CWP-677
December 2010

Interpretation of 3D seismic images using an interactive image-guided paintbrush

Christopher Engelsma

— Master of Science Thesis —
Geophysics

Defended on December 1, 2010

Committee Chair:
Dr. Andrzej Szymczak

Advisor:
Dr. Dave Hale

Committee members:
Dr. Paul Sava
Dr. Terry Young

Center for Wave Phenomena
Colorado School of Mines
Golden, Colorado 80401
(1) 303-384-2178

Abstract

Seismic interpretation is an important step when developing a model of the subsurface. In past decades, this process involved interpreting 2D seismic sections on paper with colored pencils. Over time, seismic surveys evolved to three dimensions and computational power allowed for processing on workstations. Interpreters then found it easier to interpret seismic horizons, or the boundaries between geologic layers, rather than geologic formations themselves. Unfortunately, horizons are tedious to assemble and may contain holes where the image is poor. It may be more efficient and useful to interpret volumes directly through 3D painting.

3D painting attempts to expedite the interpretation process by painting volumes with a digital 3D paintbrush. Multiple seismic slices are interpreted simultaneously as features within the image control the paintbrush's shape and orientation. This paintbrush is operated by a human interpreter who controls its location and maximum size. In this way, geologic formations are interpreted by painting voxels (3D pixels) within the seismic image.

Table of Contents

Acknowledgements

I would first like to thank my wife Jennifer for her unending support and love during my studies. I thank my advisor Dave Hale for his guidance and invaluable assistance with this research. Deepest gratitude are also due to the members of my advisory committee, Andrzej Szymczak, Paul Sava and Terry Young. Thanks to the Rocky Mountain Oilfield Testing Center and Department of Energy for providing the Teapot Dome seismic data. Also thanks to Don Herron and PGS for providing the 3D salt data shown in this thesis. Special thanks also to the students and faculty at the Center for Wave Phenomena for sharing the literature and their indispensable collaboration.

Chapter 1

Introduction

For decades, geophysicists and geologists interpreted seismic sections using colored pencils and paper, where different colors were used for different geologic layers. When painting software became widely available on personal computers in the 1980's, one could use such software to perform seismic interpretation. Digital paint has advantages over drawing on paper: it can be applied in multiple overlays that can be toggled on and off, and mistakes in digital paint are easy to undo. However, limitations in technology have allowed the geoscientist to interpret only geologic horizons, or the boundaries between geologic layers. These horizons are constructed through a process either of manual picking or using auto-tracking software, both of which involve selecting points along geologic boundaries in a seismic image. By moving through each 2D slice within a 3D seismic image, an interpreter meticulously constructs the surface which defines the boundary between two different geologic layers.

With the increasing demand for hydrocarbons, seismic imaging has proven to be an important tool for mapping the subsurface. Because of this demand, and the increased computational power that offers faster processing, seismic surveys have grown increasingly larger. The interpreter must interpret hundreds or even thousands of 2D slices of 3D images, picking along a geologic boundary while attempting to maintain consistency from slice to slice. Hence, the process of manually picking horizons for geologic interpretation is slow and tedious, especially for large data sets. Semi-automatic picking algorithms expedite the process of horizon construction, but build incomplete models that must be constantly updated to improve accuracy.

In geophysical exploration, geoscientists are ultimately interested in mapping geologic layers, for it is geologic volumes that may contain hydrocarbons or minerals of interest. Horizons are therefore incomplete because they define only the boundaries between layers. Because the interpreter relies on geologic layering apparent in noisy seismic images, these horizons are incomplete. Figure 1.1 shows a sample picked horizon. Note that the horizon contains holes which may exist because of either shortcomings in an automated tracking process or human error involved with manual picking. An alternative approach to interpretation is to interpret the geologic volumes directly using 3D painting.

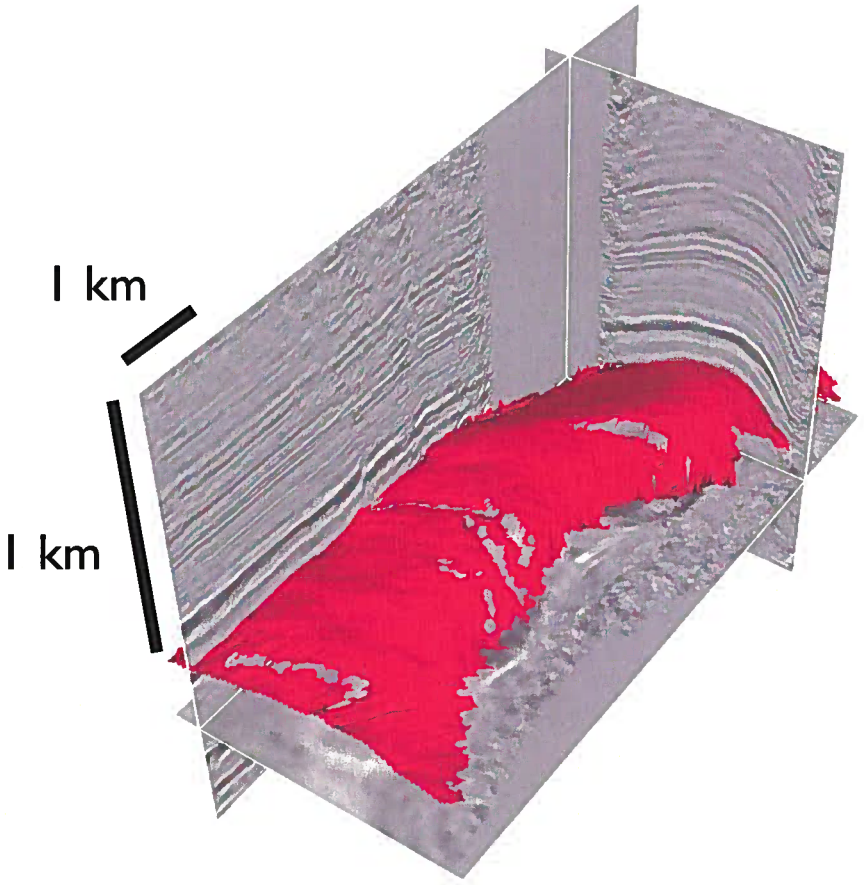


Figure 1.1. An example of a seismic horizon. Gaps in the horizon exist because of either shortcomings in automatic tracking or manual picking errors.

1.1 What is 3D painting?

Many painting software packages offer a variety of tools for creating and editing images. Most of these tools assign color values to pixels of a 2D image displayed on a 2D computer screen. Painting in 3D requires painting voxels (3D pixels) of a 3D image displayed on a 2D computer screen. Painting in 3D is inherently more difficult, in part because of the projection from 3D to 2D, but also because 3D space-filling images can seldom be displayed in their entirety. Usually one can only visualize 2D slices of 3D images and paintings.

However, recent techniques enable painting on a 2D screen with a simulated 3D environment. For example, an artist's brush stroke may be realistically reproduced by constructing virtual brushes, thereby transforming the user's cursor into a convincing paintbrush (Baxter *et al.*, 2001; Baxter & Lin, 2004). Another simulated 3D painting method involves interactively painting texture directly onto a triangulated surface with perspective projection on a 2D screen. This method enables an artist to paint textures directly onto scanned surfaces in real-time (Hanrahan & Haeberli, 1990; Agrawala *et al.*, 1995).

The techniques mentioned above use virtual brushes and surfaces to paint. While the painting environment is almost 3D, the user is unable to paint anything that does not lie within the surface in which the paintbrush is confined. Confinement of the paintbrush to a single surface inhibits efficient painting of 3D volumes, such as those filled by 3D seismic images. As suggested above, when interpreting 3D geologic structures one should paint volumes directly. Here, I refer to the process of painting seismic volumes directly as *painting in 3D* or simply *3D painting.*

In recent years, others have proposed methods for 3D painting of subsurface geology. Like the method described in this thesis, these other painting methods employ image processing algorithms to guide the painting of imaged geologic structures.

1.2 Predictive painting with lateral prediction filters

Fomel (2008) proposed a method for 3D painting using local estimates of slopes of

reflections in seismic images. The method he uses is called *predictive painting* because it uses lateral (trace-to-trace) prediction filters. This method uses reflection slopes to guide extrapolation of painted values from any reference trace to other traces in the seismic image. The method described by Fomel (2008) is interactive in that a user specifies one or more reference traces. Paint then automatically flows from the reference traces to other traces along seismic reflections. When multiple reference traces are specified, this method averages painting values extrapolated from different reference traces. In effect, paint flows laterally in directions that minimize lateral prediction errors. Therefore, this painting method works best when traces in a seismic image can be well-predicted by adjacent traces. However, this method works less well when painting across faults, unconformities, or folds. Furthermore, steeply dipping layers, salt diapirs, and stratigraphic features such as channels, are poorly described by lateral trace-to-trace prediction of seismic reflections.

