RICE UNIVERSITY

Kirchhoff Common Offset Migration Velocity Analysis via Differential Semblance

by

Jintan Li

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

Master of Science

APPROVED, THESIS COMMITTEE:

[Signatures]

William W. Symes
Professor of Computational and Applied Mathematics Department

Alan Levander
Professor of Earth Science Department

Colin Zelt
Associate Professor of Earth Science Department

HOUSTON, TEXAS

FEBRUARY, 2007
Abstract

**Kirchhoff Common Offset Migration Velocity Analysis via Differential Semblance**

by

Jintan Li

This work gives an efficient and fast approach to obtain an accurate velocity model for seismic imaging. An accurate velocity model is prerequisite for obtaining the image that represents the true underground geological structure: major evidence to explore for potential reservoirs and indicate hydrocarbons. This method optimizes a differential measure of the image gather flatness by comparing the residual moveout between neighboring traces. It stands out as a more accurate and automated inversion method among a good number of existing approaches. In this thesis, two versions of differential semblance velocity analysis are introduced: NMO-based and Kirchhoff-based. For each version, both the efficiency and effectiveness are discussed. For NMO-based differential semblance, the application is illustrated on both synthetic data set and field data set to invert both 1D and 2D velocity models. For the Kirchhoff-based differential semblance velocity analysis, the application is applied on the synthetic data set to invert a constant velocity model, providing the preliminary evidence of the effectiveness and insight for inverting depth varying or 2D velocity models for further research. Overall, this automated inversion method for velocity analysis is a potential technique for the velocity inversion for exploration.
Acknowledgements

First and foremost, I would like to thank my advisor Dr. William Symes for the guidance, support and opportunities he has provided me. He has been a model researcher and a great example to learn from. The other members of my committee have likewise played critical roles in my development as a student. I would therefore also like to thank my committee of Dr. Alan Levander and Dr. Colin Zelt for their insights and comments on my work. Additionally, I would like to thank Dr. Jianliang Qian, who provided the Eikonal solver code which played a huge role in this thesis. Finally, I would like to thank all of the members of the Rice Inversion project group for their insightful questions and comments regarding my research and talks.
## Contents

Abstract ii

Acknowledgements iii

List of Figures vii

1 Introduction 1
   1.1 Introduction .................................................. 1

List of Tables 1

2 Background 3
   2.1 Migration .................................................... 3
      2.1.1 Wave equation migration .................................. 3
      2.1.2 Kirchhoff migration ....................................... 4
   2.2 Kirchhoff migration ........................................... 5
      2.2.1 Traveltime calculation .................................... 6
   2.3 Existing versions of Kirchhoff-based DSVA ................... 7
      2.3.1 Objective .................................................. 8
      2.3.2 Traveltime calculation for Kirchhoff migration ........ 9
      2.3.3 Comparison ............................................... 9

3 Evaluation of NMO-based differential semblance 10
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Theory</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Algorithmic Details</td>
<td>14</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Data Preparation</td>
<td>14</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Velocity Representation</td>
<td>15</td>
</tr>
<tr>
<td>3.3.3</td>
<td>NMO Implementation</td>
<td>16</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Numerical Optimization</td>
<td>17</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Computational Cost</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Examples</td>
<td>18</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Example 1</td>
<td>18</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Example 2</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusions</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Kirchhoff-based Differential Semblance</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Formulation</td>
<td>32</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Computation of $d(t)$</td>
<td>34</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Computation of $\frac{\partial d}{\partial t}(t)$</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Work flow</td>
<td>35</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Image calculation</td>
<td>35</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Objective calculation</td>
<td>36</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Gradient calculation</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
<td>39</td>
</tr>
<tr>
<td>5.1</td>
<td>Synthetice data generation</td>
<td>39</td>
</tr>
<tr>
<td>5.2</td>
<td>CMP stack and Kirchhoff migration comparison</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>Traveltime implementation discussion</td>
<td>41</td>
</tr>
<tr>
<td>5.4</td>
<td>Objective function discussion</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Figures

3.1 Illustration of the Partially Irregular Grid ("PIGrid") data structure defined in the text. Velocity is given at arbitrarily spaced nodes in arbitrarily spaced “wells” (vertical lines) grouped in arbitrarily spaced “sections” (vertical planes). ........................................... 21

3.2 Example 1: 21 CMP gathers from first North Sea line. Preprocessing consists of zero-phase bandpass filter and mute of direct and head waves. 22

3.3 Example 1: A CMP before (above) and after (below) predictive deconvolution. ................................................................. 23

3.4 Example 1: NMO-Corrected CMP gathers before predictive deconvolution. 1D velocity model estimated by differential semblance velocity analysis. Note upward curvature of apparent primary at about 1.5 s. The presence of slower events (multiple reflections, presumably) has affected the velocity analysis. ........................................... 24

3.5 Example 1: NMO-Corrected CMP gathers after predictive deconvolution. 1D velocity model estimated by differential semblance velocity analysis. 1.5 s event is flattened. Apparent multiples (surface and/or pegleg) following it are suppressed by predictive deconvolution, and no longer dominate the velocity analysis. ................................. 25
3.6 Example 1: Comparison of RMS velocity with velocity scan, using data of Figure 3.2 before predictive deconvolution. Solid line = RMS velocity estimated by differential semblance velocity analysis; Grey-scale = velocity spectrum. Differential semblance RMS velocity appears to seek compromise between primary and multiple moveout velocities.

3.7 Example 1: Comparison of RMS velocity with velocity scan, using data of Figure 3.2 after predictive deconvolution. Solid line = RMS velocity estimated by differential semblance velocity analysis; Grey-scale = velocity spectrum. Suppression of coherent noise leads to more accurate velocities.

3.8 Example 2: 51 CMPs extracted from the second (Viking Graben) North Sea line. Preprocessing consisted of multiple suppression via hyperbolic Radon Transform, zero-phase bandpass filter, and mute.

