \mathbf{C}_N^H \tag{4.1}$$

This new CRSI interpolation method for seismic data offers better seismic data images as shown in the following section.

4.2 Results of NCRSI interpolation

The first result figure 4.1(c) is an example of interpolation from incomplete synthetic seismic data with NCRSI. To generate this result the initial data 4.1(a) originally composed of 480 traces are decimated by 3 at irregular positions. The interpolation gives a result close to the original data.

The second example is a comparison of NCRSI with CRSI. CRSI is not able to process nonuniform data but it is possible to preprocess the data in order to use it. This preprocessing is called the *binning*. The binning consists in an average of the positions which neighbored a regular position. It is an approximation from irregularly located values to regular values. The complete synthetic data are used to generate 160 new traces at irregular locations.

NCRSI gives a sharper result with no discontinuities. Slopes of the wavefront are smoother than with the interpolation with CRSI. Figure 4.1(e) and 4.1(f) are the differences of the original data and the recovered data with respectively NCRSI and CRSI. The difference is much less with the result from NCRSI. The signal-to-noise ratio (SNR) shows the same result. The SNR is defined by $20 \cdot \log_{10}\left(\|\mathbf{f}\|_2 / \|\mathbf{f} - \tilde{\mathbf{f}}\|_2\right)$ where $\mathbf{f}$ represents the reference data and $\tilde{\mathbf{f}}$ the reconstruction. The higher the SNR is the more faithful is the reconstruction. In case of CRSI interpolation the SNR is equal to 13.97. The SNR with NCRSI has the much higher SNR of 28.13. The NCRSI method offers a better interpolation than CRSI with binning.

