

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Heitor Henrique Martins

Extração de Isosuperfícies a Partir de Dados Meteorológicos

Bauru
2013

Heitor Henrique Martins

Extração de Isosuperfícies a Partir de Dados Meteorológicos

Monografia apresentada na disciplina Trabalho de Conclusão de Curso junto ao Programa de Graduação em Ciência da Computação do Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador: Prof. Dr. José Remo Ferreira Brega

Bauru
2013

HEITOR HENRIQUE MARTINS

Extração de Isosuperfícies a Partir de Dados Meteorológicos

Monografia apresentada na disciplina Trabalho de Conclusão de Curso junto ao Programa de Graduação em Ciência da Computação do Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

BANCA EXAMINADORA

Prof. Dr. José Remo Ferreira Brega
Professor Adjunto
UNESP – Bauru
Presidente da Banca

Prof. Dr.
Professor Adjunto
UNESP – Bauru

Profa. Dra. ??
Professora Assistente
UNESP – Bauru

Bauru, 21 de Novembro de 2013.

Resumo

H. H. Martins. Extração de isosuperfícies a partir de dados meteorológicos. 56 f. Monografia. Curso de Bacharelado em Ciência da Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho". Bauru, 2013.

O presente trabalho tem como objetivo, visualizar as informações meteorológicas por meio da construção de isosuperfícies, possibilitando usufruir das vantagens de modelos geométricos e tridimensionais para transmitir o significado dos dados utilizados de uma forma clara e eficiente. A constante evolução da tecnologia de processamentos de dados, torna possível a interpretação de massas de dados cada vez maiores, por meio de algoritmos robustos. Na meteorologia, em particular, podemos nos beneficiar deste fato, devido à grande quantidade de dados necessária para análises e estatísticas. A manipulação dos dados, por usuários de outras áreas, é facilitada pela escolha do algoritmo e das ferramentas envolvidas neste trabalho. O projeto foi desenvolvido em módulos distintos, aumentando sua flexibilidade e reusabilidade para estudos futuros.

Palavras–chave

Visualização Científica. Modelagem geométrica. Isosuperfície. Marching cubes. X3D. Meteorologia.

Abstract

H. H. Martins. Extraction of isosurfaces from meteorological data. 56 f. Monograph. Curso de Bacharelado em Ciência da Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho". Bauru, 2013.

This work aims viewing weather information, by building isosurfaces enabling enjoy the advantages of three-dimensional geometric models, to communicate the meaning of the data used in a clear and efficient way. The evolving technology of data processing makes possible the interpretation of masses of data increasing, through robust algorithms. In meteorology, in particular, we can benefit from this fact, due to the large amount of data required for analysis and statistics. The manipulation of data, by users from other areas, is facilitated by the choice of algorithm and the tools involved in this work. The project was further developed into distinct modules, increasing their flexibility and reusability for future studies.

Keywords

Scientific visualization. Geometric modeling. Isosurface. Marching cubes. X3D. Meteorology.

Lista de Figuras

Figura 2.1 Extração de uma isosuperfície a partir de um volume de pontos. 12

Figura 2.2 Caso em que duas isosuperfícies são geradas, uma interior e outra exterior. . 13

Figura 2.3 Crânio humano parcial, reconstruído a partir de uma ressonância 14

Figura 2.4 Estrutura molecular demonstrando volume e forma. 15

Figura 2.5 Moléculas de água no interior de biomoléculas. 16

Figura 2.6 Duas aplicações de separação de cargas em uma molécula 16

Figura 2.7 A partir de dados do Ground-Penetrating Radar (GPR), visão de duas pedras enterradas. 17

Figura 2.8 Visualização de uma isosuperfície com informações de pressão e altitude . 18

Figura 2.9 Visualização de uma isosuperfície referente a velocidade do vento 19

Figura 2.10 Isosuperfície sobre a salinidade do oceano Atlântico, nas proximidades da costa leste americana. 20

Figura 2.11 Restrição da triangulação, por meio das circunferências. 21

Figura 2.12 Diagrama de Voronoi em linha contínua, e Triangulação de Delaunay em linha tracejada (LEE; SCHACHTER, 1980). 21

Figura 2.13 Casos de triangulação e escolha dos vértices. 22

Figura 2.14 Reconstrução de uma tomografia, sendo triangulação pura, sombreada e transparente. 23

Figura 2.15 Visualização de um bonsai. 24

Figura 2.16 Visualização de uma perna humana, mostrando a pele, vasos e ossos. . . . 25

Figura 3.1 Aplicação do Algoritmo Marching Cubes em um Plano. 26

Figura 3.2 Representação das 15 configurações iniciais previstas. 27

Figura 3.3 Ordem de leitura e varredura de toda a cena. 28

Figura 3.4 Casos extras, além das configurações iniciais 28

Figura 3.5 Exemplo de uma redundância de pontos. 29

Figura 4.1 Objetos primitivos. 30

Figura 4.2 Regra da mão direita na disposição dos eixos. 33

Figura 4.3 Regra da mão direita na rotação de um objeto. 33
Figura 4.4 Exemplo de uma cena, objeto complexo fictício. 34
Figura 4.5 Exemplo de uma cena, tanques de água. 35

Figura 5.1 Modelagem do comportamento de uma nuvem. 36
Figura 5.2 Extrato de um arquivo GRIB após a decriptografia. 37

Figura 6.1 Diagrama de classes do projeto. 38
Figura 6.2 Visualizador desenvolvido exibindo uma isosuperfície. 39

Lista de Abreviaturas

GM	Gray Matter
WM	White Matter
MC	Marching Cubes
CSF	Cerebrospinal Fluid
CPC	Climate Prediction Center
CT	Computerized Tomography
fMRI	Functional Magnetic Resonance Imaging
GIS	Geographical Information System
GMN	Global Marine Net
GPR	Ground-Penetrating Radar
GRIB	Gridded Binary
MRI	Magnetic Resonance Imaging
NOAA	National Oceanic and Atmospheric Administration
NWS	National Weather Service
PET	Positron Emission Tomography
VRML	Virtual Reality Modeling Language
WMO	World Meteorology Organization
X3D	Extensible 3D
XML	Extensible Markup Language

Sumário

1 Introdução **10**

1.1 Objetivos 11

1.2 Organização da monografia 11

2 Isosuperfície **12**

2.1 Aplicações 13

2.1.1 Medicina 13

2.1.2 Bioquímica 14

2.1.3 Arqueologia 16

2.1.4 Meteorologia 17

2.2 Algoritmos 20

2.2.1 Delaunay triangulation 20

2.2.2 Marching tetrahedra 22

2.2.3 Shrinkwrap 23

2.2.4 Noise 24

2.2.5 Sweeping simplices 25

3 Marching Cubes **26**

3.1 Construção 27

3.2 Casos extras 28

4 X3D **30**

4.1 Funcionalidades 30

5 Grib **36**

6 Implementação **38**

7 Conclusão **40**

7.1 Trabalhos Futuros 40

Referências Bibliográficas **42**

CAPÍTULO 1

Introdução

Meteorologia é o estudo da atmosfera da Terra focada no tempo e na previsão do clima. Para que possam ser aplicados modelos matemáticos e análises estatísticas a estes estudos, é necessária uma grande quantidade de dados variados e uma ideia de como esses dados se relacionam ao passar do tempo.

Com o aumento da capacidade de processamento computacional e de aquisição de dados, cientistas conseguem gerar uma grande quantidade de dados provenientes de simulações numéricas, sensores de captação, entre outros. Consequentemente, a rápida obtenção de resultados em visualizações tem um papel importante para ajudá-los a entender estes dados.

A extração de isosuperfícies pode ajudar na visualização e interpretação desta massa de dados, ao aproveitar-se das características de modelos tridimensionais e de modelagens geométricas. Suas funcionalidades permitem, ainda, um eficiente destacamento de variáveis escalares em meio a outros valores.

A criação de uma isosuperfície envolve a utilização de um algoritmo robusto, uma vez que é realizada a manipulação de uma grande quantidade de dados. Na literatura, podemos encontrar estudos sobre diversos algoritmos. O Marching Cubes possui uma grande quantidade de material e fonte de pesquisa.

O X3D é uma linguagem que permite descrever as formas geométricas, como uma isosuperfície, a partir de estruturas em XML. Esta que é uma linguagem bastante disseminada pela internet, pela sua simplicidadade e eficácia para modelagem.

Para a exibição desta estrutura é necessário um meio de ler este arquivo em XML. Na internet, podemos encontrar uma extensa variedade de programas que fazem esta função, porém não possuem código livre e requerem uma estrutura computacional maior. Por isso o projeto possui seu próprio visualizador.

Combinando estes elementos consegue-se prover uma solução que, a partir de um determinado arquivo de dados, constrói-se uma isosuperfície e a exibe para o usuário de uma forma clara e objetiva.

1.1 Objetivos

Os objetivos deste projeto são:

- Aplicar o algoritmo Marching Cubes na extração de isosuperfícies a partir de um volume de dados;
- Facilitar a interpretação de uma grande massa de dados através de visualizações científicas e formas tridimensionais;
- Prover uma ferramenta de fácil acesso para meteorologistas e outros profissionais analisarem dados complexos.

1.2 Organização da monografia

A monografia está organizada da seguinte forma:

- No Capítulo 2 são apresentadas as principais características, métodos e aplicações de uma isosuperfície, o foco de nosso projeto. Ainda são mostrados os diversos campos de estudos que utilizam deste métodos de visualização, dando ênfase a Meteorologia;
- No Capítulo 3 é apresentado o método escolhido para a extração da isosuperfície, o algoritmo Marching Cubes. São abordados, ainda, os passos percorridos e algumas redundâncias encontradas na técnica;
- No Capítulo 4 é apresentado a ferramenta para a representação da isosuperfície, o padrão de arquivo X3D. São mostrados suas capacidades e principais funcionalidades;
- No Capítulo 5 é apresentado o formato de dados GRIB, sua estrutura básica, as vantagens em sua utilização, e as fontes de pesquisa utilizadas para a obtenção dos dados de teste;
- No Capítulo 6 é apresentado o método de implementação de todo o projeto, desde a entrada dos dados meteorológicos até a geração e visualização da isosuperfície.
- No Capítulo 7 é apresentada a conclusão final deste projeto, bem como o resumo do conhecimento adquirido ao longo do desenvolvimento e indicações de possíveis extensões do mesmo.

CAPÍTULO 2

Isosuperfície

Segundo Hansen e Johnson (2005), uma isosuperfície é um método de visualização de informação que consiste em uma superfície sólida contendo todos os pontos, de um determinado espaço, que possuem o mesmo valor escalar. Este valor pode variar entre um mínimo e máximo definido para que o usuário possa identificar características importantes na visualização. A isosuperfície pode ser considerada uma equivalente em três dimensões das linhas de contorno, como ilustrado na Figura 2.1.

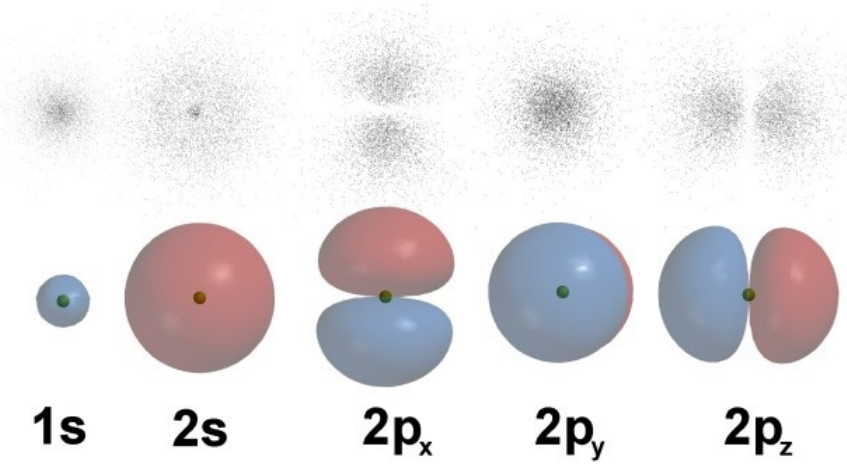


Figura 2.1: *Extração de uma isosuperfície a partir de um volume de pontos (MIELACK, 2009).*

Visualizar uma isosuperfície é uma maneira efetiva de analisar dados escalares tridimensionais. Utilizada para representar regiões onde o campo físico é constante, a isosuperfície permite a interpretação com uma vasta gama de profundidade visual. Visualizando a distribuição espacial de várias isosuperfícies, os cientistas são capazes de obter um maior entendimento da estrutura por trás de um campo (SHEN; JOHNSON, 1995).

