THREE-DIMENSIONAL FILTERING:
Environmental soundings image enhancement

Jon F. Clærbout
Cecil and Ida Green Professor of Geophysics
Stanford University

Book on Web at http://sepwww.stanford.edu/sep/prof
email: jon@sep.stanford.edu

Contents

FREEWARE, COPYRIGHT, AND PUBLIC LICENSE

This disk contains freeware from many authors. Freeware is software you can copy and give away. But it is restricted in other ways. Please see author's copyrights and "public licenses" along with their programs. I do not certify that *all* of the software on this disk is freely copyable (although I believe it to be). Thus I accept no responsibility for your act of copying this disk.

I reserve the right to distribute this book and its software for a profit. Paraphrasing The GNU General Public License, you may make modifications and copies and distribute them also, but your distribution which includes my work must be distributed free.

Jon Claerbout
April 26, 1996

Electronic introduction

Most of the figures in this book contain pushbuttons in the figure captions. Each pushbutton points to the computer code that built the figure. You not only see nice illustrations, you see, in as much detail as you are willing to pursue, how the illustrations are made. You can dig in, change data, change parameters, or even change the theory. You then press the button in any figure caption and watch the figure rebuild itself (on UNIX based computers) to your specifications.

I prepared this book for novices in geophysical data analysis, and I prepared this book for scientists and engineers who wish to preserve their results in a publishable format where readers can verify the results. Some people see reproducibility of research as a troublesome ethical issue. I see it as an everpresent, irritating practical concern. We are annoyed when we cannot easily reproduce our own work. Some of us are irritated when other professors publish grandiloquent papers whose conclusions require too much time to confirm. And we are further frustrated when new students cannot reproduce (in a year) the work of other recently graduated students. Although each student prepares a doctoral dissertation that should enable others to reproduce that work, the reality is that paper documents are outdated; they no longer fulfill the role they did a generation ago when data consisted of photographs and pencil marks on a piece of paper—when theory was several pages of Greek symbols.

Paper no longer serves us well. Our data now resides in a computer, and our theory is embodied in computer programs. Many libraries overflow and they cannot afford the materials they want. Textbooks and monographs often cost over $100; and there is the larger issue of wanting to save trees. An electronic document like this one is a "living document" that grows as people work with it. Our experience shows that it is only slightly more difficult to give birth to a "living" document than a "dead" one.

The major hurdles in preparing a doctoral dissertation or research monograph or textbook are these: (1) mastering the subject matter itself, (2) writing the ancillary technical programs, (3) learning to use a text editor and word processing system, and (4) writing the command scripts to make the illustrations. The difference between preparing a "live" document and a "dead" one, lies in point four. Will the command scripts run once and then forgotten, or will they be attached to the figure-caption pushbutton? Authors who wish to communicate will produce live documents. Authors who merely wish to advertise their scholarship without really sharing it will continue to produce dead ones. The increase in effort for the authors is minuscule—

once they learn how to file their work properly.[1] When you look beneath the surface of this book you will find everything you need to make your own living documents. And you can freely copy it from our web site or CD-ROM. At the beginning of the computer revolution, I gave young students the advice: "Learn how to type." With the mass-storage revolution now underway, and with a $10 CD-ROM able to hold all the keystrokes you can type in a lifetime, I now advise everyone: "Learn how to file."

Finally, I am particularly indebted to my colleague, Matthias Schwab, whose continuing collaboration contributed greatly to my quality control, to convenient distribution on CD-ROM, and on the world wide web.

---

[1] http://sepwww.stanford.edu/redoc/

Introduction

This book is a guide to practical application of inverse theory to large sized data sets. By large, I mean geophysical data ranging over two or more dimensions and the data space and model space sizes being larger than about 10^5 elements, about a 300×300 image. (Eigenvector methods and software such as Mathematica and Matlab are useful up until vectors of about 10^3 elements, about a 30×30 image.)

On my '93-'94 sabbatical leave I visited a variety of geophysical institutions and brought back with me several interesting data sets. I came to realize that interpolating data onto a regular mesh is itself a problem from inverse theory. For several reasons, this inverse interpolation problem turns out to be an ideal problem for an introductory book. The application of inverse interpolation to the recently acquired data exhibits in simplest and clearest manner, starkly and nakedly that: (1) Application of inverse theory is fraught with pitfalls. (2) Real data has nasty noises that demolish textbook theories. Three of four data sets worked on in my fall class had such nasty noises, and the fourth data set with no noise turned out to have missing data where some earlier worker had previously identified the bad points. (3) Application of inverse theory in large problems (data space and model space sizes are larger than about 10^5 elements) is plagued by slow convergence of iterative methods.

Data cubes

This monograph, "Applications of three-dimensional filtering" is about data cubes (Cartesian lattices of values) and how we improve our visualization and understanding of them by filtering and related operations.

We get data cubes by many methods. A signal is a one-dimensional array; an image or a frame is a two-dimensional array; and a cube or a movie is a three-dimensional array. To view a volume as a movie we slice off a sequence of frames in any of the three orthogonal directions.