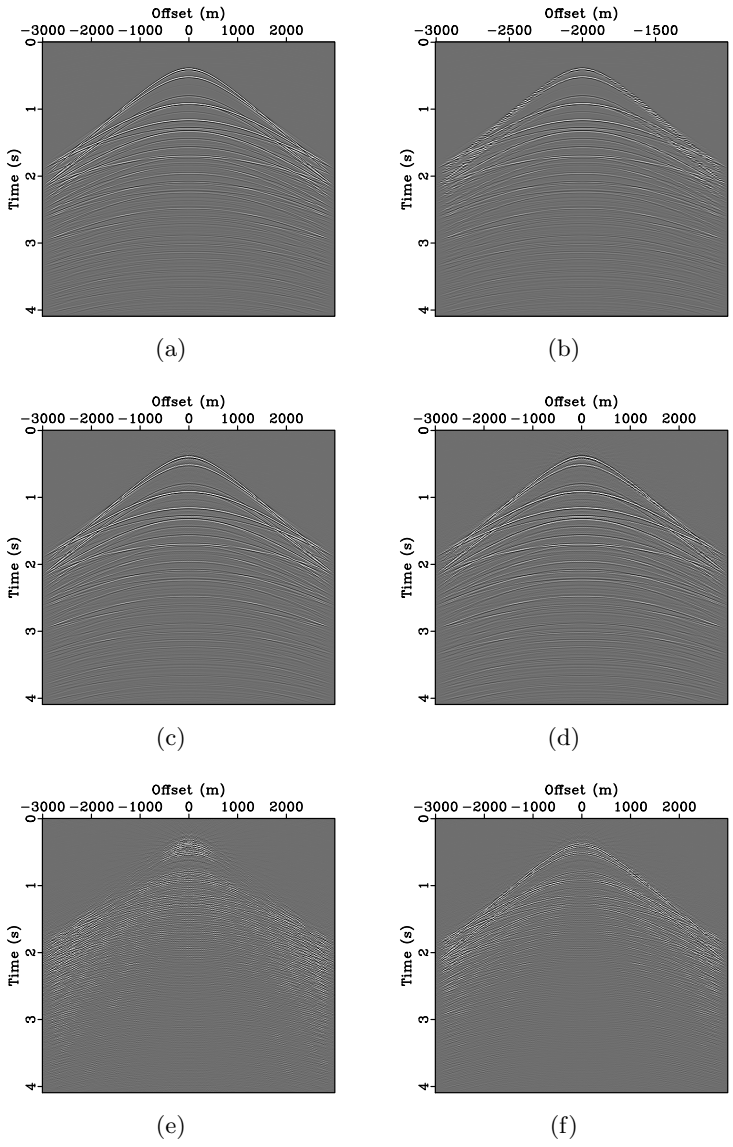


Figure 4.1: Seismic data interpolation with NCRSI. (a) Synthetic seismic data; (b) seismic data with irregular positions and randomly three times less missing traces; (c) interpolated seismic data from (b) with NCRSI, the difference between (c) and (a) is plotted in (e); (d) interpolated data with CRSI from (b) after binning. Image (f) is the difference between (a) and (d).

Chapter 5

Working environment and Engineering facilities

This chapter is an overview of working conditions as well as methods and tools that have been used to achieve the project. The Seismic Laboratory for Imaging and Modeling where the internship held is presented first with its workings. Then, descriptions of some computer software and hardware are given to highlight engineering skills that have been developed.

5.1 Seismic Laboratory for Imaging and Modeling

The SLIM is a laboratory part of the Earth and Ocean Sciences (EOS) department in the University of British Columbia (UBC), Vancouver, Canada (more details about UBC and the EOS department can be found in appendix B). It is working towards improving the recovery of images from noisy and incomplete data. The SLIM's work efforts are in response to calls of proposals from various oil companies such as Exxon Mobil, Shell, BP, BG group or Chevron.

SLIM projects

Projects in progress at SLIM aim to design the next generation of seismic processing and imaging tools like the project SINBAD (Seismic Imaging by Next-generation Basis-function Decomposition) in cooperation with the company Industry Technology Facilitator. The ultimate objective of SINBAD is to apply theoretical results to real seismic-industry driven imaging problems. SINBAD plans to bring about significant improvements to the signal-to-noise ratio and resolution of seismic images, which will be of great benefit to the entire oil and gas industry.

SLIM organization

The SLIM group is composed of thirteen people from almost the same number of countries who work on the different projects. Projects at SLIM are hold in collaboration with the faculty of Computer Science and the faculty of Mathematics. Thus weekly meetings help to plan the work :

1. SLIM group meeting.
2. Meeting joining groups from the three faculties.
3. Internship meeting. This meeting was specific to this internship.

Mailing lists and blogs are also widely used to keep in touch with news and progress. Finally, work is shared easily by using the version control system SVN partly described in the next section.

5.2 Software

All functions developed to achieve NCRSI, namely the function FFTW, NFFT and NFDCT have taken part to SLIM software development. A global view of the function coded can be found in appendix C. These tools have been integrated in the software MADAGASCAR. It is an open-source software mainly used for geophysical data analysis and reproducible numerical experiments. Thereby, every figures in this thesis related to the three functions are reproducible with MADAGASCAR. NFDCT and NCRSI have been wrapped in the SLIM software SLIMpy entirely written in python. A quick descriptions of the most relevant software used during the internship is now given. By alphabetical order :

Curvelab

CurveLab is a toolbox implementing the Fast Discrete Curvelet Transform, both in Matlab and C++. It was used with Matlab to understand the effect of the curvelet transform on seismic data. `http://www.curvelet.org/`.

LaTeX

LaTeX is a high-quality typesetting system. It includes features designed for the production of technical and scientific documentation. LaTeX is a standard for the communication and publication of scientific documents.

This thesis is entirely written in LaTeX. LaTeX is available as free software. http://www.latex-project.org/.

Google Calendar

Google Calendar is a contact- and time-management web application. It is a free software very useful to organize the planning with others planning. It runs on virtually any operating system, provided that the OS has a browser which supports the required web technologies. http://www.google.com/intl/en/googlecalendar/tour.html.