3.9 Example 2: NMO corrected gathers using 1D velocity model estimated by differential semblance.

3.10 Detail from Figure 3.9.

3.11 Example 2: NMO corrected gathers using 2D velocity model estimated by differential semblance.

3.12 Detail from Figure 3.11.

3.13 DSVA: v(z) Interval velocities

3.14 DSVA: v(x, z) Interval velocities

4.1 Work flow of the calculation of DS objective and the gradient of the objective.

5.1 Underground structure of the test model

5.2 Stack of Kirchhoff migrated common midpoint gathers, grid size: nx=200 dx=10m ox=0; nz=270 dz=10m oz=0

5.3 Common offset image, different source locations
5.4 Common midpoint image gather: different offsets, same midpoint . . . 43
5.5 Stack of NMO-corrected common midpoint gathers . . . . . . . . . 44
5.6 Stack of Kirchhoff migrated common midpoint gathers. Grid Size=5m
(left), Grid Size=10m (right) . . . . . . . . . . . . . . . . . . . . . . 44
5.7 Stack of Kirchhoff migrated common midpoint gathers. Image from
2 point ray tracing with constant background velocity model (left)by
using the eikonal solver. Grid Size=20m (right) . . . . . . . . . . 45
5.8 DS objectives for velocity ranging from 1.1km/s to 2.0km/s. The cor-
rect average velocity is 1.5km/s . . . . . . . . . . . . . . . . . . . . . 45
5.9 The common image gather at the mid point location 1000m, traveltime
calculated using analytic formula (left) and eikonal solver(right) . . . 46
Chapter 1

Introduction

1.1 Introduction

Velocity analysis is a crucial step in seismic data processing. The subsurface of the earth is often modeled as layered medium composed of different kinds of rocks with different properties. It is important to know more about these rocks in the subsurface as they contain valuable mineral resources which we need. Many methods are used to image the subsurface, the most commonly used of which is the propagation of elastic waves. Since, the properties of rocks differ based on their depth and composition, they have different responses to the elastic waves, in other words, some rocks transmit waves faster than others. Then the velocity of these elastic waves is a very important characteristic which we need to decipher from the available traveltime from source to the receiver (both on surface) through rocks of varying velocities.

Velocity analysis has been investigated in numerous ways. But broadly, it can be classified into two separate ways:

1. by minimizing the error in the predicted data set and the real data set. The most popular way to do this is by tomography which was first introduced to invert the data from the prestack transmission. It was proposed in different ways: traveltime inversion([3],[4],[15],[42]), iterated Born inversion from...
the waveform([51],[34]), optimization waveform inversion using Monte-Carlo
algorithm([24]). These algorithms were searching for the most accurate model
that can best predict the data.

2. The other type of prestack velocity analysis is to optimize the quality of the im-
age by flattening the prestack image gathers. These employed semblance to mea-
sure the flatness of primary reflection events on the migration panel([60],[16],[12],[17]).

This thesis discusses a newer method of velocity analysis, which employs a new ob-
jective function, called “differential semblance”. “Differential semblance” here means
comparing the differences in images between neighboring parameter values (traces or
image gathers) which are free of local minima. It is based on the assumption that
neighboring traces in an adequately sampled image gather are spatially non-aliased. It
has well defined and smooth high frequency asymptotics and differs from conventional
velocity analysis by being automatic and quantitative. It avoids manually picking the
peaks on the semblance panel, the general approach in conventional velocity analysis;
instead, differential semblance velocity analysis can yield useful velocity estimates
automatically without picking.

Differential semblance velocity analysis (“DSVA”) has a number of versions: hyper-
bulic or ray-trace NMO correction in CMP and plane wave domains ([48], [33],
[47]); two way reverse time ([53],[45]); prestack Kirchhoff migration ([8], [35], [5]);
shot profile and DSR migration ([39],[30]). All these versions have been investigated
and have the same requirements to obtain reliable results: effective noise suppression
(multiples, unmodeled converted waves) and a velocity model that is smooth and
compatible with imaging engine.

In the thesis, I’m going to first give an evaluation of NMO-based DSVA compared
with conventional DSVA; then, I will propose Kirchhoff-based DSVA, and give some
preliminary evidence that Kirchhoff-based DSVA may be efficient.
Chapter 2

Background

This chapter will give a brief explanation of migration and velocity analysis.

2.1 Migration

Migration is a tool used in seismic imaging to get an accurate image with true subsurface positions. It involves geometric repositioning caused by dipping reflectors or diffractions, and the methodology is based on wave equation. There are two migration methods commonly used in oil industry: wave equation migration and Kirchhoff migration. All migrations discussed here refer to prestack migration.

2.1.1 Wave equation migration

Wave equation migration has been demonstrated to produce a better image than Kirchhoff migration for complex geological structure, but it is far more expensive, especially in 3D cases, because it requires a full bin data set that covers the whole aperture to get the entire image. A typical migration, shot profile prestack depth migration, is implemented by recursively extrapolating source and receiver wavefields independently by applying an appropriate imaging condition. The final image is constructed by stacking the images from each individual shot. The governing equations
for the wave extrapolators are the one-way wave equations. This one-way wave propagator has been studied by several researchers in different forms: phase-shift([19]), split-step Fourier migration([41]), phase-shift plus interpolation (PSPI)([40]), Fourier finite difference (FFD)([37]) and generalized screen propagators (GSP)([57],[56],[58],[25]), etc. The discussion of these one-way wave propagators are beyond the scope of this thesis.

2.1.2 Kirchhoff migration

Kirchhoff migration has traditionally been one of the most widely used migration methods, and it still has very dominant use in the oil industry, mostly because of its low computational cost. However, it has difficulties in reconstructing complex images due to the high-frequency asymptotic ray approximation. Also it fails when multi-pathing occurs or where low velocity zones result in inadequate ray coverage.

Kirchhoff migration originated as a diffraction summation that incorporates the obliquity, spherical spreading and wavelet shaping factors([59]). According to the Kirchhoff migration algorithm, each subsurface point is assumed to be a point source. The reflections from such a subsurface point form a diffraction pattern known as a migration curve. Summation over such a curve with the proper weighting function gives the value of the migrated section for that point. The collection of all these summed points forms the migrated section. French([18]) and Schneider([38]) have provided good mathematical references to the Kirchhoff integral solution to the wave equation and gave the first constant velocity Kirchhoff migration algorithms. A number of researchers have since investigated the Kirchhoff migration approach to handle complex background velocities ([55],[6],[22],[26]). Geoltrain and Brac([20]) in 1992 implemented Kirchhoff migration using Eikonal equations and a finite-difference high-order (65 degree) paraxial shot-record depth migration respectively, and gave a very detailed discussion of the image artifacts caused by using Kirchhoff migration based on first-arrival traveltime. The major cause of artifacts are the energetic
events that appear in more heterogeneous layers, caused by multiple arrivals, whose travel-times are different as they take different paths from source to receiver through different subsurface velocity structures. So even using the exact velocity will result in an overmigration of the data. This will affect the accuracy of velocity analysis subsequently.

