INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Organization and imaging of three-dimensional seismic reflection data before stack

Canning, Anat, Ph.D.
Rice University, 1994
ORGANIZATION AND IMAGING OF THREE-DIMENSIONAL SEISMIC REFLECTION DATA BEFORE STACK

by

ANAT CANNING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE

G. H. F. Gardner, Professor, Director
Geology and Geophysics

M. Talwani, Professor
Geology and Geophysics

B. Bally, Professor
Geology and Geophysics

W. W. Symes, Professor
Mathematical Sciences

Houston, Texas
August, 1993
ABSTRACT

Organization and Imaging of Three-Dimensional Seismic Reflection Data Before Stack

by

Anat Canning

Pre-stack imaging of 3D seismic reflection data is at present a major challenge of seismic data processing. The main difficulty arises because of the limited power of today's computers. Manipulating volumes of 3D multi-fold seismic data and, especially, executing the large number of mathematical operations required for 3D migration is presently a very complicated and time consuming task. More than that, 3D seismic reflection surveys are characterized by very irregular acquisition geometries. This requires special migration algorithms and even new thinking.

In this thesis we present a new approach to imaging 3D pre-stack seismic reflection data. It is a practical alternative to the full 3D pre-stack depth migration and takes special care of irregular acquisition patterns. The objective of the method is three fold: (a) to perform migration before stack, (b) to correct for three-dimensional seismic effects, (c) to provide a three-dimensional velocity model and a depth image.

A 3D processing scheme is presented here. It is composed of three procedures which are different approximations to the full 3D pre-stack migration operation. The interaction between those schemes is simple, and each
procedure serves as a building block for the next module. That is due to the special derivation of the algorithms in the frequency domain. The three modules that compose this work are:

- Reorganization of multi-fold data on a regular pre-stack grid, based on Dip-Moveout and the inverse transform (DMO⁻¹).

- 3D pre-stack time migration, using the PSI technique.

- Two-pass 3D pre-stack depth migration, performed by splitting the full 3D calculation into a succession of 2D operations.
Acknowledgments

I would like to express my deep gratitude to my teacher Professor Gerald H.F. Gardner who initiated this research, for his generosity in sharing with me his knowledge and ideas, for inspiring me with his unusual way of thinking, for his guidance, encouragement and patience. This research could not have been completed without his guidance.

A very sincere gratitude I owe to Professor Manik Talwani for believing in me, encouraging me and giving me new opportunities.

The main part of this research was carried out at the Geotechnology Research Institute (GTRI) that is part of the Houston Advanced Research Center (HARC). The GTRI staff gave me companionship and support, was always willing to help and share knowledge and I am grateful for that. My experience in GTRI was truly unique. I would like to express my special thanks to Glen Denyer who helped me process enormous amounts of seismic data and to John O'Dowd for his ideas and encouragement. With Dr. Jim Allen I shared fruitful discussions that helped me along the way and I am thankful to him. Walter Kessinger generated some of the figures for me with his migration program, and helped me in many other ways. Muraly Ramaswamy processed and interpreted many seismic lines for me and offered useful suggestions. I thank both for their help and friendship. Tricia Ransom and Yvonne Gangler were always there to help in any way and I am thankful. I would also like to thank Dr. Carolyn Peddy, Dr. Ramon Carbonell and Bill McCormick for their help and friendship.
My friend Dr. David Kessler reviewed the manuscript and suggested many corrections and I thank him very much for his help.

My husband Declan and my sons Nethy and Tom deserve a special thanks for being there for me and bearing with me.

Finally, my very special thanks to my parents for everything.

This research was supported by the 3D consortium project at the Houston Advanced Research Center. I was supported by a W. M. Keck Foundation fellowship. I acknowledge the support provided by the members of the HARC consortium and by the Keck Foundation.

ELF Aquitaine Production donated a 3D dataset to this research and gave me a permission to publish the results. Arco Oil and Gas company provided the model for the physical modeling experiment. I thank them for their generosity.
# Table of content

*List of figures*  

**Chapter 1: Introduction**  
1 - Why 3D ? ................................. 1  
2 - Special problems associated with processing and migrating 3D  
  seismic data ................................. 3  
3 - Thesis objectives .......................... 8  
4 - Thesis outline ............................. 9  

**Chapter 2: Reorganization of multi-fold 3D data on a regular grid**  
1 - Introduction ................................ 11  
2 - 3D DMO, an overview ........................ 15  
3 - Using 3D DMO to organize 3D datasets .......... 17  
4 - 3D DMO ........................................ 21  
  1 - Algorithm implementation .................... 21  
  2 - Aliasing ........................................ 28  
  3 - Artifacts ...................................... 36  
  4 - Amplitude .................................... 42  
  5 - Computer implementation for 3D data .......... 44  
  5 - Inverse DMO .................................. 49
6 - Examples of reorganization and feathering correction
   1 - Synthetic example of a point diffractor
   2 - Feathering correction for the tank dataset
   3 - 3D land survey

Chapter 3: Spatial sampling in 3D - difficulties and solutions
   1 - Introduction
   2 - Correction for irregular sampling
   3 - Derivation of the weighting function - an algorithm
      1 - Introduction
      2 - Computer algorithm
   4 - Correction for insufficient sampling
      1 - Formulation of the problem
      2 - Least-Squares DMO
      3 - Examples
      4 - Discussion

Chapter 4: Pre-Stack Imaging (PSI) in 3D - a time migration
   1 - Introduction
   2 - PSI in 2D - a review
   3 - An example of PSI for a 2D marine dataset
   4 - 3D PSI
Appendix A: A verification that DMO-1 algorithm provides an exact inverse of DMO. ........................................... 168

Appendix B: Comparing 3D pre-stack migration with two-pass scheme for computation time. ........................................ 170

Appendix C: 3D seismic datasets ........................................... 172

Appendix D: About the computer and the computer programs ............ 174
List of figures

Figure 1.2.1: Comparison of 2D and 3D datasets for size ........................................ 5
Figure 1.2.2: Comparison of computation time of 2D and 3D procedures ............... 7
Figure 2.1.1: Location map for ELF dataset ....................................................... 12
Figure 2.3.1: A physical interpretation of DMO .................................................. 19
Figure 2.3.2: A physical interpretation of DMO⁻¹ ............................................... 20
Figure 2.3.3: Flow of feathering correction scheme ............................................. 22
Figure 2.4.1: General flow of 3D DMO in FX domain ......................................... 27
Figure 2.4.2: Impulse response of DMO ............................................................ 29
Figure 2.4.3: Sampling of replacement traces for DMO ....................................... 31
Figure 2.4.4: Sampling of replacement traces on regular grid ................................ 31
Figure 2.4.5: Schematic description of the tank model ......................................... 32
Figure 2.4.6: Crossline aliasing in the tank dataset ............................................. 34
Figure 2.4.7: Simulation of crossline aliasing in the tank .................................... 35
Figure 2.4.8: Line A, DMO, NMO & stack .......................................................... 37
Figure 2.4.9: Line A, NMO, DMO & stack .......................................................... 37
Figure 2.4.10: Synthetic CMP gather after DMO ............................................... 38
Figure 2.4.11: Synthetic CMP gather after DMO with correction .......................... 38
Figure 2.4.12: Linear artifact generated by DMO ............................................... 40
Figure 2.4.13: Synthetic CMP gather with AVO ............................................... 43
Figure 2.4.14: Synthetic CMP gather with AVO after DMO ............................... 43
Figure 2.4.15: General flow of a DMO ............................................................ 46
Figure 2.4.16: Computer memory organization for 3D DMO program .................. 47
Figure 2.4.17: Flow chart for the 3D DMO program ........................................... 48
Figure 2.5.1: Impulse response of DMO⁻¹ ....................................................... 51
Figure 2.6.1: Point diffractor model ........................................ 53
Figure 2.6.2: Feathering correction for a point diffractor ......... 54
Figure 2.6.3: Feathering correction for the tank dataset .......... 56
Figure 2.6.4: ELF dataset, line E1, DMO & stack .................. 60
Figure 2.6.5: ELF dataset, line E1, 2D pre-stack depth migration 61
Figure 3.1.1: Synthetic example of DMO for irregular spatial sampling .. 64
Figure 3.2.1: Polygon map .............................................. 66
Figure 3.2.2: Weighted DMO Vs regular DMO - a synthetic example .. 67
Figure 3.3.1: Polygon construction for regular dataset .......... 70
Figure 3.3.2: Polygon construction with one missing trace ....... 70
Figure 3.3.3: Polygon construction for three missing traces ...... 70
Figure 3.3.4: Polygon construction- search for live grid points .... 73
Figure 3.4.1: Synthetic example of insufficient sampling ........ 75
Figure 3.4.2: Line A: location map .................................... 77
Figure 3.4.3: Synthetic example of DMO and insufficient sampling. 81
Figure 3.4.4: Polygon map of one frequency slice ................. 81
Figure 3.4.5: Synthetic CMP gathers after Least-Squares DMO .... 83
Figure 3.4.6: Line A, CMP gathers at location L1 .................. 85
Figure 3.4.7: Line A, synthetic CMP gathers at location L1 ....... 86
Figure 3.4.8: Line A, CMP gathers at location L2 ................. 87
Figure 3.4.9: Line A, weighted DMO and stack ................... 89
Figure 3.4.10: Line A, Least-Squares DMO and stack .......... 90
Figure 3.4.11: Line A, attributes of Least-Squares DMO .......... 92
Figure 4.2.1: Cheop's Pyramid ..................................... 102
Figure 4.2.2: A 2D point diffractor after DMO ................. 104
Figure 4.2.3: A 2D point diffractor after PSI ....................... 105
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.4</td>
<td>Effect of 2D PSI on one time slice.</td>
<td>106</td>
</tr>
<tr>
<td>4.2.5</td>
<td>PSI as modeling in FK domain.</td>
<td>108</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Line B, stack.</td>
<td>110</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Line B, DMO &amp; stack.</td>
<td>111</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Line B, PSI &amp; stack.</td>
<td>112</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Line B, pre-stack depth migration.</td>
<td>113</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Line B, CMP gather # 300.</td>
<td>115</td>
</tr>
<tr>
<td>4.3.6</td>
<td>Line B, CMP gather # 450.</td>
<td>116</td>
</tr>
<tr>
<td>4.3.7</td>
<td>Line B, CMP gather # 300 after PSI.</td>
<td>117</td>
</tr>
<tr>
<td>4.3.8</td>
<td>Line B, CMP gather # 450 after PSI.</td>
<td>119</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Design of computer program for 3D PSI.</td>
<td>123</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Impulse response of PSI with v=2000 m/sec.</td>
<td>125</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Impulse response of PSI with v=3000 m/sec.</td>
<td>126</td>
</tr>
<tr>
<td>4.6.3</td>
<td>2D synthetic experiment- zero offset time section.</td>
<td>128</td>
</tr>
<tr>
<td>4.6.4</td>
<td>2D synthetic experiment- one time slice after DMO.</td>
<td>129</td>
</tr>
<tr>
<td>4.6.5</td>
<td>2D synthetic experiment- one time slice after PSI.</td>
<td>129</td>
</tr>
<tr>
<td>4.6.6</td>
<td>ELF dataset, line E2, 3D-DMO &amp; stack.</td>
<td>131</td>
</tr>
<tr>
<td>4.6.7</td>
<td>ELF dataset, line E2, 3D-PSI.</td>
<td>132</td>
</tr>
<tr>
<td>4.6.8</td>
<td>ELF dataset, line E2, velocity analysis after DMO.</td>
<td>133</td>
</tr>
<tr>
<td>4.6.9</td>
<td>ELF dataset, line E2, velocity analysis after 3D PSI.</td>
<td>134</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Crossline PSI for one time slice.</td>
<td>139</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Design of computer program for two-pass scheme.</td>
<td>142</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Stack of line # 20 from 3D-synthetic experiment.</td>
<td>145</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Stack of line # 11 from 3D-synthetic experiment.</td>
<td>146</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Stack of line # 20 after the two-pass scheme.</td>
<td>147</td>
</tr>
<tr>
<td>5.5.4</td>
<td>Stack of line # 11 after the two-pass scheme.</td>
<td>148</td>
</tr>
</tbody>
</table>
Figure 5.5.5: ELF dataset, line E3, DMO-stack .......................... 150
Figure 5.5.6: ELF dataset, line E3, DMO, crossline PSI &-stack ...... 154
Figure 5.5.7: ELF dataset, line E1, DMO-stack .......................... 154
Figure 5.5.8: ELF dataset, line E1, depth section after two-pass scheme 155
Figure 5.5.9: ELF dataset, line E1, time section after two-pass scheme . 156
Figure 5.5.10: ELF dataset, line E1, after 2D pre-stack depth migration . 157
Figure 5.5.11: ELF dataset, line E1, after 2D pre-stack depth migration . 158
Figure 5.5.12: ELF dataset, line E1, after 3D post-stack depth migration 159
Figure 5.5.13: ELF dataset, line E1, time section after 3D PSI ............ 160
Chapter 1

Introduction

1.1 - Why 3D?

3D seismic surveys are conducted in order to produce an image of the subsurface. The seismic experiments are performed by initiating an energy source at some point near the surface of the earth and recording the energy that has traveled through the subsurface. This recorded wave field is affected by the physical properties of the medium it has traveled through. The subsurface image is obtained by using a seismic migration process which is a mathematical formulation that is designed to undo the effects of wave propagation from the recorded wave field. Since seismic waves propagate, reflect and refract in the earth in three dimensions, a migration process that provides a correct image of the subsurface should be performed in three dimensions.

In the past seismic exploration was mainly performed using two dimensional surveys. But this picture is changing. There is a growing interest in 3D and today 3D exploration is quite common. When imaging a two dimensional line, one has to assume that all the seismic energy that was recorded in the section, has traveled only through a two-dimensional cross section of the subsurface. This assumption is valid only when all sources and all receivers are located on the same line on the surface, and the structure in the subsurface is two dimensional in nature, i.e., there are no variations in the relevant physical
properties of the medium in the direction perpendicular to the vertical plane through the source-receiver line.

This is a very simplified picture of the earth's crust. Obviously, many structures in the earth are three dimensional in nature. Salt domes may be the most extreme example of three dimensional bodies in the earth, but many other geological structures have severe three dimensional geometries. Furthermore, most marine surveys and many land surveys do not comply with the assumption that all sources and receivers are located on the same line.

Migration processes, applied to 2D seismic data, cannot handle the effects of wave propagation that occur out of the plane of the section, and therefore cannot provide the correct picture of the subsurface, regardless of the technique used. Sideswipes (out-of-plane seismic events) exist in 2D seismic images and cause difficulties in the interpretation. Finally, 2D migration cannot position the events in the section in a 3D sense and therefore 2D surveys have an inherent problem in tying intersecting migrated lines (Yilmaz, 1987).

In order to get a complete and correct image of the subsurface, a 3D seismic survey should be acquired and a full 3D migration should be applied to the data. The 3D migration will remove the sideswipes, bring events that were scattered in the crossline direction into their correct position, and solve the problems of missties. Furthermore, 3D migration will focus energy scattered in three dimensions and with that greatly increase S/N (Krey, 1989).

With today's computer technology, seismic migration in three dimensions is a heavy computational task. Since the value of this process is evident, processing and migrating seismic data in three dimensions is one of the current most challenging tasks in seismic exploration.
1.2 - Special problems associated with processing
and migrating 3D seismic data.

Although the advantage of a 3D seismic image over a 2D one is clear, many problems that are specific to 3D seismic processing complicate the imaging procedure in 3D. Since 3D seismic surveys are costly to acquire and process, so they are usually designed for a detailed study of complex geological areas where 2D surveys had failed to provide a good image. In these cases, simple assumptions underlying many conventional processing schemes are not valid and more accurate algorithms should be used. The conventional CDP stack method (Mayne, 1962) for example, assumes horizontal interfaces and homogenous layers and obviously will fail when imaging in complex geological areas. Therefore in 3D, pre-stack imaging is often required. Furthermore, low order migration schemes may not be suitable for imaging complex geology, and a problem of balancing the cost of the migration algorithm with computational complexity arises. Adding requirements for image resolution and quality, the above becomes an important issue in 3D.

