

# Técnicas de Visualização para Dados Espaciais

SCC5836/SCC0252 – Visualização Computacional

Profa. Maria Cristina  
*cristina@icmc.usp.br*

Instituto de Ciências Matemáticas e de Computação (ICMC)  
Universidade de São Paulo (USP)

**VICG** Grupo de Visualização,  
 **Imagens e Computação Gráfica**

# Sumário

- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas
- 7 Referências

- Visualização de dados espaciais: pressupõe que dados têm **atributos espaciais** ou **espaço-temporais** (implícitos ou explícitos)
  - Natural é mapear os atributos espaciais em canais de posição espacial (**geometria é dada**)
  - Facilita a construção e interpretação das representações gráficas
  
- Técnicas oriundas da disciplina conhecida como **Visualização Científica**
  - dados resultantes de processos de amostragem/simulação
  - associados a uma grade espacial (1D, 2D, 3D) (regular ou irregular)

- Pode-se considerar as técnicas em função da dimensionalidade espacial dos dados de entrada:
  - grade unidimensional
  - grade bidimensional
  - grade tridimensional
- Outra possibilidade seria considerar o tipo de dado (atributo)
  - escalar
  - vetorial

# Introdução

- Pode-se considerar as técnicas em função da dimensionalidade espacial dos dados de entrada:
  - grade unidimensional
  - grade bidimensional
  - grade tridimensional
- Outra possibilidade seria considerar o tipo de dado (atributo)
  - escalar
  - vetorial

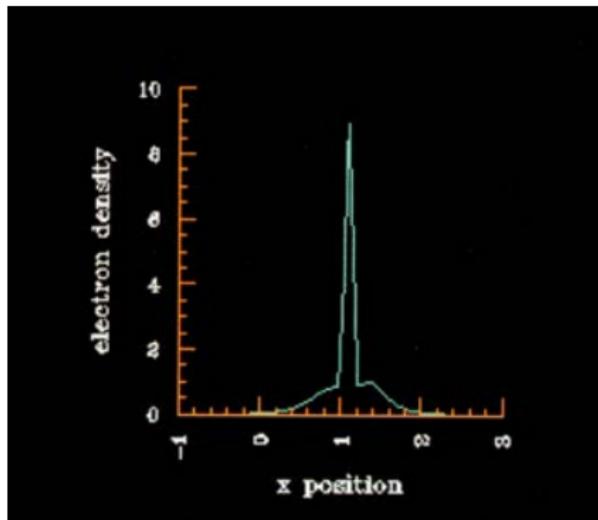
- Atributos espaciais mapeados na geometria da visualização
- **Atributos não espaciais** podem ser **mapeados** para cor, textura, formato, etc. dos marcadores gráficos
- Dados univariados: várias técnicas disponíveis para expressar um único atributo escalar

# Sumário

- 1 Introdução
- 2 Dados Unidimensionais**
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas
- 7 Referências

# Dados Unidimensionais

- Dado um conjunto de valores amostrados no espaço **unidimensional**: informação espacial mapeada em **uma das dimensões da tela** e valor associado mapeado na **outra dimensão**
  - Em geral dados precisam ser **escalados**



# Sumário

- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais**
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas
- 7 Referências

# Dados Bidimensionais

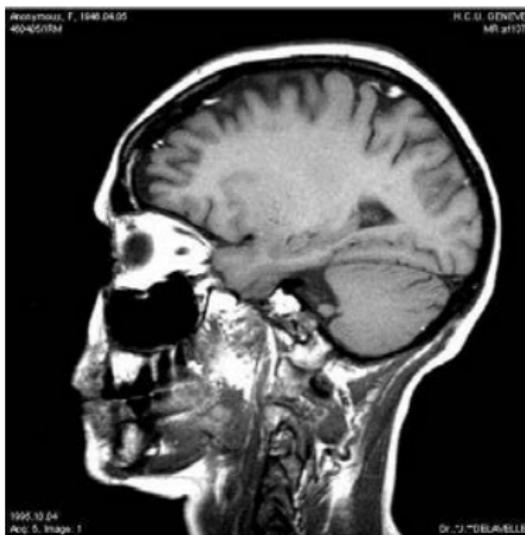
- Dados com **duas dimensões espaciais**: solução natural é mapeá-las nas **duas dimensões da tela**
- Valores em uma grade 2D (regular ou irregular)

# Dados Bidimensionais

- Dados com **duas dimensões espaciais**: solução natural é mapeá-las nas **duas dimensões da tela**
  - Valores em uma grade 2D (regular ou irregular)
- 
- O resultado pode ser
    - Imagem
    - Mapa de altura
    - Paisagem
    - Mapa
    - Contorno e isovalor

# Dados Bidimensionais

- **Imagem:** um **único valor** associado a cada posição espacial
- valores conhecidos em uma grade
- valor mapeado para intensidade (ou cor), cor dos pixels intermediários por interpolação



- **Mapa de altura:** cada valor mapeado na altura do ponto correspondente na grade 2D, formando uma **superfície 3D**



Figura: Superfície do oceano na região da Flórida.

# Dados Bidimensionais

- **Paisagem:** objetos 3D posicionados sobre a grade, os valores dos dados são mapeados nos atributos dos objetos

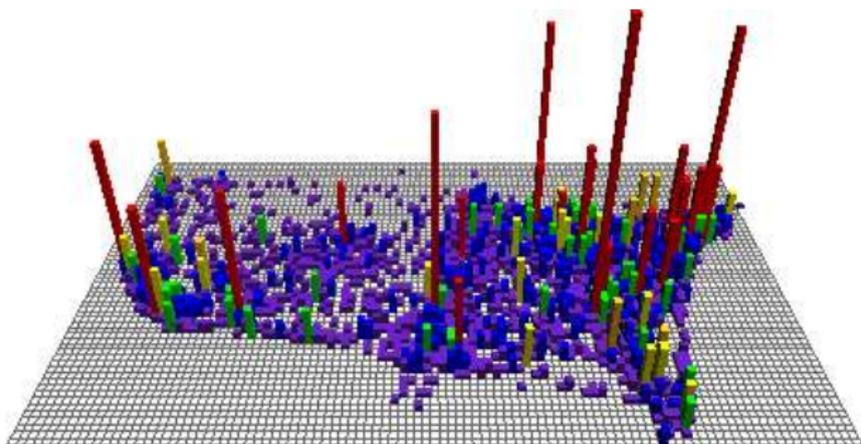


Figura: Paisagem: densidade do tráfego aéreo nos EUA em um certo período.