A common offset first-arrival Kirchhoff depth migration formula based on the diffraction-stack migration is as follows: ([38])

\[ I(x, h) = \int A(x_s, x_r, x) d(x_r, x_s, T(x_r, x) + T(x_s, x)) dx_s dx_r \] (2.1)

- \( I(x, h) \): the image at the given offset, \( h \) is the offset. This can also be called "common offset image gather"
- \( x_s \): the source; \( x_r \) : the receiver
- \( x \): the scatterer.
- \( A(x_s, x_r, x) \): the ray amplitude when the ray path gets to the scatterer;
- \( T(x_r, x) + T(x_s, x) \): the two-way traveltime from the source to the scatterer, then to the receiver
- \( d(x_r, x_s, T(x_r, x) + T(x_s, x)) \): recorded seismogram at the location \( x \) where the two-way traveltime is equal to \( T(x_r, x) + T(x_s, x) \)

In this thesis, the traveltime is computed by the first-arrival solution of the acoustic eikonal equation. We do not consider the amplitude information in the seismic wavelet as we are only interested in the kinematic representation of the subsurface.

### 2.2 Kirchhoff migration

The two most important parts in Kirchhoff migration that affect the image quality are the calculation of traveltime and amplitude. Kirchhoff migration is based on
high-frequency ray approximation, which uses *Eikonal equation* for traveltime and *Transport equation* for amplitude. Since the amplitude is not considered at the moment, only traveltime calculation is introduced.

### 2.2.1 Traveltime calculation

There are two most commonly used methods to compute the travel time. They are called ray tracing and the Eikonal solver algorithm.

#### 2.2.1.1 Ray tracing algorithm

Ray tracing methods track the ray path between source and receiver that is followed by seismic energy. Ray tracing equations calculate the ray amplitude and phase which describe how the seismic energy follows a direction dictated by velocities variations in subsurface (on the lack of such). Basic ray tracing has two kinds: ray shooting and two-point ray tracing.

Ray shooting sends a number of rays from the source point to different directions, and the traveltime at the desired receiver location is computed by interpolation of the solutions along the rays. This method is preferred when the travel time is acquired for many receiver points.

The two point ray tracing method is preferred when the desired receiver locations are limited. This is also known as ray bending. This computes the travel time at the particular receiver, which does not require interpolation.

The commonly used two-point ray shooting methods formulates the ray equation as first initial value problem, which gives an arbitrary path between source and receiver. This geometry is iteratively adjusted until the path between source and receiver satisfies Fermat's principle([14]).

Ray tracing algorithm is considered to be highly accurate and relatively efficient, but for the complex velocity variation zones, the increase of accuracy will cause the tremendous increase of the computational cost([29]).
2.2.1.2 Eikonal solver

The Eikonal solver is a relatively fast and accurate method([54],[52]). Compared with ray tracing, it is less commonly used, but is more robust and avoids travel time interpolation to a regular grid. It directly solves the eikonal equations on a regular grid without computing ray-paths. However, the Eikonal solver only computes the first arrival travel time in a continuous medium, and it ignores the multiples arrivals which may be tremendously important to achieve the accurate imaging for complex velocity structure([20],[23]).

The following is the eikonal equation the solution of which are the traveltimes in Kirchhoff modeling in an isotropic media. Given a source point $x_s$, the first arrival traveltime field $\tau(x,x_s)$ is the viscosity solution of the eikonal equation ([32]) as follows:

$$|\nabla \tau(x,x_s)|^2 - \frac{1}{c^2(x)} = 0$$

(2.2)

where $c(x)$ is velocity.

The traveltime calculation employed in this thesis solves the eikonal equation directly by using a fast sweeping method, which uses nonlinear upwind difference and Gauss-Seidel iterations with the alternating sweeping ordering ([31]).

2.3 Existing versions of Kirchhoff-based DSVA

As mentioned in the Introduction, Kirchhoff-based DSVA has been implemented in two different versions by the following researchers: Chauris and Noble, Mulder and ten Krooder([8], [35]). As compared with the version employed in this thesis (Symes's version), the difference mainly lies in the formulation of the DS objective and traveltime calculation algorithms. The following section gives a comparison of these two versions and Symes's version in terms of objective and traveltime calculation.
2.3.1 Objective

Symes's version

Symes first gave a DS formula in 1991 ([48]), which has been rewritten in this thesis in the common offset image gather domain:

\[ J[v; d] = \frac{1}{2} \sum_x \sum_h |D_h I(x, h)|^2 \]  
\[ (2.3) \]

in which: \( I(x, h) \) is the image at the given offset \( h \), and \( D_h f(h) = (f(h + \Delta h) - f(h))/\Delta h \) is the forward h-difference operator. This formula is used in this thesis as the objective function.

Chauris and Noble version

The objective function is defined as:

\[ J[v; d] = \frac{1}{2} \sum_x \frac{\sum_h |D_h I(x, h)|^2}{\sum_h |I(x, h)|^2} \]  
\[ (2.4) \]

This formula (2.4) divides the derivatives by the energy of the common image gather ("CIG"), so that DS does not depend on amplitude.

Mulder and ten Kroode version

Mulder and ten Kroode have used the same formula (2.4) for the objective function except that they transform from the image domain to data domain. The objective looks like:

\[ J[v; d] = \sum_x \sum_h |H F D_h I(x, h)|^2 \]  
\[ (2.5) \]

Here \( F \) is the Born forward modeling operator that maps the reflectors to the data space. \( H \) represents a smoothing operator to avoid increasing the high-frequency content. The migration image \( I(x, h) \) is obtained from the data \( d \) by:

\[ I(x, h) = (F^* F)^{-1} F^* d \]  
\[ (2.6) \]

\( F^* \) is conjugate transpose of \( F \).
Moreover, they also modified the amplitudes for the DSO function: they used $F^*F$ as the true amplitude operator, and also used weights in the migration by counting the number of contributing ray pairs in each point.

2.3.2 Traveltime calculation for Kirchhoff migration

1. Symes version: Eikonal solver([31])

2. Chauris and Noble version: two point ray tracing

3. Mulder and ten Kroode version: two point ray tracing

2.3.3 Comparison

The objective function used in this thesis is expected to be sufficient to invert the velocity model, and it’s implemented in the image domain instead of data domain. Considering the amplitude operator of DSO which may also involve the velocity factor, I neglected it at the moment. The traveltime calculation using eikonal solver would be a good experiment to evaluate its accuracy and efficiency for the Kirchhoff migration image, and on velocity analysis subsequently.
Chapter 3

Evaluation of NMO-based differential semblance

The differential semblance method of velocity analysis flattens image gathers automatically by updating interval velocity to minimize the mean square difference of \textit{neighboring traces} in an image volume. An implementation using normal moveout correction as the “imaging” method is relatively fast, can accommodate arbitrary acquisition geometry, and can be organized to output 1D, 2D, or 3D interval velocity models. This variant of differential semblance velocity analysis is effective within the limits of its imaging methodology: lateral heterogeneity is mild and data dominated by primary events. Application to two sets of 2D marine CMP data illustrates both the ability of the method to quickly construct 1D or 2D velocity models which approximately flatten the NMO-corrected gathers, and the tendency of coherent noise events such as multiple reflections to degrade the quality of the estimated velocity model.
3.1 Introduction

Differential semblance velocity analysis ("DSVA") is an automated approach to prestack migration velocity estimation. Its basis is the observation that pairs of nearby image traces exhibit non-aliased residual moveout, even when the migration velocity is dramatically wrong. Objective measures of residual moveout based on this concept can be optimized with respect to migration velocity to produce objective velocity analyses which flatten image gathers automatically.