Acquisition geometry of 2D seismic surveys is usually regular and simple. The sources and the receivers are located along a single line on the surface, the intervals between shots and between receivers is constant, and the result is a regular dataset. Every point on a midpoint-offset grid is represented by one seismic trace (the distribution of traces on the midpoint offset grid is usually tapered at the edges). The effects of marine cable feathering (Levin, 1983) or crooked land surveys on the position of midpoints are often ignored.

In 3D it is unrealistic to impose any geometry assumptions when processing the data. In most surveys the distribution of traces in midpoint-offset
space is very irregular, and large variations in shot-receiver azimuth are also quite common. Such irregularities in the spatial trace distribution cause many problems when one tries to migrate the data. For example, migration algorithms such as finite differences (Loewenthal et al., 1976), F-K (Stolt, 1978) and Phase-Shift (Gazdag, 1978; Kosloff and Kessler, 1987), assume that the input data are positioned on a regular grid. Pre-stack migration schemes such as source-receiver migrations (Shultz and Sherwood, 1980) or common shot migration (Reshef, 1985 and 1991) also require regular input data and therefore may not be suitable for 3D without a preliminary regularization step.

The irregularities in the acquisition parameters, especially in marine surveys, where the varying currents change the feathering angle of the cable, may cause additional problems. Some parts of the survey may not be sufficiently sampled, resulting in noise caused by aliasing. Migration in 3D should be able to provide solutions for those cases.

The most prominent problem of 3D seismic processing is related to the size of the dataset. While processing and migrating 2D datasets can be performed relatively quickly with the present computer technology, in 3D it is not the case. Figure 1.2.1 provides a comparison between the size of an average 2D survey and an average 3D survey. It shows that an increase of two orders of magnitude exists. Hence, transferring a 3D dataset in and out of the computer storage is a massive operation. The migration of seismic data in three dimensions requires thus special designs. Moreover, migration is in basis a spatial convolution operation; it means that every input trace affects many output traces which must be stored in the computer memory. But computer memories are limited and consequently, realistic performances become a major issue.
**Average size of a 2D dataset:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of midpoints</td>
<td>1000</td>
</tr>
<tr>
<td>Number of samples / trace</td>
<td>1500</td>
</tr>
<tr>
<td>Number of offsets</td>
<td>100</td>
</tr>
<tr>
<td>Total number of traces</td>
<td>$1000 \times 100 = 100,000$</td>
</tr>
<tr>
<td>Total number of samples</td>
<td>$100,000 \times 1500 = 1.5 \times 10^8$</td>
</tr>
<tr>
<td>Total number of bytes</td>
<td>$1.5 \times 10^8 \times 4 = 6.0 \times 10^8 = 0.6 \text{ Gbyte}$</td>
</tr>
</tbody>
</table>

**Average size of a 3D dataset:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of midpoints/line</td>
<td>500</td>
</tr>
<tr>
<td>Number of lines</td>
<td>500</td>
</tr>
<tr>
<td>Number of samples / trace</td>
<td>1250</td>
</tr>
<tr>
<td>Number of offsets</td>
<td>100</td>
</tr>
<tr>
<td>Total number of traces</td>
<td>$500 \times 500 \times 100 = 2.5 \times 10^7$</td>
</tr>
<tr>
<td>Total number of samples</td>
<td>$2.5 \times 10^7 \times 1250 = 3.125 \times 10^{10}$</td>
</tr>
<tr>
<td>Total number of bytes</td>
<td>$3.125 \times 10^{10} \times 4 = 1.25 \times 10^{11} = 125 \text{ Gbyte}$</td>
</tr>
</tbody>
</table>

The ratio between the size of the 2D survey and the 3D survey is : $\sim 1 / 200$

---

Fig 1.2.1: Comparison of size between 2D and 3D datasets.
Another important aspect of migration is related to the fact that migration of seismic data requires an *a-priori* knowledge of the subsurface velocity model. The quality of the migrated image, especially when depth migration is applied, is heavily dependent on the accuracy of this velocity model. Consequently, pre-stack migration is performed in an iterative manner and the velocity model is updated between migration iterations (Faye and Jeannot, 1986; Al-Yahya, 1989; Lafond and Levander, 1993). But the extension of this iterative procedure into 3D is not yet a realistic computational task, and therefore building a velocity model in 3D is currently one of the major research areas in seismic processing.

In summary, the very large size of 3D seismic surveys and the additional dimension that is introduced in the computation of 3D migration, causes large increases in computation time compared with 2D operations. Figure 1.2.2 points out the problem by comparing the computation time for some standard seismic processes, in two and three dimensions. The above shows that in 3D the derivation of processing technique is influenced by the design of the algorithm in the computer. This design is a crucial factor for the success of the application.
<table>
<thead>
<tr>
<th></th>
<th>Conservative estimation of the number of operations per input sample</th>
<th>Number of operations for a 2D application</th>
<th>Processing time for 2D application on a 1 Gflops machine</th>
<th>Number of operations for a 3D application</th>
<th>Processing time for 3D application on a 1 Gflops machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack</td>
<td>10</td>
<td>$1.5 \times 10^8 \times 10 = 1.5 \times 10^9$</td>
<td>1.5 sec</td>
<td>$3.125 \times 10^{10} \times 10 = 3.125 \times 10^9$</td>
<td>5 min.</td>
</tr>
<tr>
<td>zero offset 2D migration</td>
<td>2000</td>
<td>$2000 \times 1000 \times 1500 = 1.5 \times 10^9$</td>
<td>3 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero-offset 3D migration</td>
<td>200,000</td>
<td></td>
<td></td>
<td>$200,000 \times 500 \times 500 \times 1250 = 6.25 \times 10^{13}$</td>
<td>17.36 hours</td>
</tr>
<tr>
<td>2D Pre-stack migration</td>
<td>2000</td>
<td>$1.5 \times 10^8 \times 2000 = 3 \times 10^{11}$</td>
<td>5 min.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D pre-stack migration</td>
<td>200,000</td>
<td></td>
<td></td>
<td>$100,000 \times 3.125 \times 10^5 = 3.125 \times 10^1$</td>
<td>36 days</td>
</tr>
</tbody>
</table>

Fig 1.2.2: Comparison of computation times between 2D and 3D procedures
1.3 - Thesis objectives

Imaging multi-fold 3D seismic data was the goal of this work. An obvious way to attack this task is to extend to 3D existing 2D methods such as pre-stack depth migration (Schultz and Sherwood 1980, Reshef and Kosloff 1985, Sullivan and Cohen 1987). Facing reality, limitations in computer resources and difficulties in deriving the correct 3D velocity depth model are great obstacles on that road.

Alternatively, we have tried to provide some solutions to the problem of pre-stack imaging for 3D seismic data that are replacements to a full 3D pre-stack depth migration. These solutions represent various compromises between the accuracy of the seismic image and the cost of deriving that image. The schemes presented in this work are designed in a modular way, so that each module is in some sense, a different approximation to the full imaging process. The interaction between the modules is designed in a simple way so that every module can serve as a building block for the next one.

Because this work is about 3D seismic processing, we had to address all the special problems that are associated with 3D seismic (see section 1.2). For example, taking into account the special acquisition geometries that exist in the world of 3D seismic exploration was an important challenge that we have tried to address. A reorganization method that is suitable for pre-stack imaging is thus an important part of this work. The reorganization is based on Dip-Moveout (DMO), and so do the migration schemes. The implementation of an efficient and accurate DMO program is therefore the basis of this work.

Another challenge in pre-stack processing is the preservation of amplitude distribution with offset (AVO). This distribution can be an indicator for
the existence of gas (Ostrander, 1984) and thus an important reason for processing and migrating seismic data before stacking. Preserving AVO was one of our guidelines in developing some of the modules presented here.

Finally, our main task was to provide a practical imaging procedure that can be tested and applied with the present computer technology. Designing efficient computer algorithms was thus an important concern in this work.

1.4 - Thesis outline

In this thesis, three main 3D processing schemes are presented:

1) Reorganization of multi-fold data on a regular pre-stack grid.

The process transforms a pre-stack volume of 3D data to a set of parallel 2D lines from the acquisition point of view. After the transformation all shots and all receivers of a particular line in the volume are located on the same line on the surface. After implementation of this procedure, application of a two-dimensional procedure to every line in the volume is justified, as far as the distribution of common midpoints is concerned. This is a first-order approximation to a 3D procedure, and should be viewed as an alternative to binning. Handling the irregular spatial sampling, and infilling for missing traces is part of the reorganization scheme.
2) 3D pre-stack time migration.

This process is designed to provide a 3D time-migrated image using a pre-stack imaging (PSI) technique. The procedure is independent from velocity and is followed by a simple velocity analysis.

3) Two-pass 3D pre-stack depth migration.

A new migration scheme is designed to provide a velocity model and a depth image by splitting the full 3D migration process into a two-pass operation. It is composed of a velocity independent time migration (PSI) in the crossline direction, followed by a velocity estimation and depth migration, in the inline direction.

The following chapters step through the above operations.
Chapter 2

Reorganization of multi-fold 3D data on a regular grid

2.1 - Introduction

A simple approximation, and a very practical approach to imaging 3D data with a multi-channel technique, is to perform 2D pre-stack depth migration on selected lines from a 3D survey. Unfortunately, 3D seismic datasets often exhibit very large variations in the acquisition geometries. This means that the distribution of data traces in midpoint-offset-azimuth space can be very irregular. In fact, with 3D surveys a "line" is not a very well defined property. With marine surveys, feathering of the marine cable causes spreading of the midpoint position in the crossline direction, even if the ship is sailing along a straight line. In the case of land surveys, the acquisition geometry often does not even resemble an inline-crossline organization. An example of acquisition geometry for a 3D land survey, illustrating the chaotic distribution of sources and receivers, is shown in Figure 2.1.1. Similar behavior is very common with 3D surveys, it requires a reorganization procedure prior to the application of any 2D imaging process.

The organization of 3D multi-fold data on a regular grid is commonly performed by binning the data prior to stacking. With this method a midpoint \((X_m, Y_m)\) grid is defined, and all the traces whose source-receiver midpoints fall in a particular rectangle of the grid are assigned to its center. The method is
Fig 2.1.1.a: Location map for one line from a 3D land dataset. Notice that the sources and receivers are not located along the CMP line.
Fig 2.1.1.b: Location of sources and receivers of traces that belong to a 25'25 m bin. Notice that the source-receiver azimuth varies between traces.
suitable for obtaining a stacked section as long as there is at least one trace in each bin.

Reexamining the situation, binning may not be an adequate solution when multi-channel pre-stack processing (DMO, migration) is required, since it ignores the distribution of traces in offset and in azimuth. Binning does not ensure that all sources and receivers are located along the selected line.

Following the above difficulties, the objective of this part of the work is to reorganize the multi-fold 3D volume on a regular midpoint-offset grid with zero azimuth (as a set of parallel 2D lines). With the data organized, it is more suitable for 2D pre-stack processing, especially for inline pre-stack migration. The reorganization scheme is based on Dip Moveout (DMO) followed by the inverse transform (DMO$^{-1}$). Reorganization and DMO transformation are also pre-requisites for the subsequent imaging schemes.

In order to illustrate the reconstruction process (in which the 3D dataset is built as a set of parallel 2D lines), we use a point diffractor model and a marine acquisition design. Let the $x$ axis describe the inline direction, the $y$ axis the crossline direction and $h$ half of the source receiver offset. Let $(x_d,y_d,z_d)$ be the position of a diffraction point in 3D space. In a constant velocity medium ($v$), with a source located at $(x_s,y_s)$ and a receiver positioned along the inline direction at $(x_s+2h,y_s)$ the diffraction travel time as a function of offset is given by:

$$t = \sqrt{\frac{t_0^2}{4} + \frac{(x_s - x_d)^2 + (y_s - y_d)^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(x_s + 2h - x_d)^2 + (y_s - y_d)^2}{v^2}},$$  \hspace{1cm} 2.1.1$$

where $t_0$ is the usual two-way travel time: $t_0 = \frac{2 \cdot Z_d}{v}$. 
When the receiver cable is feathered $\alpha^0$ from the x-axis, the coordinates are 
$(x_s + 2h \cdot \cos \alpha, y_s + 2h \cdot \sin \alpha)$ and the diffraction travel time as a function of offset is 
given by:
$$t = \sqrt{\frac{t_0^2}{4} \left(\frac{(x_s - x_d)^2 + (y_s - y_d)^2}{v^2}\right)} + \sqrt{\frac{t_0^2}{4} \left(\frac{(x_s + 2h \cdot \cos \alpha - x_d)^2 + (y_s + 2h \cdot \sin \alpha - y_d)^2}{v^2}\right)}$$ \hspace{1cm} (2.1.2)

The purpose of the reconstruction is to transform the travel time $t$, given by (2.1.2) into the inline form (2.1.1). To perform this transformation we use 3D Dip-moveout (DMO) followed by the inverse DMO transform, in the inline direction.

2.2 - 3D DMO, an overview

Dip-moveout (DMO) can be defined as a transformation that maps a non-zero offset trace to its equivalent zero-offset trace. Various DMO algorithms exist. The most common techniques are F-K methods that were initiated by Hale (1983) and integral method (Kirchhoff summation type) that were suggested by Deregowski and Rocca (1981).

DMO is commonly used as a partial migration process that transforms the multi-offset dataset into a zero-offset form. Velocity analysis and stack operations follow, and finally migration is performed. In 3D, most DMO implementations generate the stacked dataset directly. Another view of DMO is that it simplifies diffraction curves. It can be therefore used to simplify pre-stack migration process. Gardner et al. (1986) used this property of DMO to develop the PSI method and Deregowski (1990) used it to perform common offset
migration. DMO can also serve as an interpolation tool as suggested by Deregowski (1986) and by Ronen (1987). We particularly use DMO as an interpolation mechanism performed to reorganize pre-stack data. The simplification that DMO introduces to diffraction curves is implicitly exploited in the reorganization scheme.

Similar to migration, DMO is a spatial convolution operation. It can be carried out as a multiplication in the wave number domain, or as a convolution in space domain. The later approach is known as integral DMO, and states that DMO is done by mapping every input trace to a finite number of replacement traces based on the DMO impulse response. The replacement traces are then summed into midpoints along the source receiver line according to the DMO equation (Forel and Gardner, 1988). Every DMO trace is the result of the accumulation of all replacement traces that fell in that position. This is clearly a trace interpolation mechanism. If the input data are sufficiently sampled in space, the output traces can be constructed at any required point, regardless of the initial data organization. Therefore, an integral implementation of DMO is a powerful trace interpolation method and thus very common in 3D.

Most DMO schemes require an NMO correction before DMO, and they operate on constant-offset sections. This means that traces are not mixed between offsets. An alternative approach was presented by Forel and Gardner (1988). They showed that when DMO is not restricted to constant offset sections, it can be performed before NMO. This approach is ideal for interpolation. First of all, interpolation that is independent from the velocity model (DMO before NMO) is obviously superior to velocity dependent methods. Secondly, mixing traces between offsets provides a more robust mechanism for interpolation allowing traces from one offset to compensate for the absence of traces in neighboring
offsets. For 3D implementation, following this approach is almost inevitable, since a constant offset section, in most 3D surveys, may not be well defined.

In the following sections, an implementation of the Forel and Gardner DMO scheme in the frequency domain, and its application for reorganizing 3D datasets is described.

2.3 - Using 3D DMO to organize 3D datasets

The travel time from a diffraction point located in 3D space at \((x_d, y_d, z_d)\), as a function of midpoint \((x_m, y_m)\), offset \((2h)\) and azimuth \((\theta)\) is given by:

\[
1 = \frac{\sqrt{\frac{t_0^2}{4} + (x_m - h\cdot\sin\theta - x_d)^2 + (y_m - h\cdot\cos\theta - y_d)^2}}{v^2} + \frac{\sqrt{\frac{t_0^2}{4} + (x_m + h\cdot\sin\theta - x_d)^2 + (y_m + h\cdot\cos\theta - y_d)^2}}{v^2}
\]

Expression (2.3.1) says that the orientation of cable \((\theta)\) on the midpoint \((x_m, y_m)\) plane changes the travel time curve. That is the reason why marine cable feathering or other irregularities in the acquisition geometry cause difficulties in stacking (Levin, 1983). The purpose of the reorganization scheme is to generate a set of parallel 2D lines from the unorganized 3D pre-stack volume, and that way to remove the effects of cable feathering.

