



Mauricio Kreczmarsky Guimarães Meinicke

**Opacidade 3D na Visualização Volumétrica
de Dados Sísmicos**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Engenharia Mecânica da PUC-Rio.

Orientadores: Marcelo de Andrade Dreux
Pedro Mário Cruz e Silva

Rio de Janeiro, abril de 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



Mauricio Kreczmarsky Guimarães Meinicke

Opacidade 3D na Visualização Volumétrica de Dados Sísmicos

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Mecânica do Departamento de Engenharia Mecânica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marcelo de Andrade Dreux

Orientador

Departamento de Eng. Mecânica - PUC - Rio

Dr. Pedro Mário Cruz e Silva

Orientador

Tecgraf/PUC - Rio

Prof. Luiz Fernando Martha

Departamento de Eng. Civil - PUC - Rio

Prof. Marcelo Gattass

Departamento de Informática - PUC - Rio

Dr. Jurandyr Schmidt

Schmidt & Associados

Prof. José Eugenio Leal

Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 13 de abril de 2007

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Mauricio Kreczmarsky Guimarães Meinicke

Graduou-se em Bacharelado em Matemática pela Universidade do Estado do Rio de Janeiro (UERJ). Kursou o mestrado no departamento de engenharia mecânica da PUC-Rio. Durante o seu mestrado participou de projetos do laboratório Tecgraf, no departamento de informática.

Ficha Catalográfica

Meinicke, Mauricio Kreczmarsky Guimarães

Opacidade 3D na visualização volumétrica de dados sísmicos / Mauricio Kreczmarsky Guimarães Meinicke ; orientador: Marcelo de Andrade Dreux ; co-orientador: Pedro Mário Cruz e Silva. – 2007.

65 f. : il. ; 30 cm

Dissertação (Mestrado em Engenharia Mecânica)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

Inclui bibliografia

1. Engenharia mecânica – Teses. 2. computação gráfica. 3. Opacidade. 4. Visualização volumétrica. 5.

CDD: 621

A toda minha família.

Agradecimentos

Aos meus pais pelo afago nas horas difíceis e pelas broncas quando necessário. A minha avó pela educação dada, por rir comigo nos momentos de alegria e pelo amparo nos momentos de dificuldade. A todos os meus familiares que torceram pelo meu sucesso nos meus estudos. A minha namorada Regina Kazumi por estar sempre ao meu lado.

Aos meus orientadores Marcelo de Andrade Dreux e Pedro Mário Cruz e Silva pela paciência, pelo voto de confiança e por toda a dedicação posta neste trabalho.

Aos professores do departamento de engenharia mecânica da PUC-Rio por oferecerem todo o apoio acadêmico necessário. Aos amigos de pós-graduação do departamento.

A todos os amigos do Tecgraf - Puc-Rio que de alguma forma participaram da minha vida acadêmica ao longo do mestrado. Um agradecimento especial a todos da equipe NeuralSis-BR e do v3o2, e em especial, ao Sérgio Álvares Maffra pela ajuda e pelas sugestões sempre muito bem vindas.

Aos amigos da TG (Tecnologia em Geofísica) da Petrobras que me acolheram durante o desenvolvimento deste trabalho, e em especial a Mauren Ruthner pela ajuda na área de geofísica.

Ao CNPq e a PUC-Rio pelo apoio financeiro sem o qual este trabalho não poderia ser realizado. A Petrobras pela autorização da publicação dos dados contidos no trabalho.

Resumo

Meinicke, Mauricio K. G.; Dreux, Marcelo; Silva, Pedro Mário C. **Opacidade 3D na Visualização Volumétrica de Dados Sísmicos**. Rio de Janeiro, 2007. 65p. Dissertação de Mestrado - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho propõe uma técnica chamada de Opacidade 3D para visualização volumétrica de dados sísmicos. O grande desafio da visualização volumétrica é definir uma função de transferência multidimensional que melhor se adapte ao dado que se deseja visualizar. Será apresentada uma função de transferência que utiliza três tabelas de cores 1D para compor a uma tabela de cores 3D. O trabalho de Silva[30] sobre opacidade 2D serviu de motivação para o desenvolvimento da técnica de opacidade 3D e ao longo deste trabalho são feitas comparações entre ambos. São apresentados exemplos reproduzindo a opacidade 2D e outros mostrando como a técnica proposta pode auxiliar no estudo de determinados eventos sísmicos.

Palavras-chave

computação gráfica, opacidade, visualização volumétrica, função de transferência multidimensional, dados sísmicos

Abstract

Meinicke, Mauricio K. G.; Dreux, Marcelo; Silva, Pedro Mário C. **3D Opacity in Volume Rendering of Seismic Data**. Rio de Janeiro, 2007. 65p. M.Sc dissertation - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

This work proposes a 3D opacity technique for the volume rendering of seismic data. The greater challenge of volume rendering is to define a multi-dimensional transfer function better adapted to the data to be visualized. This work presents a transfer function that uses three 1D color tables to compose a 3D color table. The work from Silva[30] about 2D opacity has served as a motivation for the development of the 3D opacity technique and, hence, some comparisons are made between them. Some examples are presented in order to reproduce the 2D opacity technique and to show how the proposed technique can improve the visualization of specific seismic events.

Keywords

computer graphics, opacity, volume rendering, multi-dimensional transfer function, seismic data

Sumário

| | | |
|--------|--|----|
| 1 | Introdução | 12 |
| 1.1. | Trabalhos Relacionados | 13 |
| 1.2. | Estrutura da dissertação | 15 |
| 2 | Conceitos Básicos | 17 |
| 2.1. | Conceitos Básicos de Sísmica | 17 |
| 2.1.1. | Dados Sísmicos | 17 |
| 2.1.2. | Atributos Sísmicos | 18 |
| 2.1.3. | Decomposição Espectral | 19 |
| 2.2. | Conceitos Básicos de Computação Gráfica | 21 |
| 2.2.1. | <i>Fragment Program</i> | 21 |
| 2.2.2. | Tabela de Cores 1D (<i>Lut 1D</i>) | 23 |
| 2.2.3. | Visualização Volumétrica | 24 |
| 2.2.4. | Função de Transferência | 27 |
| 2.2.5. | Opacidade em Dados Sísmicos | 28 |
| 3 | Opacidade 3D | 35 |
| 3.1. | Teoria da Opacidade 3D | 35 |
| 3.2. | Implementação | 39 |
| 3.2.1. | Construção das Texturas | 40 |
| 3.2.2. | Consulta de Textura | 41 |
| 3.2.3. | Combinação das cores | 43 |
| 4 | Resultados | 47 |
| 4.1. | Exemplo 1 – Reprodução da Opacidade 2D | 47 |
| 4.2. | Exemplo 2 – Utilização da Opacidade 3D no Dado Sintético | 50 |
| 4.3. | Exemplo 3 – Utilização da Opacidade 3D em Dados Reais. | 52 |
| 4.4. | Exemplo 4 – Opacidade 3D Utilizando a Decomposição Espectral | 56 |
| 5 | Conclusões e Trabalhos Futuros | 60 |

| | |
|------------------------|----|
| 5.1. Conclusões | 60 |
| 5.2. Trabalhos Futuros | 61 |
| 6 Bibliografia | 63 |

Lista de figuras

| | |
|--|----|
| Figura 1 – Aquisição de um dado sísmico marítimo. Adaptado de [30] | 18 |
| Figura 2 – Exemplo de um espectro de amplitude | 19 |
| Figura 3 – <i>Pipeline</i> convencional do <i>OpenGL</i> . Adaptado de [7]. | 22 |
| Figura 4 – <i>Pipeline</i> programável do <i>OpenGL 2.0</i> . Adaptado de [7] | 23 |
| Figura 5 – Exemplo de uma tabela de cores 1D | 24 |
| Figura 6 – Construção de um volume utilizando os <i>voxels</i> . Adaptado de [17] | 25 |
| Figura 7 – Visualização Volumétrica de um dado sísmico. | 26 |
| Figura 8 – Tabela de cores utilizada na visualização volumétrica do dado da Figura 7 | 27 |
| Figura 9 – Evento sísmico sendo comparado com a sísmica a sua volta e a tabela de cores utilizada na visualização. | 29 |
| Figura 10 - Tabela de cores 1D | 30 |
| Figura 11 – Dado sintético Domo, reproduzido de [30] | 30 |
| Figura 12 – Histograma Amplitude x Fase Instantânea, reproduzido de [30] | 32 |
| Figura 13 – Domo utilizando a opacidade 2D, reproduzido de Silva[30]. | 33 |
| Figura 14 - Comparação entre a opacidade 1D (a) e a opacidade 2D (b) em dados reais | 34 |
| Figura 15 – Multiplicação do canal alfa na opacidade 2D. | 38 |
| Figura 16 – Construção da textura tridimensional RGB (opacidade 2D). | 41 |
| Figura 17 – Consulta (fetch) de textura até a cor. | 43 |
| Figura 18 – Interface de edição da tabela de cores 3D com soma de canais | 46 |
| Figura 19 – Reprodução da opacidade 2D utilizando a opacidade 3D | 48 |
| Figura 20 – Reprodução da tabela de cores utilizada na Opacidade 1D | 49 |
| Figura 21 – Reprodução do histograma da Figura 12 utilizado para gerar a Figura 13 | 49 |
| Figura 22 – (a) Opacidade 1D; (b) Opacidade 2D; (c) Opacidade 3D. | 51 |
| Figura 23 – Tabela de cores associada ao volume terciário. | 52 |
| Figura 24 – Resultado da opacidade 3D com dado real | 53 |
| Figura 25 – Reprodução da comparação entre a opacidade 1D (a) e a opacidade 2D (b) | 54 |

| | |
|---|----|
| Figura 26 – (a) Opacidade 1D; (b) Opacidade 2D; (c) Opacidade 3D | 55 |
| Figura 27 – Fatias no tempo. (a) Amplitude; (b) 10 Hz; (c) 18 Hz; (d) 26 Hz | 57 |
| Figura 28 – Uso da Opacidade 3D com a decomposição espectral | 58 |
| Figura 29 – Tabela de cores 3D que originou a Figura 28 | 59 |

1 Introdução

A indústria do petróleo tem se deparado, cada vez mais, com a necessidade de explorar e desenvolver campos de petróleo em ambientes geológicos complexos e reservatórios delgados. Além disso, o volume de dados sísmicos 3D a ser interpretado tem crescido exponencialmente. Em face desta realidade, a comunidade petrolífera tem buscado o desenvolvimento de novas tecnologias que gerem imagens mais fidedignas das subsuperfícies e, conseqüentemente, reduzam as incertezas nos modelos geológicos propostos.

A computação gráfica tem oferecido ferramentas que enriquecem a visualização de dados sísmicos e auxiliam o intérprete no estudo de um determinado volume sísmico. A visualização científica tem obtido grandes avanços com as novas *GPU's (Graphics Processing Units)* programáveis, o que tem permitido uma maior interação por parte do usuário.

O objetivo deste trabalho é fazer uma análise das técnicas de opacidade existentes e propor uma implementação desta técnica de visualização, que visa contribuir com a visualização volumétrica no estudo de multiatributos em dados sísmicos. Será implementada a Opacidade 3D utilizando técnicas de programação em *hardware* gráfico.

Ao longo do trabalho é descrita a teoria que serve de base para o desenvolvimento desta nova técnica, sempre que possível fazendo uma comparação com as técnicas similares existentes na literatura e com trabalhos em áreas correlatas.

Por se tratar de um assunto multidisciplinar, que envolve conceitos de geologia, geofísica e computação gráfica, faz-se necessária uma introdução a alguns conceitos básicos, a fim de permitir uma leitura deste documento por pessoas não totalmente familiarizadas com estas áreas.

A seguir são descritos os trabalhos relacionados, que de alguma forma influenciaram ou serviram de motivação para o desenvolvimento desta

dissertação. São apresentados também os demais capítulos que compõem esta dissertação.

1.1. Trabalhos Relacionados

As pesquisas de Silva[30] foram de grande importância no desenvolvimento deste trabalho. Não só serviram de base para este trabalho, como também de motivação para o estudo de toda a teoria que está por trás da opacidade 3D e da visualização volumétrica de um dado sísmico. Silva[30] aborda, dentre outros tópicos, a opacidade 2D e o problema de gerar uma função de transferência multidimensional de forma automática¹. Silva[30] desenvolveu seu trabalho no Laboratório Tecgraf/PUC-Rio, dando continuidade a uma série de trabalhos na área de visualização de dados sísmicos (Gerhardt[11, 12] e Paiva[24]). Silva[30] propõe uma função de transferência bidimensional que leva em consideração dois atributos sísmicos. O presente trabalho é uma extensão natural do trabalho desenvolvido por Silva[30].

Kindlmann[19] mostrou que é possível a geração de uma função de transferência de forma semi-automática para uma série de dados onde a região de interesse é limitada por materiais de valores relativamente constantes. Para gerar uma função de transferência bi-dimensional, Kindlmann[19] utilizou a derivada no sentido do gradiente. Gerhardt[11] e Gerhardt et al[12] verificaram que as técnicas de visualização volumétrica, como eram aplicadas, não se adequavam à visualização volumétrica de dados sísmicos. Silva[30] utilizou a fase instantânea, como segunda dimensão, para compor uma função de transferência bi-dimensional de forma semi-automática.

Durante o desenvolvimento deste trabalho, surgiu a oportunidade de testar a opacidade 3D em diversos tipos de dados sísmicos. Estes testes foram incentivados por trabalhos de caracterização de reservatórios. Trabalhos como [13, 15, 16, 21, 26] utilizam diferentes faixas de frequência para determinar características dos reservatórios que não são visíveis quando se utiliza todo o espectro, tais como a monitoração da saturação de fluidos (Korneev[21]), zonas de fratura do reservatório (Goloshubin[16]), dentre outras.

¹ O assunto será abordado com mais detalhes na seção 2.2.4

O trabalho de Partyka[26] utiliza a decomposição espectral² e a transformada discreta de Fourier para mapear camadas de pouca espessura e identificar descontinuidades geológicas sobre uma superfície sísmica 3D. Em seu trabalho, Partyka[26] utiliza, como exemplo, a decomposição espectral de uma imagem do Golfo do México. Partyka[26] utilizou as frequências de 16-Hz e 26-Hz sobre o mapa de energia e sobre a fase do dado sísmico. Este trabalho consegue mapear finas camadas geológicas através da transformada de Fourier utilizando uma pequena janela de amplitude e fase espectral. Isto permite ao intérprete enriquecer o seu modelo no que diz respeito à quantificação relativa da espessura das camadas geológicas finas e a descontinuidades laterais das camadas.

Rutner[28] fez um estudo detalhado sobre o trabalho de Partyka[26] e constatou que a técnica apresentada permite que seja identificado um número maior de eventos sísmicos de interesse. Muitos destes eventos não foram identificados no volume sísmico após o processamento convencional. Rutner[28] sugere que a técnica seja utilizada, principalmente, para refinar o modelo geológico.

Seguindo a linha de estudo de reservatórios utilizando a baixa frequência, pode-se citar o trabalho de Goloshubin[15] que incentiva o uso da decomposição espectral para o detalhamento das imagens de reservatórios. Goloshubin[15] afirma que o uso das baixas frequências tem grande potencial para mapear áreas de grande permeabilidade.

O trabalho apresentado por Bordignon et al [5] descreve uma ferramenta de construção de uma função de transferência multidimensional, utilizando coordenadas estelares, para identificar *clusters* a serem visualizados. A interface desenvolvida possui apenas três parâmetros que são o eixo de rotação, a escala e a pintura. O trabalho de Bordignon et al[5] difere do presente trabalho pelo fato de poder ser utilizado para vários atributos, simultaneamente, não sendo limitado a três atributos. Entretanto, ao disponibilizar o uso de vários atributos, Bordignon et al[5] não possui suporte dos *hardwares* gráficos atuais, ou seja, todas as operações são feitas em *CPU (Central Processing Unit)*. Dependendo do tamanho do dado que se queira visualizar e do número de variáveis, o usuário não tem uma boa interatividade com o *software*. A interatividade é de extrema importância para

² A decomposição espectral será apresentada na seção 2.1.3

a opacidade 3D, uma vez que todo o processo de utilização da opacidade está voltado para a interação com o usuário.

Este trabalho apresenta uma “junção” entre as três linhas de pesquisas apresentadas anteriormente (Silva[30], Ruthner[28] e Goloshubin[13]), mostrando a utilização da decomposição espectral junto com a técnica de opacidade 3D aplicadas a visualização de dados sísmicos.

A técnica da opacidade 3D foi implementada, em caráter experimental, em um *software* desenvolvido pelo Tecgraf/PUC-Rio em parceria com a Petrobras. Este *software* é destinado a visualização e controle de qualidade de dados sísmicos tridimensionais, denominado *v3o2* (*Visualizador 3D, Opacidade 2D*). A maioria das figuras geradas neste trabalho é proveniente da captura de tela do *v3o2*.

1.2. Estrutura da dissertação

Esta dissertação está dividida em cinco capítulos. O Capítulo 2 descreve alguns conceitos básicos que foram necessários para o desenvolvimento teórico e aplicado do presente trabalho. Este capítulo, por sua vez, está dividido em duas partes: a primeira referente aos principais conceitos de sísmica e a segunda parte aborda os conceitos de computação gráfica. Dentre os conceitos de computação gráfica serão apresentadas as opacidades 1D e 2D, que formam a base teórica principal para o desenvolvimento deste trabalho.