Many variants of DSVA have been constructed, based on a variety of prestack imaging methodologies: hyperbolic or ray-trace NMO correction in CMP and plane wave domains, two way reverse time, prestack Kirchhoff, and both shot profile and double square root one way wave equation methods have all been used ([48, 21, 49, 8, 35, 39]). All of these methods are limited by the requirements of prestack migration imaging:

- input data must be essentially primaries-only - in particular, multiple reflections must have been effectively suppressed prior to velocity analysis;

- trial velocity models must respect the requirements of the imaging engine, usually amounting to some measure of smoothness and possibly upper and lower velocity bounds, to ensure physical correctness and to control numerical artifacts.

DSVA based on hyperbolic normal moveout suffers from the most stringent applicability limits of any of the methods described above, but is also potentially the fastest, as it relies on the simplest imaging engine. Velocity estimates based on NMO are of reasonable utility in areas of low structural relief, either in themselves or as starting models for more sophisticated model-building exercises.

This paper describes an implementation of hyperbolic NMO-based DSVA with a number of features intended to assist in its assessment for eventual use in a production environment:
• industry standard data structures for input data and diagnostic output;

• flexible velocity modeling accommodating 1D, 2D, and 3D variation within the limits implicit in the imaging methodology;

• state-of-the-art numerical optimization.

Having described the implementation of the method, we present velocities and image gathers obtained by applying NMO-based DSVA to (parts of) two North Sea lines. These two marine 2D examples exhibit common features of DSVA: convergence to reasonable velocity estimates in a small number (on the order of ten) of iterations, each involving an imaging step and some side computations; highly aligned image gathers; agreement with standard velocity analysis; measured degradation in the presence of coherent noise.

3.2 Theory

NMO-based DSVA has been described a number of times, for example by Gockenbach and Symes [21]. The implementation described here is a variant of these, somewhat simplified and adapted to accommodate typical field acquisition geometries.

Shot and receiver $x, y$ coordinates will be denoted $x_s, y_s$ and $x_r, y_r$ respectively. Data traces $d(t, h, x_m, y_m)$ are binned by CMP coordinates $x_m, y_m$ and parametrized within the bin by (3D) offset $h = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2}$. Each midpoint is assigned an interval velocity function $v(t_0, x_m, y_m)$, viewed as a function of vertical two-way time $t_0$. The corresponding RMS square slowness $u(t_0, x_m, y_m)$, defined by

$$u(t_0, x_m, y_m) = \frac{t_0}{\int_0^{t_0} v^2},$$

determines the hyperbolic approximation to two-way time

$$T(t_0, x, x_m, y_m) = \sqrt{t_0^2 + u(t_0, x_m, y_m)h^2}.$$
The inverse function of \( t_0 \rightarrow T(t_0, h, x_m, y_m) \) is denoted \( T_0(t, h, x_m, y_m) \). This inverse function is well-defined where

\[
0 < t_0 + \frac{1}{2} \frac{\partial u}{\partial t_0}(t_0, x_m, y_m)x^2.
\]

The reciprocal of this quantity determines the amount of so-called NMO stretch.

We will restrict the search for velocities to those whose values lie between upper and lower velocity envelopes \( v_{\text{max}} \) and \( v_{\text{min}} \) and whose derivatives are smaller in absolute value than an \textit{a priori} bound on the velocity slope \( v'_{\text{max}} \). We call such velocities \textit{admissible}. It is possible to determine a domain of time and offset so that \( T_0 \) is well-defined on this domain for all admissible velocities, and a mute which is nonzero only on this domain. We shall assume that this mute has been applied to the data, so that we can freely assume \( T_0 \) to be well-defined in the formulas to follow. See Symes [44] for details.

With these provisos, the differential semblance objective function may be written

\[
J[v, d] = \frac{1}{2} \sum_{x_m, y_m} \sum_h \int dt_0 |D_h I(t_0, h, x_m, y_m)|^2,
\]

where \( I(t_0, h, x_m, y_m) \) is the NMO-corrected data

\[
I(t_0, h, x_m, y_m) = d(T(t_0, h, x_m, y_m), h, x_m, y_m).
\]

The offset divided difference operator \( D_x \) is defined as

\[
D_h I(t_0, h, x_m, y_m) = \frac{I(t_0, h + \Delta h, x_m, y_m) - I(t_0, h, x_m, y_m)}{\Delta h},
\]

in which \( \Delta h \) denoted the offset increment (which need not be uniform from trace pair to trace pair, or across CMPs).

This definition differs from others [21, 35] by the absence of smoothing and amplitude factors. Symes [44] shows that

- the amplitude factor may be neglected, and the smoothing factor absorbed in the data;
• the objective $J[v, d]$ is a smooth function of interval velocity and data, with a well defined gradient with respect to velocity;

• all stationary points occur at values which differ from the global minimum value (essentially zero) by an amount proportional to the dominant wavelength in the data.

That is, asymptotically speaking $J[v, d]$ has no spurious local minimizers. Therefore a Newton-like, gradient based optimization method may be used to flatten gathers by optimizing differential semblance.

In order to employ gradient-based optimization, it is necessary to compute the gradient of $J$. A tedious but straightforward computation yields a convenient computational expression for the gradient:

\[
\nabla_v J[v, d](t_0, x_m, y_m) = -v(t_0, x_m, y_m) \int_{t_0}^{t_0+\Delta t} d\tau_0 \sum_h \frac{h^2 u(\tau_0, x_m, y_m)^2}{\tau_0 T(\tau_0, h, x_m, y_m)} D_h I(\tau_0, h, x_m, y_m) D_x I'(\tau_0, h, x_m, y_m)
\]

in which $I'$ is the NMO-corrected time derivative of the data:

\[
I'(t_0, h, x_m, y_m) = \frac{\partial d}{\partial t}(T(t_0, h, x_m, y_m), h, x_m, y_m).
\]

### 3.3 Algorithmic Details

The following subsections discuss several key ingredients of a successful DSVA implementation: data format and preparation, velocity model representation, implementation of the NMO transformation, and numerical optimization.