De acordo com Mielack (2009), duas outras vantagens relevantes neste tipo de visualização, são a facilidade de implementação dos algoritmos disponíveis, uma vez que serão utilizadas grandes quantidades de dados dificultando o mapeamento das estruturas, e o nível de interação com o modelo, que permite uma melhor interpretação e assimilação da informação.

Estas características são derivadas de outras técnicas de renderização de volume encontradas na literatura.

Uma interessante característica é que, apesar das isosuperfícies serem sólidas, é possível que durante a extração sejam geradas mais de uma isosuperfície, como mostra a Figura 2.2, sendo que uma delas pode conter as outras (HANSEN; JOHNSON, 2005).

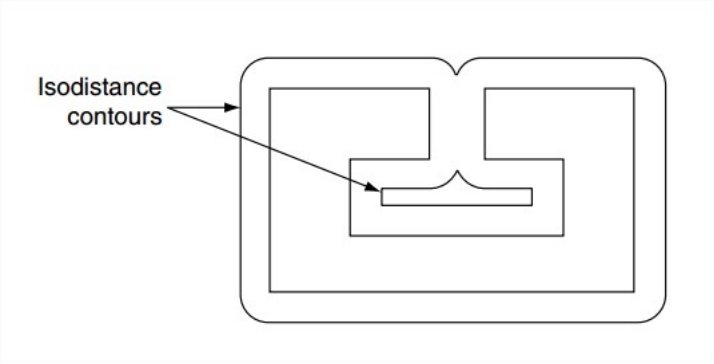


Figura 2.2: *Caso em que duas isosuperfícies são geradas, uma interior e outra exterior (HANSEN; JOHNSON, 2005).*

Por fim, modelos geométricos com um alto grau de detalhes são extraídas com técnicas de modelagem discretas, e exigem um nível elevado de esforço computacional tanto para executar o algoritmo quanto para renderizar a isosuperfície gerada (WOOD et al., 2004).

2.1 Aplicações

2.1.1 Medicina

Superfícies tridimensionais sobre a anatomia são importantes ferramentas médicas. Imagens dessas superfícies, construídas a partir de múltiplos cortes em 2D de tomografia computadorizada e ressonância magnética, ajudam os físicos a entender a complexa anatomia presente. A interpretação destes cortes necessitam de treinamento especial, e apesar dos radiologistas possuírem esta habilidade, frequentemente precisam comunicar aos físicos suas interpretações, estes que têm dificuldades em visualizar a anatomia em 3D (WALLIN, 1991).

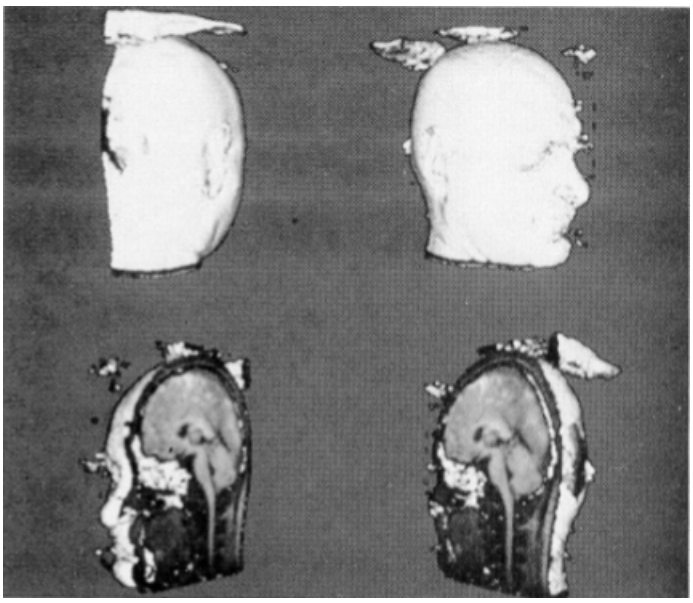


Figura 2.3: *Crânio humano parcial, reconstruído a partir de uma ressonância (LORENSEN; CLINE, 1987).*

Recentes pesquisas na visualização científica de cérebros, permitiram que as informações sobre anatomia derivadas das visualizações de alta resolução, provenientes de Ressonância Magnética, do inglês Magnetic Resonance Imaging (MRI), e Tomografia Computadorizada, do inglês Computerized Tomography (CT), fossem adicionadas a informações sobre fisiologia. Este tipo de avanço tomou como rota a obtenção de reconstruções precisas da superfície do córtex, provendo informações valiosas da geometria e características anatômicas do cérebro. Outro exemplo, é a localização das atividades cerebrais obtidas através de Tomografia por Emissão de Pósitrons, do inglês Positron Emission Tomography (PET), e Ressonância Magnética Funcional, do inglês Functional Magnetic Resonance Imaging (fMRI), poder ser mapeada na extração de superfície do córtex, proporcionando um melhor entendimento do funcionamento e organização do cérebro (DALE; SERENO, 1993).

Geometricamente, o córtex cerebral humano é uma fina camada dobrada de Matéria Cinza, do inglês Gray Matter (GM), que permanece dentro do Fluído Cérebro-espinhal, do inglês Cerebrospinal Fluid (CSF), e por fora da Matéria Branca, do inglês White Matter (WM), do cérebro. Por isso, reconstruir com precisão o córtex é um trabalho muito complexo, que pode ser facilitado através de modelos tridimensionais (MACDONALD; AVIS; EVANS, 1994).

2.1.2 Bioquímica

A bioquímica estuda grande moléculas, como as de proteínas e DNA, conhecidas como biomoléculas, e a interação entre elas. Essas moléculas são muito importantes para a des-

coberta de remédios, para isto os pesquisadores tentam achar novas moléculas para inibir a função de outras, geralmente proteínas, que são responsáveis por muitas doenças. Portanto, para entender melhor a função de grandes biomoléculas, é muito importante estudar seu volume e forma, porque, como em muitos outros campos, características de forma e função estão intimamente relacionadas (RAPOSO; QUEIROZ; GOMES, 2009).

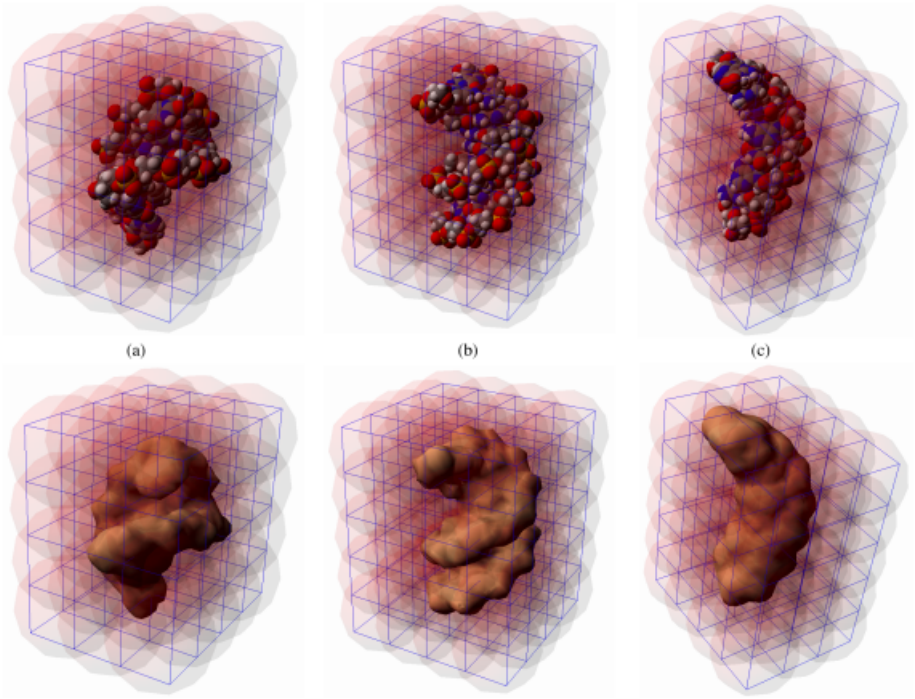


Figura 2.4: *Estrutura molecular demonstrando volume e forma (RAPOSO; QUEIROZ; GOMES, 2009).*

As moléculas de água confinadas dentro de cavidades das proteínas são muito importantes para entender a estrutura, estabilidade e funções das biomoléculas. Grandes esforços foram gastos tentando observar estas moléculas através de experimentos. É virtualmente impossível encontrar tais moléculas através de experimentos pois elas ficam presas dentro das biomoléculas por um processo de dobramento. Nestes casos, é aplicado um modelo específico de cálculo chamado 3D-RISM, para obter os resultados esperados, como vemos na Figura 2.5. Na teoria, a visualização de modelos tridimensionais tem provado ser capaz de prever a estrutura e termodinâmica das moléculas, ao menos em termos quantitativos (IMAI et al., 2005).

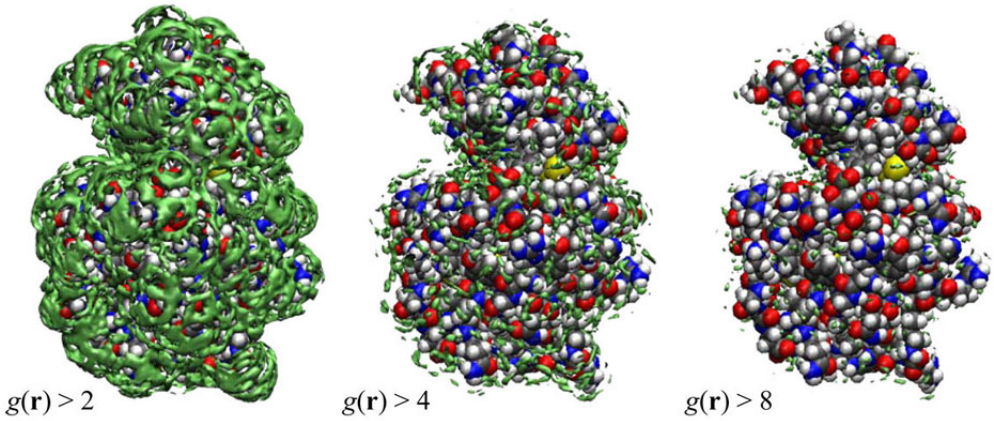


Figura 2.5: *Moléculas de água no interior de biomoléculas (IMAI et al., 2005).*

Para ajudar a interpretar a polaridade de uma molécula, a separação de carga pode ser visualizada mapeando o potencial eletrostático por meio de uma superfície, gradiente de cores ou indicando as regiões positivas e negativas do potencial eletrostático. Como mostra a Figura 2.6, estas técnicas permitem visualizar a distribuição de cargas eficientemente (SCHONBORN; HOST; PALMERIUS, 2010).

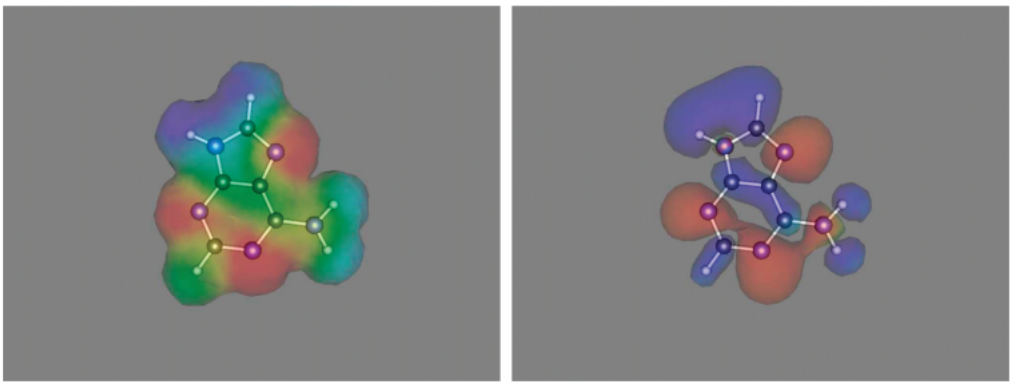


Figura 2.6: *Duas aplicações de separação de cargas em uma molécula (SCHONBORN; HOST; PALMERIUS, 2010).*

2.1.3 Arqueologia

Isosuperfícies podem auxiliar na visualização de objetos enterrados e prever suas formas integrais em conjunto com os dados extraídos via GPR. Por meio da criação de várias isosuperfícies sobrepostas, tornando mais transparentes as com valores escalares mais longes do ótimo. Os objetos resultantes podem dar uma noção do que esta soterrado, em termos de forma e tamanho. Estas imagens recriadas podem auxiliar muitos arqueologistas sem um profundo treinamento em geofísica (CONYERS, 2004).