Based on the definition of DMO, one can deduce that 3D DMO transforms this curve to its zero-offset form (Gardner et al., 1986):
\[
t = 2 \sqrt{\frac{t_0^2}{4} + \frac{(x_m - x_d)^2 + (y_m - y_d)^2}{v^2} + \frac{k^2}{v^2}} \tag{2.3.2}
\]

where \(k\) is half offset after DMO. Inspecting equation (2.3.2) reveals that DMO removes the dependence of travel time with azimuth (Jakubowicz et al., 1984). In other words, after DMO the source-receiver line can no longer be interpreted as being positioned in some direction on the \((x,y)\) plane. In fact expression (2.3.2) implies that the offset coordinate \(k\) is orthogonal to \((x,y)\) plane. Figure (2.3.1) illustrates the physical meaning of this 2D interpretation. In a more general fashion, DMO reduces the dimension of the diffraction surface from 4D to 3D (by removing the dependence with azimuth). It also changes the geometry of the surface, from being a "Pyramid of Cheops" (Claerbout, 1985) into a hyperboloid of revolution. As mentioned previously, with an integral implementation of DMO one can generate data traces at any required location. We have chosen to build the output space as a regular midpoint-offset grid. Therefore, after 3D DMO the dataset is organized on a regular grid, and the travel time is not affected anymore by source-receiver azimuth. One can choose any line from the 3D volume and process it as if it was acquired in a true 2D sense (all source receiver pairs were located on the same line).

DMO changes the geometry of the travel time surface. This effect should be removed before conventional 2D pre-stack migration can be applied to the selected lines. We derived a 2D inverse DMO operation to remove DMO from the regularized data. This operation can be viewed as a projection of the offset coordinate \(k\) back to \((x,y)\) plane and in the inline direction (Figure 2.3.2). Similar to DMO, the DMO\(^{-1}\) operator involves only the traces along the source-receiver
Ray paths associated with one CMP gather (before DMO)

Ray paths associated with the same CMP gather. After DMO offset coordinate becomes perpendicular to the original axis. In 3D offset coordinate after DMO is interpreted as being perpendicular to (X,Y) midpoint plane.

Figure 2.3.1: A physical interpretation of the effect DMO has on a 2D dataset.
Map view of midpoint plane (Xm,Ym) for a 3D survey.

Figure 2.3.2: A physical interpretation $\text{DMO}^{-1}$:
Receiver cable is projected back to the midpoint plane. For feathering correction the projection is done in the inline direction (as in this Figure). In the general case this can be done in any required direction.
line. Therefore, for zero azimuth, $\text{DMO}^{-1}$ is applied to every line separately. A summary of the reorganization scheme is presented in Figure 2.3.3.

The DMO algorithm that we have implemented assumes constant velocity medium. A more accurate DMO implementation may have to take into account the velocity variations in the medium and in fact, there are various published depth variant DMO algorithms (Perkins and French, 1990; Schwab and Gardner, 1991; Hale, 1993) that can be used to perform this task. However, after the data are positioned on the regular grid we remove the DMO, since we used it merely as an interpolation mechanism. Therefore, we assume that the overall effect of velocity variations can be ignored. This assumption enables us to use the velocity independent approach to DMO, which is an important feature of this work. More discussion of DMO in $v(z)$ media can be found in section 4.6.2.

2.4 - 3D DMO

2.4.1 - Algorithm implementation

The basic NMO-DMO equation in three dimensions is given by (Forel and Gardner, 1988):

$$t_0^2 = t^2(1 - \frac{b^2}{h^2}) - \frac{4}{v^2}(h^2 - b^2), \quad 2.4.1$$

where:
- $t$ - non-zero offset travel time,
- $t_0$ - normal incidence time,
- $h$ - half offset,
- $b$ - distance (along the source receiver line) from source receiver midpoint,
- $v$ - medium velocity.
Figure 2.3.3: Flow of feathering correction scheme. Reorganization of multi-fold 3D data as a set of parallel 2D Line.
Following Deregowski (1981), we define NMO correction by:

\[ t_{nmo}^2 = t^2 - \frac{4h^2}{v^2}, \]  \hspace{1cm} 2.4.2

and substitute it into (2.4.1). This procedure results in the known elliptic DMO impulse response:

\[ \frac{t_0^2}{t_{nmo}^2} + \frac{b^2}{h^2} = 1. \]  \hspace{1cm} 2.4.3

Forel and Gardner (1988) derived an alternative DMO operation by introducing a new variable \( k \) and defining a DMO time \( t_{dmo} \):

\[ k^2 = h^2 - b^2, \]  \hspace{1cm} 2.4.4.a

\[ t_{dmo} = t \cdot \frac{k}{h}. \]  \hspace{1cm} 2.4.4.b

Substituting (2.4.4) into (2.4.1) gives NMO equation with a new offset coordinate \( k \):

\[ t_0^2 = t_{dmo}^2 - \frac{4k^2}{v^2}. \]  \hspace{1cm} 2.4.5

With this approach DMO is performed before NMO.

The kinematics of DMO impulse response can be extracted by reorganizing (2.4.4). It is identical to (2.4.3):

\[ \frac{t_{dmo}^2}{t^2} + \frac{b^2}{h^2} = 1. \]  \hspace{1cm} 2.4.6
So both DMO algorithms are based on the same kinematics, although they operate on a different time axis. This implies that any Kirchhoff type algorithm that is based on the kinematics of (2.4.3) can be used to generate replacement traces for both methods. From this point of view, equation (2.4.4) is a generalization of the conventional constant-offset algorithms. It can be implemented either before or after NMO. A Kirchhoff implementation of (2.4.3) is described by the integral:

\[
f(t_0, x_m, h) = \int_{-h}^{h} f\left(\frac{t_0}{\sqrt{1 - \frac{b^2}{h^2}}}, x_m - b, h\right) db. \tag{2.4.7}
\]

This requires NMO correction and constant offset section (h). A similar implementation for (2.4.4) is given by:

\[
f(t_{dmo}, x_m, k) = \int_{-\infty}^{\infty} f\left(t_{dmo}, \sqrt{1 + \frac{b^2}{k^2}}, x_m - b, h = \sqrt{k^2 + b^2}\right) db. \tag{2.4.8}
\]

Application of (2.4.4) to 3D surveys was also described by Forel and Gardner (1988). According to this description, replacement traces should be positioned along the source-receiver line. Since the output DMO grid \((x_m, y_m, k)\) is a discrete function Forel and Gardner suggested to sum the replacement traces to the nearest grid point whenever it does not fall exactly on a grid location.

The above analysis was derived in order to verify that the DMO algorithm introduced by Forel and Gardner is a generalization of Kirchhoff type algorithms which are performed in constant-offset space. In the following we describe a DMO implementation that is based on the kinematics of equation (2.4.8) but derives the dynamic properties from a constant-offset algorithm. This of course is justified because of the analogy between the two algorithms.
The temporal operation (2.4.4.b) is time variant. However, a transformation of the time axis to a logarithmic scale [Bolondi, 1982] results in a time-invariant DMO operation. This simplifies the implementation of DMO in the time domain, and more than that, in the frequency domain it can be carried out as a simple phase shift. Defining:

\[ \tau = A \log(t) + B \]  

2.4.10

and applying (2.4.10) to (2.4.4.b) gives:

\[ \tau_{\text{dmo}} = \tau + A \log\left(\frac{k}{h}\right) \]  

2.4.11

which is a constant shift in \( \tau \). A and B are constants defined so \( \tau_1 = 1 \) and \( \Delta \tau = 1 \).

It turns out that [Hale, 1991] a straightforward implementation of integral DMO that uses simple time shifts is inherently aliased. The source of this aliasing problem is best illustrated by comparing impulse response of FK DMO with this simple integral implementation. The FK impulse response exhibits variations of amplitude, frequency and phase with time, offset and distance, and a simple shift in time does produce the above appearance. Most integral implementations of DMO try to address some of these problems [Deregowski, 1985, Liner, 1990, Hale 1991], but the details of their operator seem to vary from one implementation to the other. Our objective is to implement a phase-shift function that provides the kinematics of the DMO operator. It should be non-aliased and preferably be pre-calculated and stored in memory.
Based on the kinematics of constant offset DMO (2.4.3) and the log-stretch transformation (2.4.10), a dip-dependent phase-shift factor in FK domain was derived (Gardner, 1991):

\[ G(F,K) = e^{-i\pi AF \sqrt{\frac{4h^2K^2}{\Lambda^4F^2} - 1 - \log \left( \frac{1}{2} \sqrt{\frac{4h^2K^2}{\Lambda^4F^2}} \right)}} \]

where:
- \( F \) - frequency of \( \tau \),
- \( h \) - half offset,
- \( K \) - wave number (of constant offset section).

In a constant offset domain, DMO can be implemented by multiplying the input FK data by the operator \( G(F,K) \). Alternatively, one can use FFT\(^{-1}\) to transform \( G(F,K) \) to its FX form and apply the constant offset DMO as a spatial convolution. This procedure is similar to a Kirchhoff implementation of DMO.

Since \( G(F,X) \) is a function of offset, it has to be calculated separately for every offset \((2h)\). In order to implement this factor to a kinematic procedure that is based on equation (2.4.4) and also enable the general case in which the distribution of input offsets is random (as with most 3D surveys), the dependence of \( G \) on \( h \) should be removed. Defining \( G(F,K \cdot h) \) as the Fourier transform of \( g \left( \frac{F}{h} \right) \) is the key to the generalization of this algorithm so it will not be restricted to constant offset sections. Since the DMO operator is limited to the source receiver line (Equation 2.4.7), for any offset \( h \), \(-1 \leq \frac{b}{h} \leq 1\) (\( b \) is the distance along the source-receiver line). In the discrete form we divided the source receiver line to \( N \) samples (replacement traces). When using the same \( N \) for all offsets, \( g \left( \frac{F}{h} \right) \) can be calculated only once. The result is stored in
Figure 2.4.1: General flow of 3D DMO program in FX domain.
memory, and applied to every input trace, regardless of its offset \( h \). Figure 2.4.1 shows a flow chart for the velocity independent DMO algorithm, carried out as a convolution in \((F,X)\) domain. The integral representation is given by:

\[
\tilde{f}(F,x_m,k) = \int_{-\infty}^{\infty} f(F,x_m-b,h = \sqrt{k^2 + b^2}) \cdot g(F, b/h) db
\]

2.4.13

2.4.2 - Aliasing

Any DMO algorithm has to address questions related to spatial and temporal aliasing. Hale (1991) stated that in most implementation of integral DMO, the operator is aliased and therefore the result is contaminated with high frequency noise. Because DMO does not alter horizontal events, summing up all traces that construct the impulse response, should reproduce the original trace. This procedure was suggested by Hale as a necessary condition for DMO. According to Hale, most integral DMO methods do not pass this test. Figure 2.4.2 shows the impulse response of integral DMO based on (2.4.12). The sum of replacement traces does reconstruct the original input trace, and hence, is not inherently aliased.

Although a non-aliased operator is a necessary condition for a non-aliased DMO implementation, it is not a sufficient condition. Obviously, replacement traces must be generated close enough to each other so that the DMO ellipse is well sampled. Equation (2.4.4.a) says that replacement traces are generated along a circle in midpoint \((m)\) and offset \((h)\). In the implementation described in Figure 2.4.1 this circle is sampled with equal \( b \) intervals and the
Figure 2.4.2: Impulse response of DMO.

Sum of all replacement traces.

Replacement traces.

Input trace.
same number of samples (replacement traces) is used for all offsets (Figure 2.4.3). This procedure ensures that the circle is well sampled regardless of the input offset. However, as a result of this design replacement traces are not necessarily positioned on a grid point, and are therefore added to the nearest one.

Aliasing can also be related to the sampling of input data. When the input data are spatially aliased, DMO will generate artifacts. This phenomena is illustrated using the tank dataset (see appendix C). The special construction of the tank model (Figure 2.4.5) gave rise to a very steep dip reflection (90°) associated with a wave that traveled through a slow medium. Since the survey was not sampled fine enough in the crossline direction, this seismic reflection event is aliased. It is interesting to note that this event had very little moveout in the inline direction and therefore did not seem to hold any potential aliasing problem (Figure 2.4.6.a). In fact it was aliased in the crossline direction and therefore 3D DMO destroyed it while preserving ordinary, well-sampled subsurface reflections (Figure 2.6.b). To verify the conclusions from this analysis, we applied 3D DMO to synthetic data that was generated with the acquisition geometry of the tank dataset using a similar model (Figure 2.4.7). This example is of a very extreme case but it illustrates well the relationship between sampling and DMO.

In regular circumstances potential aliasing problems are often associated with sampling in the offset direction. Therefore when an NMO correction is performed before DMO, aliasing is less likely to occur. Consequently, the velocity independent DMO (2.4.4) is more susceptible to data aliasing (if applied before NMO) due to the apparent aliasing of the NMO curve. This situation
Figure 2.4.3: Sampling replacement traces at equal $b$ intervals along a circle in midpoint-offset space.

$h^2 + b^2 = h^2$

$h$ - input offset,
$x$ - position of replacement trace.

Figure 2.4.4: Sampling replacement traces at equal grid intervals.
Figure 2.4.5: Schematic description of the tank model.

The top of the physical model that was used to generate the tank dataset (see appendix) was surrounded by walls. These walls were built in order to confine a special liquid used to simulate water velocity, when the problem is scaled down. Sources and receivers were positioned in the liquid. As a result of this design the seismic dataset contained, in addition to regular subsurface reflections, reflections from these walls. This figure describes the ray paths for three types of reflections that exist in the tank dataset:

A) Regular subsurface reflection.
B) Reflection from wall traveling in the crossline direction.
C) Reflection from wall traveling in the inline direction.
(EVENT A)
Reflection from wall: Seismic wave is traveling through slow velocity medium (water) and reflects from a steeply dipping interface (90°).
Reflection event is aliased in the crossline direction due to large moveout and insufficient line spacing. On the other hand, the inline direction is parallel to the reflector and therefore, in this direction the moveout is very small.

(EVENT C)
Reflection from wall.
(Alased in the inline direction)

(EVENT B)
Regular subsurface reflection.
Figure 2.4.6: An example of the crossline aliasing problem in the tank dataset.

a) One CMP gather from the tank dataset.

b) Same CMP location after 3D DMO.

- Event B: destroyed by 3D DMO due to aliasing.
- Subsurface reflections were constructed correctly.

Reflection from the well in crossline direction (EVENT B).

Offset
a) One CMP gather from the synthetic dataset that was generated as a simulation of the tank model. Simulation was done using the actual acquisition geometry of the tank dataset. Top event is a subsurface reflection from a dipping plane ($20^\circ$). Bottom event is a reflection from the wall (event A in Figure 2.4.5), aliased in the crossline direction.

b) Same CMP location after 3D DMO. Top event was well constructed by DMO although it has larger moveout. Bottom event (A) was destroyed. It shows the same discontinuous character as event A in Figure 2.4.6.b).

Figure 2.4.7: A simulation of the crossline aliasing problem in the tank dataset.
occurs when offset intervals are large and the medium velocity is slow. After NMO correction this problem is corrected. Figures 2.4.8 and 2.4.9 compare stacked sections with DMO before and after NMO. Coherent events in the shallow part of the section (Figure 2.4.9) were destroyed when DMO was applied before NMO (Figure 2.4.8). Notice that deeper events were well preserved with both implementation. This is because NMO curves of shallow reflections are steeper than deep reflections.

Another source of aliasing is associated with resampling the time axis into the logarithmic scale. One has to make sure that the temporal sampling interval does not violate Nyquist restrictions. For a fixed $\tau$, $\Delta t$ increases with $t$, and therefore the last sampling interval must correspond to maximum frequency in the data. Based on this consideration, the number of $\tau$ samples is given by:

$$nt = \frac{\log(t_{\text{max}}) - \log(t_{\text{min}})}{\log(\frac{1}{2^{t_{\text{max}}t_{\text{max}}^{-1}} + 1})}$$  \hspace{1cm} 2.4.14

2.4.3. - Artifacts

Discretizing the grid causes inaccuracies in the computation of DMO data, mainly because replacement traces are positioned at the nearest grid-point. This produce some high frequency noise in DMO results. An example of this artifact is presented in Figure 2.4.10. Noise above the main event is the result of this miss-positioning of replacement traces. To reduce this effect we have applied an additional phase-shift that was calculated along the DMO ellipse and compensated for differences between the required position of the replacement traces and the actual position at the grid point. The correct position of a
Fig 2.4.10: CMP gather after DMO.
Input to DMO was a simulation of a multi-fold 2D seismic dataset over a horizontal interface and constant velocity medium.
Noise above the main event is the result of the inaccuracy in summing data onto a discrete grid.

