Chapter 3

Seismic Image Analysis Using Local Spectra

Problem presented by: Pierre Lemire & Rob Pinnegar (Calgary Scientific Inc.)

Mentors: Elena Braverman (University of Calgary), Michael Lamoureux (University of Calgary), Qiao Sun (University of Calgary)

Student Participants: John Gonzalez (Northeastern University), Hui Huang (University of British Columbia), Parisa Jamali (University of Western Ontario), Yongwang Ma (University of Calgary), Hatesh Radia (University of Massachusetts at Lowell), Jihong Ren (University of British Columbia), Dallas Thomas (University of Lethbridge), Pengpeng Wang (Simon Fraser University)

Report prepared by: M. Lamoureux (`mikel@math.ucalgary.ca`)

3.1 Introduction

During the one week Industrial Problem Solving Workshop held in June 2005 at the University of Calgary, hosted by the Pacific Institute for the Mathematical Sciences, our group was asked to consider a problem in seismic imaging, as presented by researchers from Calgary Scientific Inc. The essence of the problem was to understand how the S-transform could be used to create better seismic images, that would be useful in identifying possible hydrocarbon reservoirs in the earth.

3.2 Problem Description

Our group was presented with the following summary of the problem under consideration:

> Calgary Scientific Inc. is currently developing a technique to classify pixels in seismic pseudosections based on their local spectral characteristics. These are obtained from a modified Gabor transform, in which only certain (k_x, k_y) wavevectors are represented in the local spectrum of each pixel. This classification technique involves finding the dominant peak in each local spectrum, and identifying the corresponding pixel with the wavevector and amplitude of the dominant peak.

This method proved ineffective since it is inflexible when one wants to investigate any interesting features not associated with the dominant peak. The identification of secondary peaks is complicated by the fact that dominant peaks typically cover several pixels of its local spectrum; hence, the wavevector with the second largest amplitude is likely to contain a significant contribution from the primary peak. The main goals were to efficiently identify secondary peaks and to identify correlations amongst peaks from pixel to pixel.

Ideally, a fully processed seismic pseudosection should contain all the information the interpreter needs to unambiguously identify potential drilling targets, such as reefs, anticline traps, and so forth. This involves identifying layer boundaries on the pseudosection, a process that can be considered as more or less equivalent to identifying the layers themselves as continuous groups of pixels. The presence of significant amounts of noise in the data, and the accumulation of errors during the numerical steps of seismic processing, can complicate this process by obscuring important features in the pseudosection. A final goal is to improve the signal to noise ratio to facilitate interpretation of the pseudosection.

There were a number of researchers on the project with expertise in seismic imaging, and an important first step was to understand what aspect of the imaging problem we were being asked to study. However, since we would not be working directly with raw seismic data, traditional seismic techniques would not be required. Rather, we would be working with a two dimensional image, either a migrated image, a common mid-point (CMP) stack, or a common depth point (CDP) stack. In all cases, the images display the subsurface of the earth with geological structures evident in various layers. Figure 3.1 is a typical sample image.

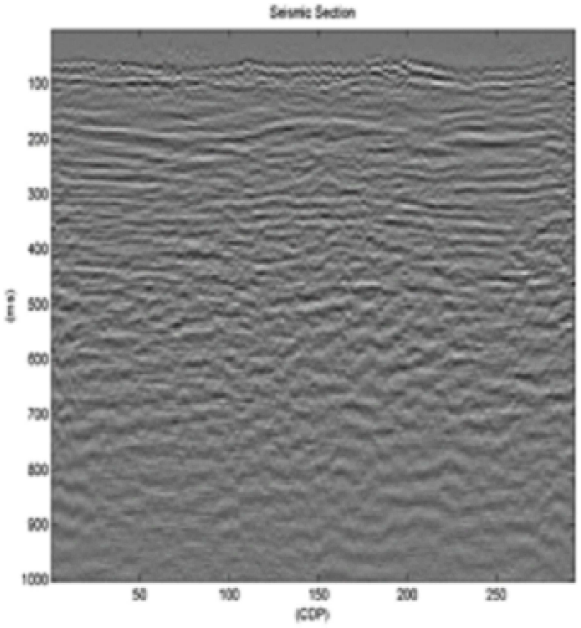

Figure 3.1: A sample seismic section.