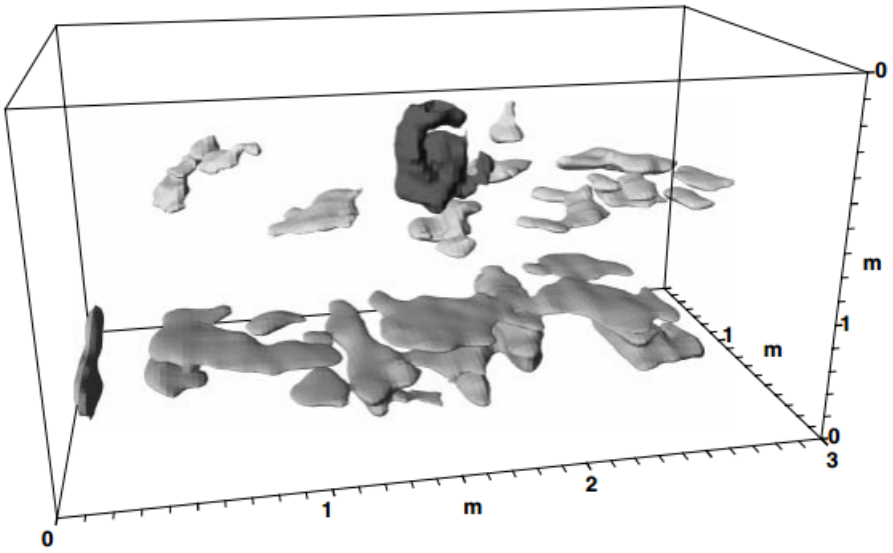


Figura 2.7: *A partir de dados do GPR, visão de duas pedras enterradas (LECKEBUSCH; PEIKERT, 2001).*

Dois blocos sobrepostos e enterrados de dimensões parecidas são um tipo de problema difícil de se resolver por modelagens numéricas. Apesar de o primeiro ser detectável, o segundo torna-se difícil de prever. Por meio da isosuperfície extraída dos reflexos de força ou amplitude dos blocos, consegue-se modelar o cenário corretamente, como visto na Figura 2.7. Este procedimento torna o modelo tridimensional mais aceitável e reduz significativamente a quantidade de dados. A isosuperfície pode então ser exportada e combinada com qualquer outro tipo de dado arqueológico em um Geographical Information System (GIS). As velocidades destas operações a tornam muito úteis para grandes pesquisas (LECKEBUSCH; PEIKERT, 2001).

2.1.4 Meteorologia

Segundo Papathomas, Schiavone e Julesz (1988), meteorologia é a área interdisciplinar de estudo da atmosfera focada nos processos e na previsão do clima. Os principais fatores dos fenômenos meteorológicos são a temperatura, a pressão e a umidade relativa do ar, além do relacionamento e das alterações que eles sofrem durante o tempo.

A atmosfera é uma massa em contínuo movimento e isto induz variações nas condições meteorológicas predominantes em uma região. O estado da atmosfera pode ser descrito por variáveis que caracterizam sua condição energética. Para um local, essa descrição pode ser tanto em termos instantâneos, definindo sua condição atual, como em termos estatísticos, definindo uma condição média (PEREIRA; ANGELOCCI; SENTELHAS, 2002).

De acordo com Papathomas, Schiavone e Julesz (1988), os tipos de dados meteorológicos disponíveis têm uma grande influência em sua visualização e posterior interpretação. No início, as medições eram feitas apenas em 2D, adquirindo pouca resolução, de modo que os especialistas utilizavam-se de, por exemplo, linhas de contorno para mostrar campos bidimensionais.

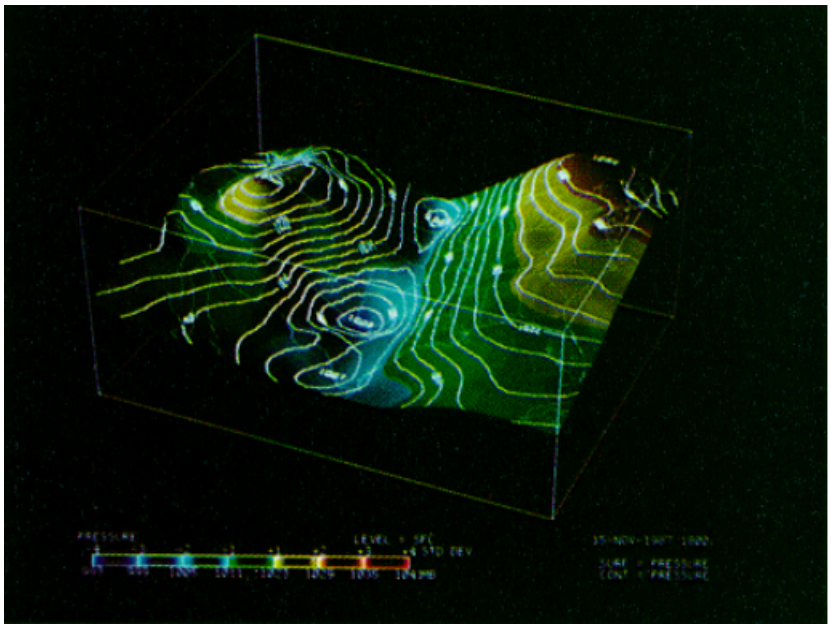


Figura 2.8: *Visualização de uma isosuperfície com informações de pressão e altitude (PAPATHOMAS; SCHIAVONE; JULESZ, 1988) .*

Com a captura de dados em 3D, os meteorologistas recebem agora a tarefa de perceber as estruturas volumétricas presentes na atmosfera. Para isso, alternavam entre várias imagens contendo as linhas de contorno de uma altura específica, para montar mentalmente uma imagem tridimensional dos dados. Finalmente, os dados passaram a chegar em uma taxa maior e os meteorologistas recebiam dados novos a cada hora, tornando possível a animação dos dados para que fosse possível um entendimento da evolução das condições do tempo (PAPATHOMAS; SCHIAVONE; JULESZ, 1988).

O conhecimento do clima e a capacidade de prevê-lo, são de grande interesse econômico não somente na agricultura, como nos meios de transporte, na prevenção de desastres, e mesmo em apólices de seguro. Esse conhecimento se torna mais útil quanto obtido com relativa antecedência, mesmo que não seja tão preciso (ZHU et al., 2002).

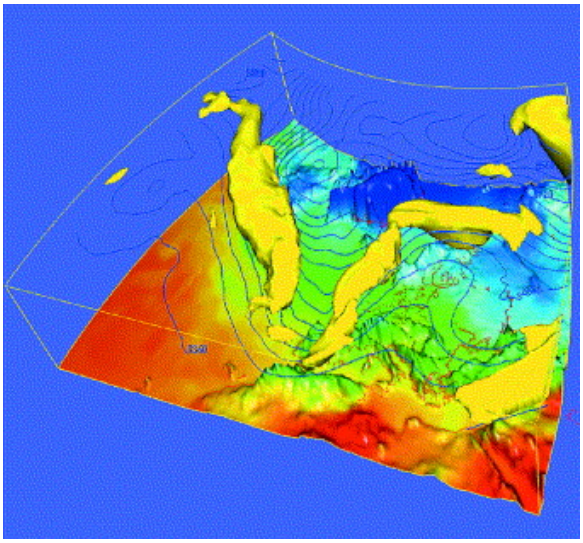


Figura 2.9: *Visualização de uma isosuperfície referente a velocidade do vento (HAASE et al., 2000).*

Na computação gráfica, trabalhos foram propostos desde o início da década de 80 (CHRISTIANSEN, 1984), focando-se na visualização de fenômenos atmosféricos de maneira a aproveitar as então novas tecnologias digitais. O uso dos modernos algoritmos de renderização, bem como do poder crescente do hardware voltado para gráficos também gerou frutos, inspirando pesquisas e soluções baseadas em um poder de processamento cada vez mais notável (HAASE et al., 2000).

De acordo com Drebin, Carpenter e Hanrahan (1988),Alguns sistemas modernos se propõem a uma representação volumétrica dos corpos meteorológicos. Essa ideia começou no final da década de 80, mas só se tornou realmente viável e capaz de produzir resultados em tempo real com o advento de novas técnicas de processamento 3D e do barateamento do hardware relacionado (ENGEL; KRAUS; ERTL, 2001).

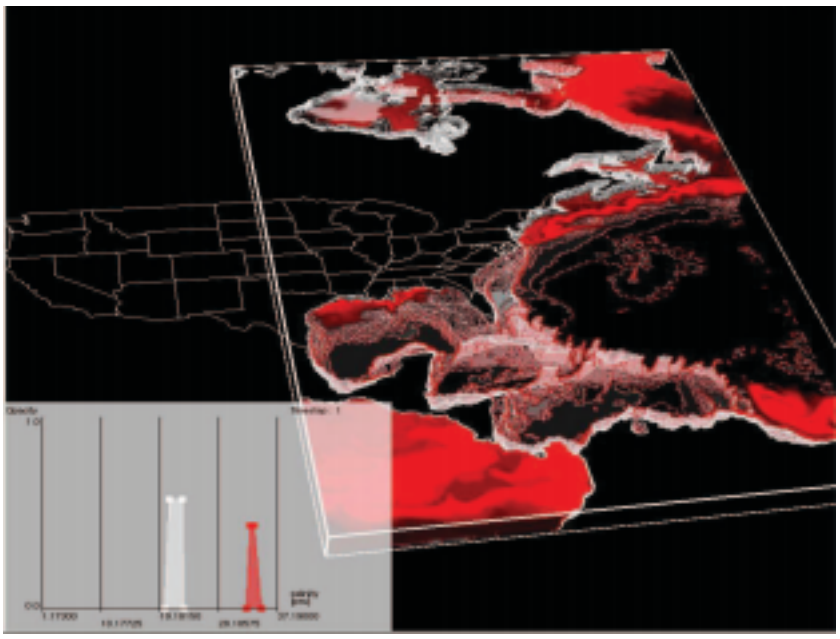


Figura 2.10: *Isosuperfície sobre a salinidade do oceano Atlântico, nas proximidades da costa leste americana (BERBERICH et al., 2009).*

Uma visualização mais detalhada e interativa alcançada através da renderização em dados 4D em tempo real, podem melhorar o entendimento de furacões, por exemplo, levando a melhores previsões no futuro (BERBERICH et al., 2009).

2.2 Algoritmos

2.2.1 Delaunay triangulation

O algoritmo Delaunay Triangulation consiste em triangular os pontos de uma área de modo que seja racionalizado os mínimos e máximos dos ângulos criados entre os triângulos. Este algoritmo é usado em inúmeros projetos, tais como: design geométrico, desenho assistido, topografia e mapeamento (CHEW, 1989).

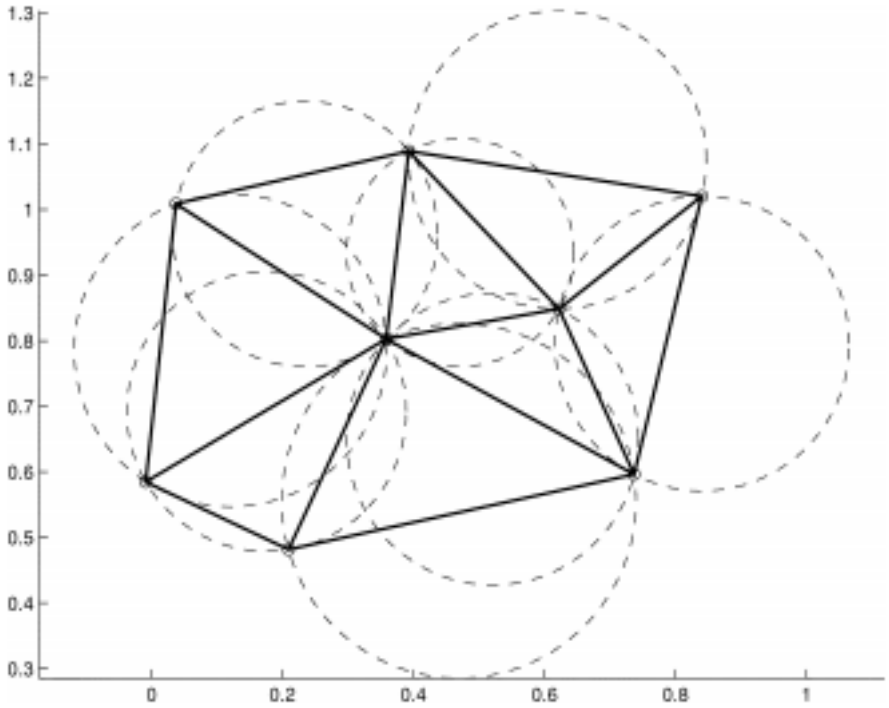


Figura 2.11: *Restrição da triangulação, por meio das circunferências (LERTRATTANAPANICH; BOSE, 2002).*

Um triângulo Delaunay pode ser restringido pela não aparição de outro vértice dentro da circunferência, resultante da junção de seus três vértices, como vemos na Figura 2.11. Nela podemos identificar a existência desta propriedade, e o resultado ao aplicarmos em vários pontos ao mesmo tempo (LERTRATTANAPANICH; BOSE, 2002).