Madagascar

Madagascar is an open-source software package for geophysical data analysis and reproducible numerical experiments. Its mission is to provide

1. a convenient and powerful environment

2. a convenient technology transfer tool

for researchers working with digital image and data processing. Madagascar is the software which make results in this thesis reproducible. http://rsf.sourceforge.net/.

Matlab

Matlab is a numerical computing environment and programming language. It allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Matlab was used with Curvelab and in many times to understand or verify signal processing experiments. This software is not free.

Python

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. Python is under open source license so that is free to use. It was used for the development of the software SLIMpy. http://www.python.org/.

SCons

SCons is an open source software build tool. SCons is a substitute for the classic Make utility with integrated functionality similar to autoconf/automake and compiler caches such as ccache. Compared to earlier tools, SCons aims to be easier to use, more reliable, and faster. `http://www.scons.org/`.

SLIMpy

SLIMpy is a Python interface that exposes the functionality of seismic data processing packages, such as MADAGASCAR, through operator overloading. SLIMPy provides a concrete coordinate-free implementation of classes for out-of-core linear (implicit matrix-vector), and element-wise operations, including calculation of norms and other basic vector operations. Algorithms NFDCT and NCRSI created during the internship were included in this software by developing the appropriate SLIMpy Application Programming Interfaces (API). `http://slim.eos.ubc.ca/SLIMpy`.

SVN

Subversion is an open source version control system. It was used to maintain current and historical versions of files such as source code, web pages, reports etc. `http://subversion.tigris.org/`.

5.3 Hardware

The SLIM hardware is based on a computer cluster, i.e. a group of linked computers working together closely so that it forms a high-performance distributed-memory system with fast inter-processor communication layer. Computers of a cluster are called the nodes of the cluster. SLIM cluster has 144 CPUs and is divided into compute nodes, storage, login nodes and management nodes. Each compute node has 8GB of memory and 4 CPUs. I.e., there is 2GB of memory per CPU on average. The CPUs and memory are shared by the tasks running on the compute node. Each login node has 8GB of memory and 4 CPUs. The CPUs and memory are shared by all users that are logged-in and running applications on the login node.

For the SLIM user, the cluster is a good tool for heavy calculations. For storage and small dataset calculations, two other servers are also used by the SLIM. Users have access to the different servers from Mac OS X via ssh.

Chapter 6

Conclusions and outlook

The conclusion is divided into two parts. The first part gives a conclusion about the research done to improve seismic data reconstruction and what results are expected on real seismic data. The second part goes back to the experience and engineering knowledge acquired during the internship. It is a more personal feeling about the internship.

6.1 Research work

The CRSI reconstruction method is improved from its lack to represent seismic data at irregular positions. This has been possible by providing CRSI a new curvelet transform more powerful and robust for this type of data. Indeed, the curvelet transform which is an outstanding transform to process seismic data is now capable to take into account singular data locations. Results shows that NCRSI is even more accurate and sharp than CRSI coupled with additional technique for seismic data recovery. Other research themes in geophysics gives for example more information about the best sampling scheme to adopt for seismic data. Combined to a reconstruction technique like NCRSI, seismic data reconstruction can hope efficient images for better interpretation. The benefit is even more interesting for companies which are looking for subsurface images with very high quality and for signal processing in general.

6.2 The engineer opinion

This internship was the best way to perfect general engineering skills and knowledge in data acquisition and signal processing. On one hand, experience have been reach thanks to the multidisciplinary topics of the internship. From geophysics and acquisition of seismic data to seismic data processing and computer sciences, multiple knowledge and adaptation have been necessary. On the other hand, the heart of signal processing have been explored with the implementation of several signal processing functions in different

essential languages, namely the C, C++ and python. This internship was also the source of scientific and human exchange on different topics, on different opinions. Mathematics, physics and geophysics are amongst the many themes seminars and weekly group meetings gave to think about. With an other approach, an other problem and hopefully an other solution. Finally, The international environment which is the university of British Colombia and the SLIM was extremely favorable to interesting and rich scientific and human sharing.

Appendix A

Sampling and undersampling

For the convenience of the reader, the calculation of the sampling function of the example figure 2.4 is derived below.