#### 3.3.1 Data Preparation

Data traces are presorted into CMP gathers. Other attributes which must be correctly defined in each trace are source and receiver $x$ and $y$ coordinates, and source and
receiver depths. CMP $x$ and $y$ coordinates (as opposed to CMP index) and (3D) offset are computed from this fundamental data. We also require that the delay after the zero phase of the wavelet be given. If this delay is identified with delay after source initiation, then implicitly a zero-phase source-signature deconvolution is assumed to have been applied to the trace.

We have implemented these requirements using utilities from the Seismic Unix ("SU") package [11], based on the SEGY standard [1]. Necessary attributes correspond to the SU header keys cdp, sx, sy, gx, gy, selev, gelev, and delrt. We have used an object oriented approach to implementation, so any other data structure assigning the appropriate attributes to traces could be substituted for SEGY, at the price of writing appropriate wrapper code.

### 3.3.2 Velocity Representation

NMO-based DSVA assumes a fixed nodal structure for velocity representation. This placement is a user decision, and is static. Research on differential semblance has suggested approaches to automatic and dynamic velocity model definition [44], but this topic remains an area for future research. We require that

- Nodal values should be bounding, i.e. if each nodal value specifying velocity model $v_1$ is greater than the corresponding nodal value specifying velocity model $v_2$, then at an arbitrary point in space $(x,y,z)$, $v_1(x,y,z) > v_2(x,y,z)$;

- RMS Velocity functions determined by the construction should be smooth, that is, without discontinuities in value or slope that would invalidate the naive geometric optics underlying the convolutional model.

The first requirement allows specification by nodal values of upper and lower envelope velocities. The second implements the bound on slope (amongst other derivatives), the other constraint imposed on admissible velocities, implicitly by the upper and lower envelopes.
We have used two velocity data structures conforming to these strictures. The simpler of the two is based on piecewise linear interpolation of nodal values, for which our implementation requires a regular grid of nodes. The second \textit{Partially Irregular Grid} ("PI Grid") data structure accommodates a less structured node geometry [13]. The PI Grid data structure permits nodes to be placed arbitrarily on arbitrary vertical lines ("wells"), the wells placed arbitrarily within vertical planes ("sections"), and the (parallel) sections placed arbitrarily in space. This node structure is illustrated in Figure 3.1. Both piecewise linear and PI Grid data structures actually define \textit{functions} of the space variables, and can be sampled on an arbitrary grid (in fact, at an arbitrary point). For use in optimization, adjoints of the sampling functions are required. Archival storage uses the SEP77 file format [9] for piecewise linear functions, and a simple ASCII disk data structure for PI Grid.

\subsection*{3.3.3 NMO Implementation}

The implementation of NMO as a map between regular grids (in $t$ and $t_0$ or $z$) inevitably requires interpolation, which is not differentiable due to conditional branches and comparisons which occur in natural interpolation code. We have adopted a local cubic interpolation model which minimizes this difficulty; Mulder and ten Kroode [35] elected a similar approach to this problem, which also occurs in DSVA based on Kirchhoff migration. We used a binning scheme to assign traces to CMPs: CMP $x_m$ and $y_m$ coordinates, averaged over CMP bin, are used to assign traces to velocity bins, and the velocity profile at the center of each velocity bin is used to move out all traces assigned to that velocity bin.

With these conventions, the computation of $J[v, d]$ can be implemented for the price of one pass through the data. The formula for $\nabla_v J[v, d]$ given in the Theory section shows that the gradient can be had for one additional pass, applying NMO to the data time derivative and accumulating the result. Instead, our implementation processes each trace only once, computing the NMO corrections of both the trace and
of its time derivative and combining these to accumulate the value and gradient of
the objective in one pass through the data.

Controlling NMO stretch requires that upper and lower bounds on velocity be
maintained, by limiting the search to an admissible set of velocities as described
above. We permitted the same number of degrees of freedom (i.e. the same regular
or PIGrid) for upper and lower envelope velocities defining the admissible set. These
envelope velocities are user specified but defaulted to \( \pm 10\% \) of the initial velocity
estimate during optimization.

\subsection{3.3.4 Numerical Optimization}

We employ the Limited Memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton
algorithm (“LBFGS”), generally considered the most effective modern unconstrained
optimization algorithm under the widest variety of circumstances [36]. We have used
the Rice Vector Library (“RVL”) simulation-driven optimization framework to im­
plement this algorithm [46]. This implementation uses a line search subalgorithm to
guarantee a reduction in the objective. The line search monitors the upper and lower
velocity envelopes, by means of a distance-to-boundary function which is a generic
attribute of the function interface in RVL.

\subsection{3.3.5 Computational Cost}

The algorithm is coded in C/C++ with RVL-defined interfaces. We used the gcc 3.4
compiler suite. A typical optimization loop of 10 LBFGS iterations on 1000-sample
traces runs at about 100 traces / s on a 1 GHz Mac PowerBook. Our implementation
pages traces from disk sequentially, and sweeps through all traces in the gather for
each LBFGS iteration.
3.4 Examples

We provide two examples illustrating the performance of NMO-based DSVA. Both are 2D marine, from the North Sea.

3.4.1 Example 1

This line overlies a nearly layered subsurface to 2 s two-way time. Figure 3.2 shows the 21 CMPs after bandpass filter and mute. Pegleg multiples from the water bottom are evident, as is clear from Figure 3.3 top, which displays a CMP from this line. To suppress the peglegs, we applied predictive deconvolution, as implemented in the SU command `supef` with appropriate parameters. This step results largely suppresses the water column reverberations, as can be seen in Figure 3.3 bottom.

Suspecting from the near-offset section that the shallow structure underlying this line is laterally homogeneous, we elected to search for a 1D velocity model. The 21 CMPs, both before and after predictive deconvolution, were then input to NMO-based DSVA. The initial velocity estimate was constant = 1.5 km/s. Approximately 10 iterations of LBFGS produced a reduction of two orders of magnitude in the objective gradient, which was our stopping criterion.