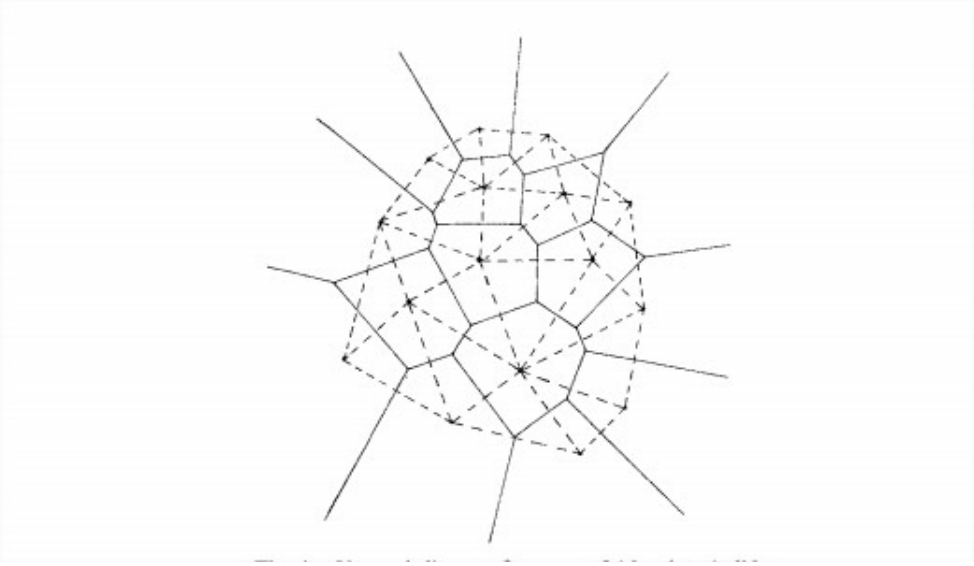


Figura 2.12: *Diagrama de Voronoi em linha contínua, e Triangulação de Delaunay em linha tracejada (LEE; SCHACHTER, 1980).*

Uma maneira eficiente de assimilar o algoritmo em questão, é pensando na sua ligação com um diagrama de Voronoi, como vemos na Figura 2.12. Neste, é traçado uma reta na distância média entre dois pontos. Esta reta se prolonga até se ligar com outra reta criada por outros dois pontos. O encontro de duas retas é chamado de Ponto Voronoi, e este ponto sempre representa um triângulo de Delaunay (LEE; SCHACHTER, 1980).

2.2.2 Marching tetrahedra

Neste algoritmo cada cubo de uma matriz é dividido em 6 tetraedros irregulares, ao dividir um cubo três vezes. Assim, cubos adjacentes dividem todas as laterais com as faces que se conectam. Esta característica é importante para garantir a uniformidade da isosuperfície criada, uma vez que os pontos são compartilhados (HANSEN; JOHNSON, 2005).

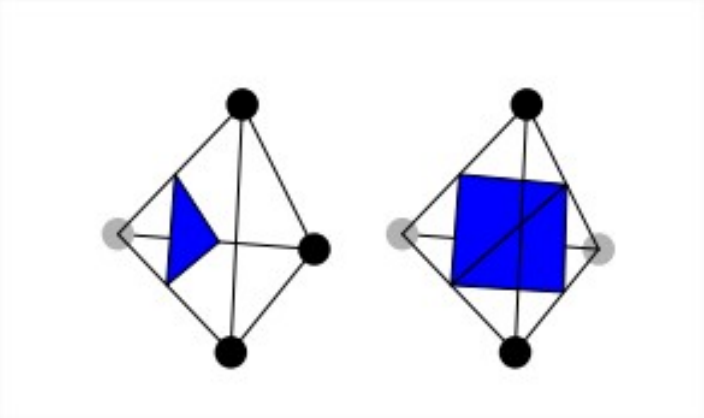


Figura 2.13: *Casos de triangulação e escolha dos vértices (TREECE; PRAGER; GEE, 1999).*

Cada tetraedro tem 16 possíveis configurações, que se dividem em três classes: sem intersecção, intersecção em um triângulo e intersecção de dois triângulos adjacentes. Na Figura 2.13, vemos dois exemplos destas configurações. Com essas configurações, cria-se uma lista indexada para comparação e definição de qual configuração será usada na extração da isosuperfície (WANG; YU, 2011).

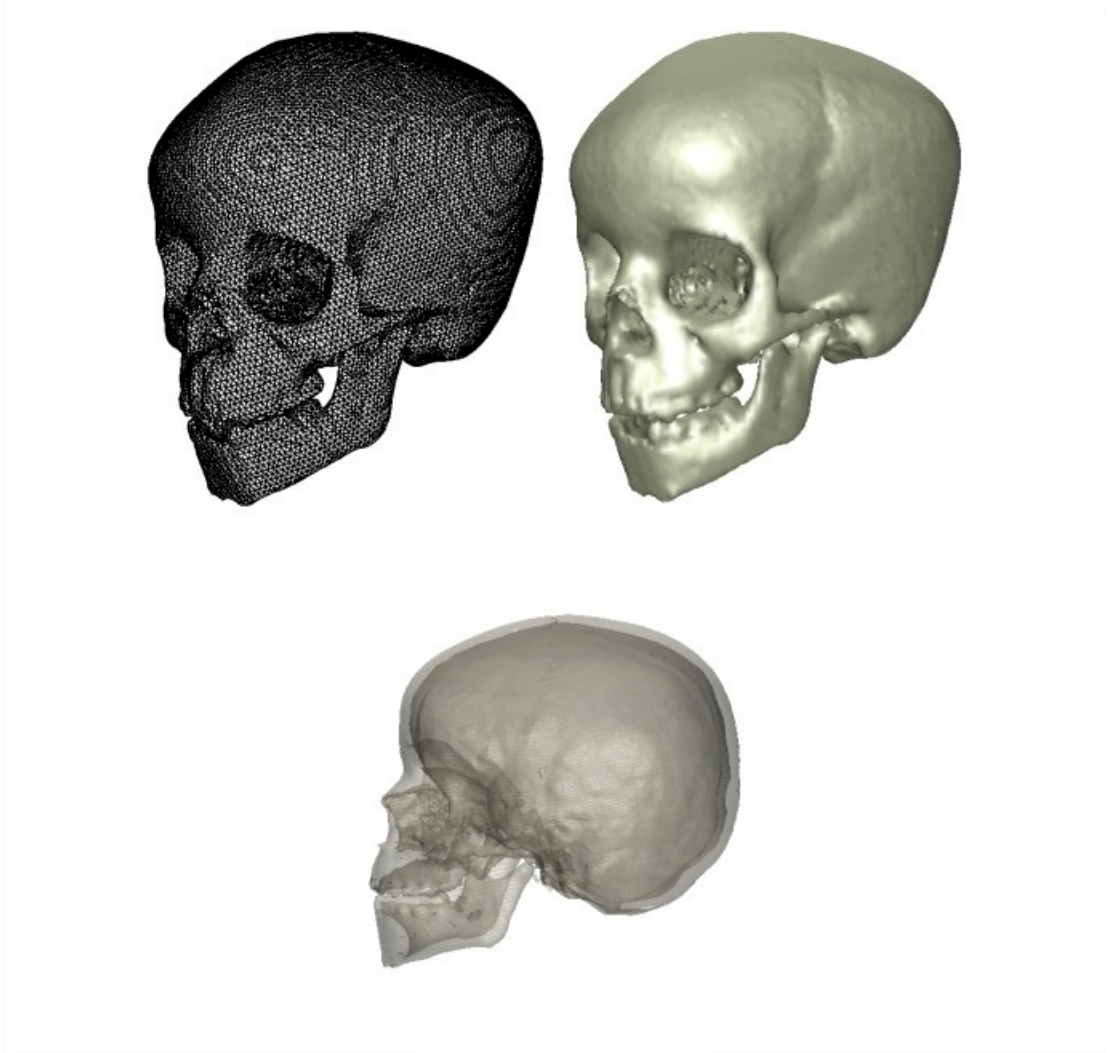


Figura 2.14: *Reconstrução de uma tomografia, sendo triangulação pura, sombreada e transparente (TREECE; PRAGER; GEE, 1999).*

Em relação a outros métodos de extração, o Marching Tetrahedra constrói modelos melhor definidos topologicamente, como vemos na Figura 2.14 e com maior definição devido seu grau de detalhe. Porém, é necessária a geração de um maior número de componentes para cada cubo simples (TREECE; PRAGER; GEE, 1999).

2.2.3 Shrinkwrap

Este algoritmo gera uma malha triangular para realizar a aproximação de uma isosuperfície. Ele começa com uma triangulação de uma esfera e, então, aplica uma série de deformações para esta triangulação, e depois transformá-lo na superfície em questão. O algoritmo é adaptativo, no sentido de que os comprimentos dos lados dos triângulos na malha variam de acordo com a curvatura local da superfície subjacente. A análise quantitativa da precisão

do algoritmo é dado juntamente com uma comparação empírica com algoritmos anteriores (BOTTINO; NUIJ; OVERVELD, 1996).



Figura 2.15: *Visualização de um bonsai, de acordo com (SZYMCZAK; VANDERHYDE, 2003).*

O método visa produzir os triângulos de tamanhos e formas diferentes de acordo com a curvatura da superfície, a fim de oferecer uma aproximação a partir de uma precisão conhecida. De acordo com os estudos as vantagens desta técnica se resumem a adaptabilidade da malha e velocidade de processamento superior a outros algoritmos. Porém, existe a desvantagem de que este algoritmo lida apenas com superfícies fechadas (OVERVELD; WYVILL, 2004).

2.2.4 Noise

O algoritmo em questão extrai rapidamente uma isosuperfície de malhas estruturadas e não estruturadas. Usando o espaço de intervalo, uma nova representação do domínio subjacente, o algoritmo de extração de isosuperfície utiliza o pior caso de complexidade para a fase de pesquisa, onde n é o tamanho do conjunto de dados e k é o número de células intersectadas pela isosuperfície. O requisito de memória é mantida a O(n) enquanto que a etapa de pré-processamento é O(nlogn) (SUTTON et al., 2000).

O Noise funciona bem para qualquer conjunto de dados, desde pequenas isosuperfícies até as maiores. O ponto chave é utilizar pontos de mínimo e máximo e a partir deles criar os intervalos de espaços. A vantagem é de que, apesar de em duas dimensões não haver uma correção entre os pontos, sobrepondo os espaços as relações aparecem por si só. Além disso, o intervalo pode servir como um ponto em comum no qual outros métodos podem ser comparados e analisados (LIVNAT; SHEN; JOHNSON, 1996).

2.2.5 Sweeping simplices

Este algoritmo acelera a extração de isosuperfícies de malhas não estruturadas, evitando a passagem de todo o conjunto de células do volume, consistindo em um esquema de decomposição e varredura. A componente de varredura incremental localiza os elementos por cruzamentos, e o esquema de decomposição de dados restringe o desempenho de pior caso do algoritmo. Para conjuntos de dados compostos por centenas de elementos, o algoritmo pode reduzir o tempo de travessia das células em mais de 90% sobre outros algoritmos de extração de isosuperfície, facilitando assim a sondagem interativa de campos escalares para problemas de grande escala em malhas não estruturadas (SHEN; JOHNSON, 1995).