For a given image the local spectrum is computed at each point resulting in a surface similar to Figure 3.2. The various peaks in the spectrum are used to classify each pixel in the original seismic image resulting in an enhanced and hopefully more useful seismic pseudosection.

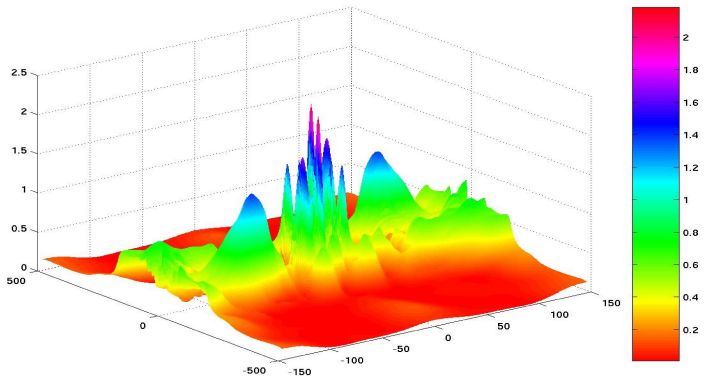

Figure 3.2: A typical local spectrum.

Thus, the objective of this project was to improve the identification of layers and other geological structures apparent in the two dimensional image (a seismic section, or CDP gather) by classifying and coloring image pixels into groups based on their local spectral attributes.

3.3 Spectral Background

The local spectra of the pixels are obtained from the Stockwell, or S-transform, which is a time-frequency spectral localization method similar to the short-time Fourier transform (STFT) [7]. The S-transform of the one-dimensional function $h(x)$ is defined as

$$S(x,k) = \int_{-\infty}^{\infty} h(x') \frac{|k|}{\sqrt{2\pi}} e^{-(x-x')^2 k^2/2} e^{-2\pi i k x'}\, dx'.$$

Because $\int_{-\infty}^{\infty} S(x,k)\, dx = H(k)$, the Fourier transform of $h(x)$, one can verify that $h(x)$ is recoverable from its S-transform. One can interpret the S-transform as a continuous wavelet transform where the mother wavelet is a Gaussian in which the wavenumber k plays the role of a dilation. This definition is easily generalized to higher dimensions and for the two-dimensional seismic images under consideration the Stockwell transform takes the form

$$S(x,y,k_x,k_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x',y') w(x-x', y-y', k_x, k_y) e^{-2\pi i x' k_x} e^{-2\pi i y' k_y}\, dx'\, dy', \tag{3.1}$$

where $h(x,y)$ is the pixel intensity of a given location and $w(x,y,k_x,k_y)$ is the corresponding two-dimensional Gaussian window. The Stockwell transform $S(x,y,k_x,k_y)$ indicates the strength of the frequencies with wavenumbers (k_x,k_y) in a neighbourhood of the point (x,y). The attributes of interest in this localized spectrum were identified to be the wavenumber (k_x,k_y) and the magnitude of any peaks in the function $S(x,y,k_x,k_y)$ as (x,y) varies across the image. These peaks are then used to classify the pixels.

The focus, then, is on identifying peaks in the local spectrum. There remained three main issues that the company wished to be addressed:

1. In Calgary Scientific's current algorithm, the four-dimensional S-transform is collapsed to a two-dimensional image by identifying each pixel with the wavevector at which its local S-spectrum is largest. That is, one particular peak is identified. This approach assumes that only one waveform is important at any pixel in the image.