Toda a teoria desenvolvida neste trabalho encontra-se no Capítulo 3, onde a idéia inovadora de opacidade 3D é descrita em detalhes, aprofundando alguns conceitos apresentados no Capítulo 2.

O Capítulo 4 descreve os testes e apresenta os resultados obtidos durante o desenvolvimento deste trabalho. Foram realizados testes com um dado sísmico sintético e em seguida com um dado real. Sempre que possível, são comparados com resultados obtidos com opacidades 1D e 2D.

No Capítulo 5, Conclusão, resume-se o que foi desenvolvido, descreve-se o que a opacidade 3D realmente traz de benefício para a visualização volumétrica de um dado sísmico e destaca-se em que tipo de estudo esta técnica deve ser

utilizada. São apresentadas algumas sugestões para trabalhos de pesquisa a serem ainda realizados.

2

Conceitos Básicos

Por ser um trabalho que engloba diferentes áreas de conhecimento, foi necessário fazer uma descrição de alguns termos de áreas como a geologia, geofísica e computação gráfica, que estão diretamente relacionados ao tema desta dissertação.

Este capítulo se divide em duas partes. A primeira parte tem por objetivo apresentar ao leitor alguns termos da área de Sísmica. A segunda parte descreve os principais conceitos de computação gráfica que foram utilizados neste trabalho.

2.1.

Conceitos Básicos de Sísmica

2.1.1.

Dados Sísmicos

A sísmica de reflexão é um método de exploração do subsolo amplamente utilizado por ser capaz de cobrir grandes áreas com um baixo custo se comparado aos métodos diretos, como por exemplo, a perfuração de poços.

A exploração de hidrocarbonetos, óleo e gás baseado em sísmica pode ser dividida em três etapas, que são aquisição, processamento e interpretação.

A aquisição é feita usando uma fonte para gerar ondas sísmicas que se propagam abaixo da superfície da terra. Em aquisições terrestres é comum usar explosões de dinamite como fonte e em aquisições marinhas são usados normalmente dispositivos pneumáticos como canhões de ar, como mostra a Figura 1.

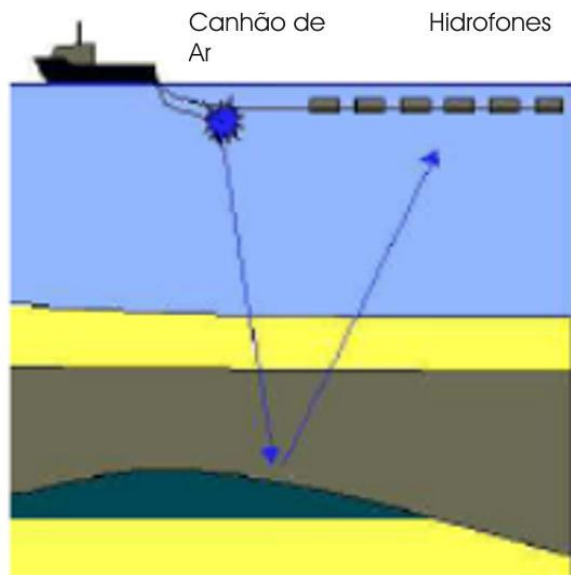


Figura 1 – Aquisição de um dado sísmico marítimo. Adaptado de [30]

Quando a onda sísmica alcança uma interface entre duas camadas de rocha com valores de impedância acústica diferentes, parte da onda é refratada, e continua viajando para baixo; enquanto outra parte é refletida e retorna à superfície. A parte refletida da onda que retorna à superfície é captada nos receptores e gravada nos sismógrafos.

O sismógrafo armazena tanto o tempo de chegada da onda quanto a intensidade medida neste momento. Após várias detonações variando a posição da fonte e dos receptores, todos os dados armazenados são enviados para serem processados.

Nesta etapa os dados obtidos na aquisição são tratados, ou seja, os ruídos são identificados e atenuados, os eventos são deslocados para a sua posição correta e são corrigidas perdas de energia. Os dados são organizados na forma de um *grid volumétrico* (ver seção 2.2.3) e o dado adquirido transforma-se realmente em um volume de dados sísmicos.

2.1.2. Atributos Sísmicos

Atributos sísmicos são todas informações obtidas a partir de um dado sísmico, seja através de medidas diretas, por lógica ou experimentalmente. Existem vários tipos de atributos, cada um com sua funcionalidade. Existe um

atributo que melhor identifica um determinado material geofísico, outro para identificar discontinuidades sísmicas e assim por diante.

O principal objetivo dos atributos sísmicos é ajudar o intérprete a gerar um modelo geológico de uma determinada área sísmica. Quanto mais detalhado for o modelo geológico feito pelo intérprete maior será o índice de sucesso na exploração da área em questão.

2.1.3. Decomposição Espectral

Os reservatórios naturais de petróleo tendem a se estender por centenas de quilômetros, mas apresentam espessuras bastante delgadas, da ordem de dezenas de metros, que se adelgaçam nas bordas até se extinguir. Estes afinamentos não são bem identificados em dados sísmicos submetidos a processamentos convencionais. Isto se deve, principalmente, à interferência das reflexões do topo e da base das camadas que ocorre à medida que as espessuras vão se tornando menores. Partyka[26] propôs uma nova técnica que investiga estas espessuras, utilizando a decomposição espectral dos dados sísmicos, feita através da transformada de Fourier.

O conceito envolvido neste método é o de que a reflexão de uma única camada fina tem uma expressão característica no domínio da frequência que é indicativa de sua espessura temporal. Por exemplo, considerando o modelo ideal de uma única camada fina e homogênea, com topo e base representados por reflexões de mesma intensidade e polaridades invertidas, a camada introduzirá uma seqüência periódica previsível de zeros no espectro de amplitude de sua reflexão composta resultante, conforme é mostrado na Figura 2.

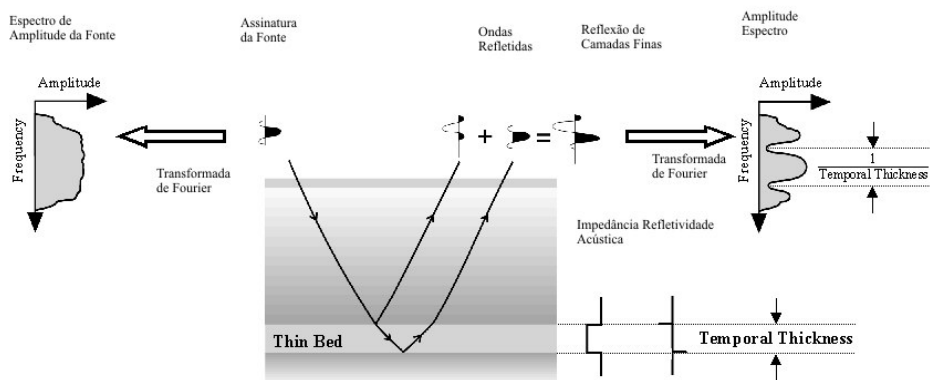


Figura 2 – Exemplo de um espectro de amplitude

Partyka[26] verificou que a frequência em que ocorrem os zeros no espectro ou a frequência em que ocorre o primeiro zero no espectro é inversamente proporcional à espessura temporal da camada. Conhecendo-se a litologia, ou ainda melhor, a velocidade da rocha, é possível estimar-se a espessura em metros da camada, dado importante no dimensionamento das acumulações, inclusive para o posicionamento de novos poços. Outro produto da decomposição espectral, o espectro de fase, traz informações de descontinuidades laterais, mas não foi abordado neste trabalho.

Na prática, a onda sísmica atravessa múltiplas camadas e a reflexão resultante torna-se bastante complexa com um registro único no espectro de amplitudes. Partyka[26] verificou que, em janelas de análise longas, uma sucessão de estratos geológicos apresenta estatisticamente um comportamento aleatório, que pode ser aproximado para um espectro plano. Já em janelas de análise curtas, a geologia deixa de apresentar um comportamento aleatório e fica então preservado no espectro o padrão de interferência de topo e base das camadas, na forma de zeros ou mínimos. Um exemplo de janela de análise longa seria considerar o traço sísmico como uma única janela e de janela curta, um intervalo de 0,100s.

Para o cálculo do espectro de amplitudes Partyka[26] utilizou a transformada de Fourier com uma janela constante. Ruthner[28] propôs em sua pesquisa a utilização da transformada S, que trabalha com janelas variáveis adequadas para cada frequência que está sendo analisada, obtendo resultados bastante satisfatórios, teoricamente com uma maior resolução e sem perder a amarração com o eixo z, dos tempos. Os dados utilizados neste trabalho foram gerados com a transformada S.

A análise e a interpretação dos espectros de amplitude são feitas visualmente e uma das formas propostas por Partyka[26] é a de cubos de frequência constante. O espectro é calculado amostra a amostra com uma janela de análise deslizante, constante para a transformada de Fourier janelada e variável, para a transformada S, e a seguir, os dados são reorganizados por frequência comum. Estes volumes são então comparados entre si. As informações de cada volume de frequência são complementares, pois cada frequência mostra melhor uma determinada espessura de camada. Outra forma de visualização sugerida é com a estrutura RGB. Cada cor recebe um volume de frequência, trabalha-se com

três frequências de cada vez. As cores puras relacionam-se diretamente com as três frequências escolhidas e as cores intermediárias, geradas das misturas das cores puras, relacionam-se a frequências intermediárias e conseqüentemente a espessuras intermediárias.

2.2. Conceitos Básicos de Computação Gráfica

2.2.1. *Fragment Program*

Com o desenvolvimento da tecnologia, surgiram várias empresas fabricantes de *hardware* gráfico. No passado cada fabricante possuía uma “linguagem” própria que deveria ser utilizada para passar as informações para o *hardware*. Devido a isto, era muito difícil para o programador conseguir criar uma aplicação que fosse independente do *hardware*. Com isso surgiu a necessidade de gerar uma padronização.

O *Open Graphics Library (OpenGL)*[23] é uma sofisticada API (*Application Programming Interface*) que contém uma série de rotinas gráficas portáteis e extremamente rápidas. No *OpenGL* é possível gerar imagens 2D e 3D com um grande grau de realismo. Esta biblioteca incorpora vários algoritmos otimizados, incluindo o desenho e manipulação de objetos gráficos 3D.

Em 1992, surgiu um consórcio, independente, formado pelos principais fabricantes de *hardware* gráfico, entre eles a 3Dlabs[1], Apple Computer[2], NVidia[22], SGI[29] e a Sun[31], chamado de *Architecture Review Board (ARB)*. O *ARB* tem como finalidade aprovar as novas funcionalidades, versões e extensões do *OpenGL*. Em 2007 o *OpenGL* encontrava-se na versão 2.0.

Comumente é utilizada a palavra *pipeline* para descrever o processo de geração de uma imagem. O *pipeline* convencional do *OpenGL* é mostrado na Figura 3.

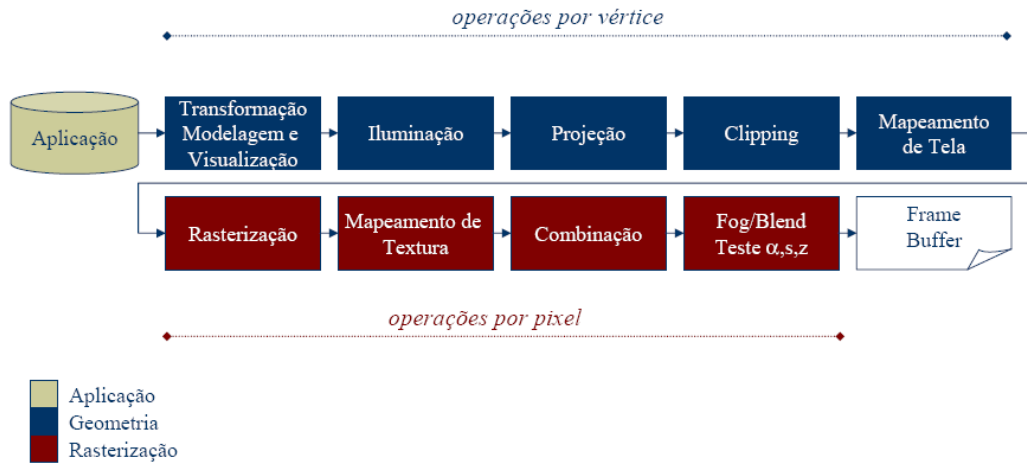


Figura 3 – Pipeline convencional do OpenGL. Adaptado de[7].

Com o avanço dos *hardwares* gráficos surgiu a opção das *GPU's* que são unidades de processamento gráfico instalados no *hardware* gráfico. Para fazer uso dessas unidades de processamento, o *pipeline* original do OpenGL deve ser modificado.

Existem dois tipos de unidades programáveis: a *vertex processor* que é a unidade processadora que faz as operações sobre os vértices dos objetos que estão sendo desenhados. As operações suportadas pelo processador de vértices são as transformações de vértices, transformações das normais, normalização, geração de coordenadas de textura, transformações de coordenadas de textura, iluminação e a aplicação de cores a materiais.

A outra unidade programável é o *fragment processor* que é responsável por fazer as operações sobre fragmento. O fragmento é equivalente ao *pixel* da imagem que será desenhado mas ele se encontra no *z-buffer*. As principais operações feitas pelo processador de fragmento são operações e interpolações com valores, acesso a texturas, aplicação de textura, neblina e operações com cores (soma).

O *pipeline* programável é apresentado na Figura 4.

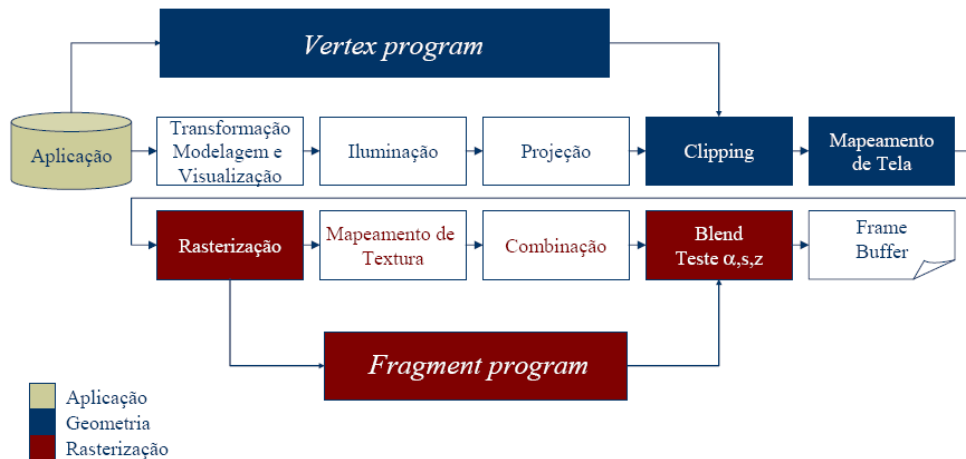


Figura 4 – Pipeline programável do OpenGL 2.0. Adaptado de [7]

As etapas de Mapeamento de Textura e Combinção, no pipeline programável, podem ser efetuadas por um *fragment program*, na GPU, reduzindo sobremaneira o tempo de processamento.

2.2.2. Tabela de Cores 1D (*Lut 1D*)

A tabela de cores, também chamada de *LUT*, acrônimo do termo inglês “*Lookup table*”, é uma estrutura de dados usada para substituir cálculos computacionais por acesso à memória. A vantagem desse método é que o acesso à memória é muito mais rápido que um cálculo repetitivo.

A chamada tabela de cores 1D é composta de quatro vetores, todos do mesmo tamanho, que são: um vetor com a intensidade da cor vermelha (canal R da cor), um vetor para a cor verde (canal G) e um para a cor azul (canal B), todos variando de zero a duzentos e cinquenta e cinco. O quarto vetor é o que armazena a transparência. O fator de transparência, que é chamado de canal alfa da cor resultante, varia de zero a um, sendo zero quando o objeto é totalmente translúcido e um para o objeto totalmente opaco.

A tabela de cores, que é utilizada neste trabalho, é armazenada como uma textura do *OpenGL* e possui uma limitação de duzentas e cinquenta e seis cores diferentes para compor a tabela.

A Figura 5 mostra uma tabela de cores 1D onde as cores foram compostas de forma simétrica. A curva preta que aparece acima da tabela de cores mostra a

transparência da cor naquele ponto. Quando a curva está embaixo o valor do canal alfa é zero e quando a curva está no ponto mais alto, seu valor é um.