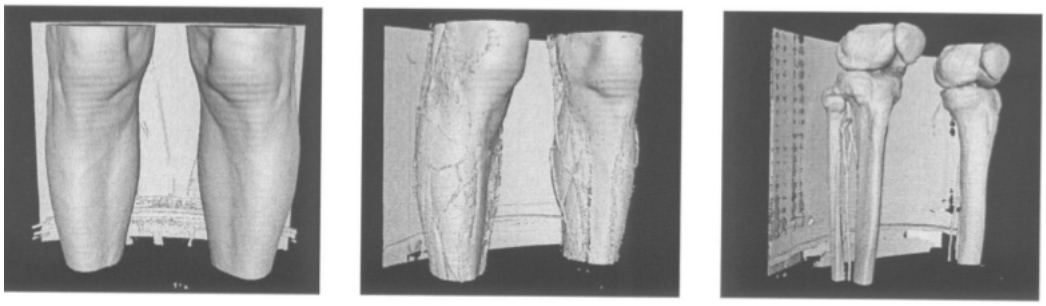


Figura 2.16: *Visualização de uma perna humana, mostrando a pele, vasos e ossos (RYMON-LIPINSKI et al., 2004).*

O algoritmo funciona melhor para casos em que o valor escalar e alterado sutilmente, porem em casos mais extremos também funciona de forma eficaz. Uma vez que os dois componentes trabalham de forma conjunta, e atualizam por meio de incrementos e não de inserções em listas (RYMON-LIPINSKI et al., 2004).

CAPÍTULO 3

Marching Cubes

Marching Cubes (MC) é um algoritmo computacional para extrair uma isosuperfície de uma malha poligonal. O algoritmo usa a técnica de dividir para conquistar, onde é tratado cada cubo unitário por vez, Figura 3.1. Após determinar a configuração da superfície neste cubo, vai para o próximo e continua a passar por um a um até completar toda a estrutura (LORENSEN; CLINE, 1987).

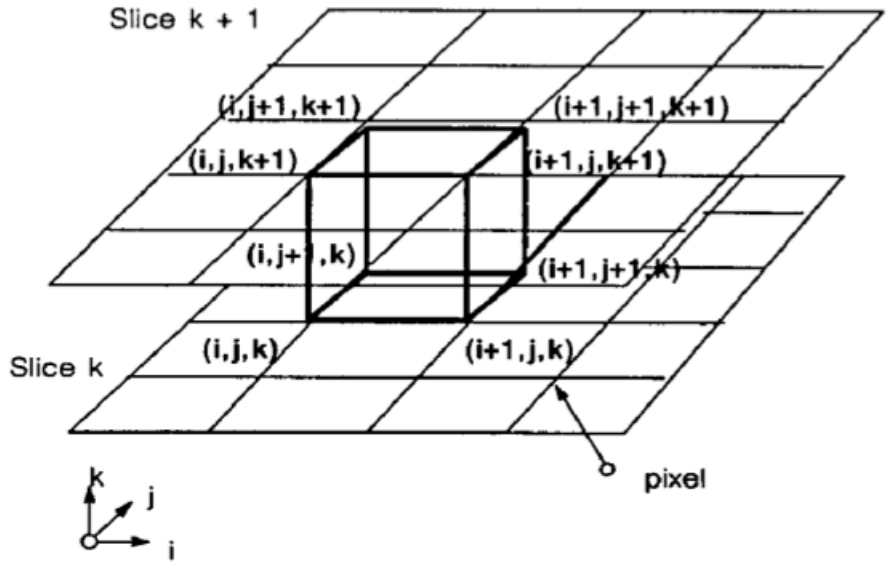


Figura 3.1: *Aplicação do Algoritmo Marching Cubes em um Plano (LORENSEN; CLINE, 1987).*

O MC é um aprimoramento da técnica de Marching Squares(MS), porém aplicado a uma malha de dados tridimensional. Neste temos um estado com 8 vértices disponíveis para marcação gerando, portanto, 256 estados possíveis. Estes estados são reduzidos a 15 casos padrões que conseguem, por meio de manipulações ou dedução reflexiva, como na Figura 3.1, chegar a todos os estados.

3.1 Construção

A Figura 3.2, com as 15 configurações iniciais, serve de base para a extração de toda a isosuperfície por meio de comparação. Cada conjunto de oito vértices formadores do cubo em questão é comparado com esta tabela, ao se chegar a configuração correta, é feita a rotação necessária para chegar ao estado, e então é passado para o próximo conjunto de vértices (LORENSEN; CLINE, 1987).

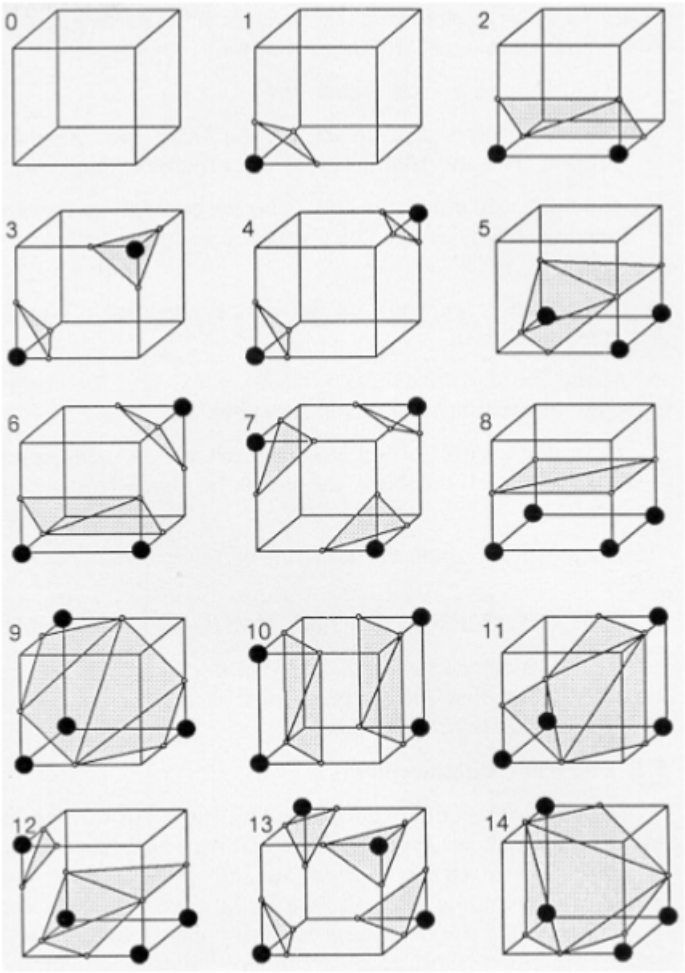


Figura 3.2: *Representação das 15 configurações iniciais previstas (LORENSEN; CLINE, 1987).*

Para uma varredura mais eficaz os cubos são lidos um a um até completar um plano em paralelo a um eixo, como na Figura 3.3 os quadrantes coloridos, para depois passar para o próximo plano, quadrantes inferiores da figura.

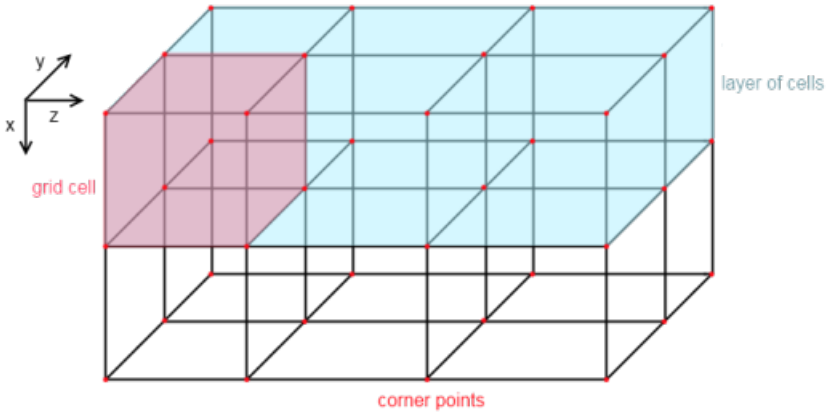


Figura 3.3: *Ordem de leitura e varredura de toda a cena.*

3.2 Casos extras

A tabela de configurações iniciais possuem 6 casos ambíguos, Figura 3.4, neles para ser realizada a correta rotação deve-se olhar os cubos ao redor para que estes sigam o mesmo padrão de escolha (ZHOU; SHU; KANKANHALLI, 1994).

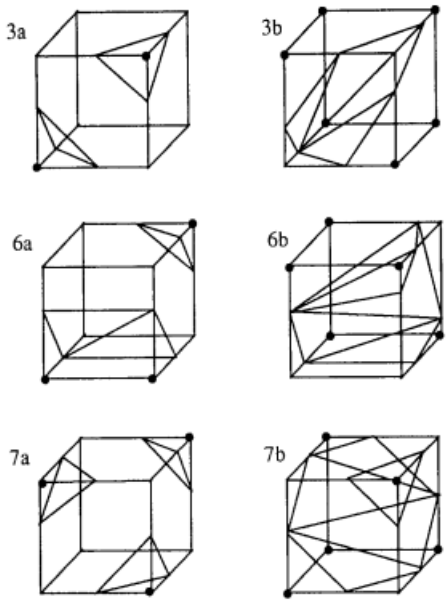


Figura 3.4: *Casos extras, além das configurações iniciais (ZHOU; SHU; KANKANHALLI, 1994).*

Na Figura 3.5, vemos como uma mesma configuração pode causar dois tipos de interpretações diferentes. Tendo isto em vista, podemos, de acordo com a literatura, alterar o algoritmo para que seja tratado caso a caso cada escolha ou, simplesmente, assumirmos um padrão

para toda a tabela de configurações iniciais, que foi a estratégia escolhida para este trabalho (LEWINER et al., 2003).

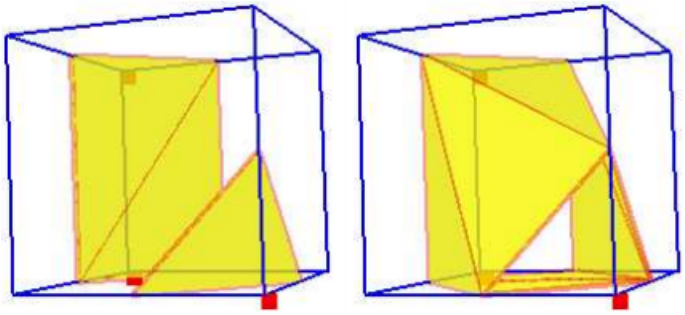


Figura 3.5: *Exemplo de uma redundância de pontos (LEWINER et al., 2003).*

CAPÍTULO 4

X3D

O Extensible 3D (X3D) é um padrão de arquivo de código aberto para a representação de cenas em 3D. Usando o Extensible Markup Language (XML) como estrutura, se é capaz de renderizar domínios e objetos em tempo real (DALY; BRUTZMAN, 2007). O X3D derivou do Virtual Reality Modeling Language (VRML) tornando sua edição e compreensão mais simples para o entendimento humano, uma vez que o VRML já era disseminado pela literatura (YU et al., 2006).

O X3D pode ser considerado uma linguagem, em forma de texto, para descrever formas tridimensionais e interação com o ambiente virtual, combinando geometrias e descrições de comportamentos em um simples arquivo de extensão ".x3d". Estes arquivos podem ser gerados a partir de um editor de textos simples, um construtor de mundos, um gerador de formas ou um modelador (CONSORTIUM, 2013).

4.1 Funcionalidades

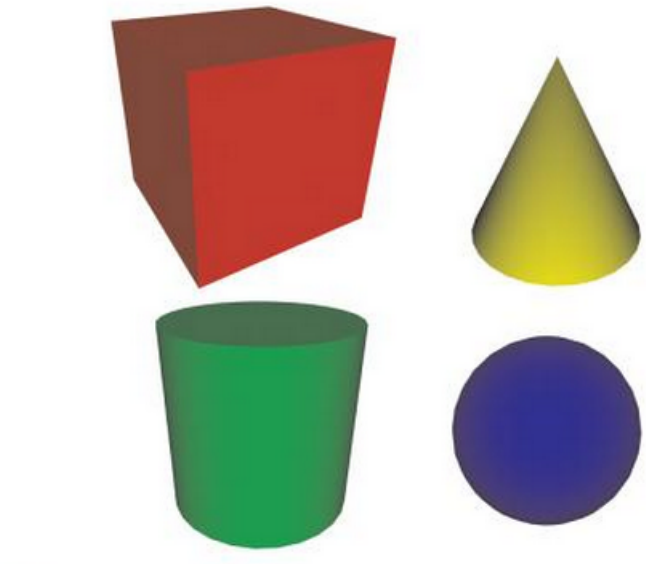


Figura 4.1: *Objetos primitivos (BRUTZMAN; DALY, 2010).*

O X3D disponibiliza uma grande variedade de propriedades na criação das cenas, como formato, textura, cor, animação e interatividade. Estas propriedades podem ser renderizadas via software, caso não exista aceleração por hardware. A texturização pode ser feita a partir de cores uniformes, imagens ou a mistura das duas. O padrão X3D possui formas básicas pritivas, como: cubo, cilindro, cone, esfera e texto (RHYNE et al., 2003).