# Dados Bidimensionais

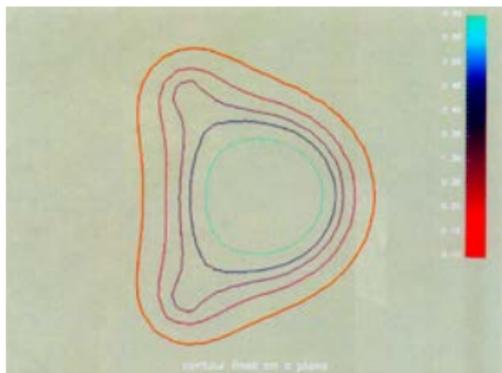
- **Mapa:** dados incluem *features* **lineares** e de **área** (contornos), bem como **objetos pontuais** (pontos de interesse)
- características próprias: visualização de dados geoespaciais



Figura: Mapa da área de São Francisco gerado pelo Google Maps

# Dados Bidimensionais

- **Mapa de contorno (isovalor):** isovalores descrevem fronteira de região 2D determinada por valores idênticos
- função contínua foi amostrada na região 2D, gerando os pontos em uma grade
- algoritmo clássico é o (*marching squares*)



**Figura:** Linhas de contorno extraídas de uma imagem de uma fatia de uma molécula de hidrogênio.

# Marching Squares

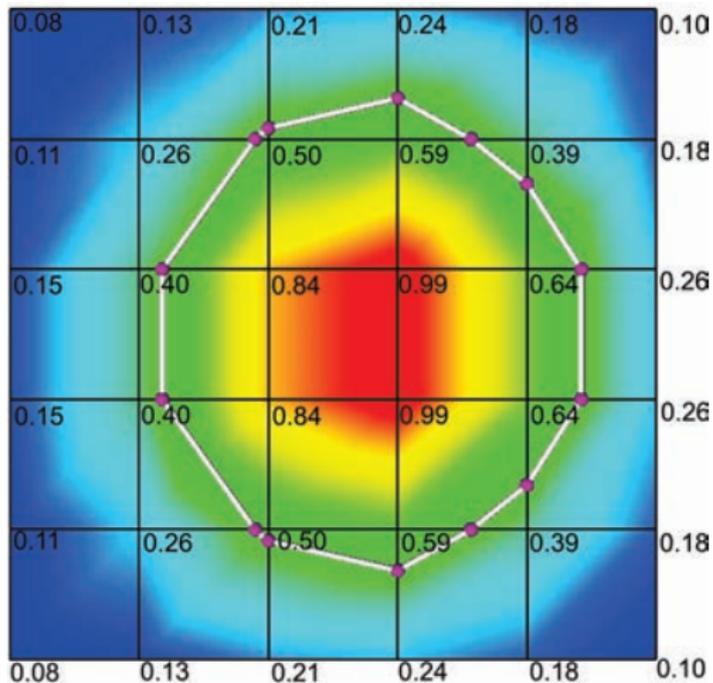


Figura: Construindo uma isolinha para um isovalor  $v = 0.48$

# Marching Squares

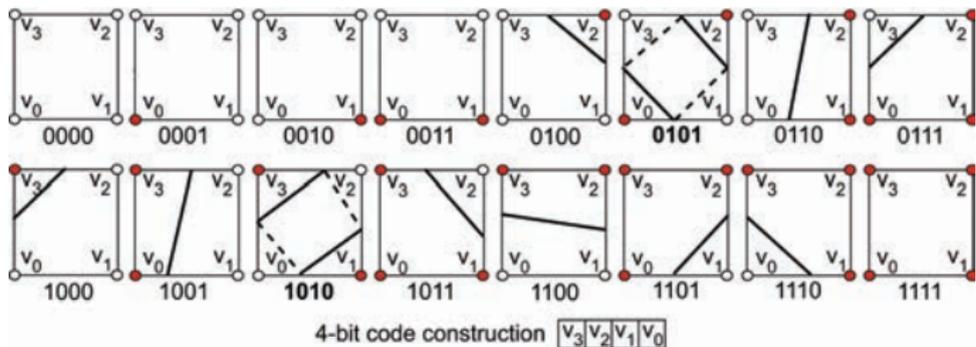


Figura: Estados topológicos de uma célula no algoritmo *Marching Squares*. Vértices “interiores” indicados em vermelho, estados ambíguos em negro.

# Marching Squares

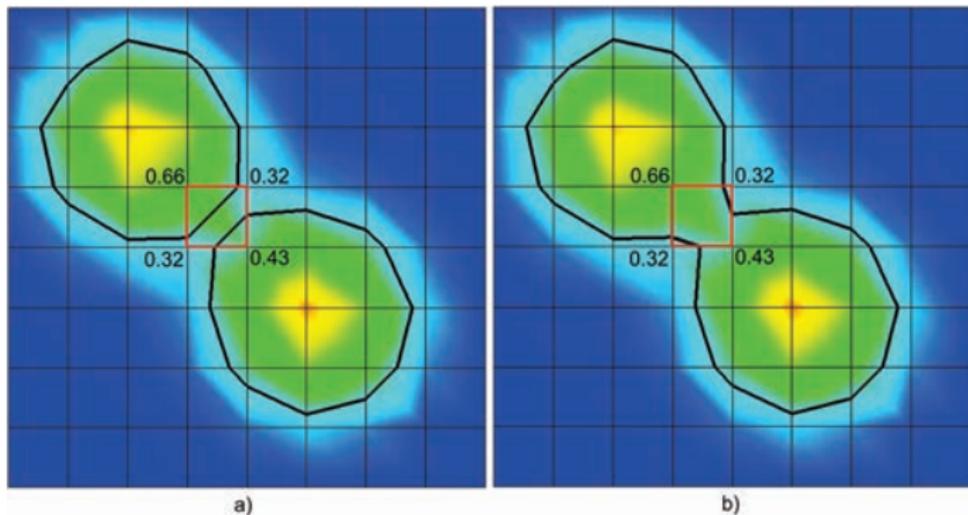


Figura: Problemas de ambiguidade na geração da isolinha: isovalor  $v = 0.37$ .

# Dados Bidimensionais

- Cenários anteriores consideram uma única variável escalar, amostrada em 2D (dados univariados)
- Se os dados descrevem múltiplas variáveis (dados multivariados, ou multidimensionais), as técnicas podem ser expandidas utilizando estratégias de **justaposição** e **sobreposição**

- Cenários anteriores consideram uma única variável escalar, amostrada em 2D (dados univariados)
- Se os dados descrevem múltiplas variáveis (dados multivariados, ou multidimensionais), as técnicas podem ser expandidas utilizando estratégias de **justaposição** e **sobreposição**
  - **Justaposição**: empilhar visualizações univariadas 2D, resultando em uma visualização 3D. Problemas com oclusão.

- Cenários anteriores consideram uma única variável escalar, amostrada em 2D (dados univariados)
- Se os dados descrevem múltiplas variáveis (dados multivariados, ou multidimensionais), as técnicas podem ser expandidas utilizando estratégias de **justaposição** e **sobreposição**
  - **Justaposição**: empilhar visualizações univariadas 2D, resultando em uma visualização 3D. Problemas com oclusão.
  - **Sobreposição**: sobrepor diferentes atributos. P.ex., na visualização de paisagem pode-se sobrepor blocos de diferentes cores, cada cor representando um atributo.