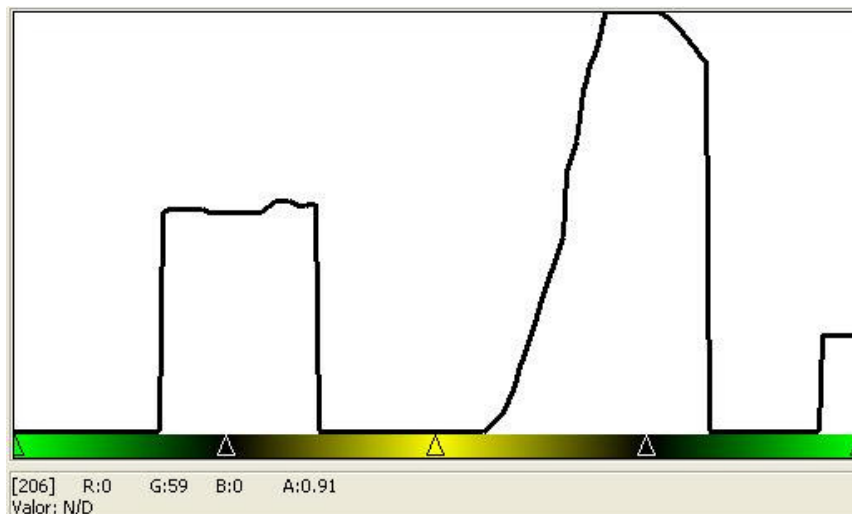


Figura 5 – Exemplo de uma tabela de cores 1D

A tabela de cores da Figura 5, apresenta triângulos na parte inferior, onde encontram-se as cores da tabela. Estes triângulos são chamados de *tags*. Na implementação do *v3o2*, o usuário pode construir a sua própria tabela de cores, bastando posicionar estes *tags* na posição desejada e escolhendo qual será a cor neste ponto. As cores dadas no intervalo entre um *tag* e outro são calculadas através da interpolação linear dos dois *tags* que limitam o intervalo.

Para otimizar o desempenho das tabelas de cores, pode-se criar, no *OpenGL*, uma textura do tipo 1D, e mapear a textura para os valores da tabela de cores.

2.2.3. Visualização Volumétrica

O termo visualização volumétrica descreve um conjunto de técnicas utilizadas para visualizar um volume de dados tridimensional. Estes dados podem ser adquiridos de diferentes fontes, tais como tomografia computadorizada, dinâmica computacional dos fluidos, dados sísmicos ou qualquer outro tipo de dado tridimensional que possa ser representado por um campo de escalares. A visualização volumétrica também é muito utilizada em dados sintéticos para simular explosões e outros efeitos em jogos 3D.

Um dado volumétrico pode ser definido como uma função matemática de domínio tridimensional, representado por um campo escalar discreto e

contradomínio unidimensional, ou seja, $f(\vec{x}) \in \mathbb{R} \therefore \vec{x} \in \mathbb{R}^3$. No contexto da visualização volumétrica, os valores do dado são armazenados na forma de uma matriz tridimensional e o acesso é feito por meio da posição do dado na matriz.

Como o domínio do dado é discreto não se tem um volume completamente preenchido, ou seja, há “buracos” ao longo do mesmo. Em um domínio contínuo os gastos com recurso de *hardware* para a visualização do dado seriam demasiadamente dispendiosos.

Ao conjunto dos pontos do domínio que contêm informações dá-se o nome de *grid volumétrico*. Para visualizar o dado sem os “buracos” criou-se uma unidade de medida chamada de *voxel* que é a menor unidade do domínio que possui informação do dado de entrada. A cada *voxel* está associado um valor que é a informação do dado de entrada no ponto do *grid volumétrico* que o *voxel* contém. Os *voxels* estão sempre alinhados com o *grid volumétrico* e as posições relativas dos *voxels*, em geral, são dadas tendo os pontos do *grid* como sendo o centro do *voxel* ou um de seus vértices.

A Figura 6 mostra um volume sendo criado a partir de um conjunto de *voxels*.

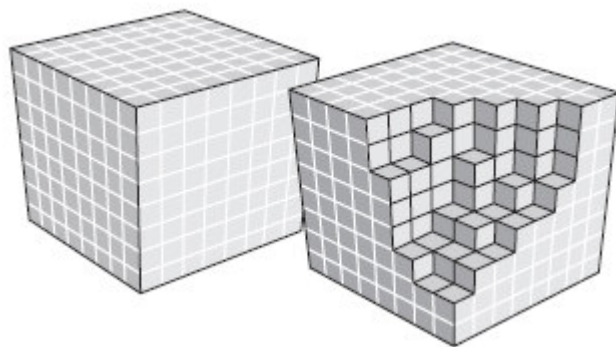


Figura 6 – Construção de um volume utilizando os *voxels*. Adaptado de [17]

Dados sísmicos são usualmente armazenados em um *grid volumétrico* regular, ou seja os pontos do domínio estão igualmente espaçados.

2.2.3.1. Visualização Volumétrica Direta

A visualização volumétrica direta constitui-se na técnica de criar uma imagem, em geral uma textura, para representar um intervalo de valores do dado volumétrico, sem se preocupar realmente com o valor do dado no *voxel*. É criado um modelo que descreve como o dado volumétrico será representado.

O modelo mais utilizado é o que faz a correspondência dos valores escalares a um conjunto de cores que deverá representá-lo. Neste modelo pode-se entender a importância da tabela de cores descrita em 2.2.2.

Um modelo usualmente utilizado é o que associa o menor valor do dado volumétrico com a cor que está na primeira posição da tabela de cores. O maior valor do dado volumétrico deve ser associado à cor que encontra-se na última posição da tabela de cores. Os valores intermediários do dado volumétrico são associados às posições intermediárias da tabela de cores.

A Figura 7 mostra a visualização volumétrica de um dado sísmico real, onde o intervalo de amostragem do dado é de cinquenta metros no eixo *in-line* (eixo *x*), vinte e cinco metros no eixo *cross-line* (eixo *y*) e de quatro milissegundos no tempo (eixo *z*).

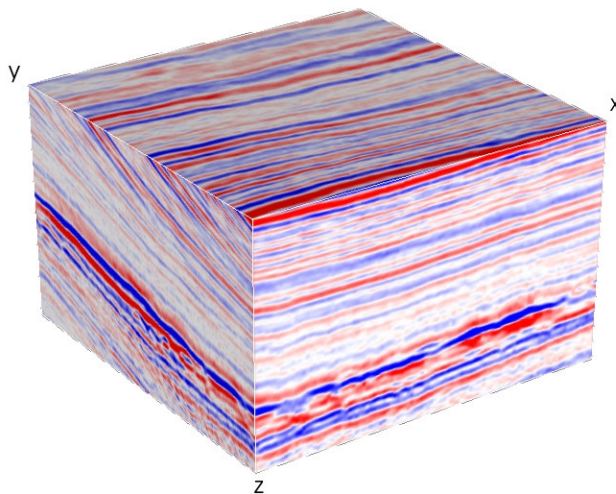


Figura 7 – Visualização Volumétrica de um dado sísmico.

O dado sísmico da Figura 7 possui a amplitude como atributo sísmico. O intervalo de valores encontrados neste dado varia de -11,89 a 10,81. A tabela de cores utilizada neste dado varia do azul (representando amplitudes mais baixas) ao vermelho (altas amplitudes). A Figura 8 mostra a tabela de cores que foi utilizada

na visualização volumétrica do dado da Figura 7. A função que mapeia o valor de opacidade dos *voxels* é a linha na parte superior da figura, bem próxima da borda.

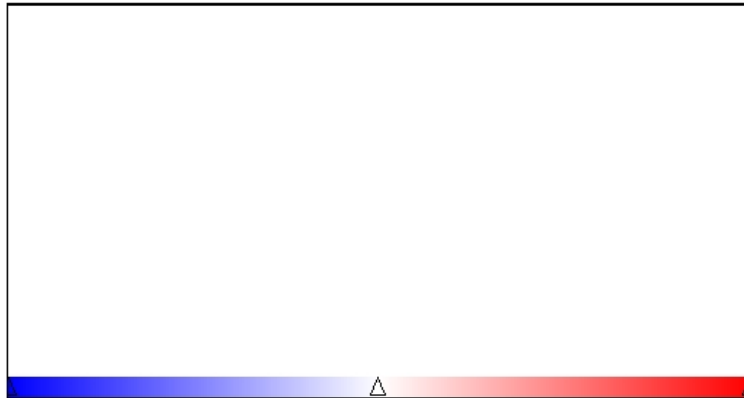


Figura 8 – Tabela de cores utilizada na visualização volumétrica do dado da Figura 7

2.2.4. Função de Transferência

A função de transferência é definida como sendo uma função matemática que transforma um sinal de entrada em um sinal de saída. A visualização volumétrica utiliza a função de transferência no dado sísmico quando faz a associação do valor escalar à posição da tabela de cores.

A transparência dos objetos também é definida por uma função de transferência. A mesma função de transferência pode ser utilizada para definir a cor de saída do objeto e a sua transparência, ou podem-se ter duas funções de transferência, uma para a cor e outra para a transparência.

2.2.4.1. Função de Transferência Multidimensional

A função de transferência multidimensional tem o mesmo objetivo da descrita anteriormente. A diferença é que neste caso leva-se em conta mais de um sinal de entrada para um único sinal de saída.

No capítulo 2.2.5.2 mostra-se que a função de transferência multidimensional tem grande valor na visualização volumétrica ao tentar combinar atributos sísmicos diferentes.

O maior desafio na visualização volumétrica é definir uma função de transferência multidimensional automática, onde não seja necessário um grande

conhecimento do usuário sobre este assunto para poder definir o sinal de saída. Na seção 3.1 será apresentada uma função de transferência multidimensional, que serviu de base para o desenvolvimento deste trabalho.

2.2.5. Opacidade em Dados Sísmicos

A opacidade é definida como sendo uma propriedade óptica da matéria e um material é considerado “opaco” quando não permite a passagem da luz em proporções apreciáveis.

A visualização volumétrica faz uso da opacidade para poder mostrar áreas que ficam escondidas durante a reconstrução do dado volumétrico. Este processo é muito utilizado na medicina, quando se faz uma tomografia computadorizada, por exemplo.

Neste trabalho a opacidade será explicada utilizando a visualização de dados sísmicos, que foi o principal objetivo do trabalho. Na visualização volumétrica de um dado sísmico, muitas vezes o intérprete tem a necessidade de visualizar apenas uma região do dado sísmico que se encontra na parte interna do volume.

Pelas características de aquisição do dado sísmico, a visualização volumétrica mostra ser uma importante ferramenta para seu estudo, uma vez que o dado sísmico está discretizado em uma malha regular, onde existe informação em apenas alguns pontos do domínio. Através da utilização da opacidade têm-se obtidos bons resultados no estudo de eventos sísmicos específicos.

Atualmente, o v3o2 disponibiliza dois tipos de opacidade para o usuário: a Opacidade 1D e a Opacidade 2D. A seguir será descrito com mais detalhes o que cada uma delas representa. No próximo capítulo será apresentada a Opacidade 3D que é o objetivo deste trabalho.

2.2.5.1. Opacidade 1D

Em certas ocasiões o geofísico precisa filtrar os atributos sísmicos de forma a realçar determinadas feições geológicas e características dos reservatórios. Por exemplo, as visualizações de determinadas faixas de frequência ou de amplitude podem realçar as estruturas geológicas, ambientes deposicionais, etc. Essa

filtragem é muito mais efetiva quando realizada interativamente, o que traz dinamismo à interpretação dos dados.

A Figura 9 mostra um evento sísmico sendo comparado com uma fatia sísmica (do mesmo atributo). Ao deslocar a fatia pelo volume sísmico é possível verificar de que forma o evento específico acompanha a sísmica a sua volta. Ao mover a fatia sísmica pode-se notar que o horizonte destacado pela seta segue a geologia da fatia.

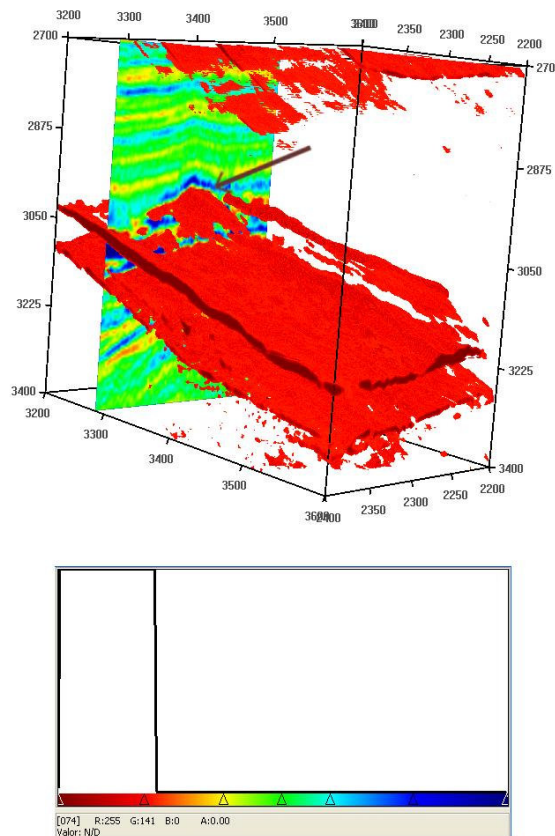


Figura 9 – Evento sísmico sendo comparado com a sísmica a sua volta e a tabela de cores utilizada na visualização.

A Figura 10 representa uma tabela de cores largamente utilizada na visualização de dados sísmicos. A função de transferência na opacidade 1D é extremamente simples. Os valores de opacidade determinados pelo usuário na tabela são associados ao canal alfa da tabela de cores.

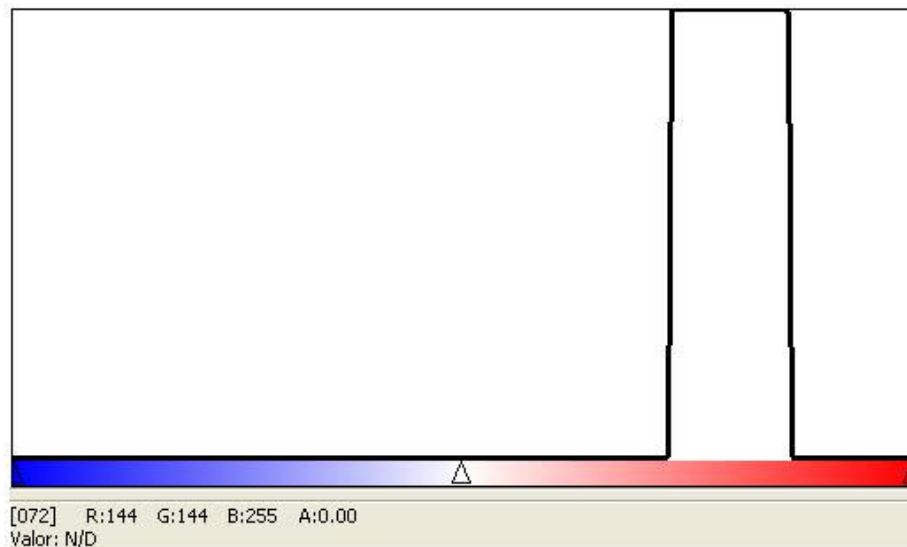


Figura 10 - Tabela de cores 1D

A Figura 11 mostra uma visualização volumétrica de um atributo sísmico utilizando uma tabela de cores 1D onde apenas uma área da tabela de cores está marcada como visível (Figura 10).

Este dado é um dado sintético chamado de Domo, que foi obtido tomando-se um traço sísmico de um dado real e replicando este traço lateralmente para gerar o dado 3D. Ao replicar o traço lateralmente foram feitos deslocamentos na direção vertical para curvar os eventos em forma de domo. Este dado foi cedido por Silva[30] e é o mesmo dado utilizado por Gerhardt et al[12]

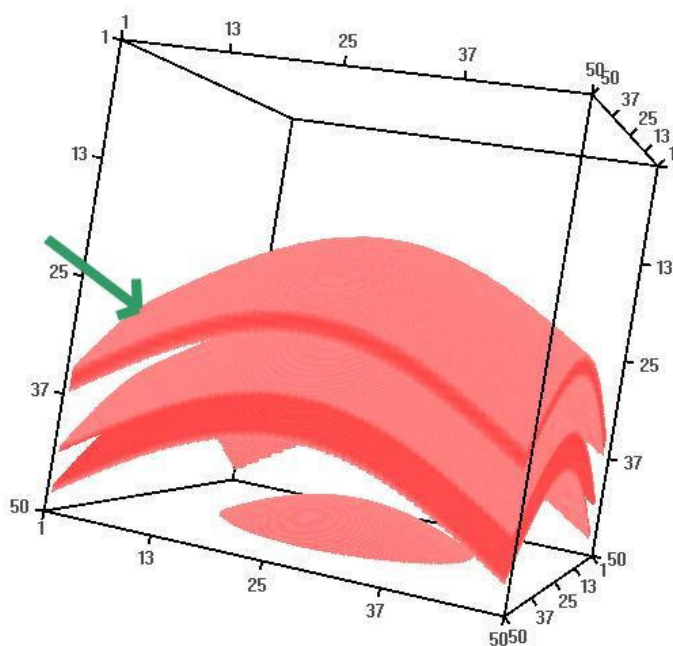


Figura 11 – Dado sintético Domo, reproduzido de [30]

O atributo utilizado no domo é a amplitude. Na Figura 11, temos a visualização de eventos de amplitude alta. A área de interesse do usuário está destacada com a seta verde, porém, devido às características do dado sísmico, não foi possível isolar completamente o evento. Ocorreu o chamado efeito de borda, que é o aparecimento de outros horizontes sísmicos com valores de amplitudes iguais ou bem próximos do valor selecionado pelo usuário, porém em datas geológicas diferentes.

2.2.5.2. Opacidade 2D

A opacidade 2D combina dois atributos sísmicos para tentar isolar o efeito de borda. De acordo com o tipo de evento que esteja sendo observado, existe um tipo de atributo que melhor auxilia no estudo em questão.