Listing 4.1: *X3D - Exemplo inicial*

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
   "http://www.web3d.org/specifications/x3d-3.0.dtd">

<X3D profile='Immersive' version='3.0'>
<head>
</head>
<Scene>
  <Shape>
    <Appearance>
      <Material></Material>
    </Appearance>
    <Cylinder radius='1.5'></Cylinder>
  </Shape>
</Scene>
</X3D>
```

A marcação "IndexedFaceSet"é uma ferramenta para a criação de formas geométricas bem flexível. A criação de uma forma é bem simples, pois é necessário apenas escolher o número de vértices a ser inserido, esta informação define o número de lados da forma, e as coordenadas de cada um desses vértices. A marcação "Solid"permite deixar a superfície visível de qualquer ponto de vista ou transparente.

Listing 4.2: *X3D - Plano*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Por meio da marcação "DiffuseColor"é possível especificar as cores do objeto no sistema RGB. Na marcação "Material"podemos controlar, ainda, a cor principal, transparência, brilho e intensidade de luz no ambiente.

Listing 4.3: *X3D - Cor*

```
<Shape>
   <Appearance>
   <Material
      diffuseColor = '1.0 1.0 1.0'
         emissiveColor = '0.4'
  transparency = '0.5'
  specularColor = '0.71 0.70 0.56'
      shininess = '0.16'
      ambientIntensity = '0.4'>
   </Material>
   </Appearance>
   <Cone height='2.0' bottonRadius='1.0'></Cone>
</Shape>
```

A marcação "Transform"cria um grupo com o seu próprio sistema de coordenadas. Possibilitando transladar, rotacionar ou dimensionar este grupo separadamente do resto da cena. A translação posiciona a cena em um sistema de coordenadas em X, Y, e Z. Tomando como ponto de partida a origem dos eixos.

Listing 4.4: *X3D - Translação*

```
<Transform
   translation = '2.0 1.0 3.0' >
</Transform>
```

A rotação orienta o sistema de coordenadas de uma cena em torno de um eixo, com um determinado ângulo de rotação, estes ângulos são medidos em radianos. A rotações são realizadas em torno dos eixos X, Y e Z.

Listing 4.5: *X3D - Rotação*

```
<Transform
   rotation = '0.0 0.0 1.0 0.52' >
</Transform>
```

Existe a possibilidade de rotacionar uma cena várias vezes e em eixo diferentes. Neste caso, o código é interpretado a partir da tag mais interna para a mais externa. Esta ordem lógica é importante pois em alguns casos o resultado final pode ser diferente do esperado se a ordem não for seguida.

Listing 4.6: *X3D - Múltiplas Rotações*

```
<Transform rotation = '1.0 0.0 0.0 0.52' >
   <Transform rotation = '0.0 1.0 0.0 0.52' >
      <Transform rotation = '0.0 0.0 1.0 0.52' >
```

```
4         </Transform>
5     </Transform>
6 </Transform>
```

A regra da mão direita é utilizada tanto para idetenficarmos a disposição dos eixos, quanto para direcionar a rotação da cena, Figura 4.3. Assim como na física, devemos posicionar o polegar da mão direita paralelo ao eixo de rotação e no sentido positivo do mesmo. Então curvar os dedos sobre o eixo, a direção desta curva é o sentido de rotação positivo (BRUTZMAN; DALY, 2010).

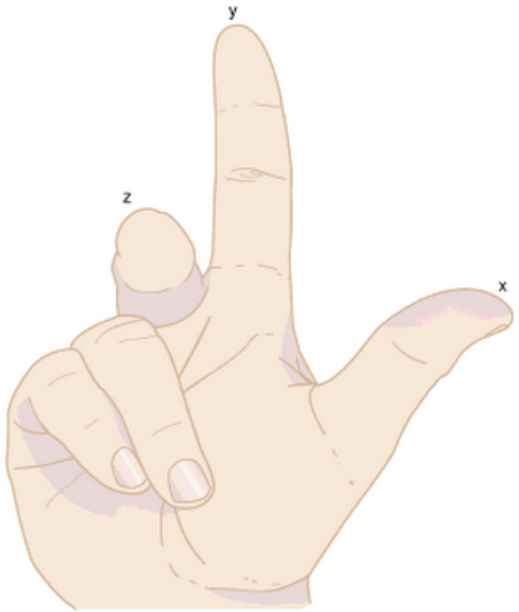


Figura 4.2: *Regra da mão direita na disposição dos eixos (BRUTZMAN; DALY, 2010).*

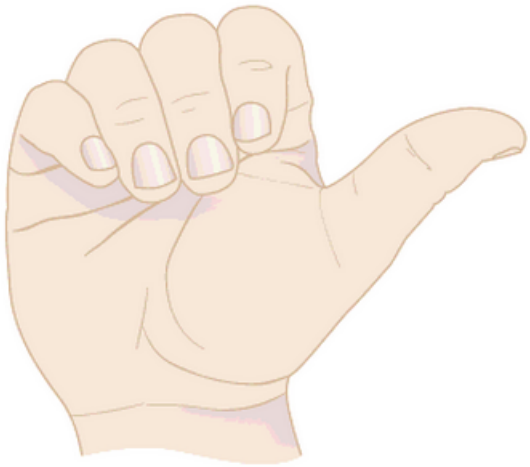


Figura 4.3: *Regra da mão direita na rotação de um objeto (BRUTZMAN; DALY, 2010).*

A marcação "Scale"possibilita aumentar ou diminuir a cena proporcionalmente como um todo ou individualmente eixo a eixo.

Listing 4.7: *X3D - Escala*

```
<Transform
   translation = '-2.0 3.0 0.0'
   scale = '0.5 0.5 0.5'>
</Transform>
```

A marcação "Def"fornece um nome para o nó, permitindo reutilizado outras vezes sem ter de contruí-lo novamente. É possível nomear qualquer tipo de nó, com letras e números, desde que o nome sejá único em todo o arquivo.

Listing 4.8: *X3D - Definição*

```
<Material DEF = 'Vermelho'
   diffuseColor 1.0 0.0 0.0>
</Material>
```

A marcação "Use"utiliza um nó nomeado anteriormente. Um nó nomeado pode ser utilizado diversas vezes, como instâncias, partilhando a mesma descrição.

Listing 4.9: *X3D - Uso*

```
<Appearance>
   <Material USE = 'Vermelho'></Material>
</Appearance>
```

Com a combinação destas ferramentas, podemos construir um objeto com várias formas combinadas em diversas posições e cores, Figura 4.4.

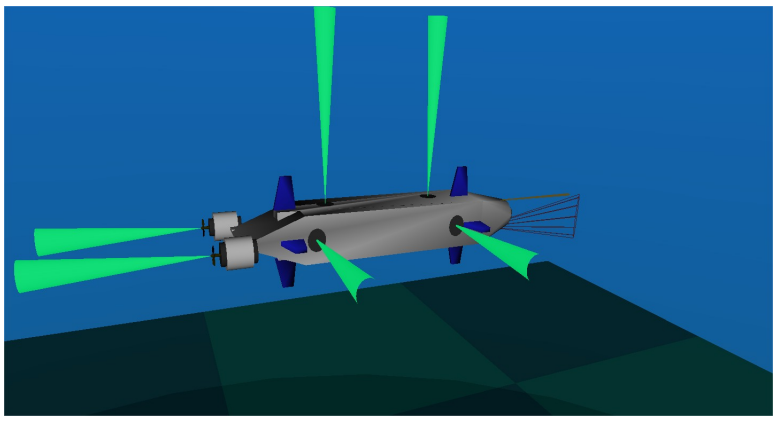


Figura 4.4: *Exemplo de uma cena, objeto complexo fictício.*

Podemos ainda criar cenas inteiras, a partir destas formas, Figura 4.5.

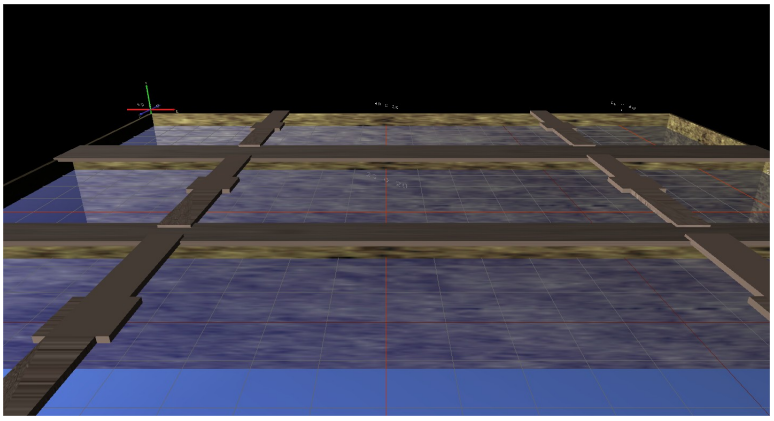


Figura 4.5: *Exemplo de uma cena, tanques de água.*

CAPÍTULO 5

Grib

Gridded Binary (GRIB) é um formato de dados usado por institutos e pesquisadores para fins meteorológicos e aprovado pela World Meteorology Organization (WMO). O projeto GRIB é de código aberto e internacional, seus arquivos são compactados, portanto, mais leves que os outros arquivos binários, e auto-descritivos, facilitando sua leitura (LUCERO et al., 2003).

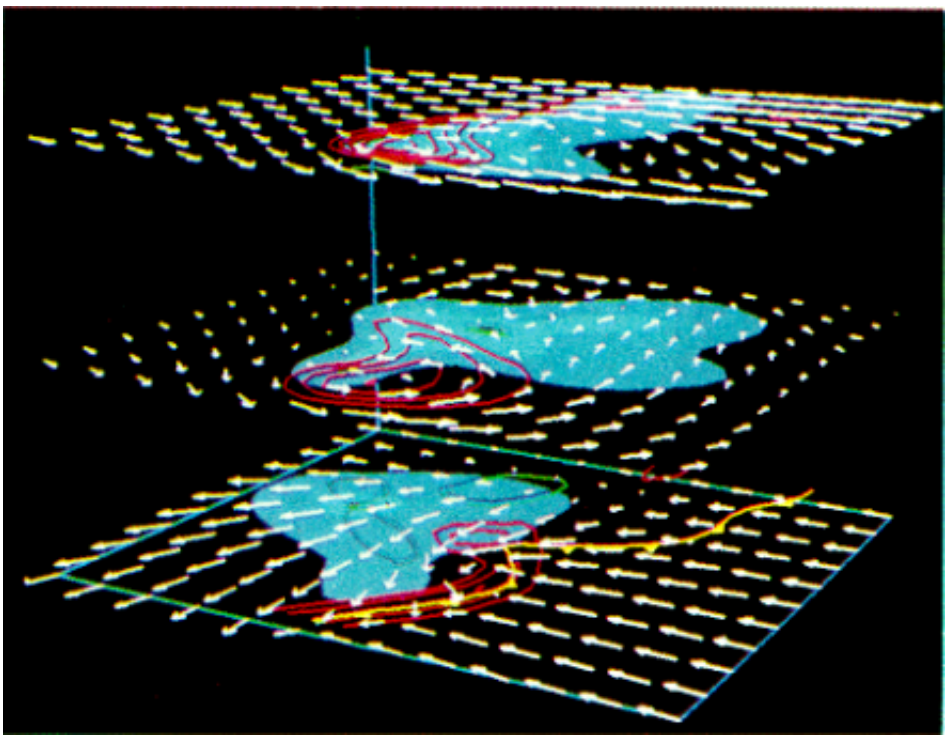


Figura 5.1: *Modelagem do comportamento de uma nuvem (PAPATHOMAS; SCHIAVONE; JULESZ, 1988).*

De acordo com Stackpole (1994), arquivos GRIB podem representar uma variedade de variáveis meteorológicas, como: temperatura, ponto de ebulição, componentes horizontais e verticais do vento, entre outros. Para este trabalho foram usados os arquivos disponibilizados no site da Global Marine Net (GMN), pois há uma grande variedade nos tipos de dados.

Cada arquivo GRIB é auto-descritivo, portanto, após sua decodificação é possível ver que sua estrutura é bem definida e de fácil compreensão, como vemos na Figura 5.2.

