Aspects of 3-D Seismic Data Volume Rendering

by

André Gerhardt, Anselmo Paiva, Ana Elisa Schmidt, Luiz Fernando Martha, Paulo Cezar Carvalho, and Marcelo Gattass

June 5, 1998
• Aspects of 3-D Seismic Data
• Quantization Effects on Visualization
• Seismic Volume Visualization Requisites
• Visualization Methods
• Conclusions
Aspects of 3-D Seismic Data

- Physical Meaning
- Noise Content
- Segmentation Based on Spatial Coherence
- Data Representation
Seismic Volume Visualization Requisites

- Interactive segmentation
- Multi-scale transfer functions
- Improved illumination model
- Interactive feedback
- Asymmetric sampling rate
- Control of sample representation
Quantization

- Quantization here is a process of mapping from a continuous set of real amplitude values to a smaller set of discrete values
- lossy compression operation
- accounts for strong reduction in the dynamic range of the data
Histogram and Quantization
Opacity Transfer Function
Land data images produced by the ray casting algorithm

Floating-point volume  8-bit data
Opacity Transfer Function
Transfer Function Sensitivity

Transfer A

Transfer B
Quantization Bin

Amplitudes

- Target Signal
- Random Noise
- Other Signals
- Quantization Bin
Visualization Methods

- Illumination Models
- Algorithms
- Ray Casting
- Shear-Warp Factorization
- 3-D Texture Mapping
- Adequacy to the Visualization Requisites
Illumination Models

• Physically based models
  – Semi-transparent gel
  – light emitting particles

• Problem
  – Gradient estimation
\[ C_\lambda = C_{amb_\lambda} + \sum_{luz} C_{luz_\lambda} \left( k_{dif_\lambda} (N \cdot L) + k_s (R \cdot V)^n \right) \]
Gradient Estimation

\[ \nabla A(X) = \left( \begin{array}{c} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \\ \frac{\partial A}{\partial z} \end{array} \right) \]

\[ \nabla A(X) \approx \left( \begin{array}{ccc} \frac{A_{i+1,j,k} - A_{i-1,j,k}}{2Dx} & \frac{A_{i,j+1,k} - A_{i,j-1,k}}{2Dy} & \frac{A_{i,j,k+1} - A_{i,j,k-1}}{2Dz} \end{array} \right) \]

6 neighbors

26 neighbors
Seismic Volume Rendering

impedance

amplitude

gradient norm

(central difference)
Amplitude Field and Gradient Estimation
Absorption Only Model

\[ \Delta I = -\frac{NA}{E} I = -\frac{(\rho(s)E\Delta s)A}{E} I = -\rho(s)AI\Delta s = -\tau(s)I\Delta s \]

\[ \frac{dI}{ds} = -\tau(s)I \quad \rightarrow \quad I(s) = I_0 e^{-\int_0^s \tau(s) ds} \]
Emission Only Model

N particles
A = \pi R^2
\rho = \text{density} = \frac{N}{V_{cil}} = \frac{N}{E \Delta s}
C = \text{emission}
Ray Casting
Ray Casting

- Para cada *pixel*, computa cor/opacidade
  - Raio paralelos são disparados de cada *pixel* através do volume

✓ Opacidades e cores ao longo do raio são acumuladas para gerar a cor final do *pixel*
Dado Volumétrico

Preparação dos Dados

Valores de voxels

Shading  Classification

Cor  Opacidade

Composição

Cor dos pixels

Rendering
\[ \alpha_{out} = \alpha_{in} + (1 - \alpha_{in}) \alpha_v \]
\[ C_{out} = C_{in} + (1 - \alpha_{in}) \alpha_v C_v \]
\[ C = C_{out} \alpha_{out} \]
Conceptual Scheme for Shear-Warp

1. shear & resample
2. project & composite
3. warp & resample

intermediate image scanline
voxel scanline
intermediate image
final image
Shear-Warp Factorization Steps

\[
M_{\text{view}} = M_{\text{warp2D}} \times M_{\text{shear3D}}
\]
Shear-Warp Optimization with RLE

Resampling and Composition

voxel scanline

intermediate image scanline

Voxel with $\alpha = 0$

Voxel with $\alpha > 0$

Opaque pixel

Non opaque pixel
[LAC95] Lacroute, Philippe.

Fast volume rendering using a shear-warp factorization of the viewing transformation.


Computer Systems Laboratory Departments of Electrical Engineering and Computer Science. Stanford University.

September/1995.
3-D Texture Mapping

Steps for rendering a volume using 2D textures:

- Load the volume data into a 3D texture.
- Choose the number of slices.
- Find the desired viewpoint and view direction.
- Compute polygons that cut through the data perpendicular to the view.
- Use the texture transform matrix to set the desired orientation of the images on the slices.
- Render each slice as a textured polygon, from back to front. A blend operation is performed at each slice; the type of blend depends on the desired effect.
- As the viewpoint and direction of view changes, recompute the data slice positions and update the texture transformation matrix as necessary.