2. A reliable means of automated edge detection is elusive. Although the approach described above can be used to produce a visually interpretable pseudo-image, it cannot be used to divide the image into layers because the peak wavevector can vary from pixel to pixel within a particular layer.

3. In Calgary Scientific's implementation, the S-transform is evaluated at a set of positions $\{(x, y)\}$ which changes as the wavevectors change. This complicates any pattern matching because the local S-spectra of adjacent (x, y) pixels are typically not defined on the same set of wavevectors. There are some standard techniques to account for this but they all introduce artifacts in the transformation. Is there a better way to compute the transform at all interesting positions (x, y) and wavenumbers (k_x, k_y)?

3.4 Addressing Issue One: Identifying Spectral Peaks

In the sample data sets that we considered, the spectral peaks were all remarkably smooth, and Gaussian-like. Whether this was a property of the data, or a property of the S-transform was not investigated in detail. However, it appears that it is a reflection of the redundancy in the S-transform. The smoothness, and Gaussian shape, suggested a number of approaches for identifying peaks. A summary of the ones we considered were:

- Reduce the two-dimensional peak selection problem to a series of one-dimensional problems. Using the slicing technique, we can consider all one-dimensional sections of the spectrum and analyze the peak structure of each slice. From this we identify the peaks of the entire spectrum. In the one-dimensional slice, the spectrum appears smooth enough (twice differentiable), to find and classify all the maxima.

- Use a discrete, derivative to find critical points, and identify peaks. Since the function is smooth, this technique should rapidly identify the few peaks there are. Ranking them by height is simple.

- Represent the function as a sum of Gaussians and use a least squares technique to specify their individual parameters. After that, the highest peaks are easily classified.

- Identify the highest peak and subtract an appropriate Gaussian-like function centered at this peak. After the subtraction, find the highest peak and repeat the procedure. It was believed that this could expose high peaks shadowed by adjacent higher peaks.

- Use clustering techniques to group the data around peaks, thus dividing all the data into several groups associated with various peaks. Later these groups can be ranged in accordance to the peak values.

Code for each of these attempted techniques can be found in the Appendix. The subtraction method was the most successful and is worth discussing in further detail. The location of the first peak is identified by searching for the highest data point in the matrix of spectral points. In a neighbourhood of this peak the function is approximated by a two-dimensional Gaussian with a width optimized to best fit the data. This best-fit Gaussian is then subtracted from the spectrum, and we have a data set with one peak removed. The process is then repeated (identify new peak, fit with Gaussian, subtract Gaussian) until all significant peaks are identified. Empirically, the first three or four peaks might be of interest.

Figure 3.3 gives a demonstration of how the method works. In the first panel, we see two large narrow peaks. By matching with an appropriate Gaussian, those peaks are subtracted leaving the spectrum in the middle panel. Two large, wide Gaussians are then identified as the next peaks and are again subtracted leaving the third panel. This sequence is repeated until all the major peaks have been identified. Symmetry in the spectrum allow the peaks to be removed in pairs.

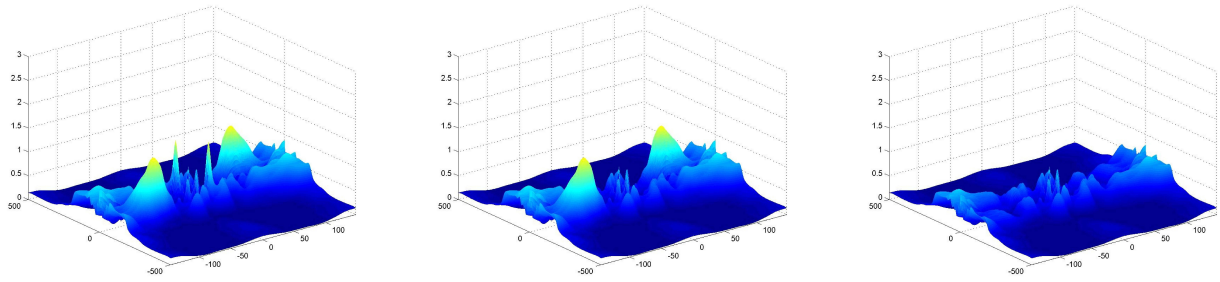

Figure 3.3: A sequence of peaks being subtracted from the data.