Silva[30] listou alguns atributos que, quando combinados, obtêm melhores resultados para uma determinada situação, como por exemplo, qual o melhor atributo para destacar eventos de maior amplitude, eventos contínuos, entre outros.

Na opacidade 1D a função de transferência tem por objetivo apenas verificar na tabela de cores qual o nível de transparência de um objeto, ou seja, a função de transferência tem apenas uma dimensão.

Já na opacidade 2D, é necessária uma função de transferência multidimensional, pois é preciso saber qual o valor da opacidade atribuído a cada atributo para compor a visualização final.

Silva[30] definiu um método interativo para compor a função de opacidade 2D, onde o usuário define a função de opacidade com o auxílio de um histograma que correlaciona os dois atributos sísmicos que estão sendo combinados. O histograma utilizado nesta função de transferência é composto tendo um atributo sísmico no eixo horizontal (a este eixo tem-se uma textura do tipo *OpenGL* associada que é a que vai ser utilizada na visualização do dado) e no eixo vertical o outro atributo que deseja-se combinar.

Para a visualização volumétrica define-se cada atributo sísmico como sendo um volume sísmico. A estes volumes sísmicos foram atribuídos os nomes de volume primário e volume secundário. Estes nomes não têm relação com a

importância do volume, mas foram usados para manter consistência com o trabalho de Silva[30]. Os nomes estão relacionados com o eixo ao qual o volume será associado na composição da visualização. O volume primário será sempre associado ao eixo horizontal e o volume secundário ao eixo vertical.

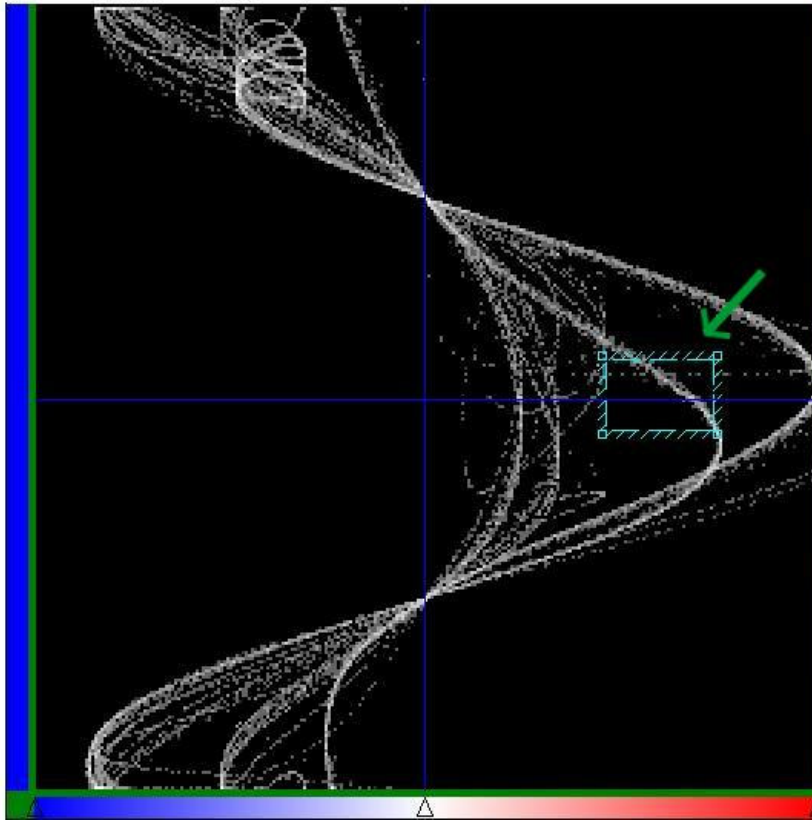


Figura 12 – Histograma Amplitude x Fase Instantânea, reproduzido de [30]

A Figura 12 mostra um histograma onde no eixo horizontal está o atributo sísmico amplitude do dado Domo e no eixo vertical encontra-se o atributo fase instantânea.

Num ponto (x, y) do histograma representa-se a frequência com que o atributo sísmico definido no eixo horizontal aparece com valor x e o atributo do eixo vertical aparece com valor y . Quanto mais branca for a cor que representa o ponto (x, y) maior é a frequência com que o par ordenado aparece e quanto mais preta for a cor que representa o par, menor a frequência deste.

Selecionados os atributos e montado o histograma, o usuário pode escolher a região do dado sísmico que há interesse em estudar. Ao selecionar uma área do histograma, a função de transferência pode atribuir opacidade zero para todos os *voxels* que estiverem fora da área selecionada e opacidade um para todos os pontos que estiverem dentro.

Na Figura 12 foi selecionada uma área retangular, indicada por uma seta. Os pontos que estão dentro do retângulo destacado são os pontos visíveis que geraram a visualização volumétrica da Figura 13.

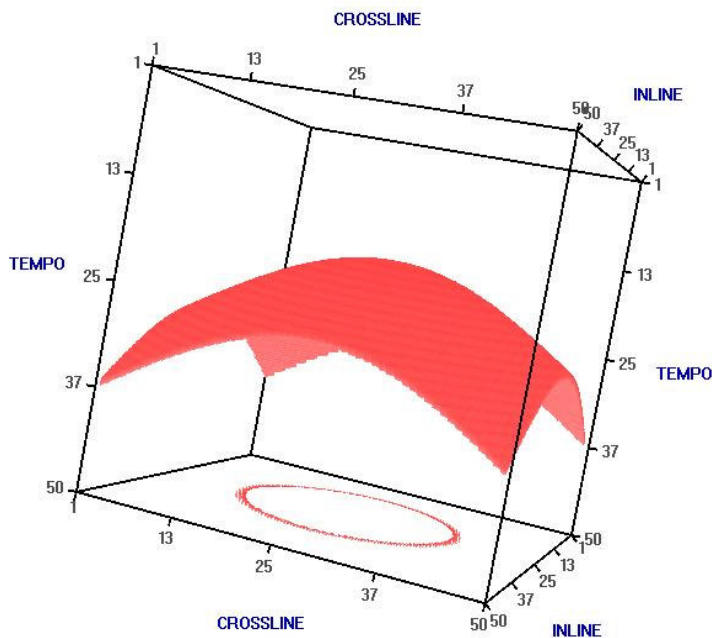


Figura 13 – Domo utilizando a opacidade 2D, reproduzido de Silva[30].

Pode-se notar que a opacidade 2D conseguiu eliminar os efeitos de borda. Na Figura 13 tem-se visível apenas o horizonte que foi destacado na Figura 11.

Os testes feitos por Silva[30] e Gerhardt[12] mostram que a opacidade 2D é muito eficaz para dados sintéticos, porém ao tentar isolar eventos de dados reais muitas vezes essa técnica não é capaz de isolá-los totalmente. Silva[30] deixa em aberto como trabalhos futuros o problema de otimização do algoritmo de visualização de multiatributos. Isto serviu de motivação para o desenvolvimento deste trabalho, que ao utilizar três atributos sísmicos para gerar a visualização volumétrica possibilitou uma melhor filtragem do dado sísmico e que junto com as novas tecnologias utilizadas pelas placas de vídeo programáveis foi possível desenvolver um algoritmo de combinação que proporciona ao usuário final uma grande interação com o programa, sendo feita à visualização volumétrica praticamente em tempo real.

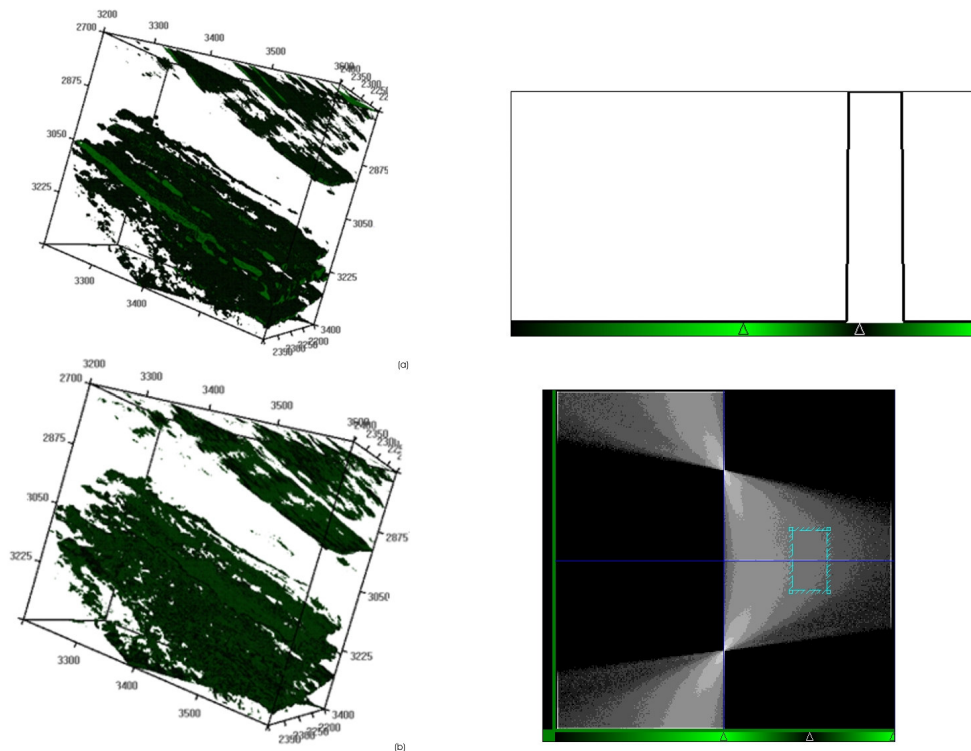


Figura 14 - Comparação entre a opacidade 1D (a) e a opacidade 2D (b) em dados reais

A Figura 14 mostra a visualização volumétrica de um dado sísmico real. No volume (a) tem-se uma faixa de amplitude visível, que envolve altos valores de amplitude. No volume (b) tem-se a mesma área de amplitude visível, porém utilizando a fase instantânea como segundo atributo, e limitando a área deste atributo como sendo um pequeno intervalo em torno dos valores de fase zero. Pode-se notar que foi possível eliminar alguns ruídos da visualização.

Com a opacidade 3D pretende-se melhorar ainda mais a filtragem do dado sísmico, fazendo com que o usuário consiga isolar com mais detalhe o evento de interesse. No capítulo seguinte, será apresentada a teoria desenvolvida sobre a opacidade 3D.

3 Opacidade 3D

Existem trabalhos na literatura que falam sobre a função de transferência multidimensional [5, 12, 19, 20, 25] e em todos a grande dificuldade é encontrar uma função de transferência multidimensional que seja adequada para o tipo de dado que esteja sendo visualizado. O problema de se trabalhar com um objeto de dimensão maior que dois está na dificuldade em se criar uma interface com o usuário que é quase tão complicado que editar diretamente o dado. Com o aumento da dimensão aumenta-se a flexibilidade, porém aumenta-se também a dificuldade do usuário definir uma função de transferência que melhor se adeque ao dado que se deseja visualizar.

A interface com o usuário é extremamente importante, pois é esta que vai conduzir o usuário durante o processo de ajuste da função de transferência e, conseqüentemente, na geração da visualização final. Caso o usuário tenha dificuldades em ajustar a função de transferência, possivelmente não terá controle sobre o que está sendo visualizado.

3.1. Teoria da Opacidade 3D

Neste trabalho, quando se trata de uma função de transferência (Opacidade 3D), trata-se de um objeto que tem a mesma dimensão do volume que se quer inspecionar. Desta forma, a tarefa de especificar uma função de Opacidade 3D com toda a sua liberdade pode ser tão difícil quanto manipular diretamente o volume de interesse. Neste trabalho sugere-se uma simplificação na especificação da Opacidade 3D e propõe-se restringir às funções de transferência 3D que podem ser obtidas através de uma combinação de três funções de transferência 1D. A opacidade 3D, desenvolvida neste trabalho, utiliza três tabelas de cores 1D para compor a chamada *lut 3D* (tabela de cores 3D). A idéia de dimensão é dada por cada tabela de cor 1D que compõe a tabela de cores 3D.

Optou-se por utilizar três tabelas de cores 1D, pois, conforme já mencionado, caso fosse criada uma interface 3D a dificuldade que o usuário teria em escolher uma região para marcar como visível seria a mesma que marcar no próprio dado volumétrico que se deseja visualizar.

Para a opacidade 3D foi criado um novo volume, que seguindo a nomenclatura dada por Silva[30], chamou-se de volume terciário. Seguindo a nomenclatura da opacidade 2D, os nomes dos volumes não têm nenhuma ligação com a importância do mesmo na visualização final.

Junto com as três tabelas de cores 1D descritas anteriormente, na seção 2.2.5.1, a tabela de cores 3D é composta por mais doze coeficientes, que serão utilizados para criar a função de transferência tridimensional. Esta, por sua vez, é responsável por criar uma tabela de cores resultante que será utilizada na visualização do dado sísmico. Esta tabela resultante não é apresentada ao usuário, pois sua criação é feita no *hardware* gráfico.

A função de transferência é dada através de uma combinação entre as três tabelas de cores utilizadas no processo e de acordo com os coeficientes (parâmetros de combinação) fornecidos pelo usuário. A combinação é feita com base nos canais *RGB* e da transparência das tabelas de cores, atendendo o grau de contribuição de cada canal no valor final da tabela de cor resultante. Neste trabalho explora-se duas maneiras de combinar as três cores provenientes de cada tabela 1D: uma por combinação linear e outra por produto.

Os doze parâmetros utilizados para definir a função de transferência que irá gerar a tabela de cores resultante são: C_{r_1} , C_{r_2} , C_{r_3} , C_{g_1} , C_{g_2} , C_{g_3} , C_{b_1} , C_{b_2} , C_{b_3} , C_{a_1} , C_{a_2} e C_{a_3} , onde os C_{r_i} , C_{g_i} , C_{b_i} e C_{a_i} representam os coeficientes do canal *R*, *G*, *B* e *A* de cada tabela de cores, respectivamente.

Uma possível função de transferência é dada por:

$$\begin{aligned}
 C_R &= C_{r_1} \times R_1 + C_{r_2} \times R_2 + C_{r_3} \times R_3 \\
 C_G &= C_{g_1} \times G_1 + C_{g_2} \times G_2 + C_{g_3} \times G_3 \\
 C_B &= C_{b_1} \times B_1 + C_{b_2} \times B_2 + C_{b_3} \times B_3 \\
 C_A &= C_{a_1} \times A_1 + C_{a_2} \times A_2 + C_{a_3} \times A_3
 \end{aligned} \tag{1}$$

Onde C_R , C_G , C_B e C_A são os canais *R*, *G*, *B* e *A* da tabela de cores resultante, respectivamente. Os coeficientes podem variar de zero a um, sendo zero quando

este parâmetro não contribui em nada para a tabela de cores resultante e um para quando o parâmetro contribui com cem por cento para a tabela de cores resultante.

Os parâmetros R_i , G_i , B_i e A_i são os valores dos canais R, G, B e A, respectivamente, das tabelas de cores 1D que irão gerar a tabela de cores 3D.

Durante a execução deste trabalho foram feitos testes com outras funções de transferência e chegou-se à conclusão que para alguns casos a multiplicação dos canais é mais vantajosa do que a soma. A combinação de multiplicação dos canais é dada por:

$$\begin{aligned} C_R &= C_{r_1} \times R_1 \times C_{r_2} \times R_2 \times C_{r_3} \times R_3 \\ C_G &= C_{g_1} \times G_1 \times C_{g_2} \times G_2 \times C_{g_3} \times G_3 \\ C_B &= C_{b_1} \times B_1 \times C_{b_2} \times B_2 \times C_{b_3} \times B_3 \\ C_A &= C_{a_1} \times A_1 \times C_{a_2} \times A_2 \times C_{a_3} \times A_3 \end{aligned} \quad (2)$$

No caso da função de transferência que utiliza a multiplicação dos canais de cores, a representação da função de transferência utilizada pode ser resumida a:

$$\begin{aligned} C_R &= C_{res_R} \times R_1 \times R_2 \times R_3 \\ C_G &= C_{res_G} \times G_1 \times G_2 \times G_3 \\ C_B &= C_{res_B} \times B_1 \times B_2 \times B_3 \\ C_A &= C_{res_A} \times A_1 \times A_2 \times A_3 \end{aligned} \quad (3)$$

onde, $C_{res_R} = C_{r_1} \times C_{r_2} \times C_{r_3}$, $C_{res_G} = C_{g_1} \times C_{g_2} \times C_{g_3}$, $C_{res_B} = C_{b_1} \times C_{b_2} \times C_{b_3}$ e $C_{res_A} = C_{a_1} \times C_{a_2} \times C_{a_3}$.

Quando o usuário seleciona uma determinada região do atributo sísmico como sendo visível, há interesse em estudar um evento sísmico que ocorreu naquela era geológica. Gerhardt[11,12] verificou que com a opacidade 1D, muitas vezes, o usuário não consegue isolar por completo este evento (Gerhardt[11,12] e Silva[30]) e para tal ele tem à disposição a opacidade 2D. Esta se utiliza de dois atributos sísmicos para tentar isolar o evento sísmico de interesse, fazendo com que apenas o que está marcado como sendo visível nos dois atributos seja visto pelo usuário.