Lets us give some sampling locations r_n

$$r_n = n\gamma + \varepsilon_n \quad \text{for} \quad n = -\infty, \ldots, \infty \tag{A.1}$$

where ε_n is a random variable, independent, identically distributed (IID), γ is the undersampling factor. The sampling operator for a finite signal of length N is given by

$$\begin{aligned} s(r) &= \sum_{n=0}^{N-1} \delta(r - r_n) \\ &= \sum_{n=0}^{N-=1} \delta(r - (n\gamma + \varepsilon_n)) \\ &= \sum_{n=0}^{N-1} \delta(r - n\gamma) * \delta(r - \varepsilon_n) \end{aligned} \tag{A.2}$$

If $\mathcal{F}$ is the Fourier operator, the Fourier transform of the sampling function is

$$\begin{aligned} S(f) &= \sum_{n=0}^{N-1} \mathcal{F}\left\{\delta(r - \gamma n)\right\} \cdot \mathcal{F}\left\{\delta(r - \varepsilon_n)\right\} \\ &= \sum_{n=0}^{N-1} \mathcal{F}\left\{\delta(r - \gamma n)\right\} \cdot e^{-j2\pi\varepsilon_n f} \\ &= \mathcal{F}\left\{\sum_{n=0}^{N-1} \delta(r - \gamma n)\right\} \cdot e^{-j2\pi\varepsilon_n f} \\ &= \frac{1}{\gamma} \cdot \sum_{n=0}^{N-1} \delta(f - \frac{n}{\gamma}) e^{-j2\pi\varepsilon_n f} \end{aligned} \tag{A.3}$$

For regular sampling the spectral amplitude of the sampling function corresponds to for $\varepsilon_n = 0$ and $\gamma = 1$

$$|S(f)| = \left| \sum_{n=0}^{N-1} \delta(f-n) e^{-j2\pi f} \right| \tag{A.4}$$

For regular undersampling ε is also equal to zero and the amplitude spectrum of the sampling operator is

$$|S(f)| = \left| \frac{1}{\gamma} \cdot \sum_{n=-\infty}^{\infty} \delta(f-\frac{n}{\gamma}) e^{-j2\pi f} \right| \tag{A.5}$$

For example if $\gamma = 3$, the product of the spectrum of the signal with the spectrum of the sampling function gives the amplitude spectrum figure 2.4(d).

The spectrum of the sampling operator for irregular undersampling is more complicated. It brings into play the probability density function $\hat{p}(f)$ of ε_n and the spectrum is formulated with the expected value of the sampling function as follow

$$E\left\{\hat{s}(f)\right\} = \hat{p}(f) \cdot \frac{1}{\gamma} \sum_{n=0}^{N-1} \delta(f-\frac{n}{\gamma}) \tag{A.6}$$

More detail about this result can be found in Hennenfent and Herrmann (2008).

Appendix B

University of British Columbia

Informations in this appendix are taken from the UBC website `http://www.ubc.ca/` and the famous free en encyclopedia Wikipedia `http://www.wikipedia.org/`. More details are available on these websites.

B.1 The university

The University of British Columbia is a Canadian public research university with campuses in Vancouver and Kelowna, British Columbia. The Vancouver campus is located on Point Grey, a peninsula about 10 km from downtown Vancouver. While the originating legislation created UBC in 1908, the first day of lectures was September 30, 1915. On September 22, 1925, lectures began on the new Point Grey campus.

In 2003, UBC had 3.167 full-time faculty teaching 33.566 undergraduate students and 7.379 graduate students. The UBC library, which comprises 4.7 million books and journals, is the second largest research library in Canada. UBC consistently ranks as one of the top three Canadian universities.

B.2 The Earth and Ocean Sciences department

The Department of Earth and Ocean Sciences was formed in April 1996 with the amalgamation of the old Departments of Geological Sciences, Oceanography, and the geophysics component of Geophysics and Astronomy. In April 2000 the Department considerably expanded its activity in atmospheric sciences.