Also interesting is why the other methods failed. The slicing technique failed because one-dimensional slices do not uniquely identify two-dimensional peaks. However, the analysis of this proposed technique lead naturally to the two-dimensional derivative method.

The two-dimensional derivative technique identified far too many local extrema. Although most of these were down in the noise level of the data, it took a significant amount of computing time to distinguish between real peaks and extrema at the noise level of the spectrum. In the end, this method was not computationally efficient.

The least squares technique also failed because of high computational costs. On the surface this seems counterintuitive since the least squares methods is well-known for its speed. However, here we were looking to optimize over a sum of Gaussians, where the heights, widths, and locations were variables to be optimized. This is a highly non-linear problem and a computationally efficient method was not obtained in the week of the workshop.

Finally, the clustering technique also failed for computational reasons. We tried using a standard implementation of the k-mean algorithm [4], and discovered that classifying pixels for one-tenth of our image took over two hours to compute. This was judged excessive, and well beyond what we hoped for in classification.

3.5 Addressing Issue Two: Edge Detection

The idea here is that near a linear feature in the original two-dimensional image, such as along an edge, the local spectra should not vary much from pixel to pixel. Consequently, it should be possible to identify correlations between spectral peaks as one moves from pixel to pixel. To test this idea, one can begin with the peak identification algorithm in Section 3.4, and apply some statistical tests for correlations.

Unfortunately, one week was not long enough to allow us to get both the peak detection algorithm working and statistically analyse the results. A larger database of seismic images and corresponding S-transforms would be required to pursue this avenue of research.

It is worth noting that the local spectral information of the S-transform varies slowly from pixel to pixel, so there should be quite a high level of correlation. What is required is a statistical test that can distinguish correlations that come from data with those that are merely a reflection of the slowly changing nature of the S-transform.

There were also suggestions to use some alternatives of S-transform. In particular, the application of methods which identify local spectra not at each individual pixel but from a group of adjacent pixels. Some adaptive methods were proposed as follows:

- The localized Fourier method. This method also has an adaptive version which can be applied to various time grids.
- Certain wavelet-like techniques for directional analysis, such as brushlets, ridgelets, or curvelets [8]. These wavelet bases were originally developed for directional image processing and identifying slanted patterns. Such techniques seem directly applicable to these seismic images.

3.6 Addressing Issue Three: Computing the S-transform

This was probably the most challenging issue to address. A direct implementation of the S-transform is highly redundant and leads to a massive amount of data, as each data point is transformed into a full spectrum. For instance, a simple 100 by 100 image transforms into 100 million data points $(100 \times 100)^2$. The company does have some impressively fast algorithms for computing an approximation to the complete transform, but with this level of redundancy, the question must be asked: are there better approaches? We did not have access to their implementation of the algorithm, so direct investigation of how it was implemented, or how it could be improved, were not possible. Some possible ideas to pursue include:

- compute a sparse subset of the transform and apply interpolation for the intermediate points;
- compute a sparse subset of the transform and use this directly to estimate peaks;
- investigate if an analog of a fast transform (FFT, wavelet, etc.) could be applied to the S-transform. In particular, the S-transform looks very similar to a phase-shifted continuous wavelet transform, so there may be some close connections to the wavelet speedup;
- consider using a different transform, such as the Gabor or wavelet transforms, which also produce local spectral information, but have a fast implementation.

Unfortunately, we did not have time to investigate these approaches in detail.