Quando a opacidade 3D utiliza a soma de canais para gerar a visualização, isto não acontece, pois ao invés de restringir-se uma área de interesse, a soma

funciona como uma “união” de áreas de interesse, fazendo com que mais dados sejam vistos. Já a multiplicação de canais restringe, somente, à área de interesse.

A Figura 15 mostra como a multiplicação dos atributos funciona como uma interseção. Somente na região em que ambos atributos estão visíveis é que o produto resulta um. Nas demais regiões este produto é zero e nada é visualizado.

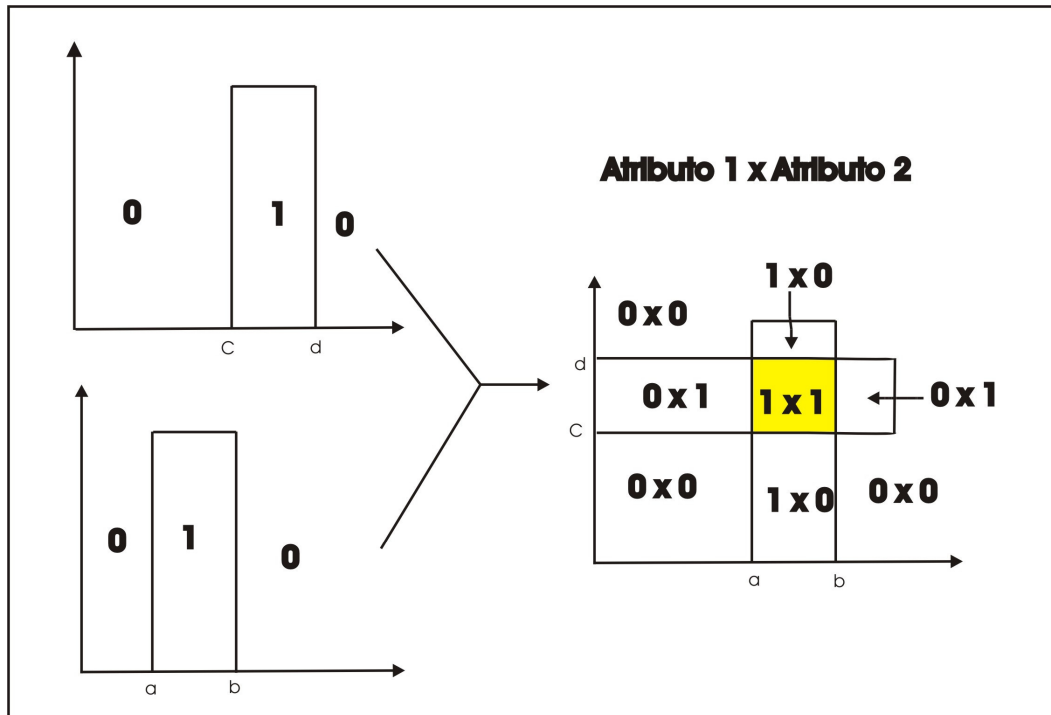


Figura 15 – Multiplicação do canal alfa na opacidade 2D.

Este trabalho tem por objetivo apresentar uma técnica que auxilie o trabalho de um intérprete, disponibilizando uma série de atributos sísmicos, onde cada um acrescente uma informação a mais para o estudo de um determinado bloco sísmico.

A interação com o usuário é de fundamental importância para a eficácia desta técnica. Para garantir uma renderização em tempo real, utiliza-se programação em placa de vídeo. Para executar o algoritmo de visualização é necessário que a estação de trabalho possua uma placa de vídeo com extensão `GL_ARB_fragment_program`. Esta extensão foi aprovada pelo comitê ARB em setembro de 2002 e incorporada ao *OpenGL* a partir da versão 1.3. Por utilizar uma extensão do *OpenGL* a implementação não é restrita a um fabricante específico de *hardware* gráfico, basta o mesmo ter a extensão mencionada.

3.2. Implementação

A implementação da opacidade 3D foi feita no *software* v3o2. No desenvolvimento deste trabalho foi necessária a criação de uma interface onde o usuário tem a oportunidade de editar a tabela de cores referente à opacidade 3D. Esta tabela de cores foi chamada de *lut 3D*. A seguir será apresentada a parte do código referente à programação em placa com uma descrição de cada linha do código. Esta parte do código foi implementada como um *fragment program*, que é a parte do código que será executada em *GPU*. O *fragment program* é executado para cada fragmento, ou seja, para cada *pixel* da imagem que encontra-se no *z-buffer* e que será renderizado no *frame buffer*.

```
(1)  "!!ARBfp1.0"
(2)  "TEMP rgba_frag;"
(3)  "TEMP lut1;"
(4)  "TEMP lut2;"
(5)  "TEMP lut3;"
(6)  "TEMP temp;"
(7)  "TEX rgba_frag, fragment.texcoord[0], texture[0], 3D;"
(8)  "TEX lut1, rgba_frag.g, texture[2], 2D;"
(9)  "TEX lut2, rgba_frag.b, texture[3], 2D;"
(10) "TEX lut3, rgba_frag.r, texture[4], 2D;"
(11) "MUL temp, program.local[1], lut1;"
(12) "MAD temp, program.local[2], lut2, temp;"
(13) "MAD temp, program.local[3], lut3, temp;"
(14) "MUL result.color, temp, 0.3333;"
(15) "END";
```

Esta seção será dividida em três partes e em cada uma, será comentado um conjunto de linhas do código. A primeira parte identifica as texturas que serão utilizadas no processo de visualização e mostra como estas são construídas. Em seguida é apresentada a parte onde são feitas as consultas das texturas (*fetch*). E por último é apresentada a combinação das cores, utilizando as funções de transferências que foram descritas na seção anterior.

Como pode ser visto o *fragment program* é uma *string*. Esta *string* é passada para a placa em tempo de execução. A placa fica encarregada de compilar este programa e de armazená-lo em memória (da *GPU*).

A primeira linha do código refere-se a versão do *fragment program* que está sendo utilizada, no caso a *ARBfp1.0*. As linhas (2), (3), (4), (5) e (6) são declarações de variáveis temporárias que estão sendo criadas pelo *fragment program*. Estas variáveis são vetores com quatro posições. Neste caso a palavra *TEMP* é uma palavra reservada do *fragment program*.

3.2.1. Construção das Texturas

Antes de explicar como ocorre a construção das texturas que são utilizadas na visualização, deve-se entender um detalhe de implementação do *v3o2*. Para compor a opacidade 2D, Silva[30] cria uma textura *RGB* tridimensional para guardar os valores dos atributos que compõem a opacidade 2D.

Ao visualizar um volume, no *v3o2*, é feita uma “transformação” do dado sísmico, pela qual os valores do atributo sísmico são convertidos para valores inteiros entre zero e duzentos e cinqüenta e cinco. Este processo é conveniente já que a visualização do dado sísmico é feita utilizando uma tabela com 256 cores. Este processo é comumente chamado de quantização. O valor do atributo quantizado refere-se a uma cor da tabela de cores 1D associada a este volume que será utilizado para visualização.

Para compor a textura *RGB* tridimensional, Silva[30] faz com que o canal R seja sempre zero, o canal G seja os valores quantizados do volume primário e o canal B como sendo os valores quantizados do volume secundário. A Figura 16 mostra o esquema de construção desta textura tridimensional.

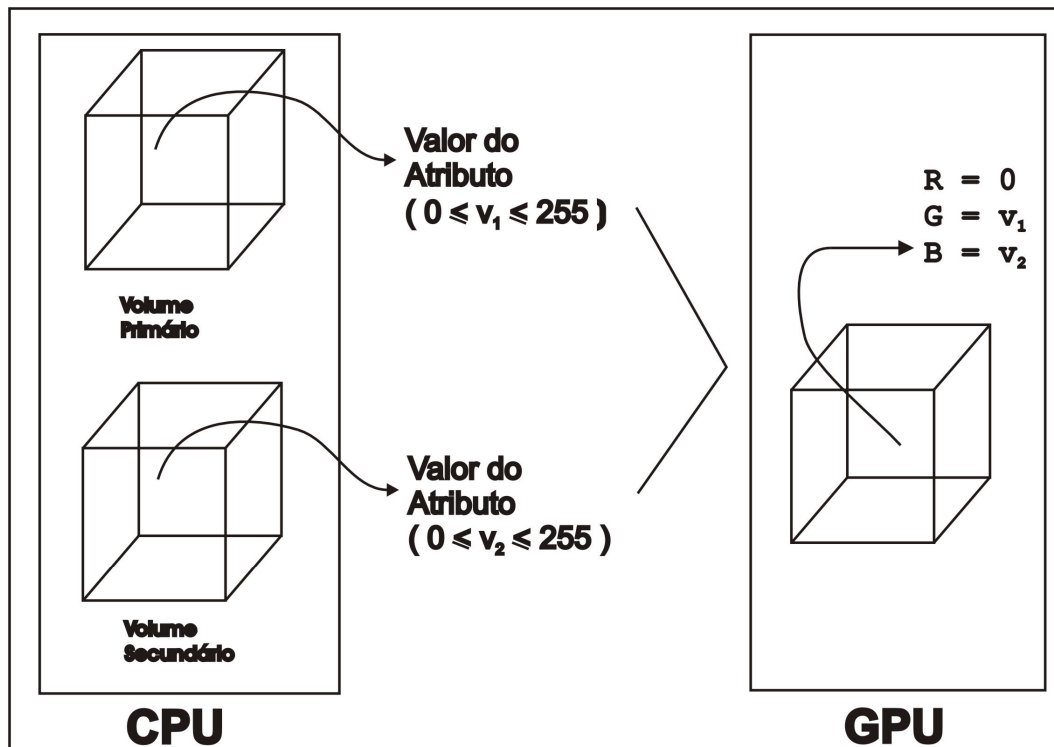


Figura 16 – Construção da textura tridimensional RGB (opacidade 2D).

A opacidade 3D faz uso desta textura tridimensional atribuindo os valores quantizados do volume terciário ao canal R da textura, que anteriormente tinha seus valores nulos. Os canais G e B não são alterados.

Antes de dar continuidade a análise do código, faz-se necessário identificar que texturas estão sendo criadas no programa. A primeira textura que aparece no código é a `texture[0]` que é uma textura 3D. Esta textura é a textura tridimensional que foi descrita anteriormente. A `texture[1]` não aparece em nenhum momento no código apresentado. Isto se deve ao fato desta textura estar sendo utilizada para a iluminação do dado sísmico, não sendo utilizada na opacidade 3D.

As texturas de número dois, três e quatro são as texturas do tipo 2D que estão associadas aos volumes primário, secundário e terciário, respectivamente. A criação destas texturas e o carregamento destas, em GPU, são feitos fora do *fragment program*.

3.2.2. Consulta de Textura

Para aplicar uma textura em um objeto, é preciso criar uma relação entre os vértices dos polígonos do objeto e os *texels* da textura. Esta relação é feita através

do mapeamento das coordenadas de textura. Cada vértice de um polígono que está sendo desenhado possui um vetor (s, t, r, w) de coordenadas de textura que identifica um ponto com seu valor *RGBA* na textura. Os vetores dos pontos são interpolados para todos os fragmentos do polígono durante a rasterização.

As linhas (7), (8), (9) e (10) do código apresentado começam com a instrução *TEX*, que é o mapeamento da coordenada de textura para a cor da textura. A instrução *TEX* possui a sintaxe

```
TEX v, u, t, tipo de textura
```

onde, *v* representa um vetor de saída com os valores de RGB. O segundo parâmetro *u* indica a identificação da imagem de textura. A textura “alvo” é indicada pelo parâmetro *t* e o tipo de textura refere-se à dimensão da textura.

A linha (7) do código apresentado usa a coordenada de textura (s, t, r) do fragmento que está em *fragment.texcoord[0]* para obter em *texture[0]* qual é o RGB que esta coordenada de textura aponta e associa ao vetor *rgba_frag*. Esta textura é uma textura dependente, pois utiliza os valores RGB como coordenada de textura. Isto pode ser visto ao analisar as linhas (8), (9) e (10).

Como foi descrito na seção 3.2.1 a textura zero possui os valores quantizados referentes aos volumes primário, secundário e terciário. Logo, o vetor *rgba_frag.g* representa o valor “quantizado” do atributo primário e este valor representa um índice da tabela de cores 1D que está associado a este volume.

Na linha de número (8), o valor que foi retornado da linha (7) e armazenado em *rgba_frag.g* é usado como coordenada de textura na consulta ao valor de *RGBA* que está em *texture[2]*. Este valor de *RGBA* é atribuído a variável *lut1*. Feito isso, a cor resultante em *lut1* é a cor definida pelo volume primário. O mesmo ocorre nas linhas (9) e (10), onde são atribuídas as cores definidas pelos volumes secundários e terciários às variáveis *lut2* e *lut3*, respectivamente.

A Figura 17 mostra o caminho que é feito pela consulta (*fetch*) de textura até a cor que contribuirá com a visualização.

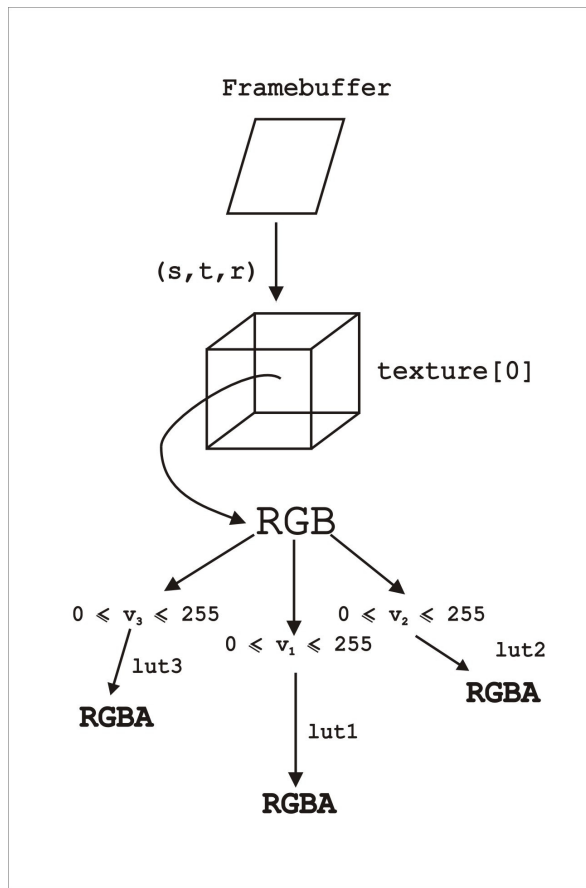


Figura 17 – Consulta (fetch) de textura até a cor.

Ao final da fase de consulta temos três vetores RGBA, um para cada volume (atributo). Esses três vetores são combinados para gerar a cor final do fragmento.

3.2.3. Combinação das cores

A instrução *MUL* aparece na linha (11). Esta instrução é uma palavra reservada do *fragment program* que tem por objetivo multiplicar duas variáveis, essa multiplicação é feita componente a componente. A sintaxe é “*MUL a, b, c;*”, onde pode-se entender como sendo $a = b \times c$. Ainda na linha (11), aparece uma variável chamada *program.local*. Esta variável refere-se aos coeficientes descritos na seção 3.1, que serão utilizados na função de transferência. Este parâmetro também é um vetor de quatro posições.

O *program.local[1]* refere-se aos coeficientes C_{r_1} , C_{g_1} , C_{b_1} , C_{a_1} , ou seja, os coeficientes que estão sendo aplicados à tabela de cores 1D associada ao volume primário. Estes parâmetros são passados para a *GPU* através do comando

glProgramLocalParameter4dARB, e toda vez que um parâmetro for atualizado, este comando deve ser chamado para informar ao *hardware* gráfico os novos valores a serem utilizados. As variáveis *program.local[2]* e *program.local[3]* são os coeficientes relacionados à tabela de cores associada ao volume secundário e à tabela de cores associada ao volume terciário, respectivamente. O resultado da linha (11) é que o vetor *temp* será representado por:

$$temp.r = C_{r_1} \times lut1.r$$

$$temp.g = C_{g_1} \times lut1.g$$

$$temp.b = C_{b_1} \times lut1.b$$

$$temp.a = C_{a_1} \times lut1.a.$$

Nas linhas (12) e (13) aparece a instrução *MAD* (*multiply and add*), que é responsável por multiplicar e somar. A sintaxe de *MAD* é “*MAD a, b, c, d*” que pode ser escrita como sendo $a = b \times c + d$. Logo, na linha (12) o vetor *temp* será representado como sendo a multiplicação dos coeficientes associados a tabela de cores do volume secundário somado com os valores encontrados na linha (11), ou seja, *temp* será representado por:

$$temp.r = C_{r_2} \times lut2.r + temp.r$$

$$temp.g = C_{g_2} \times lut2.g + temp.g$$

$$temp.b = C_{b_2} \times lut2.b + temp.b$$

$$temp.a = C_{a_2} \times lut2.a + temp.a$$

Para a linha (13) tem-se que *temp* será descrito como sendo:

$$temp.r = C_{r_3} \times lut3.r + temp.r$$

$$temp.g = C_{g_3} \times lut3.g + temp.g$$

$$temp.b = C_{b_3} \times lut3.b + temp.b$$

$$temp.a = C_{a_3} \times lut3.a + temp.a$$

A linha de número (14) novamente representa uma multiplicação. Aqui o resultado da multiplicação é atribuído a variável *result.color*. Esta variável é uma palavra reservada do *fragment program* e representa a cor com que o fragmento será desenhado no *frame-buffer*. A multiplicação por 0,3333 tenta impedir a saturação da cor resultante do fragmento.