Fig 2.4.11: Same CMP gather after phase-shift correction.
Artificial noise is reduced.
replacement trace $b_1$ is at offset $k = \sqrt{h^2 - b_1^2}$. Instead the trace is placed at grid-point $(b',k')$ which is the nearest one. The time shift applied to the trace was $\log_{\frac{k}{h'}}$ instead of $\log_{\frac{k}{h}}$. So the additional shift is:

$$\Delta \tau = \log_{\frac{k}{h'}}(k') - \log_{\frac{k}{h}}(k).$$

2.4.15

In the frequency domain this is implemented by multiplying the trace by $e^{-i\pi F \Delta \tau}$. Figure 2.4.11 shows a CMP gather in the same location after this phase correction. By applying this correction most of the artificial noise has been removed. However, the additional phase shift increased the cost of DMO, and therefore was not always implemented in the coming examples.

Another artifact of the $(k,t)$ DMO can be explained by comparing the integral representations (2.4.7) and (2.4.8). In the conventional implementation (2.4.7) the integration is restricted to constant offset planes. It is not affected by the existence or absence of other offsets. In (2.4.8) the integration is performed over all offsets $h \geq k$. Unfortunately, in practice, the offset axis is limited and therefore the integration path is artificially truncated: $|b| \leq \sqrt{h_{\text{max}}^2 - k^2}$. This produces an artifact that is seen as a linear event on a CMP gather, pointing towards the origin $(k=0,t=0)$ (Figure 2.4.12). The artificial limits on the integration path become narrower as output offset $k$ approaches maximum data offset $h$ and consequently, the amplitude of this noise increases with offset. This problem is inherent to the $(k,t)$ algorithm but, fortunately, is not serious because this noise is positioned in the muted zone.
a) One input CMP gather.
Input data was simulated using the two horizontal interface model (see chapter 3) and regular acquisition geometry. NMO correction was applied before DMO.

Figure 2.4.12: Linear artifact generated by DMO due to truncation in offset.
Artificial random noise, caused by inaccuracies in the DMO scheme.

Linear artifact due to finite input offset.

b) CMP gather after DMO.
2.4.4 - Amplitude

When DMO is used as an interpolation method generating pre-stack data traces, it is important to verify that it preserves AVO distribution. The general definition of DMO as a transformation that maps any input trace to its equivalent zero-offset trace implies that amplitude of all traces that construct a CMP gather should be identical after DMO since they are all zero-offset traces. This view is embraced in many publications (Liner, 1989; Black et al., 1993) and is the basis for the formulation of true amplitude DMO-stack operation. But with this view AVO information is not preserved and therefore not applicable for pre-stack interpolation. Amplitude preservation could be intuitively justified based on the interpretation of DMO as an operation that changes the orientation of source receiver cable (see section 2.3 and Figure 2.3.1). A synthetic example that shows that our DMO implementation preserves AVO information is presented here. A model consisting of two horizontal interfaces was used to generate a synthetic dataset. Amplitude variations with offset were introduced to both reflection events. The top event had amplitude increasing exponentially with offset and the bottom event showed amplitude decaying linearly with offset. Figure 2.4.13 displays one NMO corrected CMP gather from the input data. CMP gather in the same location but after the application of DMO is presented in Figure 2.4.14. The two sections appear to be identical and verifies that indeed AVO information was not destroyed by DMO operation.
Fig 2.4.13: One synthetic CMP gather with NMO correction and AVO (input for DMO).

Fig 2.4.14: Same CMP gather after DMO.
2.4.5 - Computer implementation for 3D datasets

A general flow of the DMO scheme is presented in Figure 2.4.15. This design requires a pre-allocation of all the output space \((t, X_m, Y_{m,k})\) in memory, so input traces could be summed into it. In practice, such large memory space is not available and this requirement must be reduced. A practical way leading to the reduction in memory requirement is based on the fact that DMO operator is limited to maximum source-receiver distance. If the order in which the input traces are fed into the computer can be controlled, this space can be reduced by limiting the number of lines to the maximum source-receiver distance in the crossline direction. In the general case, data from 3D surveys may come in any order. However, in most cases the organization of input data is known in advance, and if it is not too chaotic, this information can be used to help reduce the memory requirements. In order to do that we assume that (up to a certain extent) the data are organized in an increasing order in the crossline direction. This is normally true for marine surveys when the data are acquired along parallel lines, and therefore can be fed into the computer in an increasing line order. For land surveys, it may not be as simple as that, but often, as the acquisition process progresses, different regions in the subsurface are covered. Although these regions may span across a number of lines (band), the data can still be fed into the computer memory with an increasing band order. The maximum cable feathering, or in the general case, band width, defines the extent of the DMO operator and by that defines the number of crosslines that has to be stored simultaneously in memory. In the extreme case, the band width would be the whole survey. Figure 2.4.16 describes the memory organization of the 3D DMO program.
Unfortunately, limiting the number of crosslines is not enough. To illustrate the difficulty in processing 3D datasets, consider an example of a small dataset. Let:

\[
\begin{align*}
\text{nx} & - \text{number of CMP/line} & = 500, \\
\text{ny} & - \text{number of lines} & = 200, \\
\text{nk} & - \text{number of offsets} & = 60, \\
\text{nt} & - \text{number of time samples} & = 1000, \\
\text{nl} & - \text{number of lines across the cable} & = 10. \\
\end{align*}
\]

The memory size that is required for this small example is:

\[
\text{size} = \text{nx} \times \text{nl} \times \text{nk} \times \text{nt} \times 4 = 1.2 \text{ Gb}
\]

And this is still a very large number. To further reduce the memory requirements, the algorithm was designed in the frequency domain and the program processes frequency slices separately. The number of frequencies that are processed simultaneously can be adjusted based on the availability of memory and the size of the DMO problem. In the extreme case, one frequency slice is processed at a time, and by that the memory size is reduced by two to three orders of magnitude. Figure 2.4.17 outlines the design of the computer program.
Fig 2.4.15: A general flow of the 3D DMO scheme.
Initial Memory Setup:

After Reading Line 1:

Fig 2.4.16: An example of the computer memory organization for 3D DMO in the presence of receiver cable feathering across three output lines. The program assumes that the data is fed in inline order. Therefore, all traces that contribute to line #1 will be read in before line #4 will start to receive any contributions.
Fig 2.4.17: Flow of 3D DMO in the computer.
2.5 - Inverse DMO

To complete the reorganization procedure, effects of DMO are removed from the regularized data. In similarity to DMO, the inverse operation can be generally defined by its kinematic properties. The dynamic properties of DMO operator vary from one implementation to the other because different researchers relate different meaning to amplitude after DMO. Consequently, an exact inverse of DMO transformation is unique to every implementation. An inverse operator for the Hale FK algorithm was derived by Ronen (1987) and by Liner and Cohen (1988). Here we present an algorithm for DMO\(^{-1}\) that is designed to invert the (F,X) implementation of equation (2.4.4). Similar to DMO, the DMO\(^{-1}\) is independent from velocity, and can be performed before NMO.

Based on equation (2.4.4) DMO spreads the amplitude of a trace along an elliptic curve. To invert this operation, amplitudes are simply summed along the same curve. This of course is equivalent to spatial convolution procedure that spreads amplitudes along a hyperbolic curve. The kinematic inverse of (2.4.4) is given by:

\[ h^2 = k^2 + b^2 \]  \hspace{1cm} 2.5.1.a

\[ t = t_{dmo} \cdot \frac{h}{k} \]  \hspace{1cm} 2.5.1.b

Transforming the time axis to logarithmic scale, by applying (2.4.10) to (2.5.1) makes DMO\(^{-1}\) a time invariant operation:

\[ h^2 = k^2 + b^2 \]  \hspace{1cm} 2.5.2
\[ \tau = \tau_{dmo} + A \cdot \log \left( \frac{h}{k} \right). \]

Again, this operation can be carried out as a phase shift in the frequency domain.

To reconstruct the dynamic properties of the original input trace (before DMO) the changes in phase (2.4.12) should be undone. A phase-shift factor for DMO\(^{-1}\) that inverts \(G(F,K)\) is given by:

\[ G^{-1}(F,K) = e^{i\pi AF \left[ \sqrt{1 + \frac{4k^2K^2}{\Lambda^2F^2}} - 1 - \log \left( \frac{1}{2} \sqrt{1 + \frac{4k^2K^2}{\Lambda^2F^2}} \right) \right]}, \quad 2.5.3 \]

where:
- \(F\) - frequency of \(\tau\),
- \(k\) - half offset after DMO,
- \(K\) - wave number (of constant offset section).

For a constant offset section \(G^{-1}\) can be applied as a multiplication in FK domain. But in order to implement DMO\(^{-1}\) in a velocity independent fashion the spatial convolution is carried out along an hyperbola in midpoint-offset space (equation 2.5.1), rather than in constant offset domain. Again, a generalization of \(G^{-1}\) uses the fact that it is defined as a function of \(k\cdot K\) and therefore is the FFT of \(g^{-1}(F, \frac{b}{k})\). Combining the kinematic description of DMO\(^{-1}\) (2.5.1) and the phase-shift operator, DMO\(^{-1}\) integral is defined by:

\[ f(F, x_m, h) \approx \int_{-h}^{h} \hat{f}(F, x_m - b, k = \sqrt{h^2 - b^2}) \cdot g^{-1}(F, \frac{b}{k}) db \quad 2.5.4 \]
Fig 2.5.1: Impulse response of DM0-1

Sum of all replacement traces.

Replacement traces.

Input trace.
where: \( \tilde{f} \) - trace after DMO, \\
\( \tilde{f}^\prime \) - trace after DMO\(^{-1} \).

A verification that (2.5.4) is an exact inverse of (2.4.13) is given in appendix A.

The impulse response for DMO\(^{-1} \) is presented in Figure 2.5.1. The sum of impulse response traces show a good reconstruction of the original trace and with that verifies that the DMO\(^{-1} \) operation is not inherently aliased.

### 2.6 - Examples of reorganization and feathering correction

Reorganizing 3D datasets so that they can be imaged using a 2D procedure is demonstrated with some data examples:

#### 2.6.1 - Synthetic example of a point diffractor

The effects of source-receiver orientation on the travel times are demonstrated using a synthetic example of a point diffractor. This example serves as an explanation as to why cable feathering effects should be removed before a 2D procedure can be correctly applied. In this example we generated synthetic dataset using the acquisition geometry of conventional 3D marine survey with constant cable feathering of 25°. The diffractor was located in a constant velocity medium.
a) Point diffractor model. Input data was simulated so that source-receiver cable is oriented in a 25° angle with respect to inline direction.  → marks the position and cable orientation of the input gather shown in Figure 2.6.2.a.

b)  → marks the position and cable orientation of the CMP gathers that are presented in Figures 2.6.2.b and 2.6.2.c. The source-receiver cable lies in the inline direction.

Figure 2.6.1
Figure 2.6.2: Feathering correction - reconstruction of 3D data onto inline-crossline grid. Synthetic data was simulated over a point diffractor model.
Figure 2.6.1.a describes the model and the source-receiver configuration for one CMP location. A CMP gather from the input dataset is presented in Figure 2.6.2.a. This gather is positioned somewhat away from the diffraction point, both in the inline and in the crossline direction. The travel time for this diffraction event is affected by the feathering angle as described in equation (2.1.1). We used 3D DMO followed by 2D DMO\(^{-1}\) to reconstruct the 3D data as a set of parallel 2D lines. Now the source-receiver cable is oriented in the inline direction (Figure 2.6.1.b). The result of this reconstruction is presented in Figure 2.6.2.b. The location of this CMP gather has not changed and the change in travel time curve is the result of the change in cable orientation (equation (2.1.1)). A comparison between this gather and the synthetic desired output (Figure 2.6.2.c) verifies the accuracy of this process.

2.6.2 - Feathering correction for the tank dataset

In this example we compare binning with the feathering correction approach. We used both methods to organize the 3D dataset of the physical modeling experiment. The dataset (referred to as the "tank dataset" - see appendix C) was acquired over a complex layered model. It was collected as a simulation of a conventional 3D marine survey, and had constant cable feathering of \(11^0\). An additional 2D line was collected over the same model, at its center. This line had no cable feathering, and was used to evaluate the results from the feathering correction scheme.

Figure 2.6.3.a is a stacked section from the center line of the tank dataset. This line was stacked using conventional binning process, so the
a) Stack of center line after binning. Feathering correction was not applied.

b) Stack of center line after feathering correction.

Figure 2.6.3: Feathering correction for the Tank Dataset.
c) Stack of calibration line. This line was collected as a 2D line at the same center location. The line was acquired with no cable feathering, and should be compared with the corrected line (b).
orientation of source-receiver cable was ignored. After feathering correction the data was stacked again (Figure 2.6.3.b) and a linear event (marked with an arrow) appeared in the section. This event could be interpreted as an artifact from the feathering correction process, but the calibration line (Figure 2.6.3.c) exhibits the same event and with that verifies that this new event is a real wave phenomenon. Again, comparison between the feathering correction results and the calibration line verifies the accuracy of this process.

2.6.3 - 3D land survey

Marine datasets are normally acquired as a set of parallel 2D lines, and if one ignores the effects of cable feathering, a 2D imaging procedure may still be applied. This is in fact the normal procedure in 2D marine surveys. With land datasets acquisition geometries are usually much more complex and the trace distribution in midpoint, offset and azimuth if often quite chaotic. Therefore, imaging 3D land datasets before they are stacked can only be done with a 3D procedure that can take into account these irregularities. For those datasets, the reconstruction procedure is a necessary step if a 2D imaging scheme is to be applied.

An illustration of this procedure is provided using a small 3D land dataset collected in a very irregular way (see appendix C for description of ELF dataset). Figure 2.1.1 is the location map that show the distribution of sources and receivers with respect to the CMP line.

In this dataset the sources and the receivers that belong to the marked CMP line are not positioned on the line. In other words a 3D procedure is
required to migrate this line before stack. Figure 2.6.4 is a time section of one line (E1) from the ELF dataset after 3D-DMO and stack. We applied the reorganization scheme to this dataset, so this line also passed through 2D DMO\(^{-1}\) which organized it as a true 2D line, suitable for 2D pre-stack migration. Figure 2.6.5 is a depth image of this line acquired using a 2D pre-stack depth migration program. The main point of this example is the illustration of the mechanism that constructs a depth image from this unorganized 3D survey, using a 2D imaging procedure. A correction for 3D effects that are present in this line is presented in chapter 5.
Fig 2.6.4: ELF dataset, line E1, DMO & stack.
Fig 2.6.5: ELF dataset, line E1, depth image after 2D pre-stack depth migration. (3D-DMO, DMO⁻¹ and MIGPACK)
Chapter 3

Spatial sampling in 3D - difficulties and solutions

3.1 - Introduction

The success of the reorganization scheme relies on the ability of DMO and DMO\(^{-1}\) to reconstruct the correct kinematics of seismic events, preserve amplitudes and minimize artifacts. In the previous chapter we showed how this task can be achieved when the data are regular and well sampled. However, 3D datasets often exhibit a very irregular distribution of traces in space. With marine data the feathering angle varies along the survey, and so do the line and the shot spacing. Land surveys may exhibit large variations in source-receiver azimuth. Under these conditions a straight-forward application of DMO does not always produce good results. DMO, as well as migration, are spatial integral operations. When formulating them in a discrete form, it is implicitly assumed that the data grid is regular.

To illustrate the problem resulting from irregular sampling, we use a simple synthetic example. The model was constructed from two horizontal interfaces that separate media of identical velocity but different density. we generated a 3D dataset over this model using marine shooting geometry, constant cable feathering (10\(^{\circ}\)) and uniform line spacing. In this experiment we distributed the shots randomly along the line, while restricting the maximum distance between successive shots to twice the regular grid interval. The reflection coefficients for this model are independent of the angle of incidence
and therefore, after correction for NMO and geometrical spreading, all traces in this dataset are identical. We expect DMO and DMO\(^{-1}\) to preserve this property because there are no dip or amplitude variations in this dataset. The results were unsatisfactory. Comparing the input CMP gather (Figure 3.1.1.a), where the amplitude is constant, with the same gather after 3D DMO (Figure 3.1.1.b) illustrates the problem. The non-uniform distribution of amplitude after DMO is the result of the random trace distribution.