```
1:0:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=0:P2=0:TimeU=1:sfc:anl:NAve=0
2:1648:d=13103106:PERPW:kpds5=108:kpds6=1:kpds7=1:TR=0:P1=0:P2=0:TimeU=1:sfc:anl:NAve=0
3:3296:d=13103106:DIRPW:kpds5=107:kpds6=1:kpds7=1:TR=0:P1=0:P2=0:TimeU=1:sfc:anl:NAve=0
4:4944:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=0:P2=0:TimeU=1::anl:NAve=0
5:6502:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=0:P2=0:TimeU=1::anl:NAve=0
6:7962:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=3:P2=0:TimeU=1:sfc:3hr fcst:NAve=0
7:9610:d=13103106:PERPW:kpds5=108:kpds6=1:kpds7=1:TR=0:P1=3:P2=0:TimeU=1:sfc:3hr fcst:NAve=0
8:11258:d=13103106:DIRPW:kpds5=107:kpds6=1:kpds7=1:TR=0:P1=3:P2=0:TimeU=1:sfc:3hr fcst:NAve=0
9:12906:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=3:P2=0:TimeU=1::3hr fcst:NAve=0
10:14466:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=3:P2=0:TimeU=1::3hr fcst:NAve=0
11:15936:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=6:P2=0:TimeU=1:sfc:6hr fcst:NAve=0
12:17584:d=13103106:PERPW:kpds5=108:kpds6=1:kpds7=1:TR=0:P1=6:P2=0:TimeU=1:sfc:6hr fcst:NAve=0
13:19232:d=13103106:DIRPW:kpds5=107:kpds6=1:kpds7=1:TR=0:P1=6:P2=0:TimeU=1:sfc:6hr fcst:NAve=0
14:20880:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=6:P2=0:TimeU=1::6hr fcst:NAve=0
15:22464:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=6:P2=0:TimeU=1::6hr fcst:NAve=0
16:23946:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=9:P2=0:TimeU=1:sfc:9hr fcst:NAve=0
17:25594:d=13103106:PERPW:kpds5=108:kpds6=1:kpds7=1:TR=0:P1=9:P2=0:TimeU=1:sfc:9hr fcst:NAve=0
18:27242:d=13103106:DIRPW:kpds5=107:kpds6=1:kpds7=1:TR=0:P1=9:P2=0:TimeU=1:sfc:9hr fcst:NAve=0
19:28890:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=9:P2=0:TimeU=1::9hr fcst:NAve=0
20:30490:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=9:P2=0:TimeU=1::9hr fcst:NAve=0
21:31966:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=12:P2=0:TimeU=1:sfc:12hr fcst:NAve=0
22:33614:d=13103106:PERPW:kpds5=108:kpds6=1:kpds7=1:TR=0:P1=12:P2=0:TimeU=1:sfc:12hr fcst:NAve=0
23:35262:d=13103106:DIRPW:kpds5=107:kpds6=1:kpds7=1:TR=0:P1=12:P2=0:TimeU=1:sfc:12hr fcst:NAve=0
24:36910:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=12:P2=0:TimeU=1::12hr fcst:NAve=0
25:38518:d=13103106:SWPER:kpds5=106:kpds6=241:kpds7=1:TR=0:P1=12:P2=0:TimeU=1::12hr fcst:NAve=0
26:39996:d=13103106:HTSGW:kpds5=100:kpds6=1:kpds7=1:TR=0:P1=15:P2=0:TimeU=1:sfc:15hr fcst:NAve=0
```

Figura 5.2: *Extrato de um arquivo GRIB após a decriptografia.*

Para a leitura de um arquivo GRIB, foi utilizado o programa WGRIB (versão 1.8.0). Disponibilizado pelo Climate Prediction Center (CPC), pertencente ao National Weather Service (NWS) e vinculado ao National Oceanic and Atmospheric Administration (NOAA). É um programa multiplataforma que decodifica o arquivo GRIB e disponibiliza seu conteúdo em um arquivo de texto comum (CENTER, 2013). A documentação com as principais características e comandos do WGrib se encontram no Anexo A.

CAPÍTULO 6

Implementação

Primeiramente, o programa WGrib é utilizado na decriptografia de um arquivo de dados Grib, gerando um arquivo de texto auto descritivo. A partir deste é preenchida uma matriz para execução do algoritmo.

A classe Cube possui as configurações iniciais dos cubos, já implementadas em X3D, e as instâncias necessárias para a derivação de todas as outras possibilidades de estado, essas derivações são geradas através de rotações e reflexos.



Figura 6.1: *Diagrama de classes do projeto.*

A classe MarchingCubes percorre toda a matriz de dados, posicionando o cubo em seu

lugar dentro dos eixos X, Y e Z, atualizando-o a cada iteração. A classe ModelX3DWriter gera o arquivo no formato X3D, de acordo com o que é retornado pelas outras classes, criando o modelo 3D da isosuperfície.

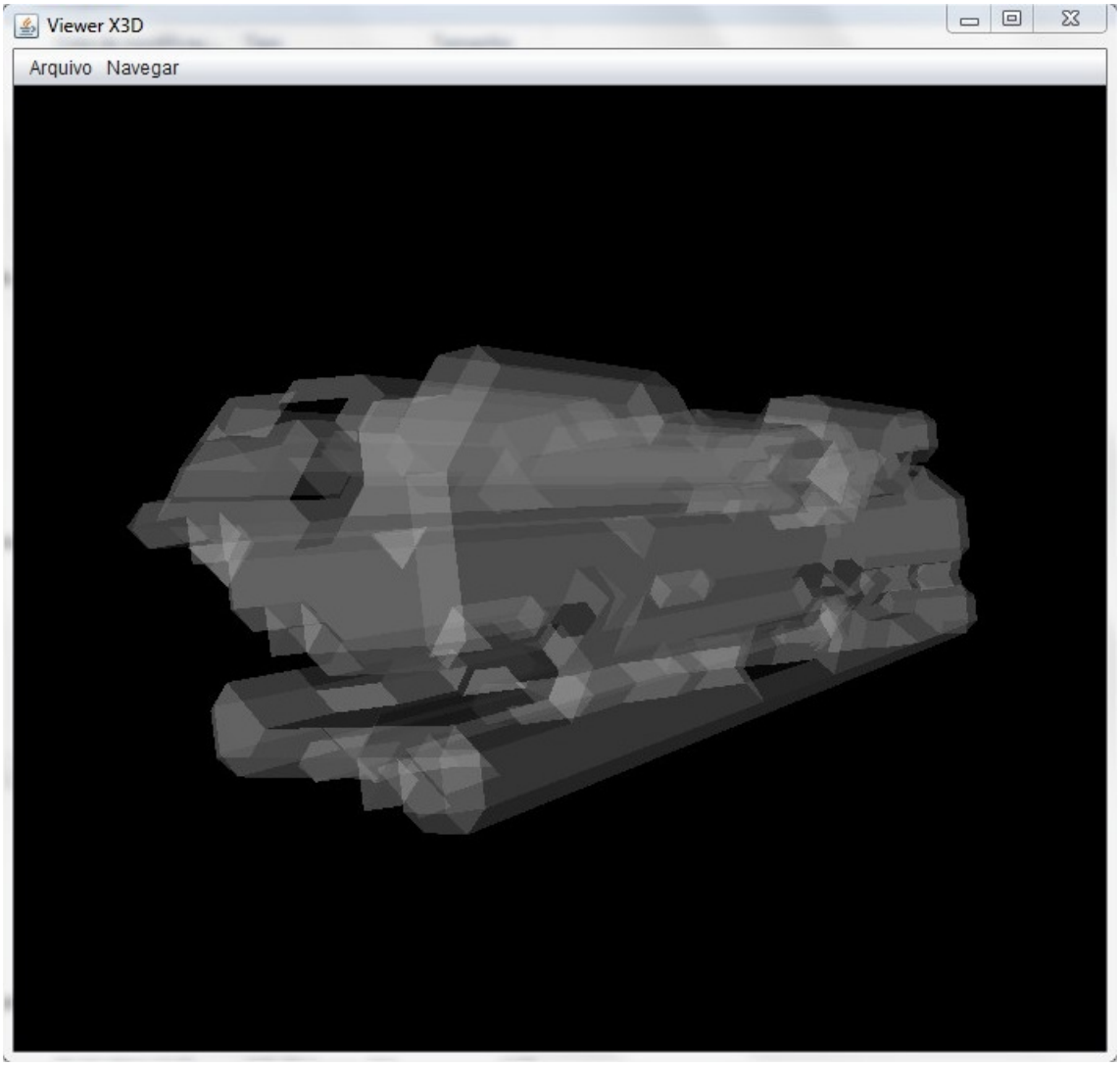


Figura 6.2: *Visualizador desenvolvido exibindo uma isosuperfície.*

Por fim, o visualizador de X3D foi construído usando tecnologia JAVA3D, permitindo a visualização e manipulação da isosuperfície gerada, como ilustrado na Figura 6.2.

CAPÍTULO 7

Conclusão

A utilização de isosuperfícies permitiu uma maior interação com os dados, que a simples visualização de seus arquivos por programas em duas dimensões. A manipulação dos dados foi melhorada, uma vez que é possível a rotação do modelo criado.

A implementação do algoritmo Marching Cubes ocorreu conforme o planejado, e de acordo com as referências técnicas. A técnica permitiu que isosuperfícies relativamente complexas fossem geradas a partir dos dados obtidos sem complicações.

A utilização da ferramenta X3D possibilitou a criação das superfícies trigonométricas utilizando o recurso IndexedFaceSet. Permitiu a criação dos 14 estados iniciais e por meio de rotações, replicar todas as 256 configurações possíveis. E, por fim, através das translações percorrer a matriz de dados e montar quadrante por quadrante a isosuperfície final.

O formato de dados GRIB se mostrou uma ótima opção, por ser de código aberto, auto-descritivo e relativamente leve se comparado a outros formatos binários. Sua ampla popularidade contribuiu para uma rápida obtenção de vários casos de testes a partir de fontes seguras, como a GRN.

A criação de um visualizador próprio para códigos em X3D e XML permitiu a melhor manipulação dos modelos tridimensionais e uma execução menos carregada em relação a outras opções encontradas.

7.1 Trabalhos Futuros

Durante o desenvolvimento do projeto foram identificados os seguintes trabalhos futuros:

- Visualização de outros formatos de dados. Aumentando a compatibilidade dos sistema e a disseminação em outras áreas da meteorologia. Exemplos de formatos: GrADS, GRID, GRAF, NetCDF e HDF;
- Aplicação de outros algoritmos conhecidos na literatura, para eventuais comparações de eficiência e desempenho para volumes de dados realmente significativos;

- Desenvolvimento de um processamento paralelo na execução do algoritmo, a fim de obter maior velocidade na extração da isosupercíficie;
- Implementação em laboratórios meteorológicos para a validação da solução por usuários da área. Possibilitando o aprimoramento da ferramenta de acordo com as impressões e sugestões.

Referências Bibliográficas

BERBERICH, M. et al. Geospatial visualization using hardware accelerated real-time volume rendering. In: *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*. [S.l.: s.n.], 2009. p. 1–5.

BOTTINO, Andrea; NUIJ, Wim; OVERVELD, Kees Van. How to shrinkwrap through a critical point: an algorithm for the adaptive triangulation of iso-surfaces with arbitrary topology. *En Proceedings of Implicit Surfaces*, v. 96, p. 53–72, 1996.

BRUTZMAN, D.; DALY, L. *X3D: extensible 3D graphics for Web authors*. [S.l.]: Morgan Kaufmann, 2010.

CENTER, Climate Prediction. *Atmospheric Data - WGrib*. nov. 2013. Disponível em: <http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html>.

CHEW, L Paul. Constrained delaunay triangulations. *Algorithmica*, Springer, v. 4, n. 1-4, p. 97–108, 1989.

CHRISTIANSEN, Hank (Ed.). *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1984. ISBN 0-89791-138-5.

CONSORTIUM, Web3D. *X3D Developers*. nov. 2013. Disponível em: <http://www.web3d.org/x3d/>.

CONYERS, Lawrence B. *Ground-penetrating radar for archaeology*. [S.l.]: Rowman Altamira, 2004.

DALE, Anders M; SERENO, Martin I. Improved localizadon of cortical activity by combining eeg and meg with mri cortical surface reconstruction: a linear approach. *Journal of cognitive neuroscience*, MIT Press, v. 5, n. 2, p. 162–176, 1993.

DALY, L.; BRUTZMAN, D. X3d: Extensible 3d graphics standard [standards in a nutshell]. *Signal Processing Magazine, IEEE*, v. 24, n. 6, p. 130–135, 2007. ISSN 1053-5888.

DREBIN, Robert A; CARPENTER, Loren; HANRAHAN, Pat. Volume rendering. In: ACM. *ACM Siggraph Computer Graphics*. [S.l.], 1988. v. 22, n. 4, p. 65–74.

ENGEL, Klaus; KRAUS, Martin; ERTL, Thomas. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In: ACM. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*. [S.l.], 2001. p. 9–16.

HAASE, H et al. Meteorology meets computer graphics — a look at a wide range of weather visualisations for diverse audiences. *Computers Graphics*, v. 24, n. 3, p. 391 – 397, 2000. ISSN 0097-8493. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0097849300000352>.

HANSEN, C. D.; JOHNSON, C. R. *The visualization handbook*. [S.l.]: Elsevier, 2005.

IMAI, Takashi et al. Water molecules in a protein cavity detected by a statisticalmechanical theory. *Journal of the American Chemical Society*, v. 127, n. 44, p. 15334–15335, 2005. PMID: 16262373. Disponível em: <http://pubs.acs.org/doi/abs/10.1021/ja054434b>.

LECKEBUSCH, J.; PEIKERT, R. Investigating the true resolution and three-dimensional capabilities of ground-penetrating radar data in archaeological surveys: measurements in a sand box. *Archaeological Prospection*, v. 8, n. 1, p. 29–40, 2001.

LEE, Der-Tsai; SCHACHTER, Bruce J. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, Springer, v. 9, n. 3, p. 219–242, 1980.

LERTRATTANAPANICH, S.; BOSE, N.K. High resolution image formation from low resolution frames using delaunay triangulation. *Image Processing, IEEE Transactions on*, v. 11, n. 12, p. 1427–1441, 2002. ISSN 1057-7149.