1.3 Seismic volume "visulation"

A different visualization and simulation ("visulation") method described by Kadlec (2009) uses structure tensors (van Vliet & Verbeek, 1995) computed from 3D seismic images to guide 3D painting. For example, an interpreter might first pick seed points on 2D slices of 3D seismic images. These seed points then serve as sources of paint in a simulation of an anisotropic fluid flow that is guided by the structure tensors. At each time step of the flow simulation, paint diffuses from the source voxels to other voxels in the 3D image. An interpreter may interactively stop the simulation when paint has spread far enough or when new seed points must be specified to fill in unpainted regions. After each iteration of the simulation, painted surfaces are rendered as a triangular mesh.

1.4 A 3D image-guided paintbrush

In this thesis, I describe a new method for interpreting 3D seismic images by directly painting seismic volumes. Like the two methods summarized above, this 3D painting al-

gorithm includes three features: an ability to interactively select and paint a 3D voxel, a mechanism for automatically painting nearby voxels, and a user-friendly interface. Of the methods outlined above, this algorithm is most similar to that of Kadlec (2009) in that the paint is guided by structure tensors computed from a 3D seismic image.

The proposed method is primarily focused on seismic interpretation in that it uses a paintbrush to directly interpret geologic structures within seismic images. The paintbrush is guided by structure tensors computed from a seismic image and is a digital representation of an interpreter's tool; it is analogous to a colored pencil for interpretation on paper. Because the paintbrush is guided by structure tensors, the brush takes the shape of features in an image. The advantage of using a 3D digital paintbrush is that it enables the user to paint outside of a plane, enabling interpretation of multiple slices of a 3D seismic image simultaneously. The user then has the ability to render 3D formations from the painted voxels.

In the following chapter, I explain how to construct a 3D image-guided paintbrush. I first discuss a method for computing structure tensors proposed by van Vliet & Verbeek (1995), and expanded upon by Fehmers & Höcker (2003). From these tensors I then define a new metric that allows paint to conform to features within an image. In the third chapter, I show how to represent these painted formations in 3D with sub-voxel precision by introducing a 3D painting data structure. In the fourth chapter, I show examples obtained from painting 3D seismic images.

Chapter 2

Constructing the paintbrush

In traditional painting software, the user is given a digital canvas (a 2D image) and a set of painting tools. (See Figure 2.1). These tools may include various shapes (or glyphs) for the paintbrush, different sizes to apply to these glyphs, and a color palette. With any of these tools, painting is interactive because the user selects pixels with a cursor, but software automatically paints the selected pixels and other pixels nearby. Automatic painting of nearby voxels is essential because users rarely want to paint every pixel one at a time.

To paint a 3D image, we should use a 3D paintbrush. We can easily extend the concept of a 2D glyph into 3D. For example, a circle in 2D becomes a sphere in 3D. However, a sphere is inadequate for complex seismic images. A spherical paintbrush is therefore difficult and cumbersome to use. It is practical and convenient to use a paintbrush that conforms to the features within a 3D seismic image. To construct this brush, I first compute structure tensors from the image.

2.1 Structure tensors $\mathbf{S}(\mathbf{x})$

Before painting a 3D seismic image, I first compute a structure tensor field. By definition, structure tensors characterize anisotropic patterns that exhibit a single local orientation. As described by van Vliet & Verbeek (1995) and Fehmers & Höcker (2003), each structure tensor in a 3D tensor field is a smoothed outer-product of image gradients.

Let $\mathbf{g}(\mathbf{x}) = \boldsymbol{\nabla} f(\mathbf{x})$ denote the gradient vector field computed from an image $f(\mathbf{x})$. Both

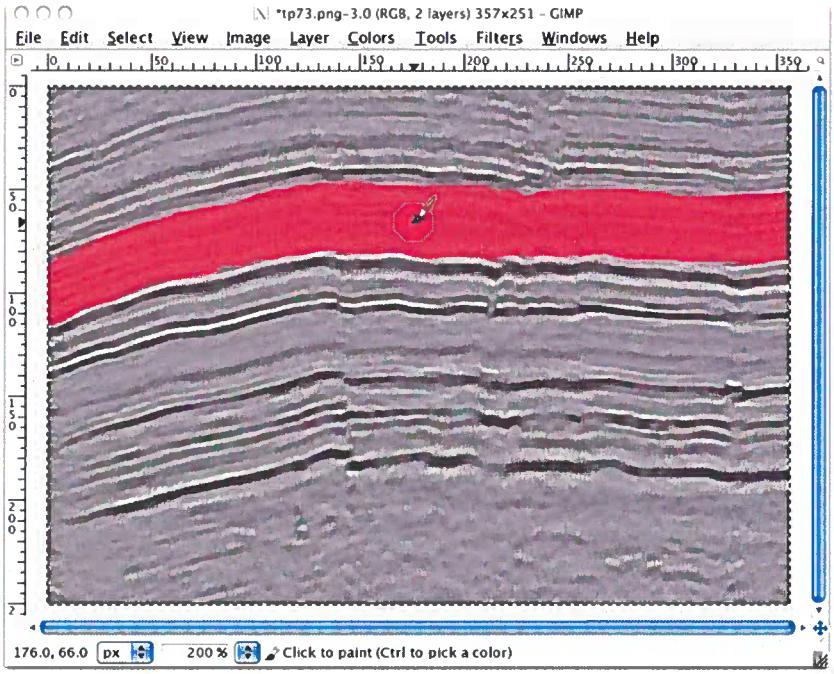


Figure 2.1. A seismic section being painted using GIMP, an open-source 2D paint program. In traditional painting software, paintbrush shapes are presented in the form of glyphs, such as circles (shown here).

the gradient $\mathbf{g}(\mathbf{x})$ and image $f(\mathbf{x})$ are uniformly sampled functions of $\mathbf{x}$, which represents the spatial coordinates of image voxels. The structure (or gradient-squared) tensor field $\mathbf{S}(\mathbf{x})$ is defined by

$$\mathbf{S}(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x})\, \mathbf{g}^T(\mathbf{x}) \rangle, \tag{2.1}$$

where $\langle \cdot \rangle$ denotes Gaussian smoothing along all spatial coordinate axes.

Gradient vectors $\mathbf{g}(\mathbf{x})$ represent estimates of both the magnitudes and directions of greatest change in the image $f(\mathbf{x})$. Structure tensors $\mathbf{S}(\mathbf{x})$ represent much of the same information averaged within Gaussian windows. This spatial averaging improves the fidelity of orientations and other attributes that I estimate from structure tensors, but it also

decreases our ability to detect abrupt changes in these attributes.

As originally demonstrated by Fehmers & Höcker (2003), the eigen-decomposition of a 3D structure tensor $\mathbf{S}$ provides a measure of orientation and dimensionality. In 3D, the eigen-decomposition of $\mathbf{S}$ is

$$\mathbf{S} = \lambda_u \mathbf{u}\mathbf{u}^\mathbf{T} + \lambda_v \mathbf{v}\mathbf{v}^\mathbf{T} + \lambda_w \mathbf{w}\mathbf{w}^\mathbf{T}, \tag{2.2}$$

where $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are orthonormal eigenvectors and corresponding eigenvalues λ_u, λ_v, and λ_w are sorted so that

$$\lambda_u \geq \lambda_v \geq \lambda_w \geq 0. \tag{2.3}$$

As stated earlier, each structure tensor $\mathbf{S}$ is positive semi-definite, so every eigenvalue is non-negative.

For any image voxel, the eigenvector $\mathbf{u}$, which corresponds to the largest eigenvalue λ_u, indicates the direction in which the image changes most. In a seismic image, the eigenvector $\mathbf{u}$ is generally orthogonal to imaged geologic layers. The eigenvector $\mathbf{w}$, which corresponds to the smallest eigenvalue λ_w, indicates the direction in which the image changes least; it may be aligned with images of buried channels. Both eigenvectors $\mathbf{v}$ and $\mathbf{w}$ tend to lie in planes of locally planar features in 3D seismic images, as shown in Figure 2.2.

2.2 Tensor-guided paintbrush

A 3D image-guided paintbrush is computed from a metric tensor field that is derived from the structure tensor field $\mathbf{S}(\mathbf{x})$. A metric tensor field defines a measure of distance between two points. For a constant metric tensor $\mathbf{D}$, the distance $t(\mathbf{x})$ to any point $\mathbf{x}$ is

$$t(\mathbf{x}) = \sqrt{\mathbf{x}^T \mathbf{D}^{-1} \mathbf{x}}. \tag{2.4}$$

When $\mathbf{D}$ equals the identity matrix, $t(\mathbf{x})$ is simply Euclidean distance.

$$\lambda_u \geq \lambda_v \geq \lambda_w \geq 0$$

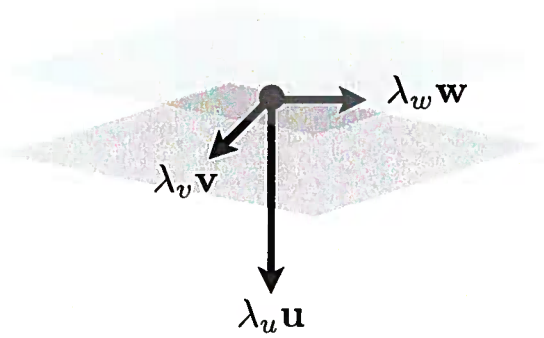


Figure 2.2. A visual representation of a 3D structure tensor computed near planar features. In a seismic image, this is analogous to a structure tensor computed within a geologic layer. The eigenvector $\mathbf{u}$ points in the direction of the greatest change which happens to be orthogonal to geologic layers. The remaining eigenvectors $\mathbf{v}$ and $\mathbf{w}$ lie within the layer.