The research focus of the new department extends from pure science studies of the earth's deep interior, through near-surface geological studies and environmental earth science, to the oceans and atmosphere. To realize their research objectives, UBC earth scientists draw on a broad base of

knowledge from the basic sciences of chemistry, physics, biology and mathematics. An integrated approach is increasingly required to understand the complex interactions due to human impact on the earth in such problems as climate change, resource exploitation and waste disposal. One specialized group is the Mineral Deposit Research Unit (MDRU) which undertakes programs supported by the mineral exploration industry.

Appendix C

Schematic view of implemented functions

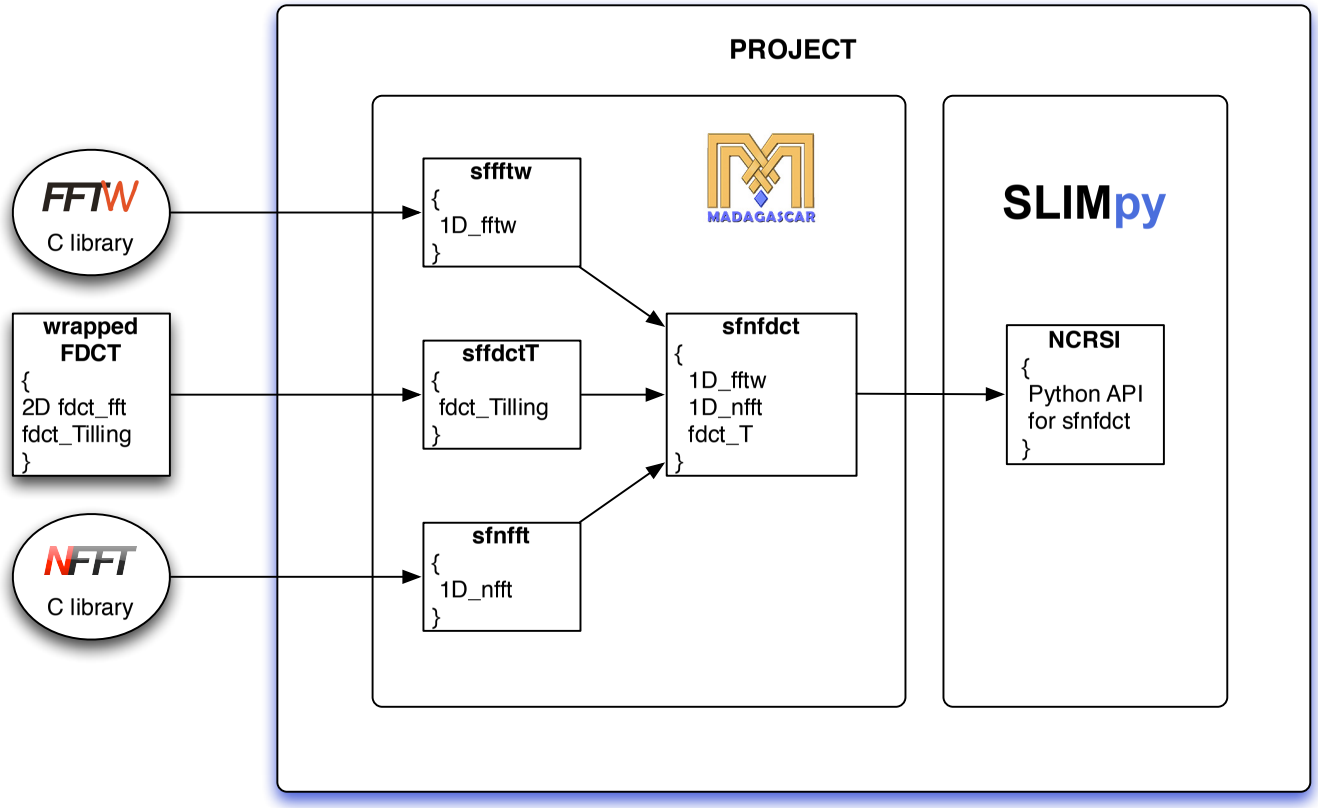


Figure C.1: Diagram of the functions created during the internship. Functions *sffftw, sffdctT and sfnffft* developed in the Madagascar software environment are respectively based on the FFTW C library, the NFFT C library and the wrapper of the FDCT function for Madagascar. The nonequispaced curvelet transform function *sfnfdct* for Madagascar uses the code of the previous functions cited. API for *sfnfdct* was written for the SLIMpy software.