A movie cube could be made from a signal such as a television signal. More typically we can directly record a cube of values in an echo sounding experiment, typically geophysical, but perhaps a medical sounding of the human body. One dimension is typically time or travel time while the other two are typically space, either (x, y) Cartesian coordinates or locations of sender s and receiver g along a survey line. Sometimes data is irregularly positioned and to prepare it for further analysis we redefine it on a uniform mesh.

Filtering and information theory are well developed for one-dimensional signals but less so for images and movie cubes. Here we skim the cream of practical methods in signal theory and carry forward the useful ones into two and three dimensions where new concepts emerge while many academic issues are left behind.

One of the more interesting things we do with data cubes is fill in the missing channels or missing frames the experiment did not reach. Such estimated data is not as good as real measurements, but when it comes to visualizing or to further analysis, it is much better than a region of zero values. Missing data could be anything but often it is predictable using principles of symmetry, translation invariance (stationarity), and scale invariance (self similarity).

Missing data makes a mess of Fourier transforms, but can easily be managed with convolutions. In some applications **Fourier transform**ation is justified on some axis. Then for the small cost of introducing complex numbers, the cube is reduced to a stack of independent planes. Here we omit Fourier transforms because (1) They are handled adequately elsewhere, and (2) when working on the space axes, experimentalists often lack the range and regularity needed to make Fourier transformation a helpful technique.

What else do we do with data cubes? The most useful thing we could do is reduce them to frames because frames can be printed on a piece of paper. Where there is too much information to reduce a volume to a plane, we try to eliminate noise so the volume can be subsampled. We try to identify striations and textures. We enhance or suppress the striations according to the application. The world is filled with waves that locally behave like plane waves. With this fact, we develop models for signal and noise and then we decompose data cubes accordingly.

For exploration geophysicists

The difference between theory and practice is larger in practice than it is in theory.[2] This is a book of techniques to bridge the gap between theory and practice. No sooner had I invented finite-difference wave-equation migration than we all became concerned with "smiles", those unrealistic images that result from "bad data quality". My finite-difference migration turned out to have clear theoretical limitations, but it was popular anyway because accidentally it is more immune to smiles than other methods. Even today, deep seismic soundings data is generally not migrated because the smiles are so overwhelming. Later, to try to overcome lack of theory for downward continuing CDP stacks I started slant stacking field gathers. I got poor results, again attributed to "data quality" though in retrospect we see that early results were poor because of operator aliasing and we now know that simple spatial interpolation of the data before slant stacking helps a lot. Then I began using wave-equation methods for making synthetic multiple reflections in hopes of predicting and removing strong

---

[2] I recall this witty remark without recalling the author's name.

multiples. I soon found that seismic energy on reflection data tends to move radially away from the source, and my prediction techniques stalled because data not recorded near zero offset is required to predict later multiple reflections. Likewise, wide offset traces always seem to exhibit signals near the aliasing frequency, frustrating many techniques. I have forgotten how many times I addressed the issue of shot-waveform estimation and the limited bandwidth of reliable data. With this history of personal experiences, I developed a deep interest in questions of "data quality" which I learned much about in the past few years, and which this book covers.

The meaning of "data quality" almost seems to come down to the question, "Can you interpolate the data?". To the neophyte, the subject of "interpolation" looks horribly dull compared to "geophysical inverse theory" (finding models from data). In reality the two are closely linked because if you can build a good model for the known data, then you should be able to find the missing data.

The goals of this book are sharper images, better velocities, fewer multiple reflections, and better calibration of amplitude versus angle. The method is to link questions of data quality with inverse theory, and do so with reproducible examples.

Preview for inverse theorists

Given a matrix relation $\mathbf{d} = \mathbf{Bm}$ between model $\mathbf{m}$ and data $\mathbf{d}$, common sense suggests practitioners should find $\mathbf{m}$ to minimize the length $||\mathbf{r}||$ of the residual $\mathbf{r} = \mathbf{d} - \mathbf{Bm}$. A theory of **Gauss** suggests that a better (minimum variance, unbiased) estimate results from minimizing the quadratic form $\mathbf{r}'\sigma_{rr}^{-1}\mathbf{r}$ where σ_{rr} is the noise **covariance matrix**. I have never seen an application where the noise **covariance matrix** was given, but practitioners often find ways to estimate it by regarding various sums as ensemble averages.

Additional features of inverse theory are exhibited by the partitioned matrix

$$\mathbf{d} \quad = \quad \begin{bmatrix} \mathbf{d}_{\text{incons}} \\ \mathbf{d}_{\text{consis}} \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{m}_{\text{fit}} \\ \mathbf{m}_{\text{null}} \end{bmatrix} \quad = \quad \mathbf{Bm} \qquad (0.1)$$

which shows that a portion $\mathbf{d}_{\text{incons}}$ of the data should vanish for any model $\mathbf{m}$, so an observed nonvanishing $\mathbf{d}_{\text{incons}}$ is inconsistent with *any* theoretical model $\mathbf{m}$. Likewise the $\mathbf{m}_{\text{null}}$ part of the model space makes no contribution to the data space, so it *seems* not knowable from the data.