# Dados Bidimensionais

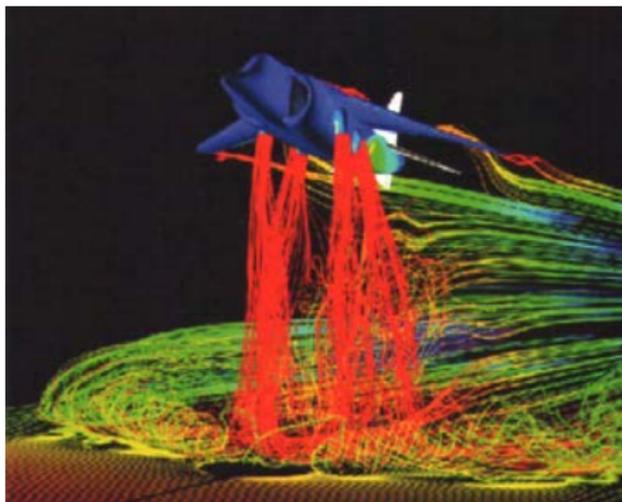
- Cenários anteriores consideram uma única variável escalar, amostrada em 2D (dados univariados)
- Se os dados descrevem múltiplas variáveis (dados multivariados, ou multidimensionais), as técnicas podem ser expandidas utilizando estratégias de **justaposição** e **sobreposição**
  - **Justaposição**: empilhar visualizações univariadas 2D, resultando em uma visualização 3D. Problemas com oclusão.
  - **Sobreposição**: sobrepor diferentes atributos. P.ex., na visualização de paisagem pode-se sobrepor blocos de diferentes cores, cada cor representando um atributo.
- Na presença de muitas variáveis (atributos) a solução é recorrer a técnicas de visualização multivariada (não espaciais)

# Sumário

- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais**
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas
- 7 Referências

# Introdução

- Dados científicos são, em geral, obtidos a partir de processos de amostragem/simulação de fenômenos contínuos
- Dados são **amostras discretas** do processo, coletadas (ou computadas) em uma grade 3D (**visualização volumétrica**)
  - Unidade básica de discretização: **voxel**



- Categorização de técnicas clássicas de **visualização volumétrica**:

- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos

- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos
  - **Técnicas de isosuperfície**: geram superfície (malha de polígonos) relativa a um valor dado, aplicam *rendering* de superfícies

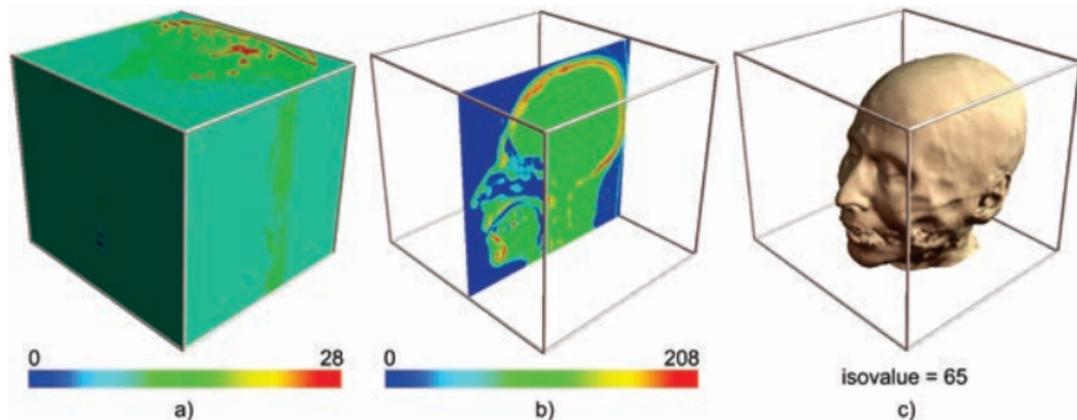
- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos
  - **Técnicas de isosuperfície**: geram superfície (malha de polígonos) relativa a um valor dado, aplicam *rendering* de superfícies
  - **Rendering volumétrico direto**: renderizam uma imagem diretamente a partir do volume

- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos
  - **Técnicas de isosuperfície**: geram superfície (malha de polígonos) relativa a um valor dado, aplicam *rendering* de superfícies
  - **Rendering volumétrico direto**: renderizam uma imagem diretamente a partir do volume
  - Para calcular a cor dos pixels:

- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos
  - **Técnicas de isosuperfície**: geram superfície (malha de polígonos) relativa a um valor dado, aplicam *rendering* de superfícies
  - **Rendering volumétrico direto**: renderizam uma imagem diretamente a partir do volume
  - Para calcular a cor dos pixels:
    - lança raios (a partir do pixel) no volume, amostragem dos diferentes materiais determina composição da cor do pixel; ou

- Categorização de técnicas clássicas de **visualização volumétrica**:
  - **Técnicas de fatiamento (*slicing*)**: exibem dados relativos a um plano de recorte 2D, alinhado a um dos 3 eixos
  - **Técnicas de isosuperfície**: geram superfície (malha de polígonos) relativa a um valor dado, aplicam *rendering* de superfícies
  - **Rendering volumétrico direto**: renderizam uma imagem diretamente a partir do volume
  - Para calcular a cor dos pixels:
    - lança raios (a partir do pixel) no volume, amostragem dos diferentes materiais determina composição da cor do pixel; ou
    - projeta conteúdo do volume em um plano, com alguma estratégia para compor a cor dos pixels

# Visualizando Dados Volumétricos



**Figura:** Visualizando um conjunto de dados volumétrico. (a) desenho da superfície (b) plano de recorte (c) isosuperfície.

# Fatiando Dados Volumétricos com Planos de Recorte

- No fatiamento, **planos de recorte** reduzem os dados: 3D  $\rightarrow$  2D

# Fatiando Dados Volumétricos com Planos de Recorte

- No fatiamento, **planos de recorte** reduzem os dados: 3D  $\rightarrow$  2D
  - Volume recortado por um plano, dada uma **direção e posição**

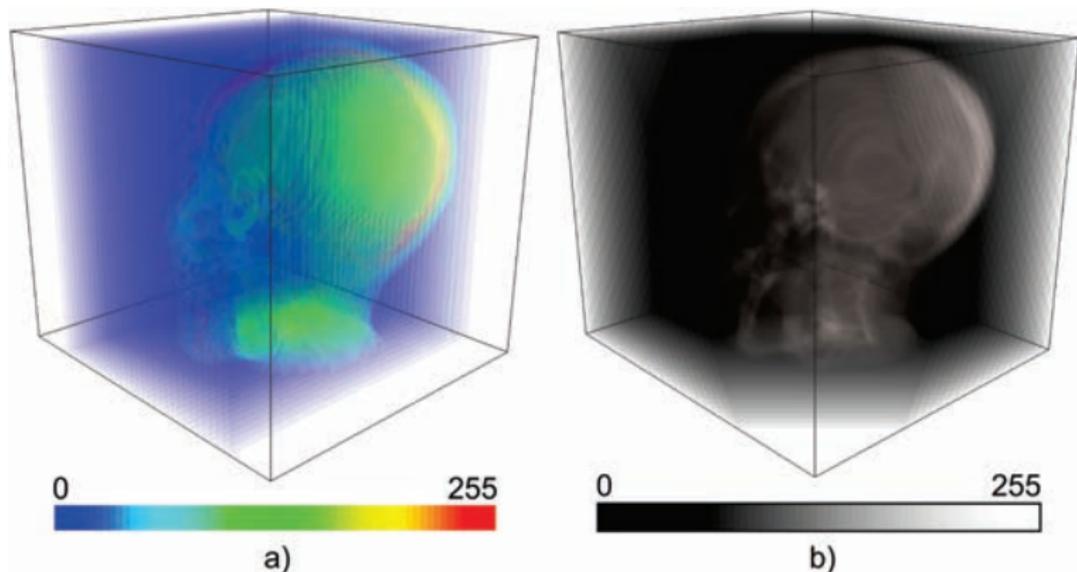
# Fatiando Dados Volumétricos com Planos de Recorte

- No fatiamento, **planos de recorte** reduzem os dados: 3D  $\rightarrow$  2D
  - Volume recortado por um plano, dada uma **direção e posição**
  - Implementação simples: **normal ao plano** coincide com **direções principais do volume**

# Fatiando Dados Volumétricos com Planos de Recorte

- No fatiamento, **planos de recorte** reduzem os dados: 3D  $\rightarrow$  2D
  - Volume recortado por um plano, dada uma **direção e posição**
  - Implementação simples: **normal ao plano** coincide com **direções principais do volume**
  - **Animação** para visualizar associações entre **fatias subsequentes**