The resulting NMO-corrected CMPs without predictive deconvolution are displayed in Figure 3.4, with predictive deconvolution in Figure 3.5. Note that in Figure 3.4, the apparent primary at about 1.5 s exhibits some upwards curvature. This overcorrection is absent in Figure 3.5. Another way to view the constructive effect of multiple suppression is via comparison with the velocity spectra [50]. Figures 3.6 and 3.7 show the RMS velocity (i.e. \( u^{-1/2} \)) computed from the interval velocity estimated by differential semblance, overplotted on velocity spectra. The RMS velocity computed from the raw data (Figure 3.6) navigates between the faster (primary) and slower (multiple) spectral peaks, illustrating the tendency of DSVA to compromise between the apparent velocities of conflicting events ([21]). In Figure 3.7, predictive
deconvolution has suppressed the slow peaks and DSVA picks the primaries.

### 3.4.2 Example 2

This example is taken from data released by Mobil in support of a workshop on seismic inversion at the 1994 SEG Annual Meeting [27]. This data exhibits a more complex pattern of multiple reflections, with much stronger water column related multiples, than does Example 1. We began with data to which hyperbolic Radon transform multiple suppression had been applied, then bandpass filtered and muted it. Considerable multiple energy is still evident, but a primary reflection series is made visible by this process. We chose 51 CMPs covering the 10 km of line, displayed in Figure 3.8, and input these to NMO-based DSVA. Approximately 20 steps of LBFGS gave convergence from a constant initial velocity (1.5 km/s), convergence once again being defined as reduction of two orders of magnitude in the gradient.

Since the velocity structure is known to be laterally heterogeneous, we used this example to compare the effects of 1D vs. 2D velocity estimation on the prediction of moveout via DSVA. The 1D velocity model was specified by four velocity nodes spaced equally from the surface to 3 km depth. The 2D model used three equally spaced wells positioned 4 km apart along the line, again each with four vertical nodes. NMO correction at the DSVA optimized 1D and 2D velocities produced the data displayed in Figures 3.9 and 3.11 respectively. Figures 3.10 and 3.12 show detail between 2.2 and 3.0 s near the right end of the line for 1D and 2D velocity estimation respectively, corresponding to the regions circled in red on Figures 3.9 and 3.11. The additional degrees of freedom in the 2D velocity structure permit a more satisfactory account of moveout, as one sees most clearly in the event at 2.5 s.

These detail plots also clearly display the residual multiple energy which limits the accuracy achievable with straightforward application of DSVA. Gockenbach and Symes [21] treated a single CMP from this data set by an extremal regularization technique, which permits DSVA to ignore all but the dominant family of moveouts.
and achieved much more accurate identification of apparent primary velocity.

It should be noted that the lack of dominant primary energy apparent in the left
hand lower part of Figure 3.8 leads to an artificially low estimate in the corresponding
region of the 2D velocity. This pathology could be remedied by constraining the
estimate in this part of the model \textit{a priori}, or (of course) by better elimination of
coherent noise.

\subsection*{3.5 Conclusions}

We have described an NMO-based implementation of DSVA and shown some 2D
examples of its use. This implementation accomodates arbitrary field geometry and
estimates 1D, 2D, or 3D velocity models, and is to our knowledge the first imple­
mentation with these features. While less flexible than DSVA based on migration,
NMO-based DSVA gives reasonable approximate interval velocities from data that fall
within its domain of applicability, at low computational cost. The (well understood)
consequences of inadequate multiple suppression are evident in both examples pre­
sented here. The results underline the importance of further research to incorporate
more physics, notably multiple reflections, into the theory and practice of automatic
velocity estimation.
Figure 3.1: Illustration of the Partially Irregular Grid ("PIGrid") data structure defined in the text. Velocity is given at arbitrarily spaced nodes in arbitrarily spaced "wells" (vertical lines) grouped in arbitrarily spaced "sections" (vertical planes).
Figure 3.2: Example 1: 21 CMP gathers from first North Sea line. Preprocessing consists of zero-phase bandpass filter and mute of direct and head waves.
Figure 3.3: Example 1: A CMP before (above) and after (below) predictive deconvolution.
Figure 3.4: Example 1: NMO-Corrected CMP gathers before predictive deconvolution. 1D velocity model estimated by differential semblance velocity analysis. Note upward curvature of apparent primary at about 1.5 s. The presence of slower events (multiple reflections, presumably) has affected the velocity analysis.
Figure 3.5: Example 1: NMO-Corrected CMP gathers after predictive deconvolution. 1D velocity model estimated by differential semblance velocity analysis. 1.5 s event is flattened. Apparent multiples (surface and/or pegleg) following it are suppressed by predictive deconvolution, and no longer dominate the velocity analysis.
Figure 3.6: Example 1: Comparison of RMS velocity with velocity scan, using data of Figure 3.2 before predictive deconvolution. Solid line = RMS velocity estimated by differential semblance velocity analysis; Grey-scale = velocity spectrum. Differential semblance RMS velocity appears to seek compromise between primary and multiple moveout velocities.
Figure 3.7: Example 1: Comparison of RMS velocity with velocity scan, using data of Figure 3.2 after predictive deconvolution. Solid line = RMS velocity estimated by differential semblance velocity analysis; Grey-scale = velocity spectrum. Suppression of coherent noise leads to more accurate velocities.
Figure 3.8: Example 2: 51 CMPs extracted from the second (Viking Graben) North Sea line. Preprocessing consisted of multiple suppression via hyperbolic Radon Transform, zero-phase bandpass filter, and mute.
Figure 3.9: Example 2: NMO corrected gathers using 1D velocity model estimated by differential semblance.

Figure 3.10: Detail from Figure 3.9.
Figure 3.11: Example 2: NMO corrected gathers using 2D velocity model estimated by differential semblance.

Figure 3.12: Detail from Figure 3.11.
Figure 3.13: DSVA $v(z)$ Interval velocities

Figure 3.14: DSVA $v(x, z)$ Interval velocities
Chapter 4

Kirchhoff-based Differential Semblance

This chapter will give a detailed mathematical derivation of Kirchhoff-based differential semblance and the workflow of how it’s implemented.

4.1 Formulation

The basic Kirchhoff imaging formula is:

\[ I(x, h) = \sum_{(r,s) \in B(h)} A(x_r, x_s, x) d(r, s, T(x_r, x) + T(x_s, x)) \]  (4.1)

Here \( B(h) = \{ (r, s) : h \leq |x_r - x_s| \leq h + \Delta h \} \) is the bin of a collection of source-receiver coordinate pairs in the offset bin \([h, h + \Delta h])\). \( A(x_r, x_s, x) \) is the amplitude, which may include factors which make \( I(x, h) \) an approximate inversion. \( T \) and \( A \) are functions of velocity. In this chapter, we will ignore the dependence of \( A \) on velocity, regarding it as fixed. Inclusion of \( A \) dependence is a lower order correction, equivalent in importance to terms already neglected in the derivation of the Kirchhoff imaging formula. Therefore this dependence may legitimately be neglected. In fact, we will drop the amplitude factor from the following formulae, on the understanding that it
may be reintroduced as a velocity-dependant factor.