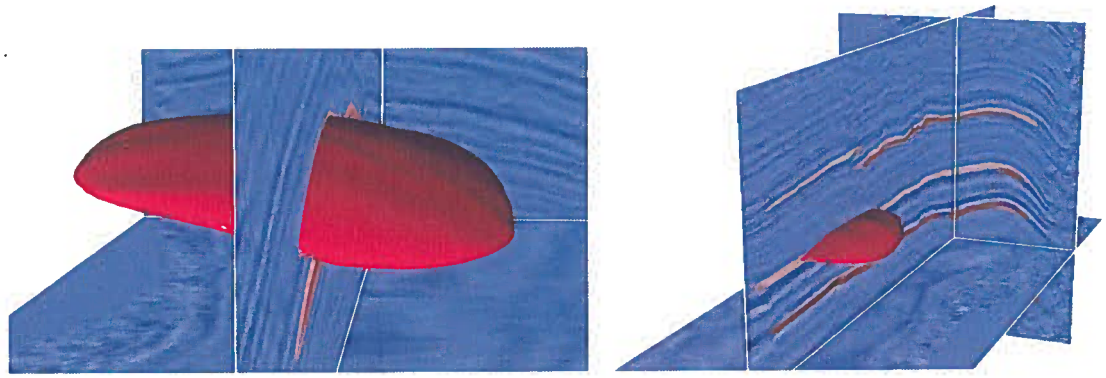
More generally, if $\mathbf{D} = \mathbf{D}(\mathbf{x})$ is a non-constant metric tensor field, we must compute distances numerically by solving an eikonal equation:

$$\nabla t(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x}) \cdot \nabla t(\mathbf{x}) = 1 \tag{2.5}$$

with the boundary condition $t(\mathbf{0}) = 0$. In this case, $t(\mathbf{x})$ denotes non-Euclidean distance between two points in a seismic image. Distances $t(\mathbf{x})$ are computed by solving a finite-difference approximation to equation 2.5 using a fast iterative method similar to that proposed by Jeong *et al.* (2007).

By rendering a surface of constant distance $t(\mathbf{x}) = t_{max}$, where t_{max} denotes a user-specified maximum brush size in voxels, I obtain the surface outline of the 3D paintbrush, as shown in Figure 2.3. In this example, the maximum brush size is $t_{max} = 58$ voxels.

To compute distances $t(\mathbf{x})$, the user first selects one voxel within a 3D seismic image.



(a) (b)

Figure 2.3. A seismic image painted using a 3D digital paintbrush that conforms to features within the image. The rendered paintbrush is a contour of constant time computed by solving an eikonal equation.

This voxel acts as the origin where distance $t(\mathbf{0}) = 0$. Voxels for which $t(\mathbf{x}) < t_{max}$ lie inside the 3D paintbrush, and voxels for which $t(\mathbf{x}) > t_{max}$ lie outside.

In geophysics, eikonal equations are often used to compute traveltimes. The eikonal equation 2.5, with anisotropic and spatially varying coefficients $\mathbf{D}(\mathbf{x})$, is the same as that used by Hale (2009a) for image-guided interpolation. In that application, as with the 3D painting algorithm, "time" is a synonym for "non-Euclidean distance" computed in a metric tensor field.

2.3 Metric tensors $\mathbf{D}(\mathbf{x})$

Because paint is guided by tensors, I alter the eigenvalues of the structure tensors $\mathbf{S}(\mathbf{x})$ so that paint is confined by features in the image. Although structure tensors $\mathbf{S}(\mathbf{x})$ define the local orientation of these features, the eigenvalues λ_u, λ_v, and λ_w correspond to the amount of change in the direction of their eigenvectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$. As paint spreads within an image, the direction and rate of flow is governed directly by these eigenvector and eigenvalue pairs. The derived eigenvalues would guide paint across, not along, features in an image. For the purpose of interpretation, we want paint to flow anisotropically within

geologic layers. I construct an anisotropic paintbrush, like that shown in Figure 2.3, by computing an anisotropic metric tensor field $\mathbf{D}(\mathbf{x})$. The new eigenvalues are computed using structure-oriented semblance (Hale, 2009b). Semblances are useful, in part, because they are an amplitude-independent measure of the coherence of features in seismic images. Semblances and, hence, the eigenvalues of $\mathbf{D}$ are normalized in the range [0,1].

I choose the eigenvectors of each metric tensor $\mathbf{D}$ to be the same as those for their corresponding structure tensor $\mathbf{S}$. The difference between $\mathbf{D}$ and $\mathbf{S}$ lies only in their eigenvalues. Specifically, the eigen-decomposition of $\mathbf{D}$ is

$$\mathbf{D} = s_3\mathbf{u}\mathbf{u}^T + s_2\mathbf{v}\mathbf{v}^T + s_1\mathbf{w}\mathbf{w}^T, \tag{2.6}$$

where eigenvalues s_1, s_2, and s_3 are semblances such that

$$0 \leq s_3 \leq s_2 \leq s_1 \leq 1. \tag{2.7}$$

Structure-oriented semblance measures local coherency in a given dimension and direction. More generally, semblance can be defined as the ratio between the squared-smoothed image and the smoothed-squared image. Using a method proposed by Hale (2009b), I compute structure-oriented semblance by smoothing along features in an image. Semblances s_1, s_2, and s_3 are computed as

$$s_1 = \frac{\langle\langle f\rangle_w^2\rangle_{uv}}{\langle\langle f^2\rangle_w\rangle_{uv}}, \tag{2.8}$$

$$s_2 = \frac{\langle\langle f\rangle_{vw}^2\rangle_u}{\langle\langle f^2\rangle_{vw}\rangle_u}, \tag{2.9}$$

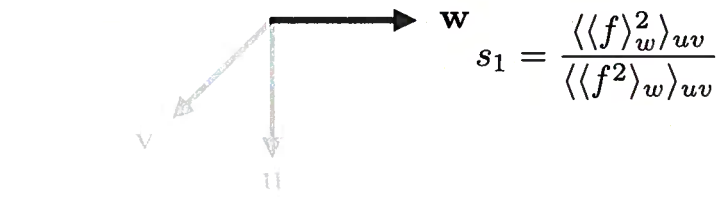
and

$$s_3 = \frac{\langle f\rangle_{uvw}^2}{\langle f^2\rangle_{uvw}}. \tag{2.10}$$

where $\langle \cdot \rangle$ denotes smoothing, and the associated subscript describes the direction of that smoothing. With structure-oriented semblance, there is a second smoothing operation performed in the directional orthogonal to the initial smoothing direction. For s_1 and s_2, these smoothing directions occur in the plane of **uv** and in the direction of **u**, respectively. Note there is only one smoothing operation for s_3 because it already smooths in all directions. In this way, the largest eigenvalue s_1, corresponding to the eigenvector **w**, is semblance computed within a locally curvilinear (1D) set of voxels aligned with **w**. Each eigenvalue s_2, corresponding to an eigenvector **v**, is semblance computed within a locally curviplanar (2D) set of voxels orthogonal to eigenvectors **u**. (The plane orthogonal to **u** contains the eigenvectors **v** and **w**). Finally, each eigenvalue s_3 represents semblance for a locally spherical (3D) set of voxels. Figure 2.4 graphically represents equations 2.8, 2.9, and 2.10.

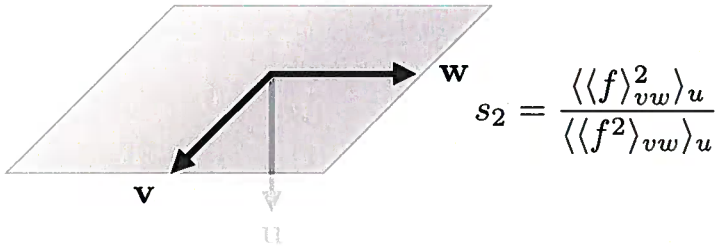
The relationship described in equation 2.7 holds true due to the anisotropic nature of a 3D seismic image with zero mean. Because s_1 describes curvilinear coherency within an image, similar voxels that lie along a curvilinear path yield higher semblance values. This typically occurs within linear features such as buried channels, or within the plane of a geologic bed, as well as in areas with more isotropy such as salt bodies. Semblance s_2 describes curviplanar coherency within an image, meaning that locally planar (but not linear) features yield higher semblance values. Therefore, features such as geologic beds have higher values for s_2. Moreover, s_2 values are typically low at fault regions because faults interrupt the planar coherency within geologic layers. Finally, semblance values s_3 describe volumetric coherency within an image. These s_3 values tend to be lower throughout a seismic image because the average of image voxels within a 3D window with a radius greater than a seismic wavelength will be nearly zero. In fact, only in volumes of nearly constant values would s_3 have a value closer to one. Figure 2.5 gives a visual representation of a 3D metric tensor **D** with eigenvectors **u**, **v**, and **w** scaled by s_3, s_2, and s_1, respectively.