Para finalizar o código a linha de número (15) indica ao *fragment program* que este chegou ao fim.

Esta foi a análise para a parte do código referente ao processamento em *GPU* utilizando uma função de transferência com soma de canais. A seguir será apresentado o código com a função de transferência que utiliza a multiplicação dos canais de cores para compor a visualização.

```
(1)  "!!ARBfp1.0"
(2)  "TEMP rgba_frag;"
(3)  "TEMP lut1;"
(4)  "TEMP lut2;"
(5)  "TEMP lut3;"
(6)  "TEMP temp1;"
(7)  "TEMP temp2;"
(8)  "TEMP temp3;"
(9)  "TEMP temp4;"
(10) "TEX rgba_frag, fragment.texcoord[0], texture[0], 3D;"
(11) "TEX lut1, rgba_frag.g, texture[2], 2D;"
(12) "TEX lut2, rgba_frag.b, texture[3], 2D;"
(13) "TEX lut3, rgba_frag.r, texture[4], 2D;"
(14) "MUL temp1, program.local[1], lut1;"
(15) "MUL temp2, program.local[2], lut2;"
(16) "MUL temp3, program.local[3], lut3;"
(17) "MUL temp4, temp2, temp1;"
(18) "MUL result.color, temp4, temp3;"
(19) "END";
```

Analisando o código acima, pode-se notar que não existe nenhuma instrução nova. São criadas variáveis temporárias novas como *temp1*, *temp2*, *temp3* e *temp4*. A maior diferença para a função de transferência que utiliza a soma dos canais é que neste caso as instruções *MAD* foram substituídas por *MUL*. As linhas (12) e (13) da função de transferência com soma de canais deu lugar às linhas (15) e (16) da função de transferência com multiplicação de canais. Nesta função de transferência não é necessário fazer a multiplicação por 0,3333 para evitar a saturação dos canais.

Na ferramenta que está sendo oferecida junto ao *software v3o2*, a interface de construção da tabela de cores 3D possui dois quadros de parâmetros. No

quadro da esquerda o usuário informa qual é a tabela de cores 1D que se deseja associar ao volume primário, a tabela de cores do volume secundário e a tabela do volume terciário. No quadro à direita tem-se a opção de editar os coeficientes que serão passados para o programa. Na Figura 18 tem-se uma visão da interface final.

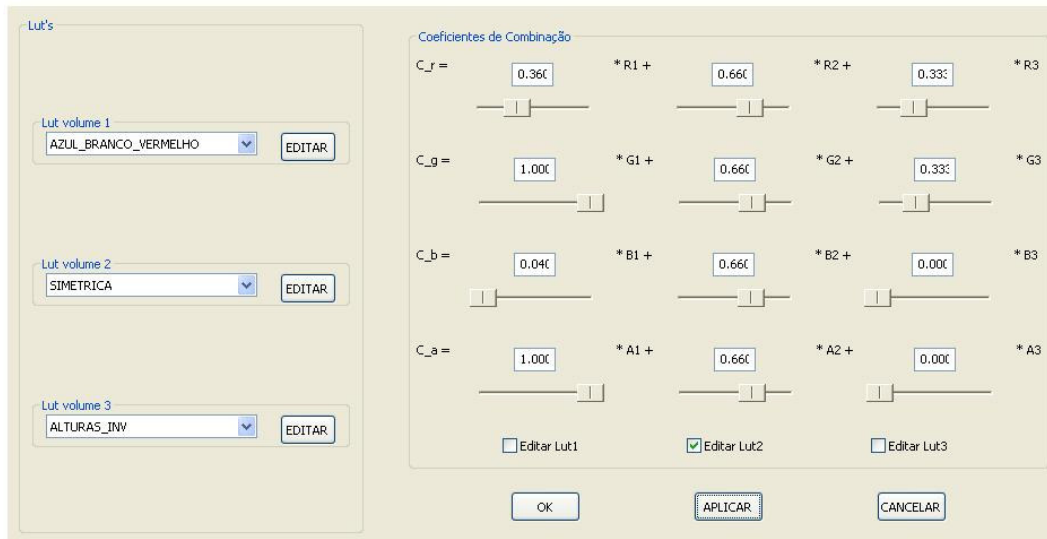


Figura 18 – Interface de edição da tabela de cores 3D com soma de canais

Percebe-se na figura que as tabelas de cores são associadas a volumes. Estes volumes são os volumes sísmicos de origem dos atributos, ou seja, são os dados volumétricos que compõem a combinação.

A tabela de cores que for associada a um volume pode ser editada como qualquer tabela de cores 1D. No quadro da direita há, ainda, três *check boxes* (caixas que podem ser marcadas como ativas ou inativas) que quando ativas, indicam que o usuário deseja editar os valores dos coeficientes referentes àquela tabela de cores de forma idêntica, ou seja, todos os quatro coeficientes referentes à tabela de cores em questão terão o mesmo valor. Caso o usuário desmarque a *check box* os coeficientes de combinação poderão ser editados separadamente.

No exemplo da Figura 18 pode-se notar que o *check box* referente a tabela de cores associada ao volume secundário está ativo e que os quatro coeficientes possuem valor 0.66. Os *check boxes* das outras tabelas de cores estão inativos e têm valores diferentes para seus coeficientes.

4 Resultados

Neste capítulo serão apresentados os testes feitos com a opacidade 3D. Nestes testes reproduzimos, inicialmente, os resultados obtidos pela opacidade 2D em Silva[30]. Serão apresentados também os resultados obtidos com a opacidade 3D sobre o dado sintético domo e sobre um dado real. Como último exemplo são geradas imagens utilizando a decomposição espectral.

No exemplo quatro, foram utilizados dados que fazem uso da técnica de decomposição espectral. Estes dados foram cedidos por Ruthner[28], gerados no aplicativo WebSintesi desenvolvido pela Petrobras em parceria com o Tecgraf/PUC-Rio. Foi utilizada a decomposição espectral, nas frequências de 10 Hz, 18 Hz e 26 Hz, sobre um dado real.

4.1. Exemplo 1 – Reprodução da Opacidade 2D

Utilizando a função de transferência que faz uso da multiplicação de canais, equação (2), é possível reproduzir os resultados obtidos com a opacidade 2D. A seção 2.2.5.2 mostra os resultados da utilização da opacidade 1D e 2D no dado sintético domo. A Figura 19 mostra o uso da opacidade 3D reproduzindo a opacidade 2D. A Figura 20 representa a tabela de cores 1D utilizada no volume primário.

Há três volumes carregados, porém apenas dois deles estão contribuindo para a imagem final. Isto se deve ao fato de que o volume terciário possui uma tabela de cores associada que é toda branca e opacidade igual a 1 em todo o seu domínio. O uso de uma tabela de cores toda branca faz com que a cor resultante não seja influenciada pelo atributo.

O volume secundário também possui associada uma tabela de cores toda branca. Entretanto, existe uma área do domínio marcada como sendo não visível. Esta área é a mesma marcada como sendo não visível no volume secundário do histograma apresentado na Figura 12.

Os atributos utilizados foram: amplitude como volume primário, fase instantânea como volume secundário e frequência instantânea como volume terciário. Neste caso o volume terciário não tem influência alguma.

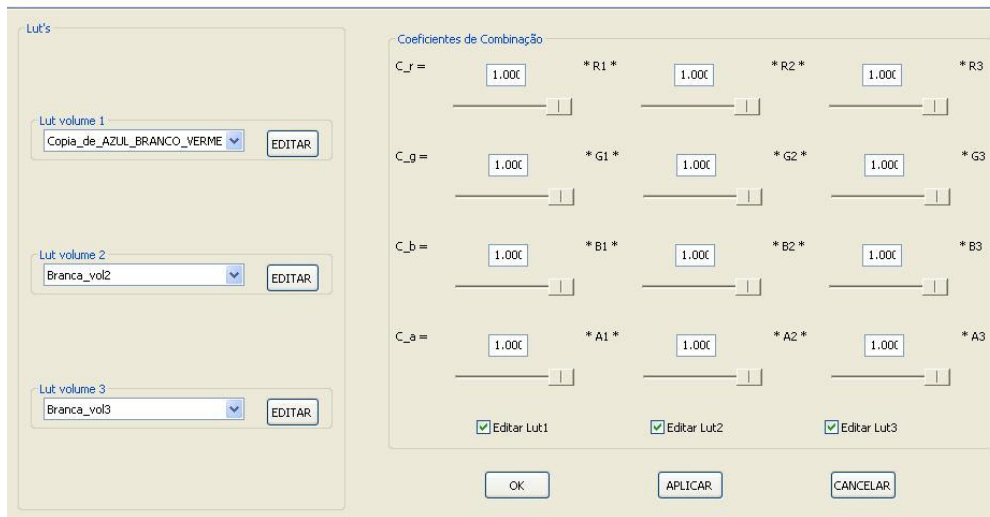
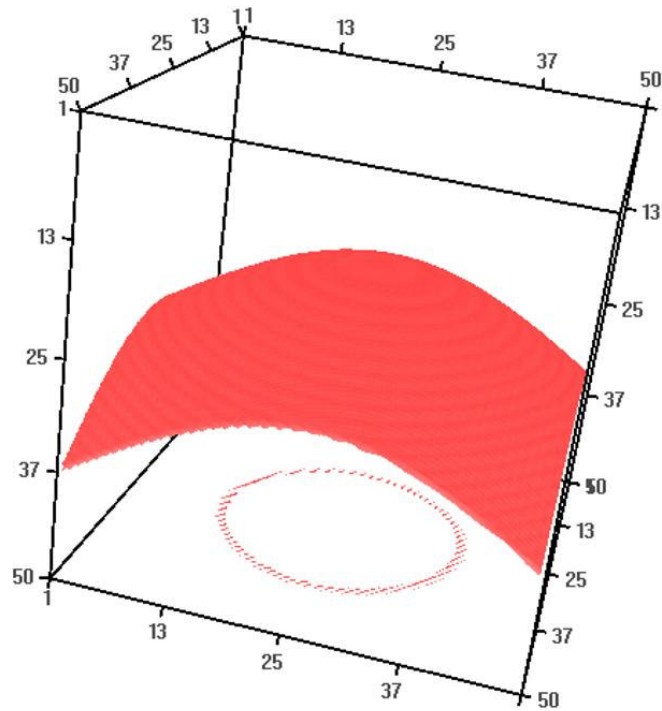


Figura 19 – Reprodução da opacidade 2D utilizando a opacidade 3D

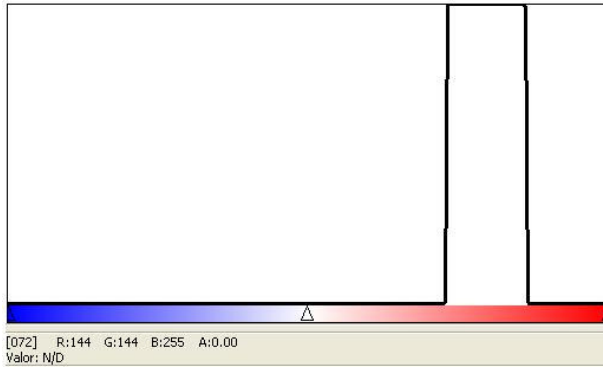


Figura 20 – Reprodução da tabela de cores utilizada na Opacidade 1D

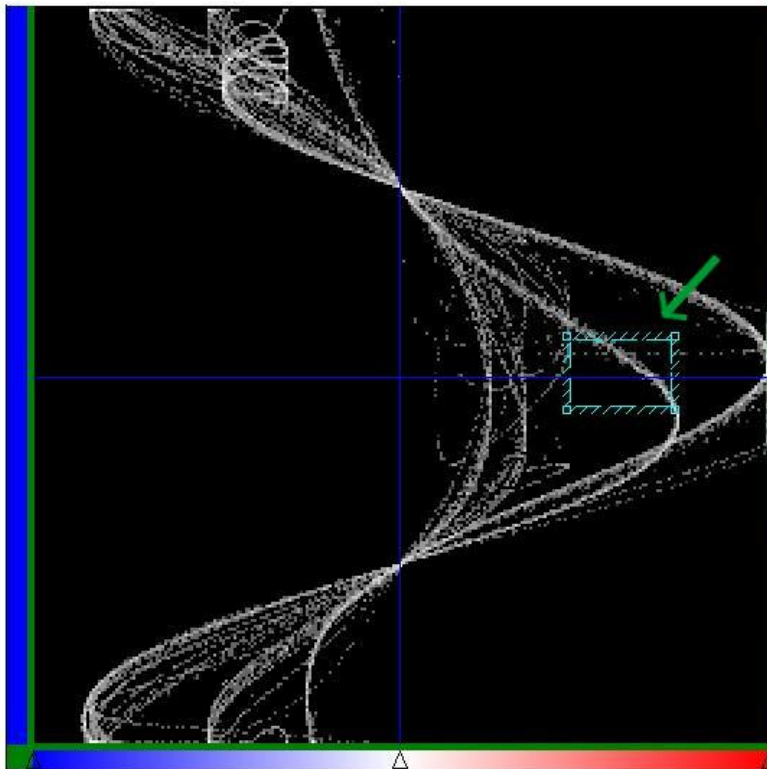


Figura 21 – Reprodução do histograma da Figura 12 utilizado para gerar a Figura 13

Os limites desta área selecionada variam da posição 186 a 221, da tabela de cores associada ao eixo horizontal (referente ao volume primário - amplitude) e da posição 116 a 140 do eixo vertical (referente ao volume secundário – fase instantânea).

4.2.

Exemplo 2 – Utilização da Opacidade 3D no Dado Sintético

A opacidade 2D trouxe um ganho para a visualização volumétrica de um dado sísmico sintético, como foi visto na seção 2.2.5.2, no dado domo, reproduzido acima no Exemplo 1.

A Figura 22 mostra a comparação entre a opacidade 1D, a opacidade 2D e a opacidade 3D. Nesta figura é utilizada uma função de transferência multidimensional que faz uso da multiplicação de canais para gerar a combinação dos volumes utilizados.

Pode-se observar na Figura 22.(a) a presença dos efeitos de borda na tentativa de isolar o evento sísmico de interesse. Na Figura 22.(b), a opacidade 2D consegue eliminar os efeitos de borda, isolando o evento sísmico de interesse. A opacidade 3D filtra ainda mais o resultado da opacidade 2D, eliminando o ruído que aparece na base do volume sísmico.

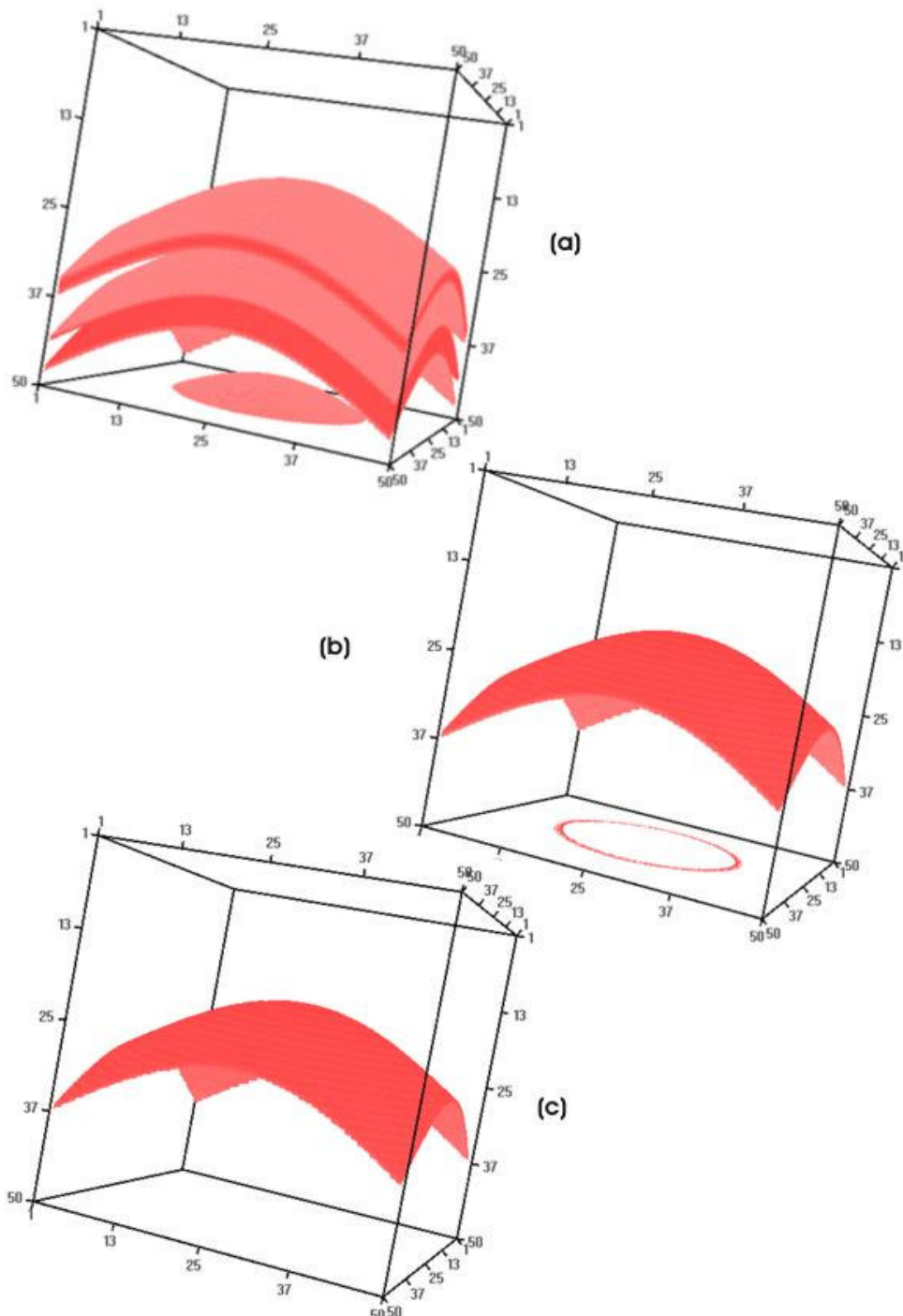


Figura 22 – (a) Opacidade 1D; (b) Opacidade 2D; (c) Opacidade 3D.