The perturbation in image volume due to perturbation in velocity, computed without taking into account any perturbation in amplitude (principal part, high-frequency leading term):

\[
\delta I(x, h) = \sum_{(r,s) \in B(h)} A(x_r, x_s, x) \frac{\partial d}{\partial t}(r, s, T(x_r, x) + T(x_s, x))(\delta T(x_r, x) + \delta T(x_s, x))
\]

(4.2)

Recall the differential semblance objective formula:

\[
J[v, d] = \frac{1}{2} \sum_x \sum_h |D_h I(x, h)|^2
\]

(4.3)
in which \(D_h f(h) = (f(h + \Delta h) - f(h))/\Delta h\) is the forward h-difference operator.

Thus the perturbation in differential semblance is:

\[
\delta J[v, d] = \sum_x \sum_h D_h I(x, h) D_h \delta I(x, h)
\]

(4.4)

\[
= \sum_x \sum_h [D_h^T D_h I(x, h)] \delta I(x, h)
\]

\[
= \sum_x \sum_{(r,s) \in B(h)} [D_h^T D_h I(x, h)] \sum_{(r,s) \in B(h)} \frac{\partial d}{\partial t}(r, s, T(x_r, x) + T(x_s, x))(\delta T(x_r, x) + \delta T(x_s, x))
\]

Introduce the linearized traveltime operator \(DT \delta v = \delta T\) and its adjoint or transpose \(DT^T\), defined by

\[
\sum_x \phi(x)(DT \delta v)(x) = \sum_x (DT^T \phi(x))\delta v(x)
\]

Then (4.4) can be re-written as

\[
\delta J[v, d] = \sum_x \left[ \sum_h [D_h^T D_h I(\cdot, h)] \sum_{(r,s) \in B(h)} (DT(x_r, \cdot) + DT(x_s, \cdot)) \frac{\partial d}{\partial t}(r, s, T(x_r, \cdot) + T(x_s, \cdot)) \right] (x)\delta v(x)
\]

(4.5)

Recall the definition of the gradient \(\nabla J\):

\[
\delta J[v, d] = \sum_x \nabla J[v, d](x)\delta v(x)
\]
In view of this definition and equation (4.5), the gradient is:

\[
\nabla J[v, d](x)
\]

\[
= \sum_h \sum_{(r,s) \in B(h)} (DT(x_r, \cdot)^T + DT(x_s, \cdot)^T) [D_h^T D_h I(\cdot, h)] \frac{\partial d}{\partial t} (r, s, T(x_r, \cdot) + T(x_s, \cdot))(x)
\]

4.1.1 Computation of \(d(t)\)

Considering the seismogram for each recorded trace, called \(d(t)\), since \(t\) is extremely unlikely to be a sample time for the recorded trace, thus an interpolation method is needed to compute the waveform amplitude which is not on a sample time. Here the calculation of \(d(t)\) is attained by using cubic Lagrange interpolation:

\[
D_2 = \frac{1}{2} \Delta t^{-3}
\]
\[
D_6 = \frac{1}{6} \Delta t^{-3}
\]
\[
k = \left[ \frac{t - t_0}{\Delta t} \right] - 1
\]
\[
l_{-1} = t - k \Delta t
\]
\[
l_0 = l_{-1} - \Delta t
\]
\[
l_1 = l_0 - \Delta t
\]
\[
l_2 = l_1 - \Delta t
\]
\[
p_0 = D_6 l_0 l_1
\]
\[
p_1 = D_2 l_{-1} l_2
\]

\[
d(t) \simeq -p_0 l_2 d_k + p_1 l_1 d_{k+1} - p_1 l_0 d_{k+2} + p_0 l_{-1} d_{k+3}
\]

here \(d_k, d_{k+1}, d_{k+2}, d_{k+3}\) represent the amplitudes at the corresponding sample time on the recorded trace.
4.1.2 Computation of $\frac{\partial d}{\partial t}(t)$

The $t$-derivative of the trace appearing in equation (4.4) should be computed by differentiating the image formula by cubic interpolation. Note that all of the quantities denoted $l_*$ in this formula have $t$-derivative $= 1$.

\[
d p_0 &= D_6(l_0 + l_1) \\
d p_1 &= D_2(l_{-1} + l_2)
\]

\[
\frac{\partial d}{\partial t}(t) \approx -(dp_0l_2 + p_0)d_k + (dp_1l_1 + p_1)d_{k+1} - (dp_1l_0 + p_1)d_{k+2} + (dp_0l_{-1} + p_0)d_{k+3} \quad (4.8)
\]

4.2 Work flow

Putting all this together gives an algorithm outline. The algorithm includes image calculation, objective calculation and the gradient calculation.

4.2.1 Image calculation

1. sort data by increasing offset, define the output image grid

2. give a starting offset and last offset to bin the data

3. give a starting gridded velocity model (SEP77 format); the velocity model can be attained by using B-spline interpolation

4. within each offset gather

   (a) for each trace

   i. precompute $d(t)$ as in the equation (4.7), accumulate the image from all the previous traces
ii. read the next trace

5. write this common offset image gather to the disk as SEP77 format

6. zero the image storage, update the current offset

7. read next offset, repeat the same step until the last offset gather is finished

4.2.2 Objective calculation

From the equation (4.3), the objective calculation based the image calculation is as follows. For each $h$:

1. read the current image gather $I_i$, and the next image gather, form difference
   
   $diff^+ = [(I_{i+1} - I_i)/\Delta h]$, $i = 1, \ldots, nh - 1$; $nh$ is the number of offsets, $\Delta h$ is the offset step

2. zero the image storage, update the current offset

3. read next offset, repeat the same step until the last offset

The accumulated $diff$ is the final objective value.

4.2.3 Gradient calculation

According to the equation (4.6), $D_h^T D_h I$, 2nd derivative of the common offset image with respect to the offset, needs to be calculated to update the velocity variation in the objective. This requires at least two consequent common offset images depending if starting from the very first offset or finishing up the very last one. Thus the gradient calculation can only be calculated after all the common offset image gathers are stored. For each offset gather:

1. get the gridded image $I_i$ for the current offset. If this is the first offset, then
   read first two common offset image gathers, form the difference: $\frac{I_i - I_{i+1}}{\Delta h}$, $i = 1$; if
this is the last offset, then form difference: $\frac{I_i - I_{i-1}}{\Delta h}$, $i = nh$ ($nh$ is the number of offsets); else, read three adjacent common offset image gathers, form difference: $\frac{2I_i - I_{i-1} - I_{i+1}}{\Delta h^2}$, $i = 2, nh - 1$; the result is $D_h^T D_h I$. Within the same offset gather, the calculation of the other terms in the gradient formula also requires re-reading the data similar to the image calculation step.