Because the eigenvalues of **D** are bounded between [0,1], distances $t(\mathbf{x})$ computed by solving equation 2.5 never exceed distances computed for a constant identity tensor $\mathbf{D} = \mathbf{I}$.



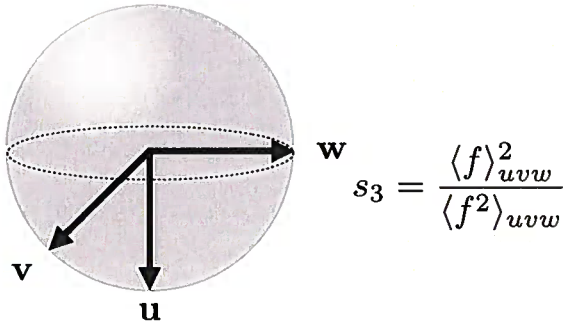
$$s_1 = \frac{\langle \langle f \rangle_w^2 \rangle_{uv}}{\langle \langle f^2 \rangle_w \rangle_{uv}}$$

(a)



$$s_2 = \frac{\langle \langle f \rangle_{vw}^2 \rangle_u}{\langle \langle f^2 \rangle_{vw} \rangle_u}$$

(b)



$$s_3 = \frac{\langle f \rangle_{uvw}^2}{\langle f^2 \rangle_{uvw}}$$

(c)

Figure 2.4. Visual representation of equation 2.8 (a), 2.9 (b), and 2.10 (c). For s_1, local coherency is measured in the direction of eigenvector **w**. For s_2, local coherency is measured within a plane perpendicular to eigenvector **u**, which contains eigenvectors **v** and **w**. Finally, for s_3, coherency is computed within a sphere that contains all three eigenvectors.

$$0 \leq s_3 \leq s_2 \leq s_1 \leq 1$$

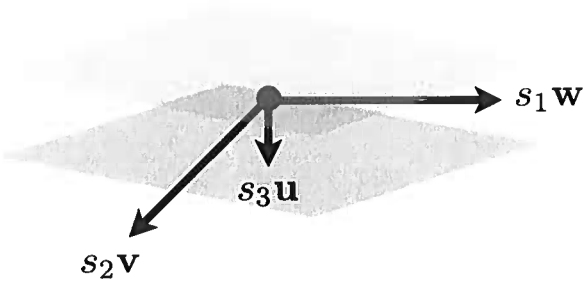


Figure 2.5. Visual representation of a 3D metric tensor computed between two bounding planes similar to Figure 2.2. The only difference between the two figures is that the metric tensor has eigenvalues computed using semblance. These eigenvalues allow paint to flow faster within geologic layers, and much slower across.

In other words, non-Euclidean distances $t(\mathbf{x})$ will always be less than or equal to Euclidean distances. Therefore, when specifying the maximum distance t_{max}, one may think intuitively of Euclidean distance and know that the 3D paintbrush, like that shown in Figure 2.6, lies inside a sphere with radius t_{max}. In noisy incoherent regions of a 3D seismic image, where all three semblances are low, the brush will be much smaller than that sphere.

The upper bound t_{max} also speeds up computation of distances $t(\mathbf{x})$. When a user selects a voxel in the 3D seismic image, that point becomes the origin for the eikonal equation 2.5. In solving that equation, I need only consider voxels at locations $\mathbf{x}$ that lie inside a sphere centered at the origin with radius t_{max}. For any voxels outside that sphere, distances $t(\mathbf{x})$ must exceed t_{max}.

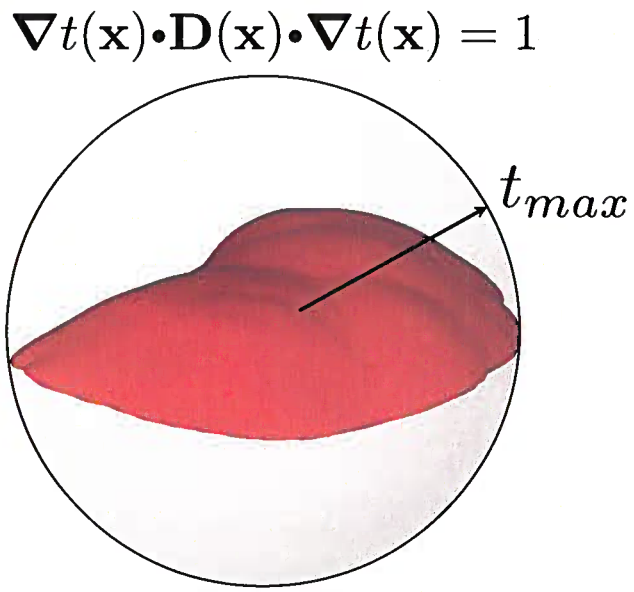


Figure 2.6. An example paintbrush within a bounding sphere with radius t_{max}. Because the eigenvalues are bounded between [0,1], the 3D paintbrush always lies within this sphere.

2.4 Visualizing metric tensors D(x) as ellipsoids

To visualize metric tensors, I display them as ellipsoid glyphs (Engelsma & Hale, 2010). While many methods have been proposed for rendering tensor information, such as assembling hyperstreamlines (Delmarcelle & Hesselink, 1993), deforming volume meshes (Zheng & Pang, 2002), and constructing superquadric glyphs (Kindlmann, 2004), ellipsoid glyphs give a discretized and intuitive representation of metric tensors. These glyphs show local directions in which paint will flow. Furthermore, I use ellipsoid glyphs because there is a unique geometric relationship between ellipsoids and symmetric positive-definite matrices. Consider the definition of a non-axis-aligned ellipsoid to be

$$\mathbf{x}^T \mathbf{D}^{-1} \mathbf{x} = 1, \tag{2.11}$$

where $\mathbf{x}$ represents any point along the surface of an ellipsoid, and $\mathbf{D}$ is a metric tensor as defined in equation 2.6. In this definition, the eigenvectors of $\mathbf{D}$ define the principle axes of an ellipsoid, and the inverses of the square roots of the eigenvalues define the equatorial radii (Strang, 2003), as shown in Figure 2.7.

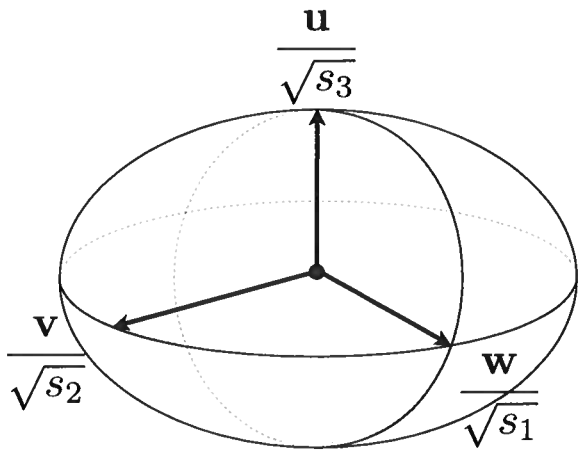


Figure 2.7. Geometric relationship between eigenvalues and eigenvectors and the three principle radii of an ellipsoid. Each principle axis of an ellipsoid is an eigenvector scaled by the square root of the corresponding eigenvalue.

Figure 2.8 illustrates three image-aligned panels of evenly-sampled ellipsoids superimposed on a 3D seismic image. The displayed ellipsoids represent only a subset of the tensors $\mathbf{D}(\mathbf{x})$ computed for every image sample. The shapes of ellipsoids represent pertinent structural geologic information. For example, in areas of a seismic image that exhibit flat geologic layering, ellipsoids tend to be oblate. In areas of dipping layers, ellipsoids exhibit the local dips of geologic structures. In areas where features are more isotropic, ellipsoids are more spherical.

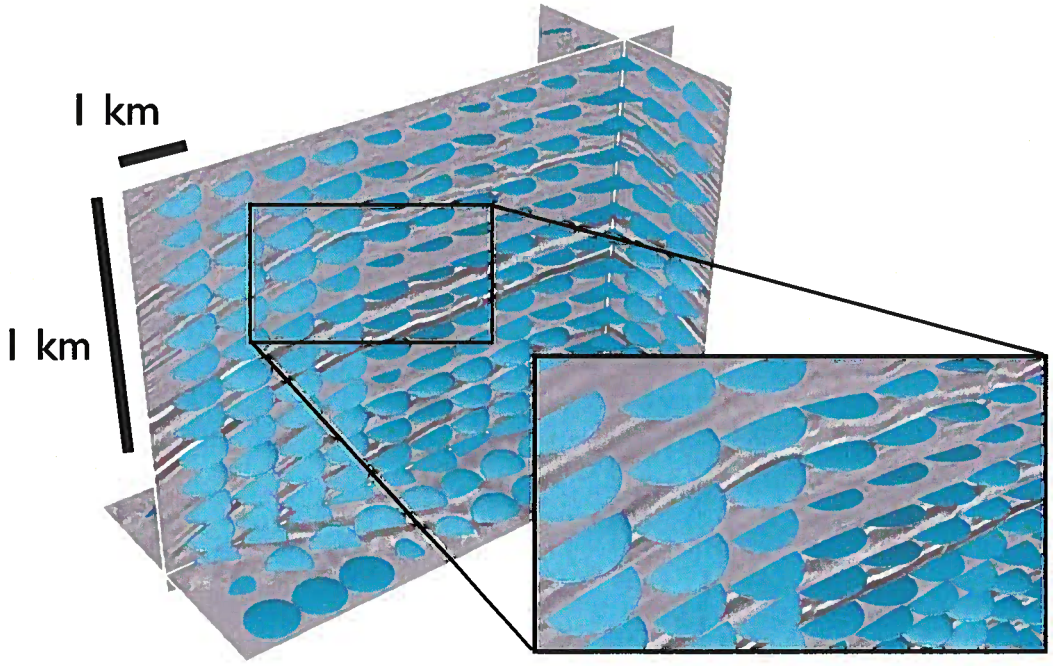


Figure 2.8. Visualization of a metric tensor field $\mathbf{D}(\mathbf{x})$. Each ellipsoid represents the local directions that paint will flow within an image. A closeup of the seismic image shows how paint will tend to flow within (rather than across) geologic layers.