The effects of irregular spatial sampling on DMO did not raise a great deal of attention in the industry, probably because most DMO implementations are directed towards achieving a good quality stacked section. The stacking process tends to cancel out most artifacts, especially when it is combined with some AGC function. In this work, pre-stack data are constructed and this last example illustrates why this artifact must be minimized. However, some discussion of the effects of irregular sampling on DMO was presented by Black and Schleicher (1989), Ronen et al. (1991) and by Beasley and Klotz (1992). Black and Schleicher suggested a method to account for the irregular sampling of data by designing and applying a correcting inverse filter to constant offset sections. Ronen et al. developed a dealiasing inversion scheme that improves DMO-stack. Beasley and Klotz suggested a method to equalize DMO amplitudes using a normalization factor based on the number of contributions each grid point receives.

We approached this problem from a slightly different angle, and considered the actual distribution of traces in midpoint-offset space as the key to deriving the correction function. The idea is to measure the distribution of input traces in the preliminary processing stages. The measurement is then passed as additional information to the DMO program, and is used to correct for the
Figure 3.1.1: An illustration of the problem associated with DMO and irregular spatial sampling, using a 3D synthetic dataset.
irregularities. The pre-processing program operates only on the coordinates and can serve as an additional tool for evaluating the survey sampling.

3.2 - Correction for irregular sampling

To analyze the problem that is introduced by irregular spatial sampling, one can investigate the trace distribution map. If a constant-offset DMO algorithm is applied, the distribution of traces in a constant-offset plane should be considered. Using k-t algorithm the whole midpoint-offset space is operated on simultaneously and therefore, one should consider the trace distribution for the whole space. When the input data are distributed irregularly in space, the vicinity of some grid points may be represented by many traces, while other grid points may not be represented at all. A conventional application of DMO treats every input trace independently from its neighbors, and therefore the overall contribution that an output point receives may not be correctly balanced. In fact, input traces that are located in finely sampled regions should be scaled down, while traces that are isolated from their neighbors should be given a larger weight. This procedure will balance the amplitude of DMO data.

This idea motivated the derivation of a weighting function that is based on the relative area represented by the trace in midpoint-offset space (Figure 3.2.1). The black circles mark the position of traces in a 2D midpoint-offset space, the polygons surrounding every point define the relative area that is occupied by that trace. In 3D, these polygons become three dimensional bodies. To balance DMO amplitudes we used the area (or volume, in 3D) as a weighting factor applied to every input trace before DMO. This procedure ensures that the
Figure 3.2.1: Polygon map - every input trace is represented by the relative area surrounding it.
Figure 3.2.2: Comparison between weighted DMO and conventional DMO when spatial sampling is irregular.
(a. is identical to Figure 3.1.1.b)
total weight of any subset of midpoint-offset space is proportional to its size, regardless of the number of points that sample it. Figure 3.2.2 presents the result of this weighting process. It demonstrates that weighting compensates for irregular distribution of input data, removes DMO artifacts and balances the amplitudes. Output data are very similar to the input data as required in this example.

3.3 - Derivation of the weighting function - an algorithm

3.3.1 - introduction

The area around any input point $P$ (Figure 3.2.1) is described by the locus of points $(m,li)$ which are closer to $P$ than to any other point. The problem of constructing these polygons is known as the "post-office problem" (Lekkerkerker, 1969). The following is a procedure that can be used to define the polygons; and it also illustrates the geometrical properties of this construction. First a line should be drawn from every point to all its adjacent points. Then a perpendicular should be built at the center of every such line. The intersections of all these perpendiculars define the break points that construct the polygons.

These areas can also be viewed by analogy to a crystal growing process (Gilbert, 1962), where every input point initiates the beginning of a circular and uniform growth of a crystal. At the end of the growing process all crystals join each other and form the required polygon setup.

Both methods can provide an insight to the construction of a computation algorithm that defines the polygons. However, in both cases the definition of the
area around every point can become very complicated and expensive, because
the mutual relationship between many millions of points (as in a typical seismic
3D survey) has to be considered. Therefore, we decided to settle for an
approximation of the weighting factors. The method we used ensures that the
total area (volume in 3D) covered by the seismic survey is distributed between
all input points.

The algorithm includes a projection of trace locations onto a regular grid,
before evaluating the weights. The grid approach greatly simplifies the
computation and provides the means for deriving a systematic and fast
computation scheme.

3.3.2. Computer algorithm

The first step is the definition a fine midpoint-offset grid. The position of
every trace in the dataset is then rounded to the nearest grid point. Obviously,
the accuracy of the final weighting function is directly related to the fineness of
this grid. In the beginning of the process the weight of every square of the grid is
set to one and if the dataset is regular, every grid point is represented by one
trace (Figure 3.3.1) and every trace receives the same weight. When one trace
is missing, the polygon construction changes (Figure 3.3.2) and the area that
belong to the missing trace is divided between four nearest grid points. In a
more general case, few adjacent traces may be missing and the construction of
polygons becomes more complex. An example is given in Figure 3.3.3 where
three adjacent traces are missing. Here the area that belongs to a missing trace
Fig 3.3.1: Polygon construction for regular 2D dataset.

Fig 3.3.2: Polygon construction for regular 2D dataset with one missing trace.

Fig 3.3.3: Polygon construction with three missing traces.
is not divided equally between the surrounding "live" grid points but the pattern is less obvious.

Since we were concerned with the construction of an efficient computer algorithm, we chose to approach this problem by considering a way to divide the unoccupied area between the "live" points. With this approach a systematic search for unoccupied grid point is performed. Once an empty grid point is encountered the area is divided between its neighbors. Although other routes exist, this approach is systematic, simple to implement in the computer in an efficient way and above all, it requires only one iteration.

The exact ratio in which the unoccupied area should be divided between surrounding points depends on the geometrical relationship between those points. But for the sake of simplicity and computational speed we choose to approximate it by dividing the area equally between surrounding live points.

The computer program that extracts the weighting function for a 2D seismic survey attaches two numbers, P and Q to every grid point of the fine grid. The first number, P, represent the number of traces that are positioned close to this grid point. The second number, Q, represent the relative area that is occupied by that grid point. The final weight is given by $Q/P$. The following is a summary of the computer algorithm for extracting weighting function:

**Step 1:** Read coordinates of every input trace and round it to the nearest fine grid point. At the end of this stage P is attached to every live grid point and represents the number of traces that fall in that rectangle. At this stage Q is set to 1 and represents the initial weight for that grid point.
Step 2: Calculate Q - search for an empty grid point and count how many live traces occupy the immediate square around it. If there are no live points around it, consider the next larger square (Figure 3.3.4) and continue this search until at least one live grid point is encountered. If $N$ is the number of live points in this surrounding square the weight ($Q$) of each of these points is increased by $1/N$. The process continues until the area of all empty grid points is divided between live grid points.

Step 3: The final weight of any grid point is computed by dividing the total accumulated weight ($Q$) by the number of traces that originally fell in that grid point ($P$).

Step 4: Every input trace receives a weight from its nearest grid point.

In 3D the process is similar but instead of squares we consider cubes. This method ensures that the total weight is proportional to the area (or volume in 3D) that is occupied by the seismic survey. On the other hand, if there are large "holes" in the survey, the weighting procedure may not solve the sampling problem (see next section) and may distort the result by assigning very large weight to traces around large holes in the data. We have limited the maximum weight that is assigned to any individual trace.
Fig 3.3.4: Search for live grid point: the first square around the center point is empty, the second square has one live point.
3.4 - Correction for insufficient sampling

3.4.1 - Formulation of the problem

When the distance between input traces becomes much larger than the regular grid distance, the application of weighting factors, as described in the previous section, fails to remove DMO artifacts. The data are insufficiently sampled for DMO. The effect of this sampling situation on DMO is demonstrated in Figure 3.4.1 where DMO and weighted DMO were applied to a synthetic experiment similar to the one previously described. This time we introduced larger distances between shots (up to 600% from the regular grid distances). It is clear that for this example, the weighting function did not remove the artifacts although it did, to some extent, regularize the amplitudes. In order to achieve good DMO results, the empty regions in the data cube should be replaced by interpolated traces.

DMO itself was suggested as an interpolation mechanism by Deregsowski (1986). Ronen (1987) described an interpolation procedure using DMO, NMO-stack, (NMO-stack)\(^{-1}\) and DMO\(^{-1}\). However, with this approach, the procedure for reconstructing true amplitude variations with offset, is not clear. A close look at the feathering correction scheme reveals that it is also an interpolation procedure that generates data traces at all grid points. This suggests some iterative application that will gradually improve DMO results and reduce the artifacts, by introducing additional interpolated traces into the original dataset. The advantage of this approach lies in the fact that DMO can preserve AVO and with that can be used to construct multi-offset traces.
Figure 3.4.1: Synthetic example demonstrating the effects of insufficient sampling on DMO results.
Figure 3.4.2: Line A - distribution map of input "live" traces.

(o - marks the position of a trace)
Fig 3.4.3.a: Input synthetic CMP gather.
At location L1

At location L2

Fig 3.4.3.b: Two CMP gathers (at different locations) after weighted DMO.
To examine this idea we used a 2D land survey that was acquired in a very irregular fashion (Line A). We have extracted the shot and receiver coordinates from the dataset and replaced the seismic information with synthetic data in the same location. The midpoint-offset distribution map (Figure 3.4.2) shows that there are many missing traces in the dataset and some regions are clearly under sampled. The horizontal interfaces model was used here again to generate synthetic data with NMO correction. This time we have added amplitude variations with offset (Figure 3.4.3.a). Amplitude of the top event increases exponentially with offset, while the bottom event decays linearly with offset. Again, since there is no dip in this model, we expect DMO to preserve the original amplitude distribution. However, the result of weighted DMO applied to this dataset (Figure 3.4.3.b) did not fulfill this expectation, and indicates that line A is not suitable for regular DMO. Some modification of the algorithm is required.

3.4.2 - Least-Squares DMO.

The modification to DMO that we have applied was motivated by some observations. The first one was that seismic data after NMO and DMO is organized along flat events within a CMP gather and in the offset direction. This suggests interpolating in the offset direction, after NMO and DMO. The second observation comes from the horizontal interface example. In this model all input traces are identical (Figure 3.1.1) and if the acquisition geometry is regular, DMO results should be identical to the input. Consequently, in the frequency domain, every input frequency slice will contain the same value at all grid points and DMO data should be the same. When DMO is applied to such a frequency
slice, the result can be used as a quality measurement. We know that if the data
was regular, all grid points after DMO would contain the same value. So the
difference between the actual value and the desired value provides a measure
of quality for the result. A combination of these observations leads to the Least-
Squares DMO algorithm. This algorithm is a modification of the regular DMO in
the F-X domain, as described in chapter 2. It is designed to interpolate pre-stack
DMO data when the original data sampling is insufficient but permits
interpolation.

Figure (3.4.4) is a schematic data distribution map in 2D midpoint-offset
space. Here the map describes one frequency slice and the black circles
represent one frequency value. If D is the data value, and Q is the relative area
around it, we mark y as the result of weighted DMO:

\[ D \cdot Q \xrightarrow{DMO} y \]

In addition, we apply DMO to the weighting function Q to obtain a new weighting
function W:

\[ Q \xrightarrow{DMO} W \]

The operation is motivated, as explained, by the horizontal reflectors example. It
is designed to acquire a quality measurement (W) for the output trace that was
generated by DMO. Interpolation in the offset direction is performed by fitting a
quadratic offset function to every frequency component. The least-squares
problem is defined using W as weight:

\[
\sum_{i=1}^{\text{max offset}} |a + bk_i + ck_i^2 - y_i|^2 \cdot |w_i|^2 = \min
\]

3.4.1

where:

k = DMO offset.
\[ D \times Q \xrightarrow{DMO} y \]
\[ Q \xrightarrow{DMO} W \]

Fig 3.4.4: Polygon map for one frequency slice
The choice of a quadratic function was motivated by the attempt to preserve AVO in multi-offset data, and is based on an approximation to the Zoeppritz equations (Shuey, 1985; Wang and Gardner, 1986) which describe the amplitude behavior with offset for seismic data.

The system of equations that formulate the least squares problems was modified from:

\[
\begin{vmatrix}
\sum |w_i|^2 & \sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2 \\
\sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2 & \sum k_i |w_i|^2 \\
\sum k_i^4 |w_i|^2 & \sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2
\end{vmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= \begin{pmatrix}
\sum y_i |w_i|^2 \\
\sum k_i y_i |w_i|^2 \\
\sum k_i^2 y_i |w_i|^2
\end{pmatrix}
\]

3.4.2

to:

\[
\begin{vmatrix}
\sum |w_i|^2 & \sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2 \\
\sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2 & \sum k_i |w_i|^2 \\
\sum k_i^4 |w_i|^2 & \sum k_i^2 |w_i|^2 & \sum k_i^4 |w_i|^2
\end{vmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= \begin{pmatrix}
\sum y_i |w_i| \\
\sum k_i y_i |w_i| \\
\sum k_i^2 y_i |w_i|
\end{pmatrix}
\]

3.4.3

so that if this DMO operation is applied to the weighting function alone, i.e. \( y_i = w_i \) for every \( i \), the solution of (3.4.3) will be: \( (a, b, c) = (1, 0, 0) \), which is the desired output for this case.

3.4.3. - Examples

Least-Squares DMO was tested using line A. When regular weighted DMO was applied to the synthetic dataset that was generated according to the
At location L1

At location L2

Fig 3.4.5: Two CMP gathers after Least-Squares DMO (compare with Figure 3.4.3.b).
geometry of line A, the result was not satisfactory and with that, pointed out the sampling problems of this dataset. Least-Squares DMO (Figure 3.4.5) corrected very well for the inaccuracies caused by irregular trace distribution, the result is free of artifacts. A comparison with the input data (Figure 3.4.3.b) shows that AVO is very well preserved.

Weighted DMO and Least-Squares DMO were also applied to the real data of line A. The trace distribution map for this example is the same as before (Figure 3.4.2). The data were recorded on land and had an AVO anomaly over a producing gas field. Figure 3.4.6.a displays one CMP gather from this input data at location L1. Notice that there are many offsets missing at this location. Weighted DMO filled the missing traces with some data (Figure 3.4.6.b) but the quality of these filled-in traces does not seem to be very good. A synthetic gather in the same location (Figure 3.4.7) shows irregularities in amplitude with offset and points out the problem at this location. Results from applying Least-Squares DMO to the same CMP gather are presented in Figure (3.4.8.c). The coherent events in this gather seems to have been honored, while random noise was removed. On the other hand, the data do not look like ordinary seismic data and therefore requires some further evaluation. Another CMP gather located over a bright spot (location L2) is presented in Figure (3.4.8.a). This gather has a coherent event that exhibits an AVO anomaly. Weighted DMO in the same location (Figure 3.4.8.b) destroyed the continuity of this event due poor sampling in the vicinity of this midpoint. However, Least-Squares DMO (Figure 3.4.8.c) constructs a continuous event and preserves the AVO anomaly well. Notice that the first event marked "A" has amplitude increasing with offset, while the event directly beneath it ("B") shows decrease of amplitude with offset. Comparing the
Fig 3.4.6: Line A, CMP gather at location L1 (see location map Figure 3.4.2)
Fig 3.4.7: Synthetic CMP gather at location L1 after DMO. Presented as a demonstration of the problem in building a CMP gather at this location with conventional DMO (The real data is presented in Figure 3.4.6)
Fig 3.4.8: Line A, CMP gather at location L2 (see location map Figure 3.4.2)
results with the input data indicates that this character of both events exist in the input data and is preserved by Least-Squares DMO.

Since Least-Squares DMO changes very much the appearance of the data, we used the stacked sections to assess the results. Comparing the DMO stack (Figure 3.4.9) with Least-Squares DMO stack (Figure 3.4.10) shows very little differences between the two sections. This occurs although the CMP gathers that constructed these two stacked sections were quite different from one another. The above leads to the conclusion that Least-Squares DMO did not destroy any coherent and significant information from the data, but it preserved and even emphasized the important amplitude information in the pre-stack data.