3.7 Additional Issues: Stepping Back to Seismic

In the end, this problem is really about seismic data imaging and analysis. It is important to realize that in all the images so created, there are the important physical aspects that should be properly modelled. That is, the images come from seismic experiments involving wave propagation; errors in the image come from errors in our physical model; the geological structures we hope to identify come from physical processes such as sedimentary deposition, geological faulting, and so on. It is a bit frustrating to begin the mathematical analysis at the end of the data processing, where we have an image that came from somewhere, and we want to analyse it. With sufficient mathematical and computational tools, it makes sense to go back to the original seismic data, and apply our research techniques to understand how this raw data can be used to reveal more accurately the features we seek to identify. This is a big task, not easily accomplished in a week-long problem solving workshop. An interested reader could refer to the work of the POTSI project (`potsi.math.ucalgary.ca`) or the CREWES project (`www.crewes.org`) for further information on research in the mathematics and geophysics of seismic imaging.

3.8 Acknowledgments

We thank Peirre Lemire and Rob Pinnegar of Calgary Scientific Inc. for all their help and guidance during our weeklong project. We also thank the Pacific Institute for Mathematical Sciences for their support of the Industrial Problem Solving Workshop.

3.9 Appendix

We include here some of our key MATLAB routines.

The main script for identifying peaks, in 2D and 3D

```
% PIXEL -- A script program for picking-up peaks in a local spectrum
%          created by the S-Transforms and removing them
%
%      By Yongwang Ma
%      University of Calgary, Alberta, Canada
%      May 20, 2005
%
%      You may copy, modify and distribute the codes for purpose of
%      research.
%

% process and plot in 2D
```

```
load SH02_LocalSpec_117_177.mat;
dimension = 2;
npick = 10; % number of pairs of peaks to remove
[F,nmovie] = pick(LocalSpec,npick,dimension);

% process and plot in 3D

load SH02_LocalSpec_117_177.mat;
dimension = 3;
npick = 10; % number of pairs of peaks to remove
[F,nmovie] = pick(LocalSpec,npick,dimension);
```

The subtractional subroutine

```
function [F,nmovie] = pick(SparseLS,npick,dimension)
% pick - finds peaks and creates humps (gaussian-like function in 2D)
%   in an image (spectrum) and substracts it from the original image
%
%   By Yongwang Ma, May 18, 2005
%      The University of Calgary
%      Calgary, Alberta, Canada
%
% Input:
% SparseLS = input image, a matrix
% npick = the number of pairs of peaks needs to be found
% dimension = dimension of final images (2- 2D images; 3-3D images)
%
% Output:
% F = a structure consisting of parameters of the frames for a movie
% nmovies = a vector storing the order of the images (figures)

pix_mat = SparseLS;
NF_half = floor(size(pix_mat,1)/2);
NK_half = floor(size(pix_mat,2)/2);
f = [-NF_half:NF_half];
k = [-NK_half:NK_half];
M = size(pix_mat,1);
N = size(pix_mat,2);
sig = 1.50; % parameter controlling the width of Gaussian-like window
figure; % using only one figure, so finally only the most updated image is sho
for ipeak = 1:npick
    pmax = max(max(pix_mat));
    % find the position of maximum (peak)  in the spectrum
```

```
    for m = 1:M
        for n = 1:N
            temp = pix_mat(m,n);
            if (temp == pmax)
                f0 = m; k0 = n;
            end
        end
    end
    % record the coordinates of peaks identified
    fpeak(ipeak) = f0;
    kpeak(ipeak) = k0;
    % transform the coordinates
    if (f0 >= NF_half)
        f_0 = f0-NF_half;
    else
        f_0 = -(NF_half-f0);
    end
    if (k0 >= NK_half)
        k_0 = k0-NK_half;
    else
        k_0 = -(NK_half-k0);
    end
    f_01 = - f_0;
    k_01 = -k_0;
    gaus_p = gaus1(f_0,k_0,M,N,k,f,sig);
    gaus_p1 = gaus1(f_01,k_01,M,N,k,f,sig);
    sub_pmat = pix_mat-gaus_p.*pix_mat;
    sub_pmat1 = sub_pmat-gaus_p1.*sub_pmat;

    % Plot out results of picking
    if (dimension ==2)
        %figure;
        imagesc(sub_pmat1);
    else % (3D)
        %figure;
        surf(k,f,sub_pmat1);
        axis([k(1) k(end) f(1) f(end) 0 3])
        shading flat
    end
    if ipeak==1
        boundz=get(gca,'CLim');
    end
    set(gca,'CLim',boundz);
%     figure;imagesc(gaus_p);
```

```
%     figure;
%     amat=gaus_p.*pix_mat;
%     plot(amat(f0,:));
%     hold;
%     plot(pix_mat(f0,:));
%     plot(sub_pmat(f0,:));
    pix_mat = sub_pmat1;
    F(ipeak) = getframe;
end
% create a movie
nmovie = zeros(1,24*npick);
nrmovie = zeros(1,24*npick);
nn1 = 1;
nn2 = 24;
nnr1 = 24*npick;
nnr2 = 24*(npick-1)+1;
for in = 1:npick
    nmovie(1,nn1:nn2) = in;
    nrmovie(1,nnr1:-1:nnr2) = in;
    nn1 = nn1+24;
    nn2 = nn2 + 24;
    nnr1 = nnr1-24;
    nnr2 = nnr2-24;
end
%play the movie forwards
movie(F,nmovie);
% play the movie backwards
% movie(F,nrmovie);
```