Chapter 3

Representing painted volumes

In this chapter, I discuss a method for representing painted (interpreted) volumes. Painting these volumes involves using a data structure that paints the formations with sub-voxel precision. This data structure is used in conjunction with the marching cubes algorithm (Lorenson & Cline, 1987) to render painted volumes as 3D meshes. Furthermore, I discuss different techniques for coloring the formations, including a method that gives painted volumes the appearance of being carved from an image.

3.1 Implicit vs. explicit surface rendering

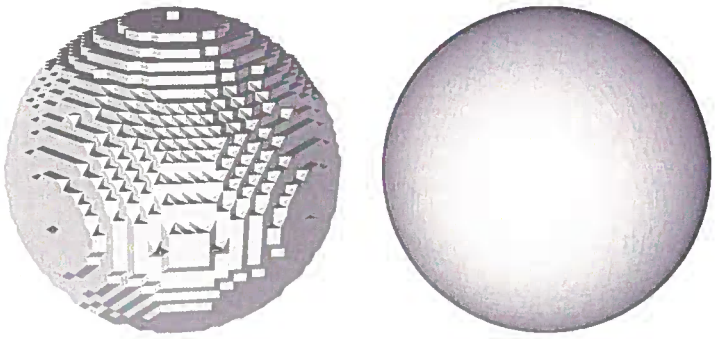
In graphics and visualization, contours may be extracted from volumetric data as either implicit or explicit surfaces. Implicit surfaces are rendered directly on the GPU. In practice, volume rendering involves using a ray casting algorithm that accumulates density values from an array of floating point numbers to render a surface. One limitation for volume rendering is that it does not allow extraction of an isosurface for further processing. Because volume rendering is simply a rendering technique, it is used only for visualization. Furthermore, the computational cost of rendering implicit surfaces is much higher compared to extracting an isosurface using an explicit surface method such as marching cubes (Kadlec, 2009; Lorenson & Cline, 1987). Another benefit of using marching cubes is that it creates a 3D mesh that can be extracted for further processing.

Marching cubes (Lorenson & Cline, 1987) is a popular method used to extract explicit surfaces from volume data. The algorithm involves scanning through each voxel within a

3D volume. At each voxel, the values of the adjacent eight voxels (or cube) are compared with a contour value to construct interpolated edge intersections. These edge intersections are then connected to form triangulated meshes. For convenience, the topology of these triangulated meshes can be determined using a table-lookup scheme. Because each cube can be processed independently, this technique can be easily run with parallel concurrency.

3.2 Painting with sub-voxel precision

Because the paintbrush is rendered from a smooth range of times computed using equation 2.5, employing a basic marching cubes algorithm renders a paintbrush with sub-voxel precision, as seen in Figure 2.3. The paintbrush is, hence, a contour of constant time, where $t(\mathbf{x}) = t_{max}$. Each voxel within the paintbrush where $t(\mathbf{x}) < t_{max}$ is assigned a value indicating that it has been painted.



(a) (b)

Figure 3.1. A spherical formation represented using voxel-precision (a) and sub-voxel precision (b). Sub-voxel precision is accomplished through using a 3D painting data structure.

Consider painting a sphere within a 3D grid of voxels. Ideally, the painted formation should appear identical to the paintbrush used to construct it. Using the marching cubes algorithm to render the painted voxels yields formations which appear terraced, as seen

in Figure 3.1(a). Typically, interpreted horizons are always represented using sub-voxel precision, because it is important to model the subterrain with high accuracy, as shown in Figure 1.1. Like horizons, the boundaries of painted volumes must be represented with sub-voxel resolution. I accomplish this by implementing a special data structure for 3D paintings.

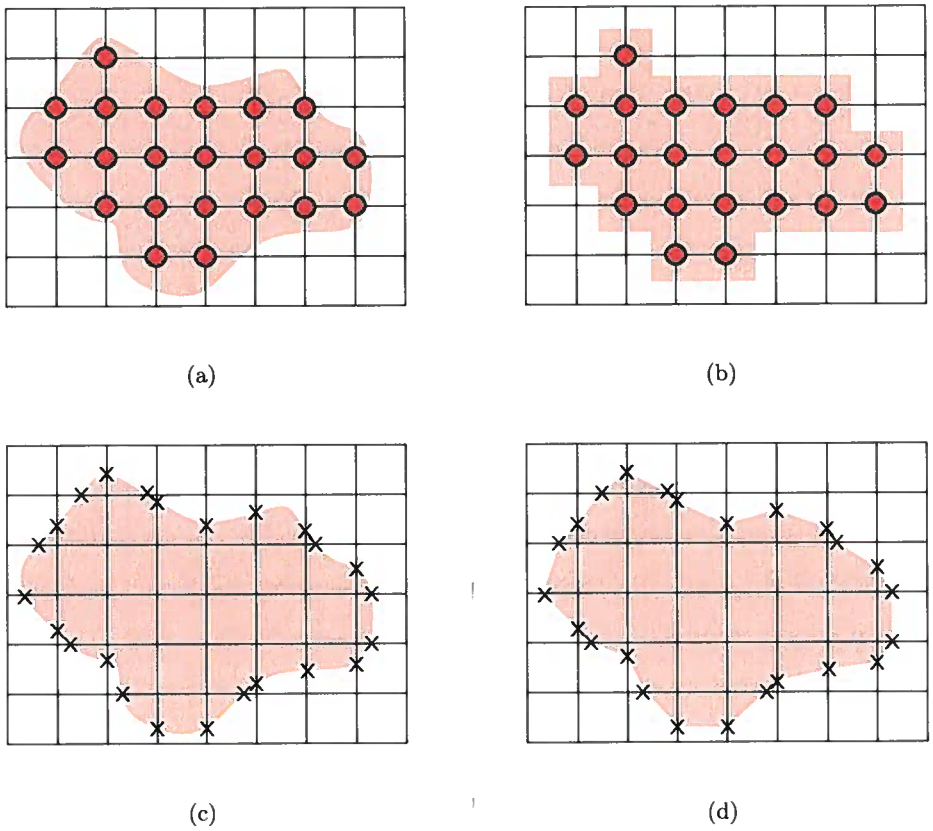


Figure 3.2. 2D demonstration of the painting data structure. A painted area represented as marked samples (a) produces a terracing artifact (b). By maintaining edge intersections that bound the painted area (c) sub-pixel precision is achieved (d).

Painting with sub-voxel precision requires more bookkeeping. By definition, a painted volume is the union of every paintbrush used to paint that formation. Each time I build

a paintbrush, I compute its boundary in terms of edge intersections within the 3D mesh. Finding each intersection requires the same procedure used in marching cubes. Once the times are constructed using equation 2.5, I compare each voxel's time to the target contour value t_{max}. If two adjacent voxels lie on both sides of t_{max}, then the intersection of t_{max} between these two samples is found using linear interpolation. Because I am using a 3D mesh, I store edge intersections in three separate arrays. For example, arrays e_1, e_2, and e_3 may store edge intersections in the first, second, and third dimensions, respectively. All computed intersections are mutable if, for example, the user continues to paint the image. Figure 3.2 shows a 2D example of this painting data structure.

The computed intersections e_1, e_2, and e_3 are later used for visualizing the painted image in 3D. The marching cubes algorithm is applied, only I now incorporate the precomputed edge intersections stored in the 3D painting data structure. Ju *et al.* (2002) have proposed dual contouring methods for rendering surfaces from Hermite data, or data that contains existing intersection points and normals. While dual contouring is not used in this thesis, it may be an alternative approach to rendering painted structures. Figure 3.3 shows a painted formation represented as painted voxels (a) and the same painted formation represented using the 3D painting data structure that stores edge intersections e_1, e_2, and e_3 (b). While both formations appear very similar, a closer inspection of these formations reveals terraced topography (c), and a smooth surface (d), respectively.

3.3 Assigning colors to formations

Each formation consists of painted voxels whose boundary is defined by pre-computed edge intersections. One option for coloring these formations is to use a solid color so that each formation is easy to distinguish within the image. (See Figure 3.3). Because each formation is assigned a single paint value, that value may be trivially mapped to a color scale. Rendering formations with constant color is equally useful when distinguishing different lithologies within an image (e.g., sands versus shales).