# Fatiando Dados Volumétricos com Planos de Recorte



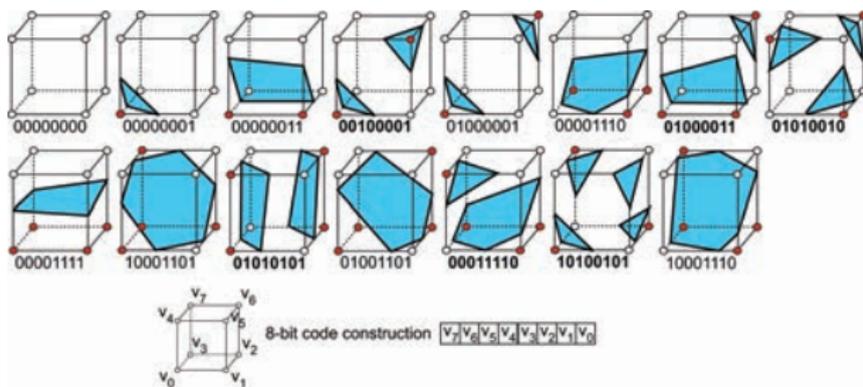
**Figura:** Visualização de um conjunto de dados volumétrico. (a) fatias alinhadas com volume (b) fatias alinhadas com a direção de observação.

# Marching Cubes para Extração de Isosuperfícies

- Outra maneira de exibir dados 3D é por meio da extração de **isosuperfícies**
  - Dado um valor de interesse, a superfície correspondente é extraída e exibida
  - Superfície descrita por uma malha de polígonos (em geral, triângulos) e renderizada com técnicas clássicas de *rendering* de superfícies
- 
- **Marching Cubes**: algoritmo mais conhecido
    - Voxel definido pelos valores associados aos seus 8 vértices
    - A contribuição do voxel na construção da superfície depende da relação entre cada um desses valores e o isovalor de interesse

# Marching Cubes para Extração de Isosuperfícies

- O algoritmo começa identificando o **estado topológico** de cada voxel, com base nos valores associados aos seus **vértices**
  - Existem  $2^8 = 256$  estados possíveis, mas somente 15 únicos – cuidado com as normais aos planos



**Figura:** Estados topológicos de um voxel. Vermelho indica vértice “interior”, casos ambíguos em negro.

# Marching Cubes para Extração de Isosuperfícies

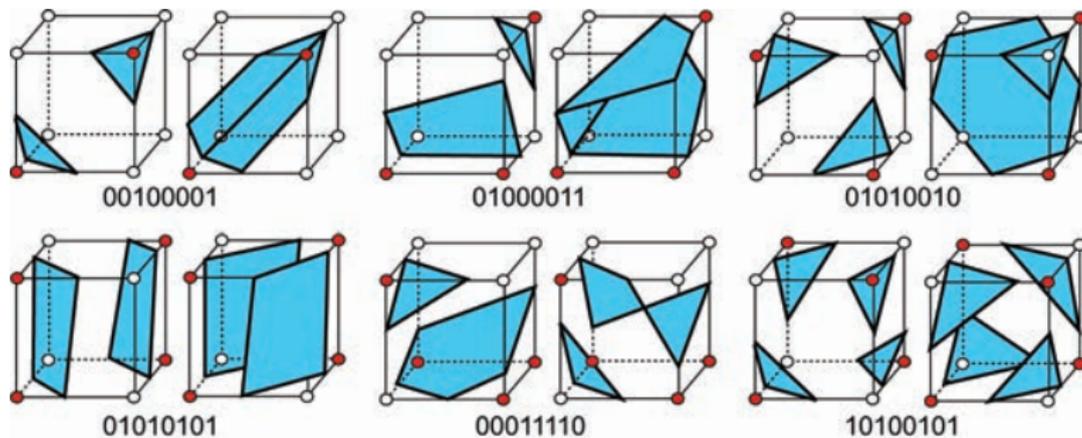


Figura: Casos ambíguos para o algoritmo *Marching Cubes*.

# Marching Cubes para Extração de Isosuperfícies

- Uma vez determinados os estados topológicos, as intersecções da superfície **com as arestas dos voxels** são calculadas, e assim identificados os triângulos que compõem a malha
  - simplificação possível: considera que a intersecção sempre ocorre no meio da aresta

# Marching Cubes para Extração de Isosuperfícies

## Problemas e Alternativas

- Consumo de memória para armazenar o volume
- Consumo de memória para armazenar a malha
- Qualidade da malha gerada
  - Definir estruturas de dados que compartilhem arestas e vértices, ou unir triângulos coplanares em faces maiores

# Marching Cubes para Extração de Isosuperfícies

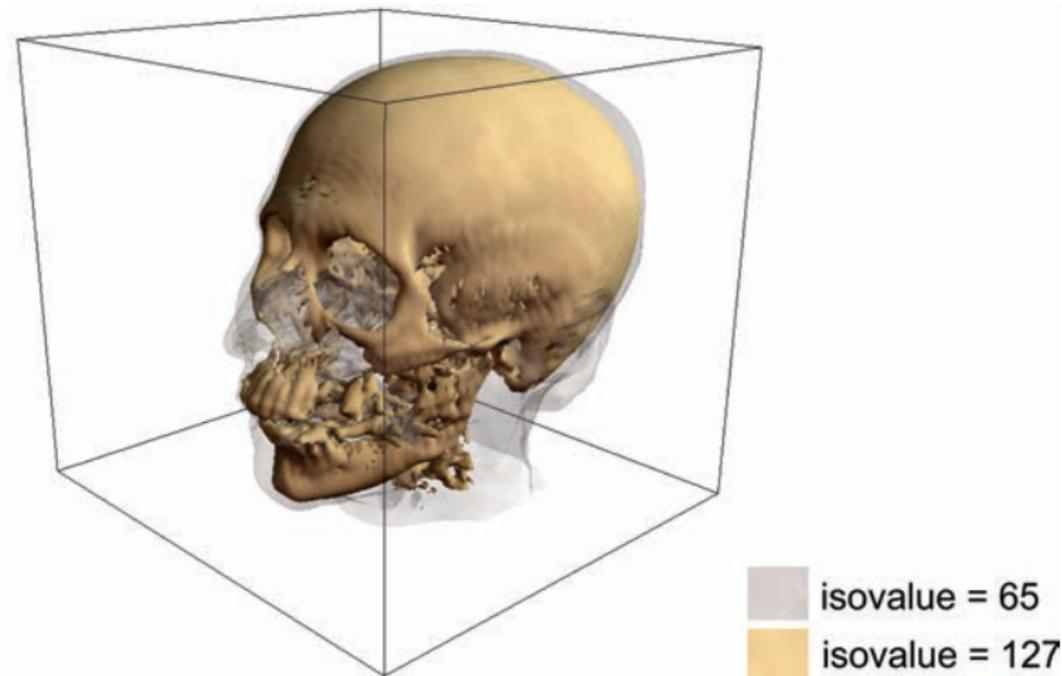


Figura: Visualização de duas isosuperfícies.