The subroutine to compute the Gaussian-like function

```
function gaus = gaus1(f0,k0,M,N,k,f,sig)

% gaus1 = subroutine to calculate a gaussian_like function (2D)
%
% f0 = frequency, coordinate at which the centre of gaussian-like function
% located
% k0 = wavenumber, coordinate at which the centre of gaussian-like
% funcition located
% M = total number of frequency samples
% N = totlal number of wavenumber samples
% k = wavenumber coordinates, a row vector
% f = frequency coordinates, a row vector
% sig = a factor, which determines the width of the gaussian-like function
```

```
p11 = zeros(M,N);
p22 = zeros(M,N);
stab = 1.e-8; % stability factor
for m = 1:M
    p11(m,:) = (f(m)-f0).^2/M^2+(k-k0).^2/N^2;
    p22(m,:) = (f(m)/M)^2 + (k/N).^2;
end
gaus = exp(-2*pi^2*(p11./(p22+stab))/(sig^2));
```

Bibliography

[1] Aki, K. & Richards, P.G. (2002). *Quantitative Seismology* (2nd edition). University Science Book: Sausilito.

[2] Averbuch, A.Z., Meyer, F.G., Strömberg, J.O., Coifman, R.R. & Vassiliou, A. (2001). *Low Bit-Rate Efficient Compression for Seismic Data*. IEEE Transactions on Image Processing, 10(12), pp. 1801-1814.

[3] Coifman R.R. & Meyer, Y. (1991). *Remarques sur l'analyse de Fourier à fenêtre*. Comptes Rendus de l'Académie des Sciences, Série I, 312(3), pp. 259-261.

[4] MacQueen, J.B. (1967). *Some methods for classification and analysis of multivariate observations*. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol I, Statistics. Edited by Lucien M. Le Cam and Jerzy Neyman. University of California Press, pp. 281-297.

[5] Meyer F.G. & Coifman, R.R. (1997). *Brushlets: a tool for directional image analysis and image compression*. Applied and Computational Harmonic Analysis, 4, pp. 147-187.

[6] Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vettering, W.T. (1992). *Numerical Recipes in C* (2nd edition). Cambridge University Press: Cambridge.

[7] Stockwell, R.G., Mansinha, L. & Lowe, R.P. (1996). *Localization of the Complex Spectrum: The S Transform*. IEEE Transactions on Signal Processing, 44(4), pp. 998-1001.

[8] Welland, G.V. (2003). Studies in Computational Mathematics, Vol 10: *Beyond Wavelets*. Academic Press.