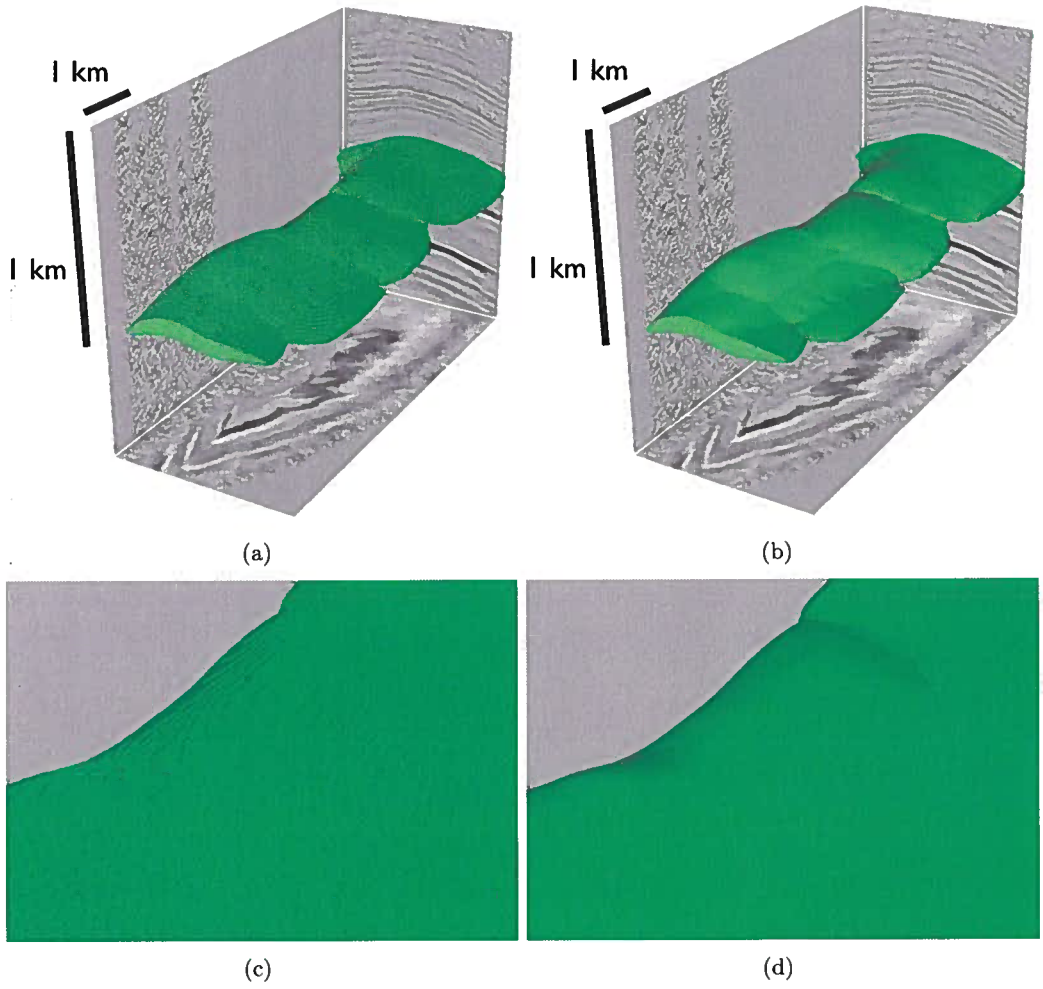


Figure 3.3. A painted formation is represented using simple marching cubes (a) and the 3D painting data structure that incorporates pre-computed edge intersections (b). The top-left surface displays a terracing artifact, which is easily observed in the close-up (c). The top-right surface is rendered with much higher precision, as shown in the close-up (d).

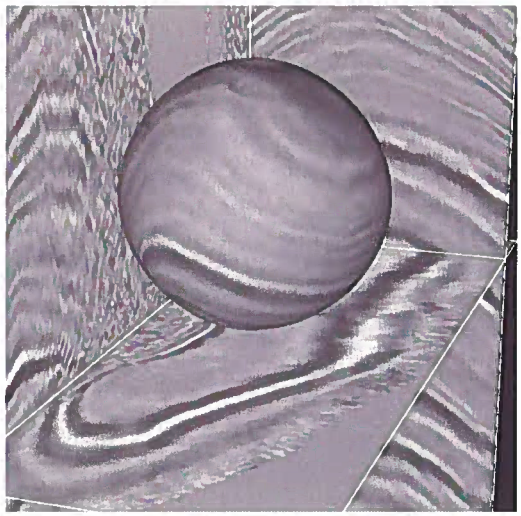


Figure 3.4. Seismic amplitudes applied to a painted sphere.

Another approach, as well as a useful quality control (QC) tool, is to apply the seismic amplitudes to the painted volume directly. Since every computed vertex lies on an edge between voxels, I need only to linearly interpolate color values from the two adjacent samples. By using the same color scale as the seismic image, this technique gives the formation the appearance of being carved from the seismic volume. Figure 3.4 shows a sample sphere formation that has been represented in this way.

One can use this carving effect as a QC tool to check a 3D painting. Assuming that an interpreter has painted a seismic image with well-imaged layers, one would expect that the top and bottom boundaries of the formation would have a constant color. This color is the seismic amplitude at the boundary between layers. It is therefore easy to identify areas where the paint has extended beyond the formation by observing the color. For example, portions of the painted volume that extend beyond their geologic boundaries may appear to have rings. Furthermore, regions that are painted where there is no geologic information within an image are easy to identify because the formation will have the same background color as the image.

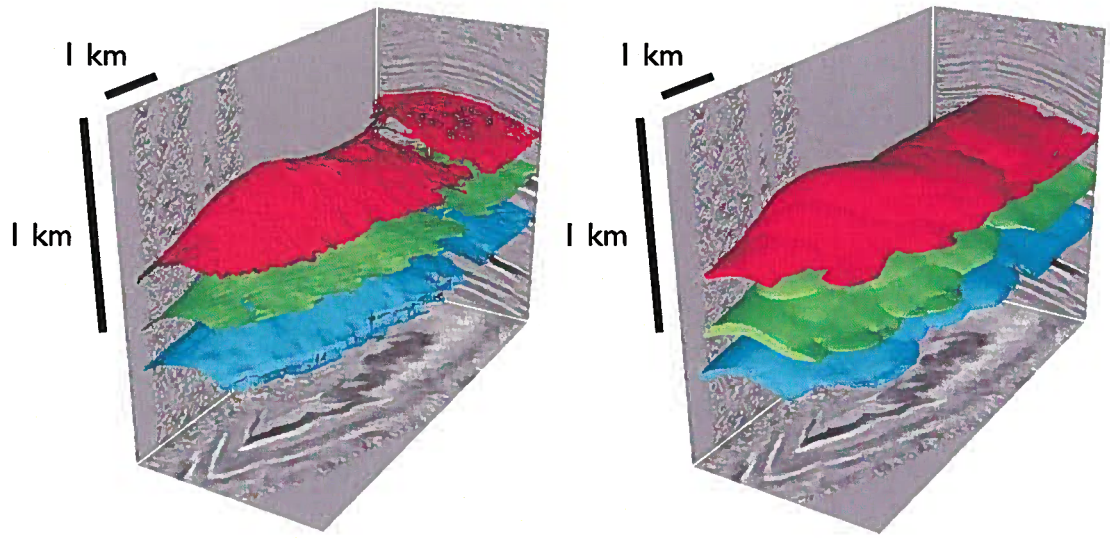
Chapter 4

Examples

4.1 Teapot Dome

I have applied seismic painting to data acquired at Teapot Dome which is located in Natrona County, Wyoming, about 55 miles north of Casper. These data were provided by the Rocky Mountain Oilfield Testing Center, which is part of the US Department of Energy. The data show a sequence of anticlinal layers cut by normal faults.

Picked horizons were also provided with these data, as seen in Figure 4.1(a). To compare my 3D painting technique with horizon picking, I painted formations corresponding to three provided horizons, as shown in Figure 4.1(b). The formation in Figure 4.2 has been rendered using seismic amplitudes as color values. Figure 4.3 shows a top-down view of the middle formation in Figure 4.2. Note that each rendered volume appears bulbous because every formation is the union of all paintbrushes used. Hence, the sparsity and size of paintbrushes affects the structure and appearance of the resulting volume. Also note that the horizons have more topographic detail than the painted formations. When picking horizons, an interpreter shifts each pick to the nearest zero-crossing using an automated method. The painted volumes do not have the same level of rugosity as horizons because each formation is ultimately constructed using metric tensors computed from the seismic image. Therefore, the shape and smoothness of each painted volume is dependent on these tensors. However, shifting to the nearest zero-crossing is achievable since zero-crossings are visible in the image. There are several ways to perform this shift. One approach is to shift the boundary of the volume in the vertical direction. This shift involves scanning the



(a) (b)

Figure 4.1. Three horizons (a) that were picked using the Teapot Dome data are shown alongside three painted formations (b). Horizons have more detail and appear more rugose than painted formations because they are typically snapped to zero-crossing. However, picked horizons contain holes because of either the lack of information visible to the interpreter or the inability of auto-tracking software to identify coherent trends. Painted formations appear smoother and fill in the holes that horizon picking misses. However, painted formations contain a bubbling artifact that is a result of the size and distribution of paintbrushes used.