# Visualização Volumétrica Direta

- No **Rendering Volumétrico Direto** os pixels da imagem resultante são computados diretamente, sem criar polígonos

# Visualização Volumétrica Direta

- No **Rendering Volumétrico Direto** os pixels da imagem resultante são computados diretamente, sem criar polígonos
  - mapeamento direto do conteúdo dos voxels em pixels: *rendering* do volume

# Visualização Volumétrica Direta

- No **Rendering Volumétrico Direto** os pixels da imagem resultante são computados diretamente, sem criar polígonos
  - mapeamento direto do conteúdo dos voxels em pixels: *rendering* do volume
- Existem dois tipos de mapeamento

# Visualização Volumétrica Direta

- No **Rendering Volumétrico Direto** os pixels da imagem resultante são computados diretamente, sem criar polígonos
  - mapeamento direto do conteúdo dos voxels em pixels: *rendering* do volume
- Existem dois tipos de mapeamento
  - **Mapeamento direto: projeta cada voxel** no plano de projeção e determina quais pixels são afetados, e como

# Visualização Volumétrica Direta

- No **Rendering Volumétrico Direto** os pixels da imagem resultante são computados diretamente, sem criar polígonos
  - mapeamento direto do conteúdo dos voxels em pixels: *rendering* do volume
- Existem dois tipos de mapeamento
  - **Mapeamento direto: projeta cada voxel** no plano de projeção e determina quais pixels são afetados, e como
  - **Mapeamento inverso (*ray casting*): lança um raio a partir de cada pixel** no plano de projeção, os raios percorrem o volume, valores são amostrados ao longo do caminho e combinados para determinar o valor do pixel

# Visualização Volumétrica Direta

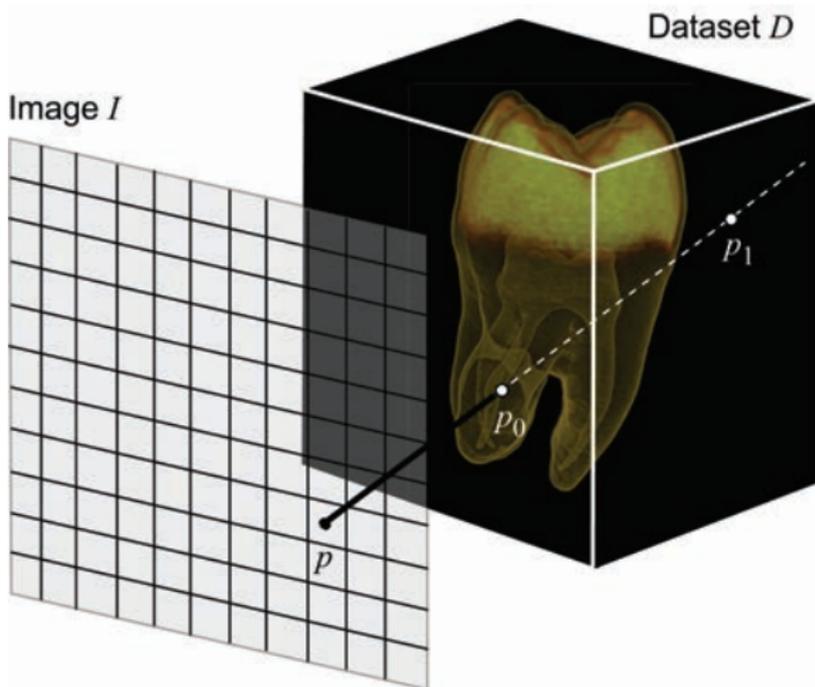


Figura: Princípio conceitual da visualização volumétrica.

# Visualização Volumétrica Direta

- **Composição** para tratar pixels afetados por **múltiplos voxels** (MD) e **múltiplos valores** encontrados ao longo do caminho de um raio

# Visualização Volumétrica Direta

- **Composição** para tratar pixels afetados por **múltiplos voxels** (MD) e **múltiplos valores** encontrados ao longo do caminho de um raio
- Cada voxel tem uma opacidade, opacidades são combinadas para determinar a cor do pixel
  - Determinar a **transparência acumulada** entre o plano de projeção e o voxel ( $\prod_{j=0}^{i-1} (1 - o_j)$ ) e usar essa informação para **ajustar a intensidade** do pixel ( $c_i \times o_i$ ) (cor e opacidade)

# Visualização Volumétrica Direta

- **Composição** para tratar pixels afetados por **múltiplos voxels** (MD) e **múltiplos valores** encontrados ao longo do caminho de um raio
- Cada voxel tem uma opacidade, opacidades são combinadas para determinar a cor do pixel
  - Determinar a **transparência acumulada** entre o plano de projeção e o voxel ( $\prod_{j=0}^{i-1} (1 - o_j)$ ) e usar essa informação para **ajustar a intensidade** do pixel ( $c_i \times o_i$ ) (cor e opacidade)
- Vor final de um pixel dada por:

$$I(x, y) = \sum_{i=0}^n c_i \times o_i \times \prod_{j=0}^{i-1} (1 - o_j)$$

- em que  $c_i$  e  $o_i$  são cor e opacidade do voxel  $i$ , respectivamente

# Visualização Volumétrica Direta

- **Problema de classificação:** como mapear um valor associado a um voxel nos canais de opacidade e de cor (RGB)

# Visualização Volumétrica Direta

- **Problema de classificação:** como **mapear** um **valor associado a um voxel** nos **canais de opacidade e de cor** (RGB)
- Mapeamento definido por **Funções de transferência**
  - Pode-se, p.ex., analisar voxels em que acontecem transições significativas de valor (mudança de cor e opacidade)
  - Definir tais funções não é trivial: pode demandar interação do especialista

# Visualização Volumétrica Direta

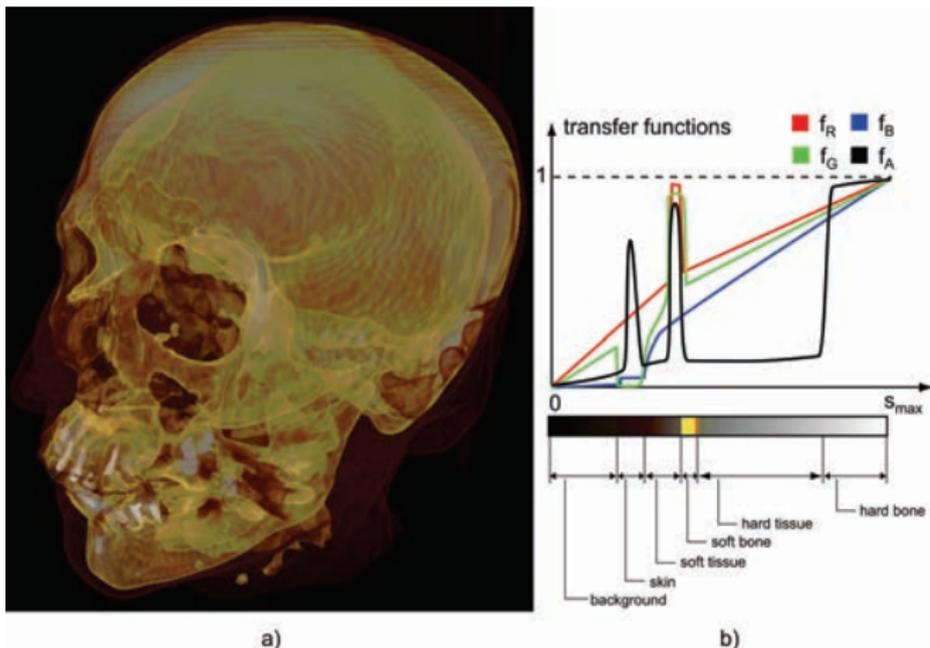


Figura: (a) rendering de volume representando uma cabeça. (b) função de transferência que enfatiza tecido mole e as partes mole e dura dos ossos.