3.4.4 - Discussion

The last example demonstrated that DMO fills in missing traces, but it also indicated that in some cases this infilling may not be so good. Following Ronen (1987), a simple way to interpolate for missing data could be carried out by stacking the data and distributing the stacked traces at all offsets. However, if few traces in a CMP gather are "good", fitting some curve to the amplitude along the offset can do a similar job i.e., preserve the average of all values along the offset. The advantage of this approach is clear: AVO information is maintained. Comparing stacked sections (3.4.9) and (3.4.10) is a confirmation that this can be achieved. Least-Squares DMO has an additional important feature. It determines \textit{a-priori} the quality of the trace, based on the geometry of the survey, and attaches this quality measurement to the trace in the interpolation process. The investigation of the survey geometry provides an idea of the relation
Fig 3.4.9: Line A, Weighted DMO & stack.
Fig 3.4.10: Line A, Least-Squares DMO & stack.
between the actual output and the desired output. This information is imposed on the interpolation process and provides an additional correction.

Fitting a quadratic $a + bk + ck^2$ provides the recipe for calculating the amplitude of every trace as a function of offset, but it also supplies three additional attributes ($a, b,$ and $c$) for every sample in a CMP. These attributes can be plotted in similarity to a stacked section, indicating the amplitude of normal incidence reflection ($a$), and the rate of amplitude increase in the offset direction ($c$). This is a by-product of Least-Squares DMO that may be useful for detecting AVO anomalies and for building true amplitude time sections. This idea is demonstrated with line A. A small part of this line, over the AVO anomaly, is presented in Figure 3.4.11.a. Those data had passed through NMO, DMO and stack. Figure 3.4.11.b is the normal incidence section. The display was generated by plotting the coefficient $\varepsilon$ from equation (3.4.1). However, $a$, $b$ and $c$ are complex numbers generated in the frequency domain. Before the section was displayed, the coefficients were converted from frequency to time. It is interesting to compare the normal incidence section to the stacked section. The stack is the average of all amplitudes in the offset direction. This is equivalent to fitting a constant function in a least-squares sense. The normal incidence section is based on a similar idea but here we take into account the variations of amplitude with offset, and use the normal incidence amplitude (amplitude at offset zero) instead of the average amplitude. Compared to conventional stacking the amplitude of the normal incidence section can be more accurately related to the reflection coefficient. And since both processes are based on least-squares fit they have similar power of enhancing signal and reducing noise. And indeed, with line A the overall quality of the normal incidence section
a) Line A NMO, DMO and stack.

Fig3.4.11: Attributes of Least-Squares DMO.  
Least-Squares DMO is performed by fitting a quadratic in the offset direction (trace = a+bk+ck² and k=offset). a is the amplitude at normal incidence and c is the rate of AVO change.
b) Normal incidence section (coefficient a).
c) Plot of coefficient $c$. 
d) Normalized section \((c\cdot\text{sign}(a))\).
e) Comparison between the sections using one trace through the AVO anomaly. (Same trace is repeated 5 times).
is not inferior to the stack; on the contrary. Comparing the normal incidence section with the regular stacks shows that the amplitude values are different. Compared with the regular stacked section, the bright spot in the normal incidence section covers a narrower area. A close inspection of the AVO anomaly that is associated with this bright spot indicates that the amplitude information, as seen in the normal incidence section, is more likely to be correct.

Another interesting section is the display of coefficient of the quadratic variable, \( c \), which indicates the rate of amplitude change with offset. Figure 3.4.11.c displays coefficient \( c \). Here the bright spot is very pronounced, indicating a probable AVO anomaly. However, some shallow events that are not related to AVO anomalies are also quite bright. This is because the polarity of the normal incidence section was not taken into account in \( c \). So a fast AVO decrease is also positive on this section. To separate AVO anomalies from the other events, we multiplied the section by the polarity of the normal incidence section (a). Figure 3.4.11.d is the result and here a positive trace excursion indicates an AVO anomaly. Indeed in this location the AVO anomaly stands out very well. A summary of these displays with one trace through the bright spot is presented in Figure 3.4.11.e.

Least-Squares DMO operates in the same frequency domain as regular weighted DMO and therefore is easily applied to D datasets. When operating in the frequency domain, every slice can be computed in memory. This feature of the algorithm simplifies the process and could be very useful within an iterative procedure that involves DMO and DMO\(^{-1}\). Such a procedure can be used to gradually improve the quality of a DMO dataset. All iterations can be performed
in memory without repeating the costly steps that involve moving data in and out of memory.

The main drawback of this algorithm relates to the fact that NMO has to be applied beforehand. This of course requires knowledge of the velocity function. The fact that interpolation is performed after NMO breaks the flow of the full migration procedure that is described in the next chapters. So, if Least-Squares DMO is to be applied before imaging, an additional step that includes conversion of frequency to time, NMO$^{-1}$ and conversion back to frequency, is required. This complicates the procedure and in practice defines the interpolation as a separate step.

Another problem that should be pointed out is that the Least Squares DMO implicitly assumes horizontal interfaces. In steep dip situations this mechanism may require modification.
Chapter 4

Pre-Stack Imaging (PSI) in 3D - a time migration

4.1 - Introduction

The second scheme that is presented in this thesis is equivalent to pre-stack time migration in 3D. The migration is based on PSI technique (Gardner et al., 1986) and fits naturally into the frame of this work. PSI requires a regularized data and pre-processing with DMO, so the reorganization scheme is a necessary first step. One could stop there and settle for a 2D imaging procedure or, without a great increase in cost, continue to 3D PSI. The implementation of PSI that is described in this chapter is based on a simple interaction with DMO, it does not require a large increase in computer resources. Therefore, compared to many other 3D pre-stack migration methods, it is a very attractive approach.

The main advantage of PSI over other pre-stack migration techniques is its independence from velocity. With PSI, imaging precedes velocity analysis. The result is organized on a regular grid, as a set of migrated midpoint-offset gathers. Following the imaging step, a conventional velocity analysis is used to determine the velocities, and conventional application of NMO and stack yields a zero-offset migrated time image.

The fact that the velocity analysis is the very last step in this migration scheme is the main advantage of PSI. The imaging procedure, that is the expensive part, is independent from the velocity field and thus performed only once. The velocity analysis can be performed iteratively without repeating the
migration operation. After PSI, seismic events are migrated and consequently, velocity is estimated at the migrated position. Moreover, CMP gathers after PSI are free from diffractions and thus exhibit less conflicts between seismic events. PSI gathers also show higher S/N due to focusing of seismic energy. As a result, the resolution of the velocity panels is increased.

Because DMO was implemented in a velocity-independent way, the velocity estimation step, that is naturally built into PSI, is the very last step in the imaging procedure. 3D PSI is suitable for application in complex geology setting because it is performed before stack. On the other hand, being a time migration it does not handle correctly the lateral velocity variations, and from that aspect, is inferior to depth migration. So the issue of choosing between cost and accuracy becomes an important factor when selecting this approach. However, with today's computer technology, PSI in 3D maybe one of the few realistic approaches to imaging 3D seismic data before stack.

4.2 - PSI in 2D - a review

Time migration with the Kirchhoff approach (Schneider, 1978) is performed by an integration process. In the pre-stack form (Sullivan and Cohen, 1987) Kirchhoff migration is schematically described by:

$$V(X_{d},t_{0}) = \iint \text{Scal} \cdot X_{\text{Ph}}d\int d(X_{s},X_{t},t_{s}+t_{r})dX_{s}dX_{t},$$

4.2.1

where: $X_{s}$ - source location, $X_{r}$ - receiver location, $U(X_{s},X_{r},t)$ - pre-stack data, $V(X_{d},t_{0})$ - migrated image, Scale - scaling factor, Phase - phase operator,
\[ t_g = \sqrt{\frac{t_0^2}{4} + \frac{(X_d - X_g)^2}{v^2}}, \]

\[ t_r = \sqrt{\frac{t_0^2}{4} + \frac{(X_d - X_r)^2}{v^2}}. \]

From a kinematic point of view, equation (4.2.1) describes a procedure that sums all amplitudes along a diffraction surface. This surface is known as the "Pyramid of Cheops" and is described by the double square root equation:

\[ t = \sqrt{\frac{t_0^2}{4} + \frac{(X_g - X_d)^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(X_r - X_d)^2}{v^2}}. \]  \hspace{1cm} 4.2.2

Based on this diffraction surface we generated a synthetic 2D multi-offset dataset, it is displayed in Figure 4.2.1 using a 3D visualization program. Notice the intersection of one time slice with this curve, it is similar to a square. DMO simplifies the diffraction surface and converts it to a surface of revolution (Figure 4.2.2):

\[ t = 2 \sqrt{\frac{t_0^2}{4} + \frac{(X_m - X_d)^2}{v^2}} + \frac{k^2}{v^2}, \]  \hspace{1cm} 4.2.3

where: \( X_m \) - midpoint location,
\( k \) - offset (after DMO).

Pre-stack migration after DMO can be performed by changing the integration surface from (4.2.2) to (4.2.3). PSI is based on this simplification that DMO introduces to diffraction surfaces and is performed in two steps. In the first step, which is the "imaging step", individual time slices are processed separately. The
Pre-Stack Volume

Time Slice

Fig 4.2.1: Point Diffractor: Cheop's Pyramid.
second step involves the time axis and is equivalent to a standard velocity analysis, NMO and stack.

PSI is based on the fact that the intersection of any time slice with the DMO curve (Figure 4.2.2) is a circle in midpoint-offset space \((X_m, k)\). This property is expressed in equation (4.2.4) by fixing the time \(t\) and rearranging equation (4.2.3):

\[
(X_m - X_d)^2 + k^2 = \frac{v^2}{4} (t^2 - t_0^2) = R^2.
\]  

4.2.4

The center of the circle lies at the diffractor position \(X_d\) on the zero offset axis \((k=0)\). In the first step of PSI the amplitudes are summed along that circle and the result is assigned to its apex (Figure 4.2.4):

\[
U_{ps}(X_d, R, t) = \sum_{X_m} \sum_{k} U(X_m, k, t), \tag{4.2.5}
\]

where \(U(X, k, t)\) is the amplitude at point \((X, k)\), and time slice \(t\). The radius \((R)\) is a function of the velocity \((v)\), the depth of the diffractor \((t_0)\) and the time slice \((t)\), but operation (4.2.5) is well defined for every \(X_d\) and every \(R\) regardless of the velocity \(v\). At the end of this summation process all the amplitudes of the original diffraction curve are concentrated along a hyperbolic curve on a single CMP gather that is located at the diffraction point (Figure 4.2.3). The final step of this migration process involves the time axis; now all partial sums are added along the hyperbolic curve in offset-time space. The result is assigned to the apex of the hyperbola which is the final migrated position. This procedure is equivalent to a standard velocity analysis, NMO and stack.

In summary, PSI operates on separate time slices and is performed as a summation of amplitudes over circles. Sarmiento and Gardner (1986) pointed out that this procedure is equivalent to an exploding reflector forward modeling,
Pre-Stack Volume

Time Slice

Fig 4.2.2: Point Diffractor: Pre-Stack Seismic Data After DMO
Figure 4.2.3: Point Diffractor: Pre-Stack Seismic Data After PSI.
Figure 4.2.4: 2D PSI collapse a circle around (Xd,0) on a time slice into a point at (Xd,R).
if we substitute the offset coordinate $k$ of PSI with the depth/time coordinate of the wave propagation scheme. If the circle is an exploding reflector, a forward modeling process with constant velocity ($v=1$) will propagate the waves into the center, and the time would correspond to the radius of the circle ($R$), precisely as PSI requires (Figure 4.2.5). This analogy between PSI and constant velocity forward modeling motivated the application of an FK forward modeling algorithm that is analogous to the Stolt FK migration (Stolt, 1978).

In summary, PSI is carried out after the application of DMO, simply as an FK forward modeling operating on constant time slices (Sarmiento and Gardner, 1986). The advantage of the FK approach is its ability to preserve amplitude and phase that otherwise would require special considerations. The time slices after FK modeling should be scaled by $1/k$ because FK performs direct sums and causes amplitude increases with the circumference of the circle. This PSI procedure ignores the scaling factor in Kirchhoff integral (4.2.1). This factor is a combination of geometrical spreading and the obliquity factor. A simple compensation for geometrical spreading factor can be applied by multiplying input data by some function of time. But obliquity factor is not applied in this procedure. Generally PSI preserves phase but AVO distribution is destroyed. This issue is further discussed in the next section (4.3).

PSI was described here based on the assumption that the medium velocity is constant. Under this assumption the operation is independent from the velocity itself. But in fact the only significant property of the diffraction curve that is used in this derivation is the fact that after DMO the diffraction becomes a surface of revolution. It turns out that this property is not very sensitive to change in the vertical velocity function (see section 4.6.2) and therefore PSI is quite successful in more general cases.
Fig 4.2.5: PSI as modeling in F-K domain. Waves are propagating from the exploding reflector towards the zero offset axis. At "time" corresponding to the radius of the circle all the energy will concentrate at a point.
4.3.- An example of PSI for a 2D marine dataset

Before describing the extension of PSI into 3D, a 2D example is presented. This example serves as a tool for analyzing and discussing the problems that are related to PSI. It is also used as a verification of the quality of PSI images and as an illustration of the velocity analysis procedure.

To demonstrate the effects of PSI we used a 2D line (Line B) from the Gulf of Mexico. A stack of the input data is presented in Figure 4.3.1. The velocities used to stack this section were derived by analyzing the input dataset. This section is composed of a thick sedimentary with faulted blocks. The positions of the faults cannot be identified clearly in this section. The first step in the imaging procedure is DMO (Figure 4.3.2). The regions that exhibit improvements by DMO are marked on the section. The changes are attributed mainly to the improvement in the stacking velocity function after DMO. The result from the next step (PSI) is presented in Figure 4.3.3. PSI has clarified the picture, defined the gross fault and imaged the dipping layers beneath it (marked on the section). The velocity function that was used to stack this section was derived after PSI. To assess the quality of the PSI results we also applied pre-stack depth migration to the same dataset (Figure 4.3.4) and it turned out that the depth migration had resolved the fault surface better. This is associated with the fact that depth migration handles lateral velocity variations. But, PSI was performed without any velocity information, while the depth migration required a detailed velocity function and several migration iterations. This comparison is important because it provides us with the insight of what can be expected from PSI in 3D.
Fig 4.3.2: Line B, DMO and stack.
Fig 4.3.3: Line B, PSI and stack.
Fig 4.3.4: Line B, Pre-stack depth migration.
Observing PSI before it is stacked provides another view of the algorithm. Figure 4.3.5 displays a CMP gather (#300) from the input data. The dominant event around 3.1 seconds is probably a reflection from the fault zone (marked on the stacked section in Figure 4.3.2) and thus associated with a very steep dip. Another CMP gather (#450) is displayed in Figure 4.3.6 and does not contain this fault zone reflection. The effect of PSI on the steep dip event can be observed using the same two CMP locations. The fault reflection that was originally positioned in CMP 300, was migrated out by PSI (Figure 4.3.7). This event can now be identified in CMP 450 (Figure 4.3.8). But PSI had an additional effect on the dipping events, it migrated them to the far offsets (as can be observed with CMP 450). So, in order to preserve dipping events, PSI should generate more offsets then originally were present in the dataset. With line B, 80 offset/CMP were originally collected. At the PSI stage we generated additional 50 offsets. Analyzing this effect of PSI on the pre-stack data leads to the following conclusions:

- Muting PSI data before stack serves as a dip filter, and therefore should be applied cautiously.
- Velocity analysis after PSI is more complex than regular stacking velocity analysis. Dipping events lack the information in the near offsets and therefore it is more difficult to fit them on a hyperbolic curve. Changing the velocity function will tend to move the zero offset time of these events.
- For dipping events, AVO information is not preserved by PSI.
Fig 4.3.5 Line B, CMP gather #300.
   a) raw data, b) with NMO correction and mute.
Fig 4.3.6 Line B, CMP gather #450.
   a) raw data, b) with NMO correction and mute.
Fig 4.3.7.a: Line B, CMP gather #300 after PSI.
Fig 4.3.7.b: Line B, CMP gather #300 after PSI, NMO and mute.
Fig 4.3.8.a: Line B, CMP gather #450 after PSI.
Fig 4.3.8.b: Line B, CMP gather #450 after PSI, NMO and mute.
4.4 - 3D PSI

For 3D cases PSI is derived in a similar way. After DMO the travel time surface of a point diffractor in 3D space is given by:

$$t = 2\sqrt{\frac{t_0^2}{4} + \frac{(X_m - X_d)^2 + (Y_m - Y_d)^2}{v^2} + \frac{k^2}{v^2}},$$ \hspace{1cm} (4.4.1)

where: \((X_m,Y_m)\) - midpoint coordinates, 
\((X_d,Y_d,Z_d)\) - location of diffractor, 
\(k\) - offset, 
\(v\) - velocity, 
\(t_0 = 2\cdot Z_d/v\).