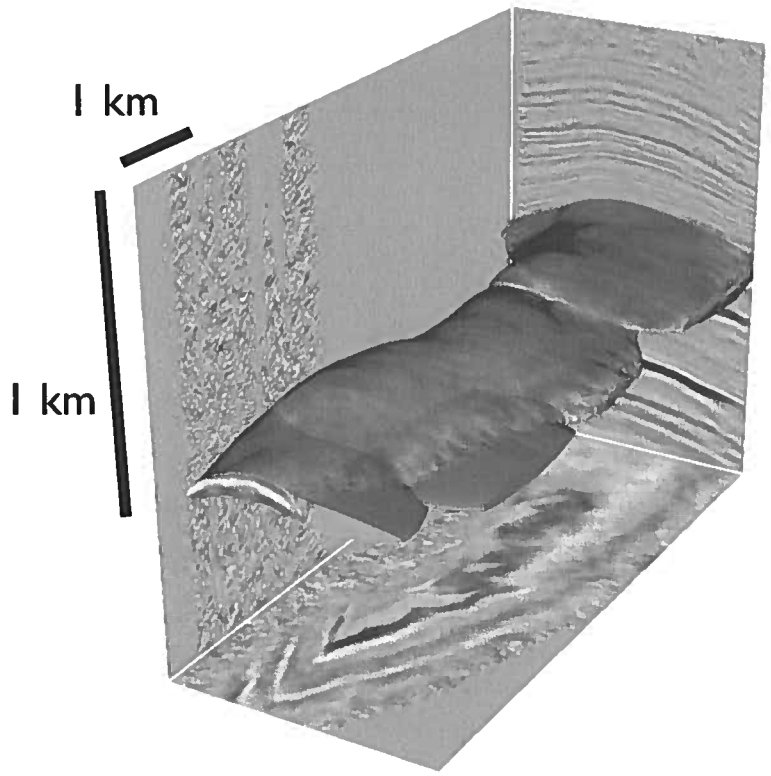


Figure 4.2. A painted formation colored using amplitudes from the seismic image. The purpose of representing geologic layers in this way is to highlight painted voxels that lie outside the layers.

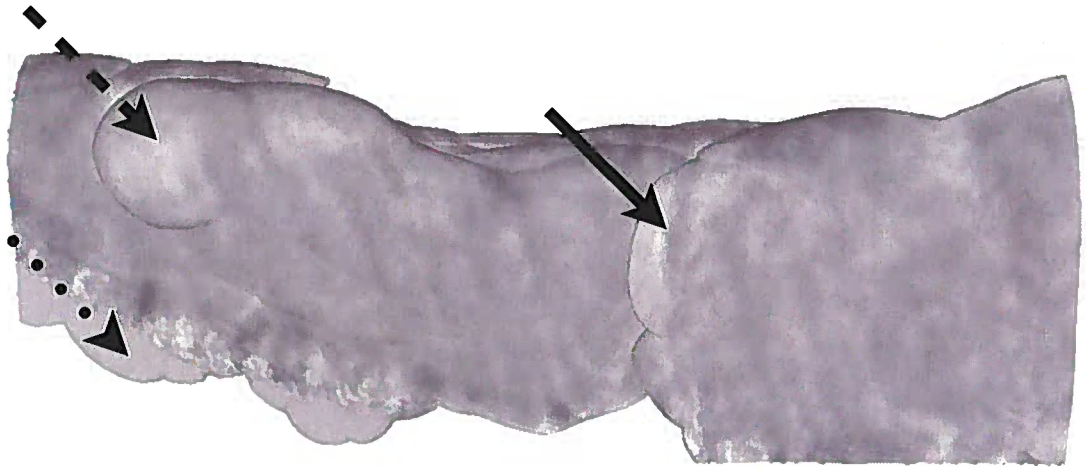


Figure 4.3. A closer examination of the formation shown in Figure 4.2. The dashed arrow points out an area where the outline of the paintbrush is obvious. This artifact is a result of the size and sparsity of paintbrushes used. The dotted arrow highlights regions of the volume that were painted where there was no information in the image. The solid arrow points at the fault region which is left as a hole in the picked horizon.

boundaries of the volume and then searching for the nearest zero-crossing within a vertical window. This method may fail for steeply dipping layers or noisy parts of an image. Another approach is to adjust the boundary of the painted volume in the direction perpendicular to the layers (in the direction of eigenvector $\mathbf{u}$). This may produce artifacts in areas where the shift requires a large jump, but placing a threshold on this shift would reduce these artifacts. A third approach is to update the tensors at zero-crossings during paint construction. This process requires the eikonal solver to set the eigenvalue λ_u to zero at zero-crossings before continuing further iterations. In this way, paint will stop flowing across geologic boundaries and continue to flow within the geologic layer.

4.2 Painting salt

I have applied 3D painting to a seismic image of subsurface salt. These data were provided by Don Herron and PGS. Typically, salt is interpreted in two parts: a top-salt and

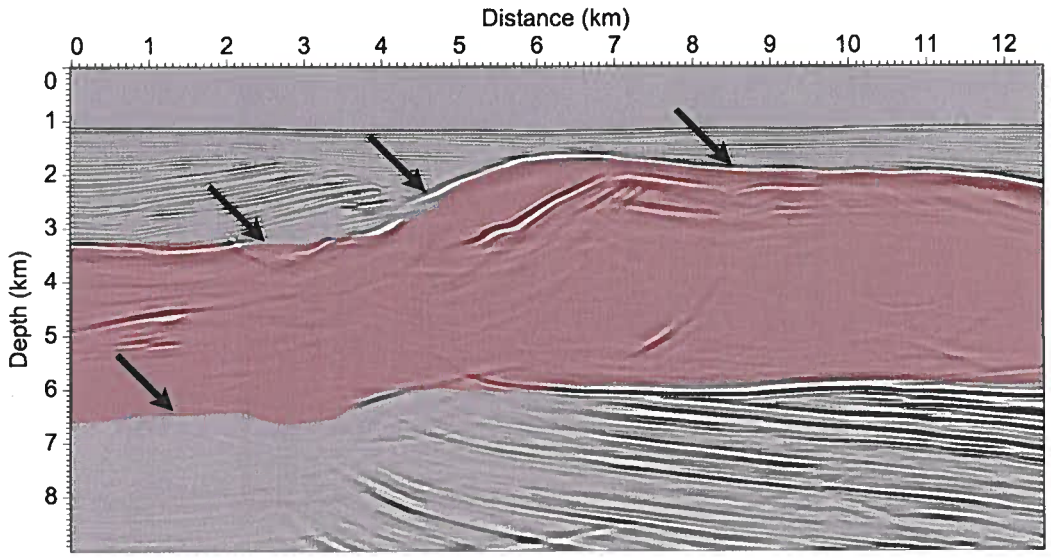


Figure 4.4. A 2D section of the painted salt body. The painted area is shown in red, and areas where the paint is not completely confined to the salt body are shown by the arrows.

a bottom-salt. While some geologic entities may be easily defined by their top and bottom boundaries, salt bodies tend to be more complex. For example, salt bodies may be steeply dipping or have multiple overhangs that can not have a defined top or bottom boundary. Furthermore, salt bodies may resemble vertical pillars that are equally difficult to define in this way. Therefore, defining a salt body by its vertical boundaries is insufficient in some cases. Defining a salt body by its volume is more intuitive, and we can quickly define that volume by employing 3D painting.

Figure 4.4 shows a 2D section of painted 3D salt data. The red painted area represents the interpreted salt. While some boundaries of the salt are well-defined, there are some areas that are problematic. Each arrow in Figure 4.4 points to an area where the paint has extended into the surrounding layers. One reason for these leaks is that the semblance parameters were poorly chosen for this geologic setting, creating a poor definition of the local coherency. Another reason is simple human error. As Figure 4.4 is intended to show,

because this method involves a level of interactivity, the result is subject to mistakes. As I previously mentioned, one of the benefits of digital paint is that it can be easily undone.

Figure 4.5 shows the salt body rendered in 3D. Seismic amplitudes were used to color the formation. As with the Teapot Dome data, the formation lacks a high level of detail. The resolution of the painted volume may be improved by shifting the boundaries of the formation to the nearest zero-crossing. Figure 4.6 shows a top-down view of the same formation in Figure 4.5. The marble appearance of the formation is caused by paint extending outside the boundaries of the salt. The white portions of the volume indicate regions where the salt boundary was not well defined and paint extended into the layers above.