Para gerar a Figura 22.(c) foram utilizadas as tabelas de cores descritas no Exemplo 1, com a tabela de cores referente ao volume terciário tendo apenas uma região marcada como sendo visível, com valores entre 0 (zero) e 72 (setenta e dois). Esta tabela de cores está reproduzida na Figura 23. A tabela de cores é toda representada pela cor branca.

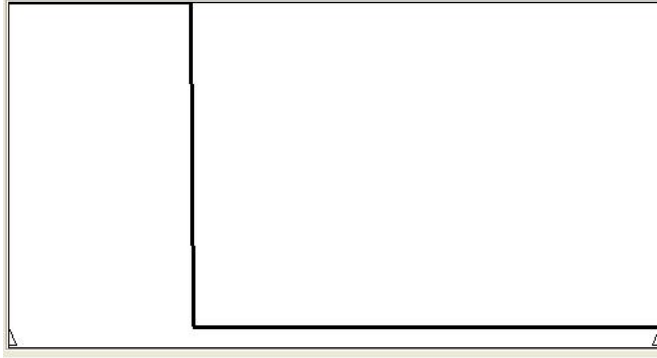


Figura 23 – Tabela de cores associada ao volume terciário.

4.3.

Exemplo 3 – Utilização da Opacidade 3D em Dados Reais.

A seguir será apresentado o resultado da opacidade 3D em um dado sísmico real, utilizando a função de transferência com multiplicação de canais. Os volumes utilizados tiveram como atributos a amplitude, a fase instantânea e a frequência instantânea. Porém, a opacidade 3D dá liberdade ao usuário de escolher quaisquer três dos atributos sísmicos disponíveis. Fica sob a responsabilidade do intérprete avaliar qual combinação de três atributos obtém melhor resultado.

A Figura 24 mostra o resultado da opacidade 3D utilizando tabelas de cor branca para os volumes secundário e terciário. Os intervalos marcados como visíveis seguem os intervalos utilizados na visualização da opacidade 1D e 2D da Figura 25.

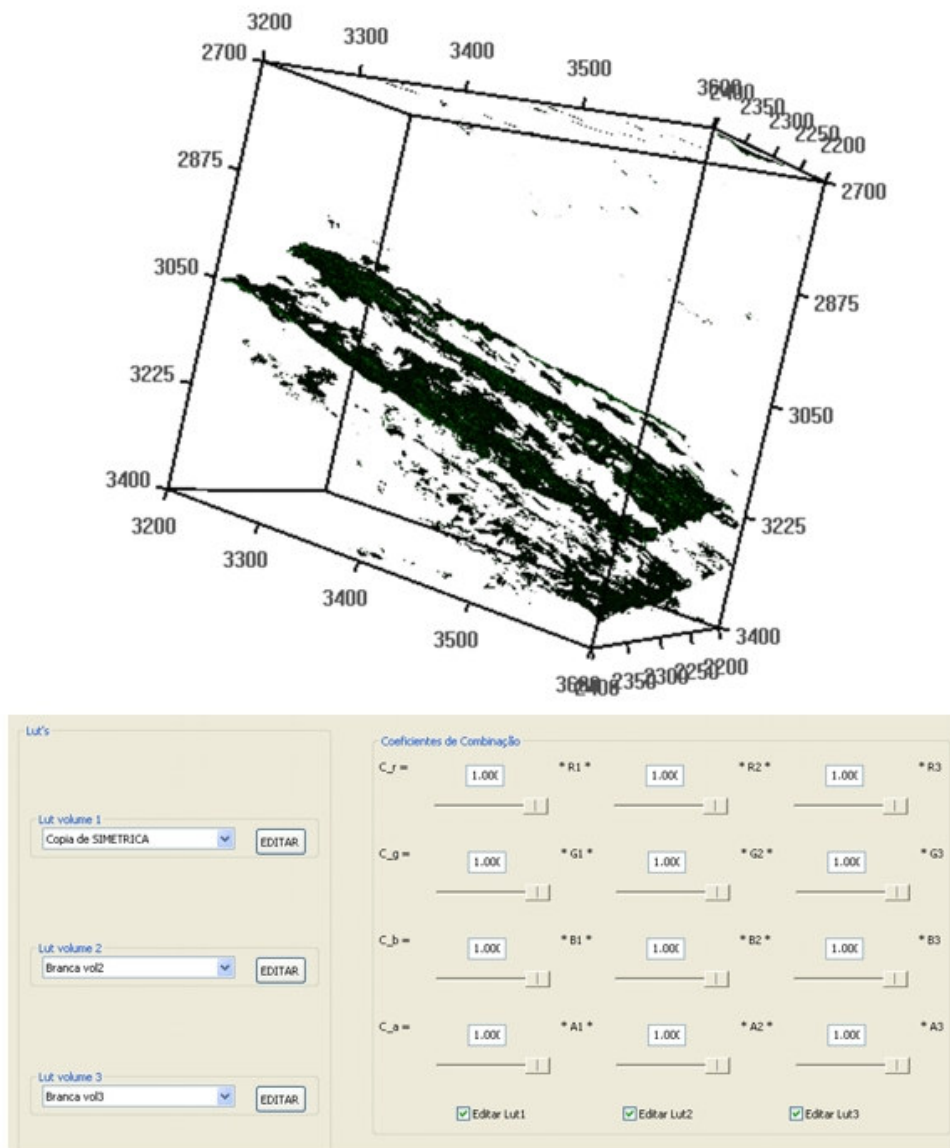


Figura 24 – Resultado da opacidade 3D com dado real

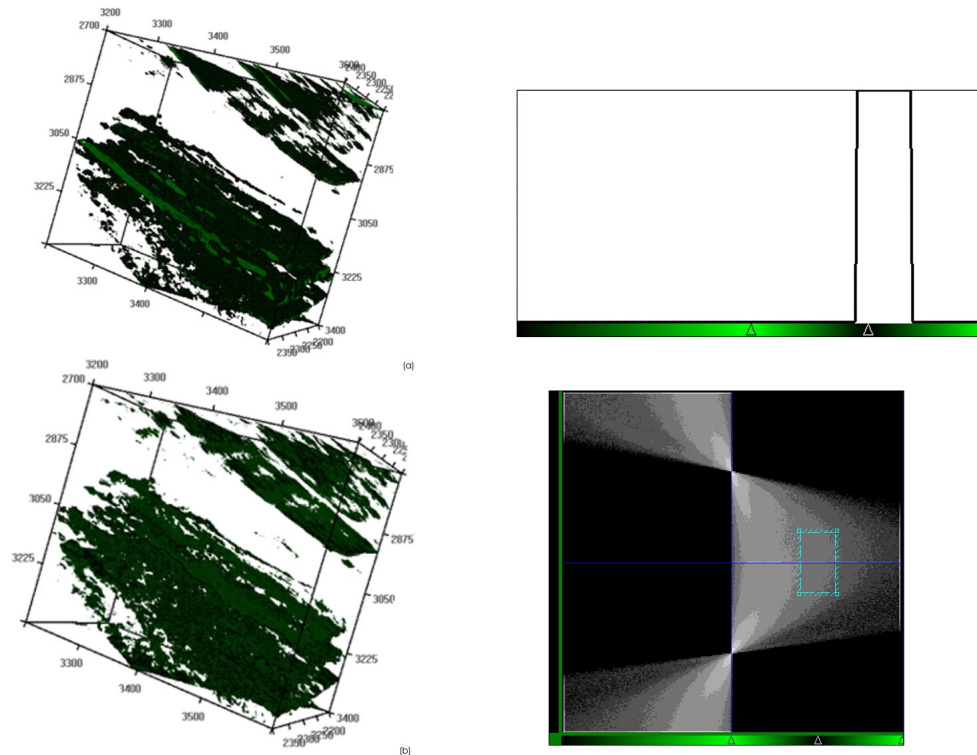


Figura 25 – Reprodução da comparação entre a opacidade 1D (a) e a opacidade 2D (b)

Também é possível reproduzir os resultados obtidos pela opacidade 2D na 3D, utilizando a tabela de cores do volume terciário branca e com todos os valores marcados como visíveis.

Na Figura 26 tem-se a comparação entre os resultados das três opacidades. Na opacidade 1D existe muito ruído na parte superior do dado e o evento de interesse está escondido entre outras camadas. A opacidade 2D conseguiu filtrar um pouco do ruído presente na parte superior do dado. O evento de interesse já começa a tomar forma.

Na opacidade 3D o ruído da parte superior do dado é quase que filtrado por completo. O evento de interesse aparece mais destacado.

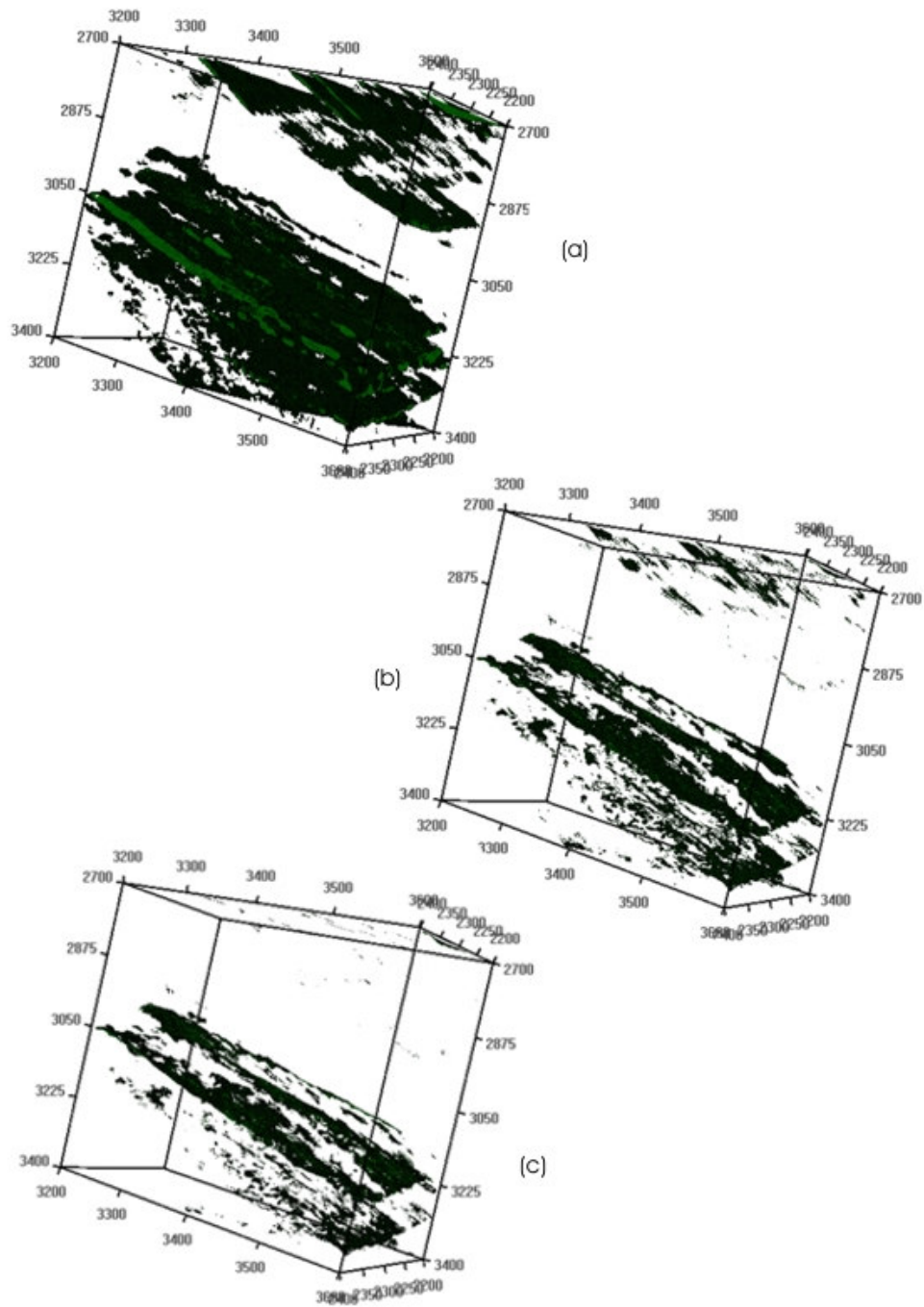


Figura 26 – (a) Opacidade 1D; (b) Opacidade 2D; (c) Opacidade 3D

4.4.

Exemplo 4 – Opacidade 3D Utilizando a Decomposição Espectral

Uma outra utilização da técnica de opacidade 3D é realizar uma combinação de atributos provenientes da decomposição espectral, a fim de prover uma visualização de duas ou mais faixas de frequência, simultaneamente.

Partyka[26] utiliza a decomposição espectral na faixa de 16 Hz e 26 Hz. Partyka[26] define um “cubo 4D de tempo-frequência”. Para montar o cubo 4D, Partyka[26] define uma fatia no tempo e, em seguida, faz a decomposição espectral dessa fatia. O resultado é inserido no cubo, fazendo com que o mesmo passe a ter os pontos x , y e *tempo* originais e mais as frequências provenientes da decomposição espectral. Com isso o intérprete pode navegar no espaço, no tempo ou na frequência do dado sísmico. O dado resultante desta técnica é muito maior que o dado original, o que pode vir a ser um problema para dados sísmicos maiores.

Esta idéia está presente na opacidade 3D, sendo que não é necessário criar o “cubo 4D”. Todos os valores da decomposição espectral estão presentes no volume sísmico que está sendo visualizado. A visualização de uma amostra é dada pela combinação dos três atributos sísmicos que compõem a amostra.

Na Figura 27 estão representadas quatro fatias sísmicas, todas provenientes do mesmo volume sísmico e na mesma amostragem de tempo. A figura (a) representa o atributo sísmico amplitude, que é a forma mais comum de obtenção do dado sísmico. Na figura (b) encontra-se a decomposição espectral na frequência de 10 Hz. Nesta imagem pode-se observar mais à direita um canal sísmico tomando forma. A decomposição espectral na frequência de 18 Hz se encontra na figura (c), onde um reservatório aparece na parte superior do dado com um pequeno canal à direita da imagem. Utilizando a frequência de 26 Hz, representada na figura (d), além do canal e do reservatório observam-se falhas.

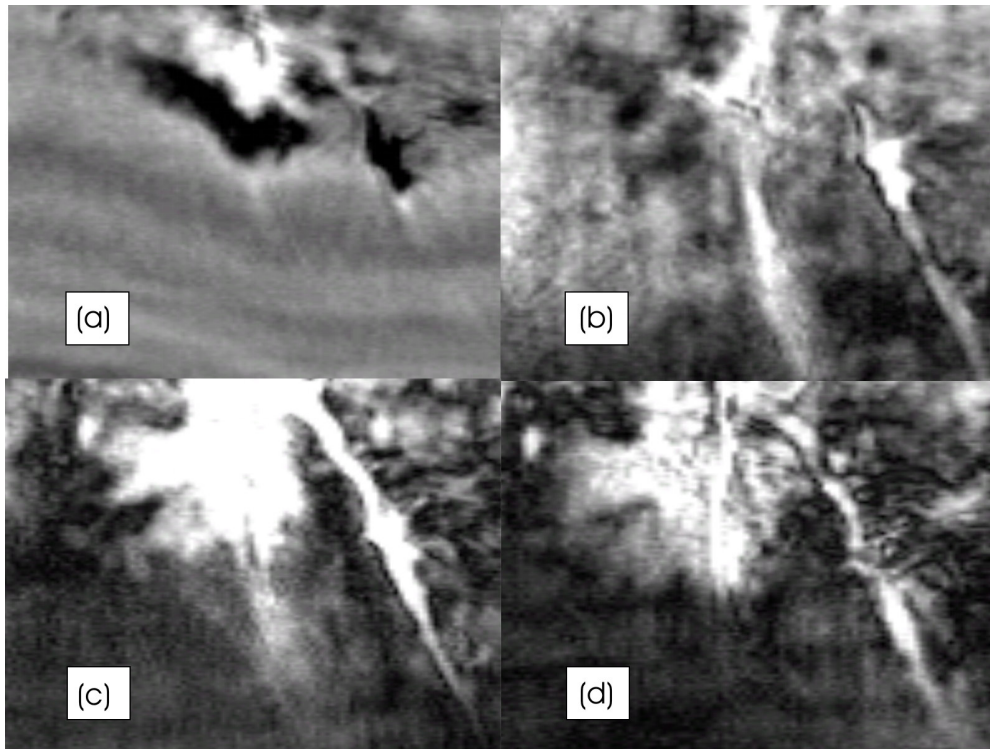


Figura 27 – Fatias no tempo. (a) Amplitude; (b) 10 Hz; (c) 18 Hz; (d) 26 Hz

A Figura 28 apresenta duas técnicas de interpretação utilizando a opacidade 3D para interpretação da decomposição espectral. Na Figura 28(a) e Figura 28(b) foram utilizadas a tabela de cores PRETO_BRANCO para os três volumes associados à visualização. A utilização desta tabela de cores é explicada pela Figura 28(b). A cor preta é formada pela combinação de (R, G, B) igual a $(0, 0, 0)$, enquanto que a cor branca é formada pela combinação de (R, G, B) igual a $(255, 255, 255)$. Os tons de cinza que se encontram no meio da tabela de cores são formados também por iguais valores de vermelho, verde e azul.