In analogy with the 2D case, the intersection of this surface with a constant time slice is a sphere, centered around the diffraction point at offset zero:

$$(X_m - X_d)^2 + (Y_m - Y_d)^2 + k^2 = \frac{v^2}{4}(t^2 - t_0^2) = R^2.$$ \hspace{1cm} (4.4.2)

PSI is performed in 3D space by summing amplitudes along the sphere and assigning the result to its apex:

$$U_{psi}(X_d,Y_d,R,t) = \sum_{Y_m} \sum_{X_m} \sum_{k} U(X_m,Y_m,k,t).$$ \hspace{1cm} (4.4.3)

Once this is performed, all energy is concentrated at \((X_d,Y_d)\) with

$$t = \sqrt{t_0^2 + \frac{(2\cdot R)^2}{v^2}}.$$ \hspace{1cm} (4.4.4)
which is a hyperbola in time (t) and offset (R). At this stage, in similarity to the 2D case, standard velocity analysis, NMO and stack will complete the migration procedure.

However, based on the symmetry of the sphere, 3D PSI can be simplified by reducing (4.3.3) into a two-pass operation, i.e., a set of 2D operations in one direction followed by a set of 2D operations in the perpendicular direction. By introducing an intermediate coordinate Q, expression (4.4.2) can be separated into:

\begin{align*}
\text{a.} & \quad (Y_m - Y_d)^2 + k^2 = Q^2 \\
\text{b.} & \quad (X_m - X_d)^2 + Q^2 = R^2
\end{align*}

4.4.5

The operation that is described by equation (4.4.5.a) is equivalent to 2D PSI (in the crossline direction) that transforms \((X_m,Y_m,k)\) to \((X_m,Y_d,Q)\). Expression (4.4.5.b) is the inline operation and transforms \((X_m,Y_d,Q)\) to \((X_d,Y_d,R)\). The ability to separate 3D PSI into successive 2D operations is another attractive feature of PSI that makes this algorithm relatively easy to apply in 3D.

4.5 - The interaction between DMO and PSI - a computer implementation

Any time domain implementation of 3D pre-stack migration encounters problems similar to those of 3D DMO (see section 2.4.5). Therefore a realistic implementation should be formulated in the frequency domain. Fortunately, PSI is naturally applied in the frequency domain. PSI operates on separate time slices and the operation is identical for every slice. In other words, PSI is independent from time and therefore independent from frequency. Moreover,
Fig 4.5: Design of the computer program for 3D PSI in the frequency domain.
PSI is also independent from $\log(t)$ and consequently independent from the frequency components of $\log(t)$. This property is described simply by applying 2.4.10 to equation 4.4.3:

$$U_{\text{psi}}(X_d, Y_d, R, \tau) = \sum_{Y_m} \sum_{X_m} \sum_{k} U(X_m, Y_m, k, \tau),$$  \hspace{1cm} 4.5.1

and FFT:

$$\tilde{U}_{\text{psi}}(X_d, Y_d, R, f) = \sum_{Y_m} \sum_{X_m} \sum_{k} \tilde{U}(X_m, Y_m, k, f),$$  \hspace{1cm} 4.5.2

where:  
- $f$ - frequency of log (time),  
- $\tilde{U}$ - FFT($U$).

This is a very important characteristic of PSI, it makes the interaction between DMO and PSI in 3D very simple. Figure 4.5.1 is a summary of the 3D DMO - 3D PSI program. This design makes 3D PSI a simple and very fast operation. In fact after the frequency slices are generated, PSI is much faster than DMO.

4.6. - Examples

4.6.1 - Impulse response

To test the impulse response from the PSI program we generated a dataset that was composed of one "live" trace, placed at the center midpoint at zero offset. This trace contains a spike every 0.5 sec. All other traces were zero out. After applying PSI the dataset was stacked, first with $v = 2000$ (Figure 4.6.1) m/sec and then with $v = 3000$ m/sec (Figure 4.6.2). This experiment illustrates the velocity independent nature of PSI. PSI was performed only once, then, by
Fig 4.6.1: Impulse response from the PSI program (v=2000 m/sec).
Fig 4.6.2: Impulse response from the PSI program (\(v=3000\) m/sec).
changing the stacking velocity, the image has changed. This test of PSI impulse response show correct kinematic properties and is similar to impulse responses derived from phase-shift migration programs.

4.6.2 - Diffractions in vertically varying medium

DMO and PSI were derived based on a constant velocity assumption. However the only property that is used in the derivation of PSI is that after DMO diffraction surfaces become surfaces of revolution, so they have a circular trajectory on fixed time slices. This 2D example is a designed to show that PSI is not very sensitive to vertical velocity variations and works quite well in the presence of vertical velocity gradient.

The input data was generated with a Kirchhoff modeling routine using a simple model with three diffraction points at different locations in \((x,z)\). The velocity model had a substantial vertical gradient of 1500 m/sec in every 1000 m. Figure 4.6.3 displays a zero-offset picture of this dataset. Notice that the diffractions are not hyperbolic, this is due to the velocity gradient. However, a cross section through the pre-stack data volume after DMO (Figure 4.6.4) shows that the diffraction appear to be circular on the time slice. This is the reason why PSI is successful in collapsing the circle to a point (Figure 4.6.5).
Fig 4.6.3: Zero-offset time section of 2D synthetic experiment.
input model contained three point diffractors and constant velocity gradient (1.5 1/sec)
(The line marks the location of the time slice that is presented in the next figure)
Fig 4.6.4: One time slice through the 2D multi-fold synthetic dataset - after PSI.

Fig 4.6.5: One time slice through the 2D multi-fold synthetic dataset - after DMO.
4.6.3 - A 3D land dataset

3D PSI was also applied to the ELF dataset. Figure 4.6.6 is a time section of line E2 from this 3D dataset. The same section, after 3D-PSI is presented in Figure 4.6.7. PSI had successfully migrated the section and defined the position of the faults. Although deeper parts of the section are not very well resolved due to poor S/N, the upper half shows improvement after PSI. This improvement in the continuity of the reflectors (marked A on the section) is probably due to imaging before stack. The repositioning of events that occurred after 3D PSI had some clear 3D components, for example, the position of the right fault (marked B on the section) with respect to its unmigrated position (Figure 4.6.6) has been substantially displaced. This is likely to be a 3D effect and could not have been explained with a 2D procedure alone.

Another important aspect of 3D PSI is the enhancement of S/N. This is important for the velocity analysis procedure, and is also demonstrated with this line. Figure 4.6.8 is a velocity analysis semblance plot from line E2. This plot was generated after DMO. A similar plot at the same location was generated after 3D PSI (Figure 4.6.9). The comparison between these two displays shows clear enhancements of S/N due to 3D-PSI. There are two important implications from this increase of S/N. The first one is of course the increase in resolution of the seismic image by PSI. The second implication is related to the improvements of semblance plots themselves by PSI. It provides a better definition of the velocity function and consequently a better seismic image.
Fig 4.6.6: ELF dataset, line E2, 3D-DMO & stack.
Fig 4.6.7: ELF dataset, line E2, 3D-PSI.
Fig 4.6.8: ELF dataset, line E2, velocity analysis after DMO.
Fig 4.6.9: ELF dataset, line E2, velocity analysis after 3D PSI.
Chapter 5

Two-pass 3D pre-stack depth migration

5.1 - Introduction

In the previous chapter, a scheme for performing pre-stack time migration was presented. However, time migration does not handle correctly lateral velocity variations (Hubral, 1977; Judson et al., 1980) and for complex geology pre-stack depth migration is called for. Experience with pre-stack migration indicates that the quality of the depth image is very sensitive to the accuracy of the velocity model and therefore pre-stack depth migration is also used for velocity estimation. The final depth image is usually obtained in an iterative procedure, and the velocity model is updated between iterations.

This procedure is very successful in 2D and therefore extending it into 3D seems like a good idea. Ideally the velocity analysis procedure should also be extended into 3D, because the complexity of three-dimensional structures effect the velocity estimation process (Levin, 1971). However, pre-stack depth migration in 3D, especially in the iterative mode, is limited by computer resources. The number of operations required for the full 3D procedure is very large and computation may take many weeks to complete. In addition, limitation in computer memory may impose restrictions on the maximum size of the output volume. Under these present conditions, a two-pass approach to 3D pre-stack imaging is an attractive idea. It can simplify the operation by reducing the size of
the data that has to be processed simultaneously. It also greatly reduces the overall computation time by requiring a smaller number of operations (see appendix B).

The term "two-pass 3D migration" is used to describe a procedure that separates the 3D migration into successive 2D operations in perpendicular directions. The idea of separation was discussed by Brown (1978). Gibson et al. (1983) suggested a two-pass approach to zero-offset 3D migration. Jakubowitz and Levin (1983) used the wave equation approach to verify that the two-pass approach, in the zero-offset case, is exact for constant velocity media. The extension of the two-pass method to include velocity variations with depth was derived by Pan and French (1989). Two-pass 3D migration of zero-offset data turned out to be a very successful approach and it is still commonly used for migrating large stacked 3D datasets.

For pre-stack implementations a two-pass approach has very attractive features. It offers greater flexibility in data manipulation, it speeds up computation time, it enables us to choose different procedures in different directions and it can be also used to split the 3D velocity estimation procedure.

However, the extension of two-pass approach into 3D is not obvious. Unlike zero-offset migration that is based on a single square-root equation:

\[ t = \sqrt{\frac{t_0^2}{4} + \frac{(X_m - X_d)^2}{v^2} + \frac{(Y_m - Y_d)^2}{v^2}} \]  
\[ 5.1.1 \]

the kinematics of pre-stack migration is described by a more complex DSR equation:

\[ t = \sqrt{\frac{t_0^2}{4} + \frac{(x_s - x_d)^2 + (y_s - y_d)^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(x_r - x_d)^2 + (y_r - y_d)^2}{v^2}} \]  
\[ 5.1.2 \]
The separation of (5.1.2) into a cascade of operation in \( X \) and \( Y \) is not obvious. Nonetheless, there are some methods that split the pre-stack migration procedure. Berryhill (1991) suggested a method for applying crossline migration to 3D multi-fold data. Gardner (1991) used a kinematic argument to show that after DMO, 3D pre-stack migration can be separated into a two-pass operation. This is done by applying zero-offset migration to constant-offset sections in the crossline direction and completing the process by 2D inline pre-stack depth migration.

We have chosen to use a two-pass approximation to 3D migration with PSI as the crossline operation. This approach has many merits. First of all it utilizes the simplification that DMO introduces to simplify the splitting of the pre-stack migration procedure. Secondly, it is independent from the velocity model and therefore performed only once. In addition, PSI can be easily implemented in the frequency domain, and with that solve the massive sorting problem that is involved in a crossline operation. Sorting can be done in memory for separate frequency slices. Formulating PSI as the crossline operation is the key that enables the splitting of the velocity estimation procedure.

So the idea is to separate the 3D pre-stack depth migration into two steps. The first step is a velocity independent time migration, applied in the crossline direction. The second step includes velocity estimation and 2D pre-stack depth migration in the inline direction. With this approach, the crossline migration is viewed as a pre-conditioning procedure. It is used first to organize the 3D data volume on a regular grid, and then to position seismic energy in the correct 2D plane. The crossline migration removes 3D effects from the pre-stack volume before the migration velocity model is defined. The velocity estimation process, as well as depth imaging, are applied as 2D procedures.
5.2 - Kinematic justification

Following the pre-stack migration procedure from the PSI point of view one has to apply DMO as the first step. DMO transforms a 3D diffraction surface from (5.1.2) to (4.4.1). The next step is to consider one time slice and sum all amplitude over a sphere (Equation (4.4.2)). With the two-pass approach we want to migrate the data first in the crossline direction. Fixing the inline position \( x_m \) in equation (4.4.2) results in a circle with radius \( Q \) centered at \( (y_d, k=0) \):

\[
(y_m - y_d)^2 + k^2 = R^2 - (x_m - x_d)^2 = Q^2. \tag{5.2.1}
\]

Expression (5.2.1) describes the kinematics of crossline PSI for any time slice. It is performed by adding all amplitude values along this circle and assigning the result to its apex:

\[
U_{cl-psl}(x_m, y_d, Q, t) = \sum_{y_m} \sum_k U(x_m, y_m, k, t). \tag{5.2.2}
\]

Operation (5.2.2) is defined for any point \((x_m, y_d, Q)\) regardless of the velocity \( v \). At this point the amplitude of the original migration surface (4.4.2) is concentrated in one plane of the 3D space at \( y_m = y_d \). Figure 5.2.1 shows how by successive application of 2D PSI in the crossline direction, the original sphere is collapsed into a circle (in \( m \) and \( k \)) centered over the diffraction point. The operation up to this stage follows the recipe of the full 3D PSI. At this point the PSI process is not completed but this intermediate circle is exactly equivalent to a 2D diffraction surface after DMO. This means that the 3D surface has been transformed into a 2D one. In other words, after the crossline summation step of PSI the energy is concentrated along
Fig 5.2.1: Crossline PSI collapses a sphere into a circle with coordinates y (crossline) and k offset, centered over the diffraction position. (This is a description of one time slice from a 3D diffraction surface).
\[
\frac{v^2}{4}(t^2 - t_0^2) - (X_m - X_d)^2 = Q^2
\]

or:

\[
t^2 = t_0^2 + \frac{4(X_m - X_d)^2}{v^2} + \frac{4Q^2}{v^2}.
\]

Expression (5.2.4) describes the hyperbolic curve that is identical to a 2D diffraction after DMO (in X (inline) direction). Out-of-plane effects from the original diffraction have been collapsed by the crossline operation, and velocity information was not used at all to perform the operation. The last step in this formulation is to remove DMO and get:

\[
t = \sqrt{\frac{t_0^2}{4} + \frac{(X_s - X_d)^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(X_t - X_d)^2}{v^2}}.
\]

This expression is the familiar DSR equation in 2D. It describes the migration in X direction and can be performed with any pre-stack migration program.

5.3 - Computer algorithm and implementation

The kinematic description of the two-pass migration is exact for constant velocity medium. The procedure is composed of the following steps:

- 3D DMO: Organizes the data on a regular grid and transforms the 3D multi-fold data into a separable form.
• Crossline PSI: Performs time migration in the crossline direction.

• 2D DMO⁻¹: Removes DMO from the data to prepare it for regular depth migration.

• 2D inline pre: The final step that includes a velocity estimation procedure stack depth and is performed as a 2D operation for all the inlines in the migration 3D volume. The result of this procedure is a depth migrated image in which 3D effects have been taken into account.

The computer implementation of the two-pass scheme is very similar to the full 3D PSI procedure. DMO, crossline PSI and DMO⁻¹ are independent from velocity and performed only once. This feature enables the formulation of all three steps in the frequency domain (Figure 5.3.1). The data are converted from frequency back to time only after the last step which is DMO⁻¹. At this stage the 3D data volume can be considered as being composed of a set of parallel multi-offset 2D lines. Ideally, all 3D effects have been removed, so the final step which provides the depth image is performed with the regular 2D pre-stack depth migration program. It can include the velocity estimation procedure that normally accompanies these migration techniques (Faye and Jeannot, 1986; Al-Yahya, 1989; Lafond and Levander, 1993).
Fig 5.3.1: Design of the two-pass computer program in the frequency domain.
5.4 - Discussion

The two-pass scheme should be placed between 3D pre-stack time migration and 3D pre-stack depth migration. It is in fact time migration in one direction and depth migration in the other direction. The two-pass scheme is only an approximation to the full 3D procedure. It is exact only for constant velocity medium. However, the next example in \( \nu(z) \) medium (section 5.5.1) was designed to show that it works well also in the presence of significant vertical velocity gradients. To assess the results in complex geological situations one has to try it on various datasets. The result should be compared to a 2D pre-stack depth migration, 3D zero-offset migration and 3D PSI (see section 5.5.2).