(a) For each trace:

- compute $\frac{\partial t}{\partial x}(t)$ as in the equation (4.8)
- compute the adjoint operator $DT(x_r, x)^T + DT(x_s, x)^T$ which involves source and receiver for each trace respectively
- do the multiplication of the results above (a), (b) with $D_h^T D_h I$. Note: all these three results have the same size as $I_i$, accumulate this result for each trace

(b) write the gradient result for this offset gather out to the disk

2. zero the perturbation gridded image storage, the gradient storage, update the offset

3. repeat the same step until the last offset gather is finished

These three steps are implemented sequentially. The gradient calculation contributes to the velocity update in the iteration, while the objective value gets to the minimum when the correct estimated velocity is found. However, this flow only provides the necessary steps for one estimated velocity model, which does not give an automated update for a series of trial velocities as an iterative process. This will be included in the future research.

The following figure may give a better illustration of how this works.
I define an image grid

give an initial velocity model

loop over each gather

loop each trace
read each trace, compute traveltimes, amplitude image and derivative of the image
write the image gather to the disk
read the previous and current offset gather, compute objective and keep summing it up
read the previous two offset gathers and the current one, compute the gradient of objective, keep summing it up
zero the image storage, update the current offset

Data input
(sorted into common offset gathers)

Figure 4.1: Work flow of the calculation of DS objective and the gradient of the objective
Chapter 5

Results

This chapter will discuss the Kirchhoff-based differential semblance objective on the synthetic data.

5.1 Synthetic data generation

The data used was generated by using 2D synthetic Seismic Unix script susyn1v with a constant background velocity model \( v(x, z) = 1500m/s \). The source wavelet is the Ricker wavelet with a peak frequency of 25hz. This dataset is composed of 61 shots with a shot spacing of 20 meters, and each shot has 100 offsets. The first offset starts at 100 meters, with an incremental offset step of 20 meters. Figure (5.1) gives an underground structure with four layers.

5.2 CMP stack and Kirchhoff migration comparison

The prestack migration provides a higher quality image as compared with CMP stack section. However, CMP stacked sections are always a good first approximation of the image. The following will give a comparison between the CMP stack and the Kirchhoff
Migration image from the same data set.

The steps are as follows:

• generate synthetic data set from the given geometry
  – sort the data set into common offset gather → kirchhoff migration → image
  – sort the data set into common mid point gather → NMO correction → stack → stack section

• compare the two outputs

From this comparison, it’s evident that Kirchhoff migration can provide a more accurate image with the right subsurface location than CMP stack section, which can somehow illustrate a roughly correct underground structure. This shows that migration is necessary for the dipping structure.
5.3 Traveltime implementation discussion

Here I give a comparison of the images by using two different traveltime algorithms with this constant velocity model.

- first algorithm: simply computes the twoway traveltime directly: \(2 \times \text{distance} / \text{velocity}\).

- second algorithm: calculates traveltime using the eikonal solver by Qian ([31]). This algorithm requires several parameters to be set appropriately to get the solution with high accuracy, such as grid size, tolerance value and iteration number.

Examples

Here, I list two examples using two different sets of parameters. mesh size: 5m and 10m. Note I used the same mesh size in the traveltime calculation and in the
These examples have shown that the traveltime calculated from Eikonal solver in this case is quite accurate; as to the efficiency, the speed of traveltime calculation is as fast as using the analytical formula by using the constant velocity model. The parameters of the image grid can somehow affect the image resolution, but the outcome is still reliable since it can give accurate locations of the reflectors.

### 5.4 Objective function discussion

According to the DS formula, for each common image gather, the summation over the differences of the flat events will be minimum at the correct velocity model. This is the criterion for the velocity iteration in the gradient calculation for velocity updating. As discussed in the previous section, the accuracy of traveltime calculation may have effect on the DS objective as well. Two experiments using analytic formula and eikonal solver respectively for traveltime calculation were performed. Figure (5.9) illustrated
the difference of these two set of objective values by using a trial of velocities starting from 1.1km/s to 2.0km/s.

It has been predicted that there would be difference between the objective values by using two different types of traveltime calculation. The difference is trivial, which proves that using eikonal solver for traveltime calculation may cause some error, but this error is within the tolerance and acceptable.

Figure (5.9) exhibits two common image gathers at the correct velocity (1.5km/s) based on different types of traveltime calculation. Note the image gather on the right will have a slightly upward curvature with larger offsets. The events are more flat on the left than those on the right. However, the flatness of the gather (right) is still sufficient for judging the correct velocity because the DS objective is minimum at the correct velocity model.
Figure 5.5: Stack of NMO-corrected common midpoint gathers

Figure 5.6: Stack of Kirchhoff migrated common midpoint gathers. Grid Size=5m (left), Grid Size=10m (right)
Figure 5.7: Stack of Kirchhoff migrated common midpoint gathers. Image from 2 point ray tracing with constant background velocity model (left) by using the eikonal solver. Grid Size=20m (right)

Figure 5.8: DS objectives for velocity ranging from 1.1km/s to 2.0km/s. The correct average velocity is 1.5km/s.
Figure 5.9: The common image gather at the mid point location 1000m, traveltine calculated using analytic formula (left) and eikonal solver(right)
Chapter 6

Conclusion

Kirchhoff migration is a useful and efficient imaging tool and provides a good platform for velocity analysis. The cost efficient of Kirchhoff migration is unneglectable factor providing a big advantage of the velocity analysis. However, due to the artifacts caused by the multi-pathing in the Kirchhoff migration, Kirchhoff-based DSVA is limited to regimes where multi-pathing is unimportant, for example, shallow sediments. Unfortunately, Kirchhoff migration can hardly do a very good job for the deep water or salt structure, which are very popular “targets” in the oil industry nowadays, and thus this can somehow limit the application of Kirchoff-based DSVA in these areas. Within this limitation, Kirchhoff-based DSVA still has potential use in finding the correct velocity model for industry use and is expected to provide some enlightenment for the imaging methods. This thesis provided the preliminary evidence that DSVA objective with the eikonal solver traveltimes is effective and usable to judge the correct velocity model for Kirchhoff migration velocity analysis. The thesis gives the first derivation of an effective algorithm for a grid-base (as opposed to ray-based) computation of gradient. This computation of gradient can be employed in the future research for automatically updating the velocities.
Bibliography