Ao utilizar apenas o canal de cor vermelho para visualizar o volume primário é como se houvesse uma tabela de cores associadas que tivesse os canais verde e azul sempre iguais a zero e o canal vermelho crescendo de zero a duzentos e cinquenta e cinco. Fazendo o volume secundário tendo apenas o coeficiente do canal verde igual a um e o volume terciário tendo apenas o coeficiente do canal azul igual a um, tem-se a Figura 28(b).

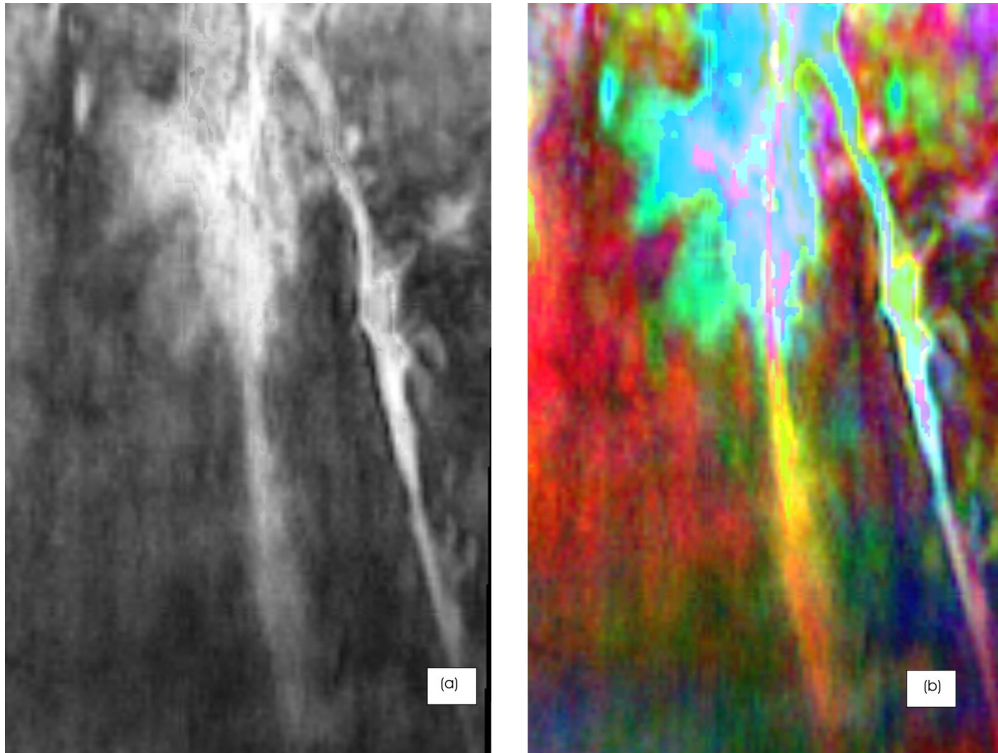


Figura 28 – Uso da Opacidade 3D com a decomposição espectral

Para melhor compreender a Figura 28 deve-se lembrar como são compostas as cores no sistema *RGB*. Por exemplo, o amarelo é uma combinação do vermelho e do verde, logo os *pixels* que estão em amarelo na Figura 28 possuem um valor alto no volume de 10 Hz (vermelho) e também no de 18 Hz (verde). Na parte da imagem que se encontra em vermelho temos uma resposta baixa nas frequências de 18 Hz (verde) e 26 Hz (azul) e uma resposta alta para a frequência de 10Hz (vermelho). A cor magenta é formada pela combinação das cores vermelha e azul, enquanto que a cor ciano é formada pelas cores azul e verde. Vale lembrar que a cor preta é formada pela ausência de cores, ou seja, pelos canais *R*, *G* e *B* iguais a zero e a cor branca pela soma de todas as cores, ou seja, pelos canais *R*, *G* e *B* iguais a duzentos e cinquenta e cinco. É interessante observar como o reservatório e os canais estão realçados nesta visualização.

A Figura 29 mostra a tabela de cores que resultou a Figura 28.

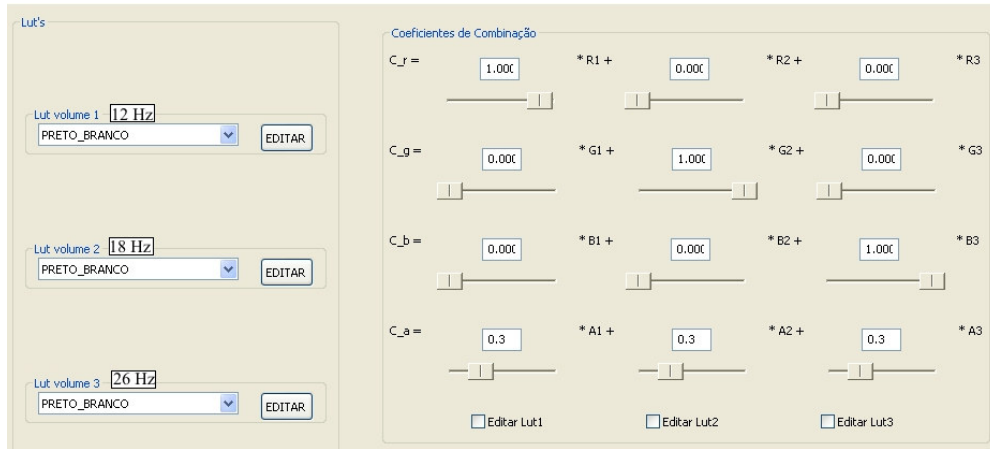


Figura 29 – Tabela de cores 3D que originou a Figura 28

Este exemplo mostra como a técnica de opacidade 3D se mostra poderosa quando se necessita dar ênfases diferenciadas a diferentes atributos.

5 Conclusões e Trabalhos Futuros

O objetivo principal deste trabalho foi desenvolver uma técnica de visualização científica, chamada de opacidade 3D, que auxilie o intérprete a montar um modelo geológico de uma determinada área sísmica. Durante o desenvolvimento da mesma teve-se a preocupação em utilizar o que há de mais moderno em termos de *hardware* dentro do parque de máquinas da Petrobras, até a presente data.

Buscando uma maior interatividade com o usuário, optou-se por utilizar as *GPU's* programáveis. Dentre as máquinas disponíveis para o desenvolvimento e teste da opacidade 3D optou-se por utilizar *fragment program*, em *OpenGL*, que é independente de sistema operacional. Optando por utilizar extensões do *OpenGL* reconhecidas pelo *ARB*, a opacidade 3D também tornou-se independente do fabricante do *hardware* gráfico.

5.1. Conclusões

Foi visto que a opacidade 2D em dados sintéticos é capaz de isolar um evento de interesse, porém ao ser aplicado em dados reais os resultados obtidos podem ser sensivelmente melhorados. Silva[30] sugere como trabalhos futuros a extensão da opacidade 2D para multiatributos. Este trabalho é a extensão natural proposta por Silva[30].

A opacidade 3D consegue reproduzir os resultados obtidos pela opacidade 2D no dado sintético domo, que foi o mesmo utilizado por Silva[30]. Ao combinar os atributos sísmicos de amplitude, fase instantânea e frequência instantânea, a opacidade 3D consegue filtrar ainda mais os resultados obtidos pela opacidade 2D.

Com relação aos dados reais, foi possível verificar que a opacidade 3D contribui para a melhoria da visualização volumétrica de um evento sísmico específico.

A opacidade 3D mostrou, ainda, que pode contribuir para o estudo de multiatributos, pois estes podem ser combinados de forma a dar uma idéia inicial para o intérprete do grau de influência de um atributo sísmico sobre outro.

Foi visto que a decomposição espectral junto com a opacidade 3D obtém bons resultados na identificação de falhas geológicas. A opacidade 3D pode servir de ferramenta auxiliar para a decomposição espectral na mensuração de reservatórios delgados.

5.2. Trabalhos Futuros

A opacidade 3D é uma ferramenta que será integrada ao *software* v3o2 que é desenvolvido pelo Tecgraf – PUC/Rio para a Petrobras. Os atributos testados neste trabalho são os que estão disponíveis dentro do v3o2. Um trabalho futuro a ser realizado é a utilização de outros atributos sísmicos para compor a opacidade 3D.

As placas gráficas têm apresentado um desenvolvimento tecnológico muito rápido, o que possibilitou um grande avanço na área da visualização volumétrica direta e indireta. Este trabalho faz uso desse desenvolvimento, já que a parte da visualização volumétrica é feita através do *fragment program*. Hoje uma limitação do *fragment program* é a utilização de apenas quatro texturas. Com essa limitação, só se tem apoio do *hardware* gráfico para o uso de três atributos. Com o avanço dos *hardwares* esta limitação será certamente superada e mais atributos poderão ser combinados.

A função de transferência multidimensional com a soma dos canais mostrou-se muito importante com a decomposição espectral. Para a utilização da opacidade 3D é necessário que se tenha três volumes sísmicos carregados. Em alguns testes foram utilizados para compor a visualização apenas dois volumes para facilitar a interpretação do mesmo, porém os três volumes tinham que ser carregados. Este fato deu origem à sugestão de num trabalho futuro fazer a adequação da opacidade 2D para a soma de canais.

Como trabalho futuro pode-se testar outras funções de transferência. Uma sugestão é fazer a combinação de uma tabela de cores 2D com uma tabela de cores 1D.

Recentemente tem-se trabalhado com a sísmica 4D, que consiste em comparar um levantamento sísmico feito há alguns anos com um levantamento recente, para averiguar a precisão do modelo geológico de origem. A opacidade 3D, utilizando a função de transferência com soma de canais, parece ser uma boa ferramenta para auxiliar na sísmica 4D.

6 Bibliografia

- 1 3Dlabs, produtora de placas de vídeo e desenvolvedora de tecnologia para processadores gráficos. Disponível em <<http://www.3dlabs.com/>>
- 2 APPLE, Apresenta a linha de produtos da Apple, além de informações sobre os hardwares e softwares desenvolvidos pela linha .Mac. Disponível em <<http://www.apple.com/br/>>, 2007
- 3 BAVOIL, L.; CALLAHAN, S. P.; LEFOHN, A.; COMBA, J. L. D.; SILVA, C. T. **Multi-Fragment Effects on the GPU using the k-Buffer**, SCI Institute, Technical Report, November 2006
- 4 BERNARDON, F. F.; CALLAHAN, S. P.; COMBA, J. L. D.; SILVA, C. T. **Interactive Volume Rendering of Unstructured Grids with Time-Varying Scalar Fields**, Eurographics Symposium on Parallel Graphics and Visualization (2006), The Eurographics Association, May 2006
- 5 BORDIGNON, A.; CASTRO, R. P.; LOPES, H.; LEWINER, T.; TAVARES, G.; **Exploratory Visualization Based on Multidimensional Transfer Functions and Star Coordinates**. SIBGRAPI XIX, 2006
- 6 CHENG, C.; WAKEFIELD, G.H. **A Tool for Volumetric Visualization and Sonification of Head-Related Transfer Functions (HRTF's)**, University of Michigan, Department of Electrical Engineering and Computer Science, 2000
- 7 CELLES, W. **Renderização em Tempo Real**. Notas de aula, 2006
- 8 COHEN, M.; MANSSOUR, I.H. **OpenGL: Uma abordagem prática e objetiva**, Novatec, 2006, 478 p.
- 9 DORN, G. A. **Modern 3-D Seismic Interpretation**, The Leading Edge, v.17, n.9, p.1262-1272, 1998
- 10 GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J., **Padrões de Projeto**, Bookman, 2005, 364 p.
- 11 GERHARDT, A. **Aspectos da Visualização Volumétrica de Dados Sísmicos**, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro – RJ, 1998.

- 12 GERHARDT, A. et al. **Two-Dimensional Opacity Functions for Improved Volume Rendering of Seismic Data**. In: Congresso da Sociedade Brasileira de Geofísica, VII, Salvador, 2001
- 13 GOLOSHUBIN, G. M.; KORNEEV, V. A. **Seismic low-frequency effects from fluid-saturated reservoir**, Technical Program Expanded Abstracts: Society of Exploration Geophysicists - SEG, p. 1671-1674, 2000.
- 14 GOLOSHUBIN, G. M.; KORNEEV, V. A.; VINGALOV, V. M. **Seismic low-frequency effects from oil-saturated reservoir zones**, International Exposition and 72nd Annual Meeting: Society of Exploration Geophysicists - SEG, October 2002, Salt Lake City – Utah
- 15 GOLOSHUBIN, G. M.; KORNEEV, V. A.; VINGALOV, V. M.; SILIN, D.; VanSCHUVYER, C. **Reservoir imaging using low frequencies of seismic reflections**. The Leading Edge, Volume 25, Issue 5, p. 527-531, May 2006
- 16 GOLOSHUBIN, G. M. **Using frequency-dependent seismic attributes in imaging of a fractured reservoir zone**, Annual Meeting: Society of Exploration Geophysicists - SEG, 2005, p. 1417 - 1421, Houston – Texas
- 17 HADWIGER, M.; KNISS, J. M.; ENGEL, K.; REZK-SALAMA, C. **High-Quality Volume Graphics on Consumer PC Hardware**, In: SIGGRAPH 2002, Course Notes, No.42 (2002)
- 18 JOHANN, P; RAGAGNIN, G.; SPINOLA, M. **Spectral Decomposition Reveals Geological Hidden Features in the Amplitude Maps from a Deep Water Reservoir in the Campos Basin**, Technical Program Expanded Abstracts: Society of Exploration Geophysicists - SEG, p. 1740-1743, 2003
- 19 KINDLMANN, G.; DURKIN, J. W. **Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering**. In: IEEE Symposium On Volume Visualization, 1998 North Carolina, p. 79-86. Proceedings
- 20 KNISS, J.; KINDLMANN, G.; HANSEN, C. **Multi-Dimensional Transfer Functions for Interactive Volume Rendering**. IEEE Transactions on Visualization and Computer Graphics, v.8, n.3, p.270-285, July-September 2002
- 21 KORNEEV, V. A.; GOLOSHUBIN, G. M.; DALEY, T. M.; SILIN, D. **Seismic low-frequency effects in monitoring fluid-saturated reservoirs**. Geophysics, v.69, n.2, p.522 - 532, March/April 2004

- 22 NVIDIA, **Possui informações técnicas sobre as placas de vídeo que desenvolve, além de uma área específica para desenvolvedores de aplicações gráficas**, Disponível em <<http://maintenance.nvidia.com/>>, 2007
- 23 OPENGL. The Industry's Foundation for High Performance Graphics. Apresenta a documentação sobre a API do OpenGL, fóruns de discussão e artigos sobre o tema. Disponível em <www.opengl.org>, 2007
- 24 PAIVA, A. C.; SEIXAS, R. B.; GATASS, M. **Introdução à Visualização Volumétrica**, Tecgraf, PUC-Rio, Rio de Janeiro, 1999
- 25 PARK, S. W.; BUDGE, B.; LINSEN, L.; HAMANN, B.; JOY, K.I. **Multi-Dimensional Transfer Functions for Interactive 3D Flow Visualization**. IEEE Computer Society, Proceedings of the 12th Pacific Conference on Computer Graphics and Applications, 2004
- 26 PARTYKA, G.; GRIDLEY, J.; LOPEZ, J. **Interpretational applications of spectral decomposition in reservoir characterization**. The Leading Edge, Volume 18, Issue 3, p. 353-361, March 1999
- 27 ROST, R. J. **OpenGL Shading Language**, Second Edition, Addison Wesley, 2006, 740 p.
- 28 RUTHNER, M. P. **Aplicação da Transformada S na Decomposição espectral de Dados Sísmicos**, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro-RJ, 2004
- 29 SGI, **Silicon Graphics**, Disponibiliza informações sobre os hardwares desenvolvidos, além de possuir uma área destinada a desenvolvedores de aplicações gráficas. Disponível em <<http://www.sgi.com/>>, 2007
- 30 SILVA, P. M. C. **Visualização Volumétrica de Horizontes em Dados Sísmicos 3D**, Tese de doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro-RJ, 2004
- 31 SUN, Sun Microsystems, Possui softwares para downloads, artigos sobre as tecnologias que desenvolve. Disponível em <<http://br.sun.com/>>, 2007

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)