On the other hand, the two-pass procedure that we have described has many advantages. It is computationally fast (see appendix B), relatively easy to implement in the computer and provides the required output, i.e., velocity depth model and a depth image. The main advantage of the two-pass scheme is with the velocity estimation procedure. With a full 3D migration, velocities should be estimated by an iterative 3D procedure, which is still waiting for the new generation of computers. However, without this accurate velocity model the quality of the depth image maybe poor. In the two-pass scheme the migration algorithm is not exact, but the velocity estimation requires a 2D operation and therefore is easy to perform iteratively for a detailed velocity information. This velocity model should be a good approximation of the subsurface model, because 3D effects were corrected for.

Since the quality of the depth image is so sensitive to the accuracy of the velocity model, the two-pass approach may prove to be a very attractive solution
to pre-stack imaging of 3D seismic data, at least until the computation power that is required for an iterative 3D pre-stack depth migration become available.

5.5 - Examples

5.5.1- Synthetic example: line diffractors within a vertical velocity gradient

A simple synthetic example was designed to show that the two-pass scheme is successful in imaging 3D data in the presence of vertical velocity gradient. The 3D model contained three line diffractors within a medium that has 0.75 1/sec constant velocity gradient. One line diffractor was positioned parallel to the X axis with 20° dip. Second line diffractor, parallel to the Y axis, had 30° dip and a third line, with no dip, was positioned diagonally. A 3D multi-fold dataset was generated over this model. Figure 5.5.1 is a stacked section of one line from the input data. This cross section was taken along the first line diffractor (in the inline direction). A stack of another line that is located somewhat away from the diffractor position is displayed in Figure 5.5.2. The same two lines, after the application of the two-pass migration scheme, are presented in Figures 5.5.3 and 5.5.4 respectively. The data had passed through 3D-DMO, crossline PSI, Inverse DMO and inline pre-stack depth migration. This test was performed using MIGPACK program which is a finite-difference pre-stack depth migration program, based on downward continuation in common-shot and common-receiver domain. Depth positioning of all events in those final sections is correct and coincides with the input model.
First line diffractor
Second line diffractor
Third line diffractor

Fig 5.5.1: Stack of line #20 from 3D synthetic experiment
5.5.2 - A 3D land survey

The two-pass pre-stack migration procedure was applied to the ELF dataset and is illustrated using two lines from this dataset:

a) Line E2, effects of crossline migration:

Figure 5.5.5. displays a DMO-stack of line E3 from the ELF 3D land dataset. In the shallow section a folded structure is clearly defined but the deeper parts suffer from poor S/N and continuous reflectors cannot be identified. The purpose of this 3D experiment was the structural definition of the deeper strata and this section indicates that this cannot be achieved merely with DMO and stack. Figure 5.5.6 displays a stacked section of the same line after crossline PSI. This dataset was not migrated in the inline direction and therefore diffracted energy is noticeable and faulted planes are not very well defined. But the crossline migration had a very clear effect on this section: the deeper parts (marked with an arrow) now show continuous reflectors. This of course is due to the repositioning of events by the crossline migration, and could not have been achieved with a pure 2D procedure. Notice the boxed area in Figure 5.5.6 which shows great improvement compared to the DMO section. This improvement is important for the definition of the fault plane by inline migration.

b) Two-pass 3D pre-stack depth migration of line E1, a comprehensive comparison:
Fig 5.5.5: ELF dataset, line E3, DMO-stack.
Fig 5.5.6: ELF dataset, line E3, DMO, crossline PSI &-stack.
Line E1 is a good example of what can be expected from the two-pass scheme. Line E1 was used to assess the two-pass method versus 2D pre-stack procedures and 3D post-stack procedures. Figure 5.5.7 displays the DMO-stack of line E1 and Figure 5.5.8 displays a depth section of line E1 after the two-pass scheme. This last section has passed through 3D-DMO, crossline PSI, DMO⁻¹ and inline pre-stack depth migration that included migration velocity analysis. This final migration step was performed using the MIGPACK program. The migrated (two-pass) section converted back to time is presented in Figure 5.5.9. Comparing DMO-stack with the two-pass results illustrates the power of this migration scheme. The target area (marked around 2.2 sec) was very poorly defined and reflectors are broken up. The two-pass scheme clarifies the picture, defined a clear unconformity and brought out the reflectors that onlap on this marker (marked on the section). Even deeper than that, a trace of a fault may be identified on this section.

This result is compared with other imaging procedures. The first comparison is with a 2D pre-stack procedure that includes 3D DMO, inline DMO⁻¹ and inline pre-stack depth migration (Figure 5.5.10). Figure 5.5.11 is the same data converted back to time. Comparing 2D results with 3D results indicates the importance of 3D imaging procedures. The 2D migration, that was performed before stack was not successful in defining the unconformity zone.

The next possible solution is with a 3D procedure. We applied to this dataset a 3D time migration after stack (Figure 5.5.12). This experiment did not improve the result at the target zone. It was not sufficient to image beneath the complex faulted blocks and was inferior to the results from the two-pass imaging scheme.
The third and final comparison is made with 3D pre-stack time migration - 3D PSI (Figure 5.5.13). This comparison is also in favor of the two-pass section. This is probably because PSI is a time migration, it fails to image beneath complex structures since it does not take into account the complex ray-path through the faulted overburden.

This comparison indicates that the two-pass scheme provided a better picture compared to all the methods we tested. Of-course, to further establish this result more testing should be performed using other datasets and different geological settings. The computer algorithms that were used to generate the comparison sections may also have some influence on the quality of the results, and therefore additional comparisons versus other computer programs should be considered in order to learn more about this new method.
Fig 5.5.7: ELF dataset, line E1, DMO-stack.
Fig 5.5.8: ELF dataset, line E1, depth section after two-pass 3D pre-stack depth migration.
Fig 5.5.9: ELF dataset, line E1, time section after two-pass 3D pre-stack depth migration.
Fig 5.5.10: ELF dataset, line E1, depth section after 2D pre-stack depth migration.
Fig 5.5.11: ELF dataset, line E1, time section after 2D pre-stack depth migration.
Fig 5.5.12: ELF dataset, line E1, time section after 3D post-stack time migration.
Fig 5.5.13: ELF dataset, line E1, time section after 3D PSI.
Chapter 6

Conclusions

Two main routes are commonly considered for processing and migrating 3D seismic reflection information. The practical one uses binning to organize the data and zero-offset migration to image it. This approach ignores the special geometries of 3D surveys and fails in complex geological settings. The second path is a full 3D pre-stack depth migration. This approach is very promising but unfortunately is still an unrealistic computational undertaking. Both methods originated as an extension of successful 2D procedure.

We have successfully implemented an alternative approach for processing and imaging 3D seismic reflection data. This method was specifically designed to answer the special problems that are encountered when processing 3D datasets. It is based on an approximation of the full 3D migration procedure but provides answers to problems that are inherent to the other methods. Although it is a pre-stack operation it has proven to be a realistic computational task, we have successfully applied the procedure to a 3D seismic survey that was acquired on land with very complex acquisition geometry.

An important feature of the program is the reorganization scheme. Before imaging, the data are reorganized according to a new coordinate system that is independent from the original geometry of the survey. This is one key that enables the simplification of the migration procedure.
The scheme that was presented in this thesis is composed of different programs, but it should be observed as one modular scheme. Each one of the modules is in fact a building block to the next one, and the interaction between the modules is simple and efficient. This view of the imaging procedure was made possible by the special design of the program in the frequency domain. This design was another key that enables the efficient and practical application of the program in the computer. In the procedure that we have described, the most difficult computer task is the generation of data in the frequency domain. This is due to the large size of seismic datasets and limitations of computer I/O systems. Once the data are organized in the required way, four optional multi-fold data volumes can be generated without a large increase in computation cost. The following are optional output volumes from the complete scheme:

- DMO- 3D pre-stack volume,
- Pre-conditioned data volume (3D DMO, Crossline PSI & DMO\(^{-1}\)),
- 3D PSI (Pre-stack time migration),
- Depth image (two-pass 3D pre-stack depth migration).

The interaction between the different stages is describes in Figures 4.5.1 and 5.3.1. Every step is in some sense a higher approximation to the exact solution.

The question of spatial sampling, interpolation and infilling of missing data was also addressed within this work. This issue is of great importance when 3D seismic processing is concerned, since migration techniques fail when spatial sampling is not adequate. We have presented a solution to this problem that is suitable for application is some cases, but by no means a general one. It
is limited to simple geological setting with fairly small dips. However, as 3D pre-stack imaging procedures become more realistic, this problem will probably generate more interest and research. The effects of azimuth variations on the pre-stack migration process is left an open question, and so is the issue of utilizing DMO and migration as Q.C for spatial sampling. All these issues were encountered during the execution of this work and provide interesting and important subjects for future research.
References


Hale, I.D., 1991, Dip moveout processing, course notes series: v. 4, SEG publication.


Jakubowicz, H., Beasley, C., and Chambers, R., 1984, A comprehensive solution to problems in the processing of 3D data: presented at the SEG 54th annual meeting.


Appendix A

A verification that the DMO⁻¹ algorithm provides an exact inverse of DMO

For constant offset section DMO and DMO⁻¹ are exact inverse of one another because:

\[
\text{DMO}(F,K) = \text{INPUT}(F,K) \cdot G(F,K)
\]

\[
\text{DMO}^{-1}(F,K) = \text{DMO}(F,K) \cdot G^{-1}(F,K)
\]

and \(G^{-1}(F,K)\) (equation (2.5.3)) is defined as the inverse of \(G(F,K)\). There are no singularities in the forward or in the inverse functions. But the integral implementations (2.4.12) and (2.5.4) do not follow exactly the convolution analog of this scheme, and thus require verification.

To normalize DMO phase-shift function, we define \(K' = h \cdot K\). The normalized functions \(G(F,K')\) and \(G^{-1}(F,\frac{k}{h}K')\) are defined by equations (2.4.12) and (2.5.3) respectively. For every \(F\) and every \(K'\)

\[
G(F,K') \cdot G^{-1}(F,\frac{k}{h}K') = 1.
\]

and therefore

\[
\int_{-h}^{h} g(F, \frac{b - b'}{h}) \cdot g^{-1}(F, \frac{b'}{k}) db' = \delta(b). \tag{A.1}
\]
To verify that $DMO^{-1}$ is the exact inverse of $DMO$ one must show that the operation defined by equation (2.5.4) inverts equation (2.4.13), in other words $f = f$. By substituting (2.4.13) into (2.5.4) we get

$$f(F, x, h) = \int_{-h}^{h} \int_{-\infty}^{\infty} f(F, x - b - b') \cdot g(F, \frac{b}{h}) \cdot g^{-1}(F, \frac{b'}{k}) db db'. \quad \text{(A.2)}$$

Defining $a = b + b'$ we get:

$$f(F, x, h) = \int_{-\infty}^{\infty} f(F, x - a, h) \int_{-h}^{h} g(F, \frac{a - b'}{h}) \cdot g^{-1}(F, \frac{b'}{k}) db' da. \quad \text{(A.3)}$$

and substituting A.1 into A.3 gives:

$$f(F, x, h) = \int_{-\infty}^{\infty} f(F, x - a, h) \cdot \delta(a) da = f(F, x, h).$$

Q.E.D.
Appendix B

Comparing 3D pre-stack migration with two-pass scheme for computation time

To compare the computation requirements of 3D pre-stack migration with the two-pass pre-stack migration, an estimation of the number of operations in pre-stack migration is required. One way to derive this quantity is by considering the Kirchhoff approach. With this approach every input trace is spread along the output space. For pre-stack migration, the input space is all the pre-stack volume and the output space is equivalent to a zero offset volume.

Let:

- $N_x$ - number of midpoints in $x$ direction,
- $N_y$ - number of midpoints in $y$ direction,
- $N_h$ - number of offsets,
- $M$ - number of operation required for migration per input trace.

So, the number of operation in 3D pre-stack depth migration ($N_1$) is:

$$N_1 = M \times \text{number of input trace} \times \text{number of output traces}.$$ 

with:

- $N_x \times N_y \times N_h$ = number of input traces,
- $N_x \times N_y$ = number of output traces.

To simplify this evaluation let us assume that $N_x = N_y$ and thus $N_1 = M \times N_h \times N_x^4$.

The number of operation in the two-pass scheme ($N_2$) is given by:

$$N_x \times 2D \text{ migration} + N_y \times 2D \text{ migration}$$
and equals to:

\[ N_2 = 2 \cdot M \times Nh \times Nx^3. \]

The ratio between the number of operation in the two schemes is \( \frac{N_1}{N_2} = \frac{Nx}{2} \) and for a regular survey this factor could be 50-500.

The above figure is only a conservative approximation. For example, it took into account a simple implementation with one iteration. The difficulties that are associated with the computation of travel times (required for full 3D pre-stack migration) were also ignored. The assumption that \( M \) is identical for 2D and 3D is another simplification that was made in this estimation. Therefore, for an iterative implementation \( \frac{N_1}{N_2} \equiv Nx \) and if one considers the travel time computation problem then \( \frac{N_1}{N_2} \equiv Nx^2 \).
Appendix C

3D Datasets

In the course of this work we used two 3D datasets to test the computer programs, to study the results and to demonstrate the various algorithm that were used. The generation of synthetic 3D pre-stack data for this purpose is not possible at present since it is an enormous computational task. Therefore we performed a physical modeling experiment and used this dataset in addition to a real seismic dataset.

The physical modeling experiment was performed as a 3D experiment especially for this project. It was carried out at the Seismic Acoustic Laboratory in the University of Houston. The model itself was provided by Arco. This experiment was designed to simulate a 3D marine shooting geometry, i.e., it was collected as parallel shot lines with constant cable feathering. The model was composed of five layers with very severe geometry. The following are some parameters of the dataset, referred to as the "tank dataset":

Number of lines ------------------: 100
Number of shots/line -----------: 170
Number of receivers/shot ------: 60
Cable length -------------------: 7291 ft
Minimum offset ----------------: 1274
Receiver interval ---------------: 102 ft
Shot interval ------------------: 100 ft
Line interval ------------------: 170 ft
Feathering angle --------------: 11°
Sampling interval --------------: 0.002 sec
Maximum time -----------------: 4 sec
Data volume -------------------: 1,020,000 traces.

The second dataset was made available to us by ELF Aquitaine Production. This dataset is a small part of a land survey collected over an interesting complex of faulted blocks. Those data were collected using the following acquisition pattern: receiver lines were laid along the inline direction, shots were positioned perpendicular to the receiver lines and the acquisition progressed diagonally. The result was a survey that include a broad band of azimuths and offsets but their distribution between midpoints was not regular (Figure 2.1.1). The following are some parameters of the "Elf dataset":

Number of lines ---------------: 250
Number of CMP/line ------------: 208
Number of traces/CMP (bin) ----: ~25
Maximum offset ----------------: 2600 m
CMP interval -------------------: 25 m
Line interval ------------------: 25 m
Sampling interval ---------------: 0.004 sec
Maximum time ------------------: 5 sec
Data volume -------------------: 1,200,000 traces.
Appendix D

About the computer and the computer programs

In the course of this work we have developed five computer programs to process 3D multi-fold seismic data: 3D-DMO; Least-Squares DMO; Inverse DMO; 3D-PSI and 3D-weights which is a program that calculates the weighting function for 3D multi-fold data. All the programs were developed for the NEC-SX3 super-computer that was made available for this work at the Houston Advanced Research Center (HARC). The following are some details about the SX3:

Number of vector processors -----------: 2
Peak performance ---------------------: 2 Gflops
Memory -------------------------------: 512 Mb
Extended memory ----------------------: 2 Gb
Disk space (available for this work) ---: 7 Gb

The most complex programs were the 3D-DMO and 3D-PSI. Both programs are spatial convolution operations that process 3D pre-stack datasets, and produce a 3D pre-stack data volume. We have applied both processes to the ELF dataset and the following are some details related to the execution of the programs:
3D DMO:
Input data size: 1.2 M traces
Output data size: 1.3 M traces
CPU time: 2.5 hours
Elapsed time: 12 hours
Performance: ~500 Mflops

3D PSI:
Input data size: 1.3 M traces
Output data size: 1.7 M traces
CPU time: 2 hours
Elapsed time: 6 hours
Performance: ~800 Mflops