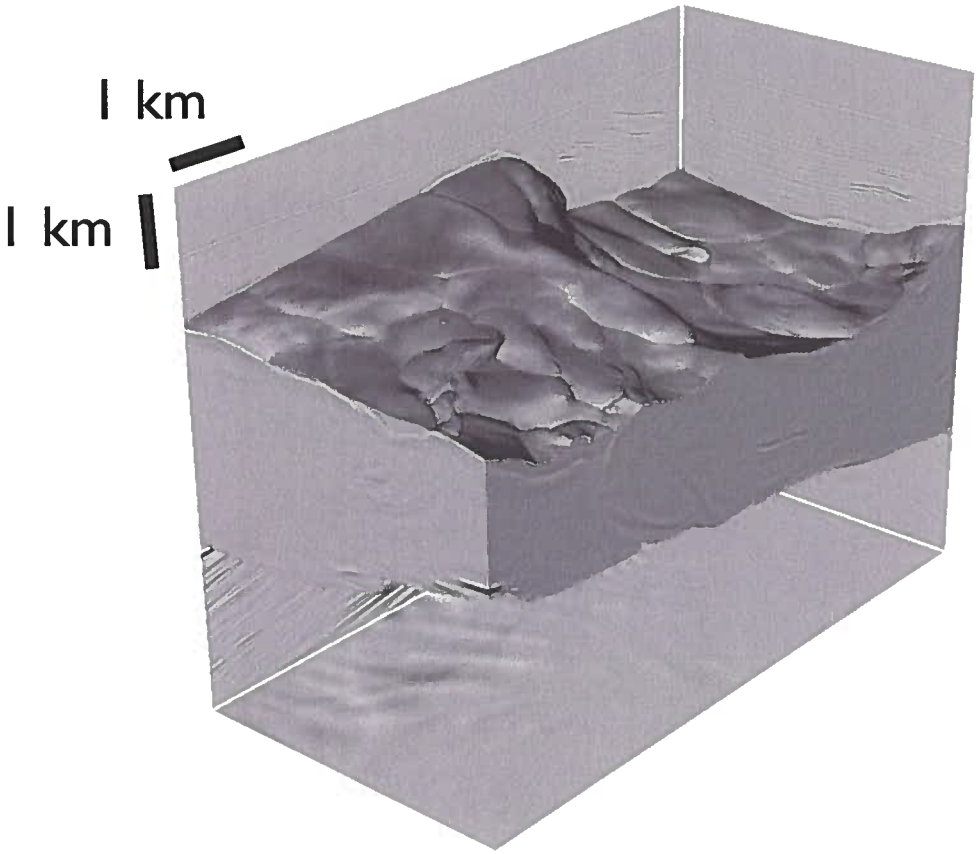


Figure 4.5. A salt body that has been painted using 3D painting. Seismic amplitudes have been used to color the formation. While the volume demonstrates a generic shape of the salt body, it still lacks a fine level of detail.

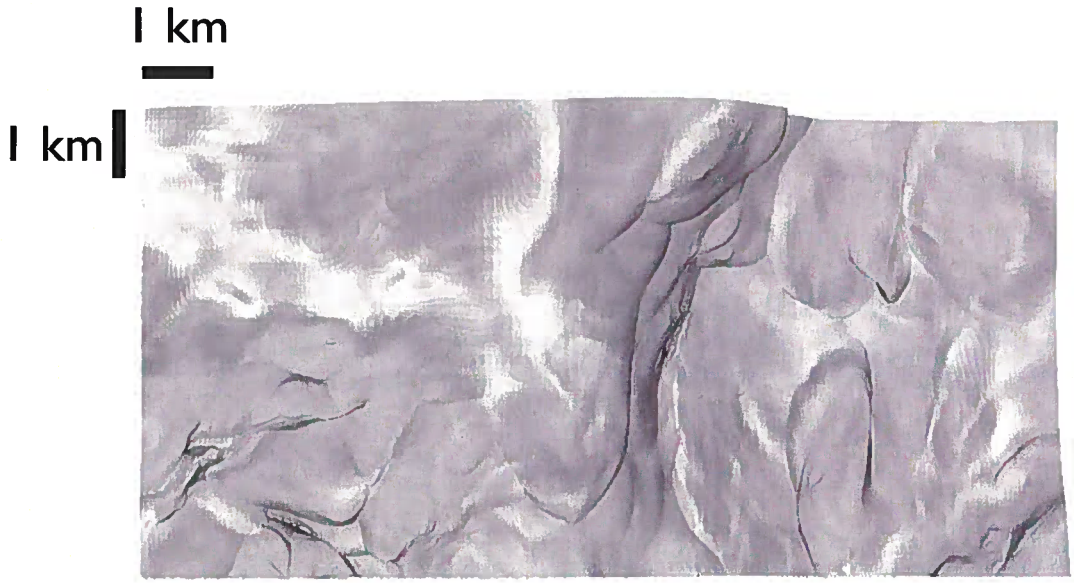


Figure 4.6. A top-down view of the same salt body in Figure 4.5. Using the seismic amplitudes to color the formation helps identify areas that were poorly painted. The boundary of a poorly painted formation deviates from zero-crossing. Some deviations include the white areas on the formation. For a volume that is properly painted, we expect to see the same color along the boundary of the formation.

Chapter 5

Conclusions

I have proposed a method to paint images in 3D. This method requires the construction of a 3D paintbrush that conforms to features within an image. The local orientations of features in the image are first estimated by computing structure tensors $\mathbf{S}(\mathbf{x})$ (van Vliet & Verbeek, 1995). The eigenvalues of structure tensors are replaced with structure-oriented semblance values, thereby constructing a new metric $\mathbf{D}(\mathbf{x})$ (Hale, 2009b). By solving an anisotropic eikonal equation, I compute non-Euclidean distances that paint voxels within an image. Rendering a contour of constant time will display a paintbrush in 3D.

Painted volumes are represented with sub-voxel precision for accuracy comparable to picked seismic horizons. This sub-voxel precision is maintained by a 3D painting data structure that stores edge intersections in a 3D grid. Using seismic amplitudes to color the formation gives the formation the appearance of being carved from the image. This coloring technique may be used as a QC tool to check painted formations for potential errors.

Painting 3D images has numerous applications. Painting seismic images could be a useful interpretation tool. By painting geologic volumes directly, geologic entities such as salt become easier to define. In seismic interpretation, salt bodies are traditionally picked as a top-salt horizon, and a bottom-salt horizon. 3D painting instead considers the entire salt body as one entity. Furthermore, because this method paints volumes directly, estimating the quantity of economic resources can be trivially accomplished. Assuming an interpreter has identified a volume of interest, directly painting the volume results in a fast computation of resource volumes.

References

Agrawala, M., Beers, A.C., & Levoy, M. 1995 (April). 3D painting on scanned surfaces. *Pages 145–150 of: Proceedings of the Symposium on Interactive 3D Graphics SI3D.*

Baxter, W.V., & Lin, M.C. 2004. A versatile interactive 3D brush model. *Proceedings of Pacific Graphics 2004*, 319–328.

Baxter, W.V., Scheib, V., Lin, M.C., & Manocha, D. 2001 (August). DAB: Interactive haptic painting with 3D virtual brushes. *Pages 461–468 of: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH.*

Delmarcelle, T., & Hesselink, L. 1993. Visualizing second-order tensor fields with hyper-streamlines. *IEEE Computer Graphics and Applications*, **13**, 25–33.

Engelsma, C., & Hale, D. 2010. Visualization of 3D tensor fields derived from seismic images. *CWP Report 655*, 187–192.

Fehmers, G.C., & Höcker, C.F.W. 2003. Fast structural interpretations with structure-oriented filtering. *Geophysics*, **68**, 1286–1293.

Fomel, S. 2008. Predictive painting of 3D seismic volumes. *SEG Expanded Abstracts 27*, 864–868.

Hale, D. 2009a. Image-guided blended neighbor interpolation. *CWP Report 634*, 247–260.

Hale, D. 2009b. Structure-oriented smoothing and semblance. *CWP Report 635*, 261–270.

Hanrahan, P., & Haeberli, P. 1990. Direct WYSIWYG painting and texture on 3D shapes. *SIGGRAPH Computer Graphics*, **24**(4), 215–223.

Jeong, W.K., Fletcher, P.T., Tao, R., & Whitaker, R.T. 2007. Interactive visualization of volumetric white matter connectivity in DT-MRI using a parallel-hardware Hamilton-Jacobi solver. *IEEE Transactions on Visualization and Computer Graphics*, **13**, 1480–1487.

Ju, T., Losasso, F., Schaefer, S., & Warren, J. 2002. Dual contouring of Hermite data. *Pages 339–346 of: Proceedings of the 29th annual conference on Computer Graphics and Interactive Technologies.*

Kadlec, B. 2009. *Interactive GPU-based "visulation" and structure analysis of 3-D implicit surfaces for seismic interpretation.* Ph.D. thesis, University of Colorado at Boulder.

Kindlmann, G. 2004. Superquadric tensor glyphs. *Pages 147–154 of: Proceedings of the Joint Eurographics - IEEE TCVG/EG Symposium on Visualization '04.*

Lorenson, W.E., & Cline, H.E. 1987. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, **21**, 163–169.

Strang, G. 2003. *Introduction to linear algebra.* Wellesley–Cambridge Press.

van Vliet, L.J., & Verbeek, P.W. 1995. Estimators for orientation and anisotropy in digitized images. *Pages 442–450 of: Proceedings of the First Annual Conference of the Advanced School for Computing and Imaging ASCI '95.*

Zheng, X., & Pang, A. 2002. Volume deformation for tensor visualization. *Pages 379–386 of: Proceedings of the conference on Visualization '02.*