# Visualização Volumétrica Direta

- Outro **problema** é o cálculo da **iluminação e tonalização** – não existem normais
  - Pode-se considerar o **gradiente** (taxa de mudança) em cada direção do volume

# Visualização Volumétrica Direta

- Outro **problema** é o cálculo da **iluminação e tonalização** – não existem normais
  - Pode-se considerar o **gradiente** (taxa de mudança) em cada direção do volume
- Gradiente na direção  $x$  de um voxel  $(v_x, v_y, v_z)$  pode ser calculado com base nos seus vizinhos  $(v_{x-1}, v_y, v_z)$  e  $(v_{x+1}, v_y, v_z)$

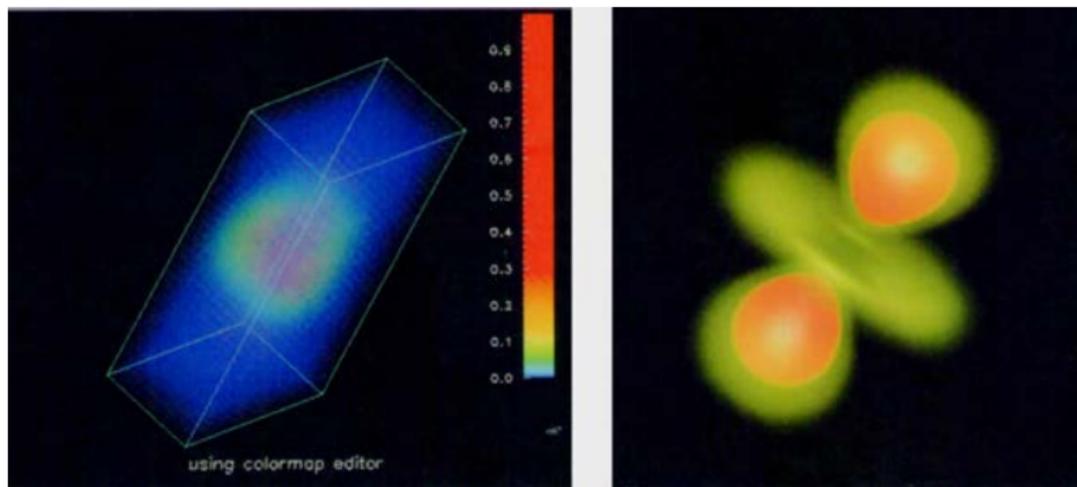
# Visualização Volumétrica Direta

- Outro **problema** é o cálculo da **iluminação e tonalização** – não existem normais
  - Pode-se considerar o **gradiente** (taxa de mudança) em cada direção do volume
- Gradiente na direção  $x$  de um voxel  $(v_x, v_y, v_z)$  pode ser calculado com base nos seus vizinhos  $(v_{x-1}, v_y, v_z)$  e  $(v_{x+1}, v_y, v_z)$ 
  - Uma maneira simples de determinar  $g_x$ , o componente  $x$  do gradiente, é fazer  $v_x - v_{x-1}$  (**operador intermediário de diferença**)

# Visualização Volumétrica Direta

- Outro **problema** é o cálculo da **iluminação e tonalização** – não existem normais
  - Pode-se considerar o **gradiente** (taxa de mudança) em cada direção do volume
- Gradiente na direção  $x$  de um voxel  $(v_x, v_y, v_z)$  pode ser calculado com base nos seus vizinhos  $(v_{x-1}, v_y, v_z)$  e  $(v_{x+1}, v_y, v_z)$ 
  - Uma maneira simples de determinar  $g_x$ , o componente  $x$  do gradiente, é fazer  $v_x - v_{x-1}$  (**operador intermediário de diferença**)
  - Realizando essa operação nas três direções  $x$ ,  $y$ , e  $z$  tem-se uma aproximação da direção de maior variação do valor do voxel

# Visualização Volumétrica Direta



**Figura:** Exemplos de rendering volumétrico direto. (a) modelo emissivo, sem cálculo de tonalização (b) abordagem baseada em textura incluindo componentes especulares e difusas.

# Sumário

- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos**
- 6 Combinando Técnicas
- 7 Referências

- **Visualização de fluidos** (*flow visualization*): estudo de métodos para a visualização do comportamento dinâmico de líquidos e gases

# Visualização de Fluidos

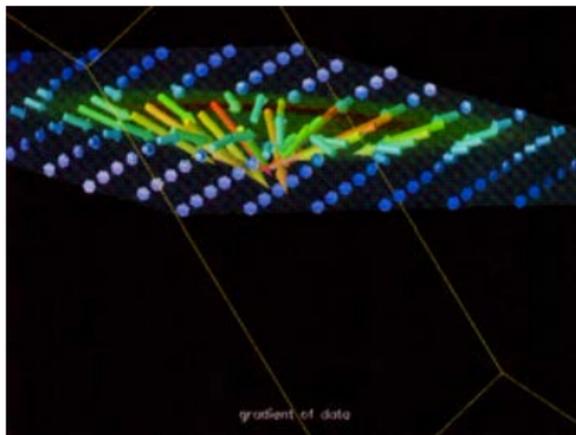
- **Visualização de fluidos** (*flow visualization*): estudo de métodos para a visualização do comportamento dinâmico de líquidos e gases
- **Computational Fluid Dynamics (CFD)** define métodos capazes de simular fluidos em uma vasta gama de condições
  - Simulações computadas em grades 2D ou 3D: **vetores de velocidade** (dados vetoriais)
  - Análise do campo vetorial para identificar **turbulência, vórtices**, pontos de sela (*saddle points*) e outras estruturas

# Visualização de Fluidos

- **Visualização de fluidos** (*flow visualization*): estudo de métodos para a visualização do comportamento dinâmico de líquidos e gases
- **Computational Fluid Dynamics (CFD)** define métodos capazes de simular fluidos em uma vasta gama de condições
  - Simulações computadas em grades 2D ou 3D: **vetores de velocidade** (dados vetoriais)
  - Análise do campo vetorial para identificar **turbulência**, **vórtices**, pontos de sela (*saddle points*) e outras estruturas
- Campos vetoriais podem ser:
  - **Estáticos**: o campo de velocidade é estável
  - **Variantes no tempo**: vetores de velocidade variam, mas suas posições não; ou valores e posição variam (*unsteady flows*)

# Técnicas de Visualização de Fluidos

- Visualização mais simples: mostrar o campo de velocidade usando glifos representando os vetores
- Outra estratégia é gerar um campo escalar da magnitude, e utilizar técnicas de visualização de **valores escalares**