To choose the **null space**, simple inverse theory suggests practitioners should also minimize $||\mathbf{m}||$ which amounts to setting the null space to zero. Baysian inverse theory says practitioners should use the model **covariance matrix** σ_{mm} and minimize $\mathbf{m}'\sigma_{mm}^{-1}\mathbf{m}$ for a better answer although it would include some nonzero portion of the null space. Never have I seen an application where the model-covariance matrix was a given prior. Specifying or estimating it is a puzzle for experimentalists. For example, with a model space $\mathbf{m}$ being a signal (having components that are a function of time) or being a stratified earth model (components a function of depth z) a practitioner might supplement the regression $r(z) \approx 0$ by a regression like $dm(z)/dz \approx$

0 which means supplementing the noise quadratic form $\mathbf{r}'\sigma_{rr}^{-1}\mathbf{r}$ by a model quadratic form $\mathbf{m}'\mathbf{D}'\mathbf{D}\mathbf{m}$ where $\mathbf{D}$ is a finite-difference representation of the first z derivative. This implies that the model **covariance matrix** is something like $\sigma_{mm} \approx (\mathbf{D}'\mathbf{D})^{-1}$. Neither the model **covariance matrix** nor the null space $\mathbf{m}_{\rm null}$ *seems* learnable from the data and equation (0.1).

In fact, both the **null space** and the model **covariance matrix** can be estimated from the data and that is one of the novalties of this book. To convince you it is possible (without launching into the main body of the book) I offer a simple example of an operator and data set from which your human intuition will immediately tell you what you want for the whole model space including the null space:

Take the data to be a sinusoidal function of time (or depth) and take $\mathbf{A} = \mathbf{I}$ so the operator $\mathbf{B}$ is a delay operator with truncation of the signal shifted off the end of the space. Solving for $\mathbf{m}_{\rm fit}$, the findable part of the model, you get a back shifted sinusoid. Your human intuition, not any mathematics here, tells you that the truncated part of the model, $\mathbf{m}_{\rm null}$, should be a logical continuation of the sinusoid $\mathbf{m}_{\rm fit}$ and the null space should not be taken zero $\mathbf{m}_{\rm null} = \mathbf{0}$, nor should it be taken to be the constant function (with $dm(t)/dt = 0$).

Prior knowledge we exploit in this book is that unknowns are functions of time and space (so their **covariance matrice**s have structure). The "prediction-error filter" (PEF) tends to the inverse spectrum. We explore the form of PEFs in higher dimensions and the further structure arising from unknowns being wavefields and images of sedimentary sections.

Reproducibility

Most of the figures in this book contain pushbuttons in the figure captions. Each pushbutton points to the computer code that built the figure. You not only see nice illustrations, you see, in as much detail as you are willing to pursue, how the illustrations are made. You can dig in, change data, change parameters, or even change the theory. You then press the button in any figure caption and watch the figure rebuild itself (on UNIX based computers) to your specifications.

I am particularly indebted to my colleague, Martin Karrenbach, whose continuing collaboration is making possible these interactive documents on CD-ROM.

Subroutines

This book uses mathematics to explain the given runnable computer codes. The code itself is in Ratfor[3] a dialect of Fortran more suitable than standard Fortran for exposition.[4] People often ask me why we do not use commercial languages like Matlab

---

[3] See Appendix A for details about Ratfor.

[4] In the summer of 1993 Dave Nichols and I and about 6 other people converted many of the operators of this book into C++.

and Mathematica. The reason is that in imaging applications the matrix operators are too large for any storage scheme. Your computer screen holds an image of about $1000 \times 1000 = 10^6$ elements. If this is an output space from an equivalent sized input space, then the matrix transfer operator is $10^6 \times 10^6 = 10^12$ elements. By the end of chapter **??** we see that the way to charactorize an operator is by a subroutine pair, not by a data structure.

There is little or no mathematics or code that is not carried through with examples with both synthetic and real data. The code shown is that used to generate the examples given. Additional code not shown in the paper version of the book handles data input, output, graphics and interaction.

Computer code in the paper book uses nomenclature resembling the accompanying mathematics so the two illuminate each other. The codes are in a consistent style being documented by the text in the book and the examples. Some codes have been heavily tested while others have only been tested by the preparation of the given illustrations.

Adjoint (conjugate-transpose) operators

The subroutines here that incarnate linear operators are written in a coding style that also incarnates the adjoint (conjugate-transpose) operator thus enabling both modeling and data processing with the same code. Besides being concise and avoiding redundancy, this style of coding ensures the consistency required for parameter fitting by conjugate-gradient optimization as described here and in my 1992 textbook, “Earth Soundings Analysis, Processing versus Inversion” (PVI).