Bibliography

Candès, E. J., 2006, Compressive sampling: Presented at the Proceeding of the international Congress of Mathematicians, Madrid, Spain.

Candès, E. J., L. Demanet, D. L. Donoho, and L. Ying, 2006a, Fast discrete curvelet transforms: SIAM Multiscale Model. Simul., **5**, 861–899.

Candès, E. J. and D. L. Donoho, 2000, Curvelets – a surprisingly effective nonadaptive representation for objects with edges: Presented at the Curves and Surfaces, Vanderbilt University Press.

Candès, E. J., J. Romberg, and T. Tao, 2006b, Stable signal recovery from incomplete and inaccurate measurements: Communications on Pure and Applied Mathematics, **59**, 1207–1223.

Clearbout, J. and S. Fomel, 2004, Image estimation by example.

Clearbout, J. and D. Nichols, 1991, Interpolationbeyond aliasing by (tau,x)-domain pefs: European Association of Exploration Geophysicists, Expended Abstracts, 2–3.

Donoho, D. L., Y. Tsaig, I. Drori, and J.-L. Starck, 2006, Sparse solution of underdetermined linear equations by stagewise orthonormal matching pursuit.

Frigo, M., 1999, A fast fourier transform compiler.: Presented at the Proc. 1999 ACM SIGPLAN Conf. on Prog. Lang. Design and Implementation.

Frigo, M. and S. G. Johnson, 1997, The fastest fourier transform in the west: Technical Report MIT-LCS-TR-728, MIT Laboratory for Computer Science.

———, 1998, FFTW: An adaptive software architecture for the FFT: Proceedings of International Conference on Acoustics, Speech and Signal Processing, Volume 3, 1381+.

Hennenfent, G. and F. J. Herrmann, 2005, Sparseness-constrained data continuation with frames: Applications to missing traces and aliased signals in 2/3-D: Presented at the SEG International Exposition and 75^{th} Annual Meeting.

———, 2008, Simply denoise: wavefield reconstruction via jittered undersampling: Geophysics, **73**.

Hennenfent, G. and F. J. Herrmannn, 2006, Seismic denoising with nonuniformly sampled curvelets: Computing in Science and Engineering, **8**, 16–25.

Herrmann, F. J. and G. Hennenfent, 2008, Non-parametric seismic data recovery with curvelet frames: Geophysical Journal International. (doi:10.1111/j.1365-246X.2007.03698.x).

J. Keiner, S. Kunis, D. P., 2006, Nfft 3.0 - tutorial: Technical report, University of Lubeck.

Kuehl, H., 2002, Least-squares wave-equation migration/inversion: PhD thesis, University of Alberta.

Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208–221.

Sacchi, M. D., T. J. Ulrych, and C. Walker, 1998, Interpolation and extrapolation using a high-resolution discrete Fourier transform: IEEE Transaction on Signal Processin, **46**, 31–38.

Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, **67**, 890 – 794.

Trad, D. O., T. Ulrych, and M. D. Sacchi, 2003, Latest views of the sparse radon transform: Geophysics, **68**, 386–399.

Xu, S., Y. Zhang, D. Pham, and G. Lambare, 2005, Antileakage Fourier transform for seismic data regularization: Geophysics, **70**, V87 – V95.

Zwartjes, P. M. and M. D. Sacchi, 2007, Fourier reconstruction of nonuniformly sampled, aliased data: Geophysics, **72**, no. 1, V21–V32.