**Figura:** Visualização de uma nuvem de chuva: setas indicam a direção e força do vento.

# Técnicas de Visualização de Fluidos

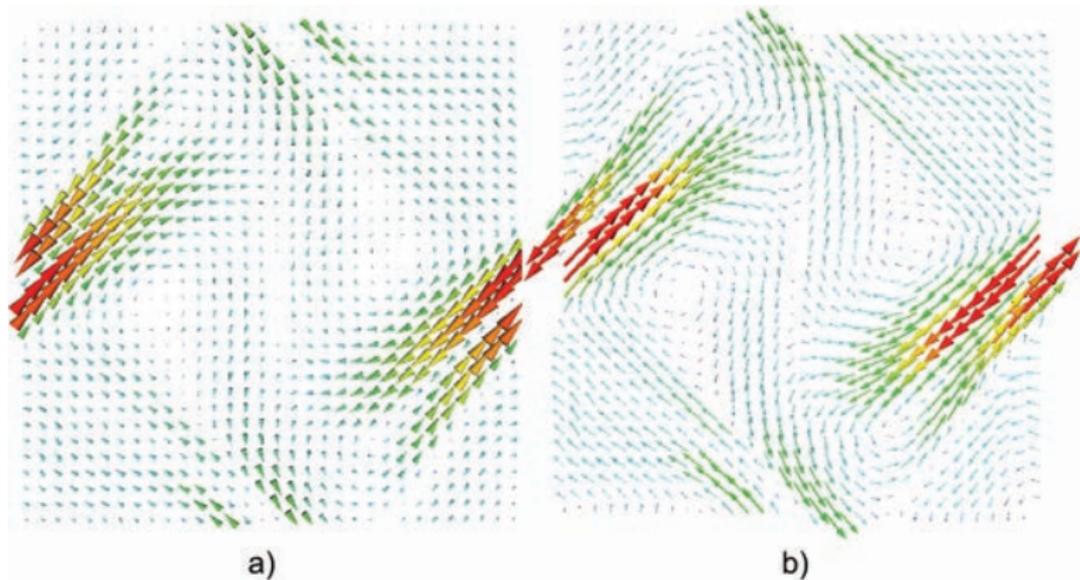


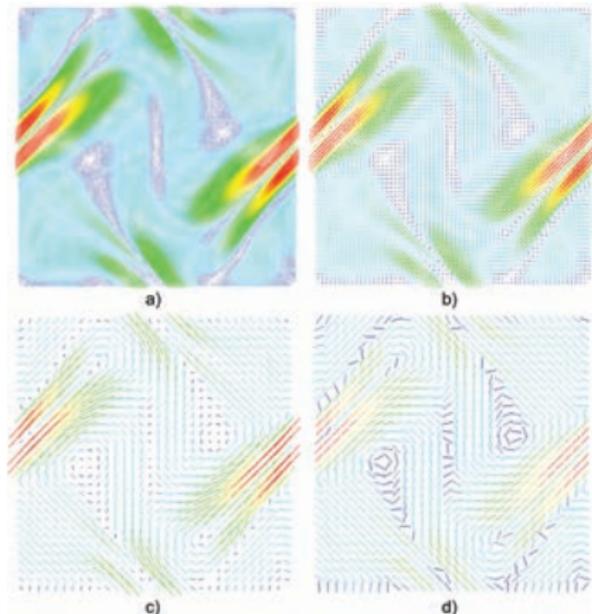
Figura: Diferentes tipos de glifos. (a) cones (b) setas.

- Uma das **dificuldades** é controlar o **número de elementos** exibidos
  - Muitos: **obscurece** a visualização
  - Poucos: características importantes podem **passar despercebidas**

# Técnicas de Visualização de Fluidos

- Uma das **dificuldades** é controlar o **número de elementos** exibidos
  - Muitos: **obscurece** a visualização
  - Poucos: características importantes podem **passar despercebidas**
- Pode ser amenizada com **controle interativo** pelo usuário ou por técnicas que identifiquem **regiões de interesse**

# Técnicas de Visualização de Fluidos

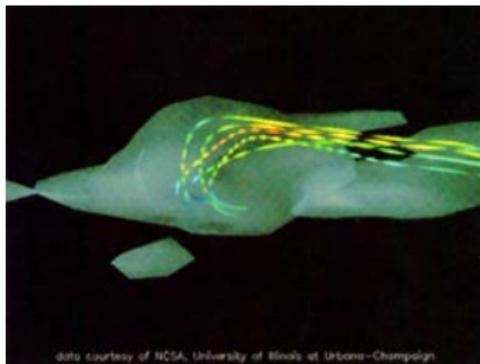


**Figura:** Visualização do campo de velocidade com diferentes amostragem de glifos. (a) metade (b) um quarto e (c) um oitavo. (d) Um quarto com glifos escalados para o mesmo tamanho.

# Técnicas de Visualização de Fluidos

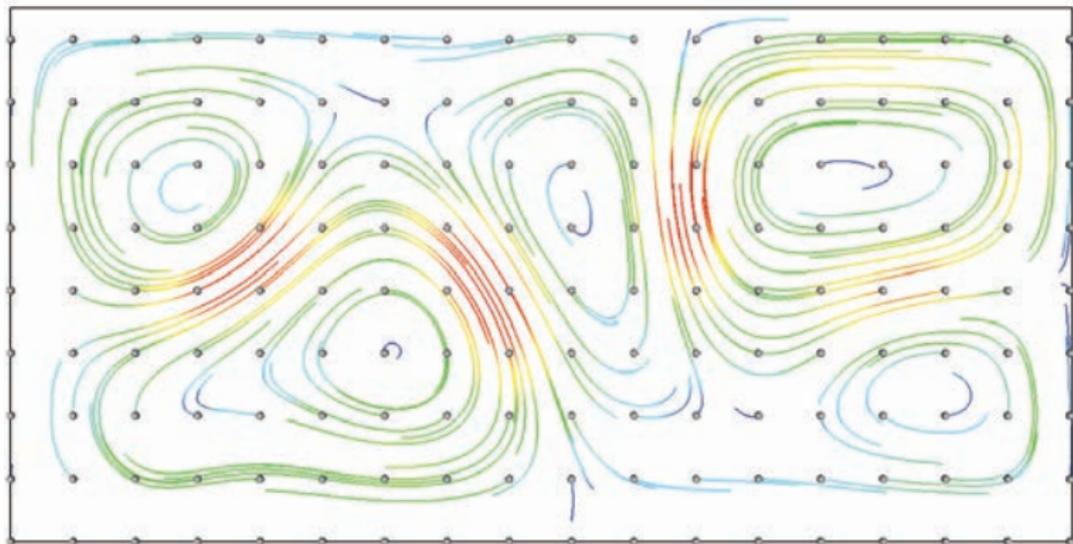
## Streamlines

- Dadas algumas localizações (sementes), calcula para cada semente um caminho tangente ao campo de fluido
  - Caminho pode ser exibido como linhas, ou fitas
  - Outros **atributos**, como magnitude e vorticidade, mapeados em canais de cor, tamanho e torção



**Figura:** Visualização de fluidos usando fitas, com a vorticidade mapeada para a torção da fita.

# Técnicas de Visualização de Fluidos



**Figura:** *Streamlines* em um campo de fluido 2D. Os pequenos círculos cinza indicam as sementes.

# Convolução Integral de Linha

- Outro método interessante para visualização de fluidos é o **Line Integral Convolution (LIC)**
  - Usa um campo **vetorial** e um campo **aleatório** (textura)

# Convolução Integral de Linha

- O valor de cada pixel na imagem é dado pela **média ponderada** de uma sequencia de pixels adjacentes no campo aleatório (da textura), ao longo de um **caminho** linear centrado no pixel e seguindo a *streamline* que passa por ele
  - Pixels ao longo de linhas com vetores similares adjacentes terão somas similares da imagem de textura

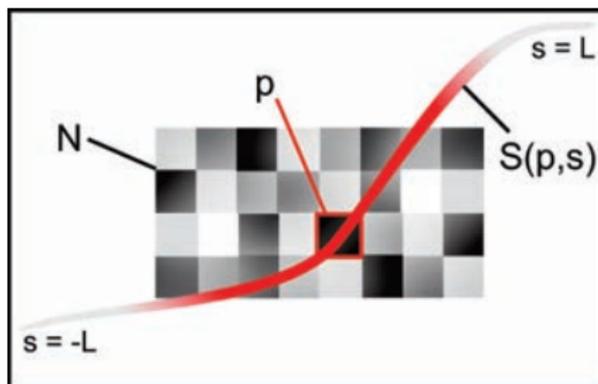


Figura: Ilustração do princípio do LIC.

# Convolução Integral de Linha



**Figura:** A textura à esquerda é combinada com o campo vetorial no centro, criando a imagem à direita.

# Sumário

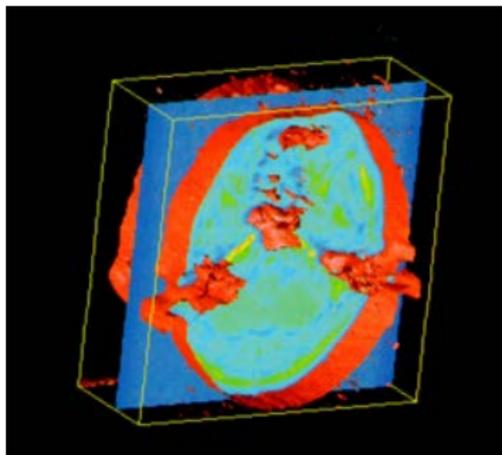
- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas**
- 7 Referências

# Combinando Técnicas

- Desde que seja possível evitar a oclusão, pode ser uma **boa estratégia combinar múltiplas técnicas** para uma visualização mais efetiva
  - Na verdade, pode ser **necessário** em algumas aplicações, p.ex. previsão do tempo: temperatura da superfície, velocidade do vento, umidade relativa, etc.

# Fatias + Isosuperfícies

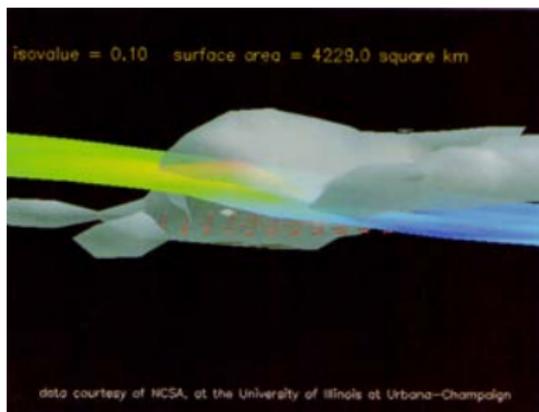
- É possível unir **isosuperfícies** e **fatias** de um conjunto de dados empregando cores diferentes para cada uma das visualizações
  - O mapeamento das cores é o ponto crucial nessa junção



**Figura:** Conjunto de dados volumétricos da área médica: exhibe simultaneamente uma isosuperfície e uma fatia 2D.

# Isosuperfícies + Glifos

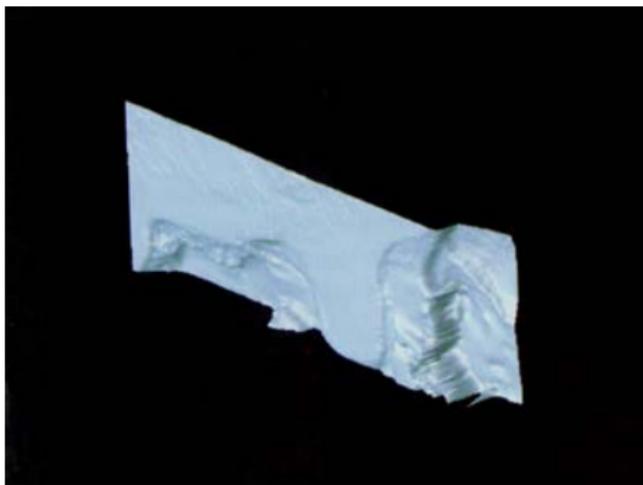
- Para tornar uma **isosuperfície** mais representativa **glifos** podem representar informação adicional sobre os dados



**Figura:** Visualização de uma nuvem de tempestade: combina uma isosuperfície e um plano de recorte mapeando a densidade da água, e glifos mostrando a direção e força do vento.

# Rubber Sheet + Linhas de Contorno + Cor

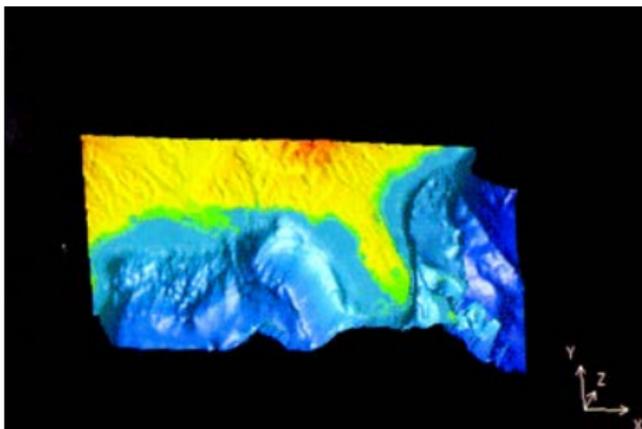
- Mapear dados de maneira **redundante** para diferentes canais visuais pode favorecer a percepção do usuário



**Figura:** Visualização *rubber sheet* da altura acima e abaixo do nível do oceano.

## Rubber Sheet + Linhas de Contorno + Cor

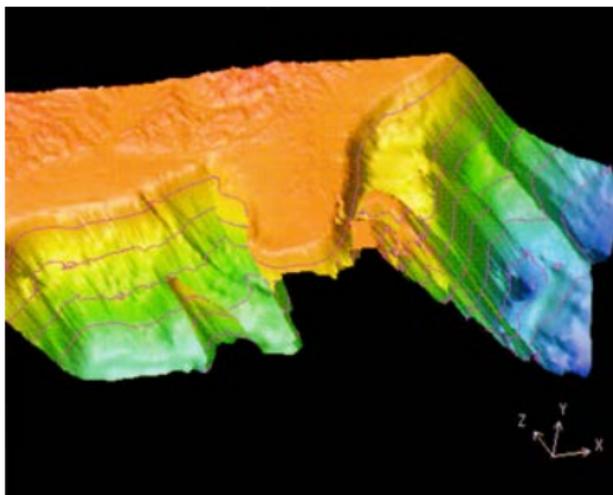
- Mapear dados de maneira **redundante** para diferentes canais visuais pode favorecer a percepção do usuário



**Figura:** Visualização *rubber sheet* da altura acima e abaixo do nível do oceano: usa cor para mapear redundantemente a altura.

# Rubber Sheet + Linhas de Contorno + Cor

- Mapear dados de maneira **redundante** para diferentes canais visuais pode favorecer a percepção do usuário



**Figura:** Zoom da figura anterior com linhas de contorno adicionais (informação do gradiente mais visível).

# Sumário

- 1 Introdução
- 2 Dados Unidimensionais
- 3 Dados Bidimensionais
- 4 Dados Tridimensionais
- 5 Dados Vetoriais: Fluidos
- 6 Combinando Técnicas
- 7 Referências**

# Referências

- Ward, M., Grinstein, G. G., Keim, D. **Interactive data visualization foundations, techniques, and applications.** Natick, Mass., A K Peters, 2010.
- Alexandru C. Telea. **Data Visualization: Principles and Practice.** A K Peters, 2008.