LEWINER, Thomas et al. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, Taylor & Francis, v. 8, n. 2, p. 1–15, 2003.

LIVNAT, Y.; SHEN, Han-Wei; JOHNSON, C.R. A near optimal isosurface extraction algorithm using the span space. *Visualization and Computer Graphics, IEEE Transactions on*, v. 2, n. 1, p. 73–84, 1996. ISSN 1077-2626.

LORENSEN, W. E.; CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In: ACM. *ACM Siggraph Computer Graphics*. [S.l.], 1987. v. 21, n. 4, p. 163–169.

LUCERO, A. et al. Compressing three-dimensional grib meteorological data using klt and jpeg 2000. In: *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*. [S.l.: s.n.], 2003. v. 3, p. 1836–1838.

MACDONALD, David; AVIS, David; EVANS, Alan C. Multiple surface identification and matching in magnetic resonance images. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Visualization in Biomedical Computing 1994*. [S.l.], 1994. p. 160–169.

MIELACK, C. Isosurfaces. In: ZIB. *ZIB Zuse Institut Berlin*. [S.l.], 2009. p. 20.

OVERVELD, Kees van; WYVILL, B. Shrinkwrap: An efficient adaptive algorithm for triangulating an iso-surface. *The Visual Computer*, Springer-Verlag, v. 20, n. 6, p. 362–379, 2004. ISSN 0178-2789. Disponível em: <http://dx.doi.org/10.1007/s00371-002-0197-4>.

PAPATHOMAS, Thomas V; SCHIAVONE, James A; JULESZ, Bela. Applications of computer graphics to the visualization of meteorological data. In: ACM. *ACM SIGGRAPH Computer Graphics*. [S.l.], 1988. v. 22, n. 4, p. 327–334.

PEREIRA, A.R.; ANGELOCCI, L.R.; SENTELHAS, P.C. *Agrometeorologia: fundamentos e aplicações práticas*. Agropecuária, 2002. ISBN 9788585347710. Disponível em: <http://-/books.google.com.br/books?id=8U89ywAACAAJ>.

RAPOSO, A.N.; QUEIROZ, J.A.; GOMES, A.J.P. Triangulation of molecular surfaces using an isosurface continuation algorithm. In: *Computational Science and Its Applications, 2009. ICCSA '09. International Conference on*. [S.l.: s.n.], 2009. p. 145–153.

RHYNE, T. et al. Information and scientific visualization: separate but equal or happy together at last. In: *Visualization, 2003. VIS 2003. IEEE*. [S.l.: s.n.], 2003. p. 611–614.

RYMON-LIPINSKI, B. von et al. Efficient point-based isosurface exploration using the span-triangle. In: *Visualization, 2004. IEEE*. [S.l.: s.n.], 2004. p. 441–448.

SCHONBORN, Konrad; HOST, Gunnar; PALMERIUS, Karljohan. Visualizing the positive-negative interface of molecular electrostatic potentials as an educational tool for assigning chemical polarity. *Journal of Chemical Education*, v. 87, n. 12, p. 1342–1343, 2010. Disponível em: <http://pubs.acs.org/doi/abs/10.1021/ed100417c>.

SHEN, Han-Wei; JOHNSON, C.R. Sweeping simplices: a fast iso-surface extraction algorithm for unstructured grids. In: *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on*. [S.l.: s.n.], 1995. p. 143–150, 447. ISSN 1070-2385.

STACKPOLE, J. D. A guide to grib (edition 1): The wmo format for the storage of weather product information and the exchange of weather product messages in gridded binary form. *Natl. Meteorol. Cent., Natl. Weather Serv., Silver Spring, Md*, 1994.

SUTTON, Philip M et al. A case study of isosurface extraction algorithm performance. In: *Data Visualization 2000*. [S.l.]: Springer, 2000. p. 259–268.

SZYMCZAK, A.; VANDERHYDE, J. Extraction of topologically simple isosurfaces from volume datasets. In: *Visualization, 2003. VIS 2003. IEEE*. [S.l.: s.n.], 2003. p. 67–74.

TREECE, Graham M; PRAGER, Richard W; GEE, Andrew H. Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, Elsevier, v. 23, n. 4, p. 583–598, 1999.

WALLIN, A. Constructing isosurfaces from ct data. *Computer Graphics and Applications, IEEE*, IEEE, v. 11, n. 6, p. 28–33, 1991.

WANG, Jun; YU, Zeyun. Adaptive and quality tetrahedral mesh generation. In: *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. v. 3, p. 323–327.

WOOD, Zoë J et al. Removing excess topology from isosurfaces. *ACM Transactions on graphics*, v. 23, n. 2, p. 190–208, 2004.

YU, Tianbiao et al. Application research of x3d in individualized products customization. In: *Technology and Innovation Conference, 2006. ITIC 2006. International*. [S.l.: s.n.], 2006. p. 1699–1703. ISSN 0537-9989.

ZHOU, Chen; SHU, Renben; KANKANHALLI, Mohan S. Handling small features in iso-surface generation using marching cubes. *Computers Graphics*, v. 18, n. 6, p. 845 – 848, 1994. ISSN 0097-8493. Disponível em: <http://www.sciencedirect.com/science/article/pii-/0097849394900116>.

ZHU, Yuejian et al. The economic value of ensemble-based weather forecasts. *Bulletin of the American Meteorological Society*, v. 83, n. 1, p. 73–83, 2002.

Anexo A

wgrib v1.6.0 Wesley Ebisuzaki

Portable Grib decoder

"Wgrib"is a portable program to read grib files that were created by the NCEP/NCAR Reanalysis Project. Of course, the program is not restricted to Reanalysis files but Eugenia Kalnay is happy whenever she sees the phrase "NCEP/NCAR Reanalysis".

The documentation for wgrib is spread over several files, readme, readme.dos, formats.txt, grib2ieee.txt, notice, porting.txt, tricks.wgrib and usertables.txt and changes.

Running wgrib without any arguments displays a short help message.

Portable Grib decoder for NCEP Operations etc. it slices, dices v1.6.0 prelim 2 (7-01-97) Wesley Ebisuzaki usage: ./wgrib [grib file] [options]

Inventory/diagnostic output selection -s/-v/-V short inventory/verbose inventory/very verbose non-inventory (default) regular inventory Options for inventory/diagnostic output -PDS/-PDS10/-GDS/-GDS10 print PDS/GDS in hex/dec -verf print forecast verification time -4yr/-ncep$_o pn/-ncep_r eanseedocumentation$

Decoding Grib selection -d [record number] dump record number -p [byte position] dump record at byte position -i dump controlled by stdin (inventory list) (none) no decode .. inventory only Options for decoding Grib -text/-ieee/-bin/-grib dump to a text/ieee/bin/grib file -h/-nh dump will have headers (default)/no headers -H dump will include PDS and GDS (-bin/-ieee only) -append append to dump file -o [file] output file name, 'dump' is default

*** Standard Inventory ***

WGRIB's first duty is create an inventory. This inventory also serves as an index file. Using the test file land.grb you should be able to enter:

% wgrib land.grb Using NCEP reanalysis table, see -ncep$_o pn, -ncep_r eanoptions 1 : 0 : $d = 87010100 : LAND : kpds5 = 81 : kpds6 = 1 : kpds7 = 0 : TR = 0 : P1 = 0 : P2 = 0 : TimeU = 1 : sfc : anl : NAve = 1$

The first line indicates that wgrib couldn't figure out whether to use the reanalysis or operational grib tables. Since land.grb is from reanalysis, we should use the reanalysis tables.

Trying again, we get

% wgrib land.grb -ncep$_r ean1:0:d=87010100:LAND:kpds5=81:kpds6=1:kpds7=0:TR=0:P1=0:P2=0:TimeU=1:sfc:anl:NAve=1

The inventory consists of several fields separated by colons. The contents of the fields are:

1. Record number 2. Position in bytes 3. Date (YYMMDDHH). 4. Parameter name (LAND=land/sea mask) 5. Indicator of parameter and units (grib PDS octet 9) 6. Type of level/layer (grib PDS octet 10) 7. Height, pressure, etc (grib PDS octets 11-12) 8. Time Range (grib PDS octet 21) 9. Period of time 1, (grib PDS octet 19) 10. Period of time 2, (grib PDS octet 20) 11. Forecast time unit (grib PDS octet 18) 12. level 13. anl=analysis, fcst=forecast 14. NAve (number of grids used to make average)

*** Short Inventory ***

The short inventory can be obtained using the -s option. This inventory is easier to read the the previous inventory and can also be used as an index file.

%wgrib -s land.grb -ncep$_r ean1:0:d=87010100:LAND:sfc:anl:NAve=1

1. Record number 2. Position in bytes 3. Date (YYMMDDHH). 4. Parameter name (LAND=land/sea mask) 6. Type of level/layer (grib PDS octet 10) 7. Forecasts, analysis, etc 8. For an average, the number of fields averaged together

*** Verbose Inventory ***

The small verbose inventory can be obtained using the -v option. This inventory can be used as an index file.

% wgrib -v land.grb -ncep$_r ean1:0:D=1987010100:LAND:kpds=81,1,0:sfc:anl:"Land-seamask[1=land;0=sea]

1. Record number 2. Position in bytes 3. Date (YYYYMMDDHH). 4. Parameter name (LAND=land/sea mask) 5. KPDS5, KPDS6, KDPS7 (PDS Octets 9, 10, 11-12) 6. Type of level/layer (grib PDS octet 10) 7. Forecasts, analysis, etc 8. Description of parameter type

*** Verbose Description ***

The fourth type of file description can not be used as an index file. However, it is more human readable. It gives you information that is not normally available such as grid dimensions. Using the test file land.grb, you should be able to enter:

%wgrib land.grb -V -ncep$_r ean$

rec 1:0:date 1987010100 LAND kpds5=81 kpds6=1 kpds7=0 levels=(0,0) grid=255 sfc anl: LAND=Land-sea mask [1=land; 0=sea]

timerange 0 P1 0 P2 0 TimeU 1 nx 192 ny 94 GDS grid 4 num$_i n_a ve$1$missing$0$center$7$subcenter$0$proces
lat88.542000to−88.542000$long$0.000000to−1.875000by1.875000, (192x94)$scan$0$bdsgrid$1

min/ max data 0 1 num bits 4 BDS$_R ef$0$DecScale$1$BinScale$0

The first line states the record 1 starts at byte position 0 the initial date is January 1, 1987 at 00Z the parameter is "LAND"(numeric code 81, PDS octet 9) with a level type 1 (kdps6=1, PDS octet 10) and value 0 (PDS octets 11-12) or levels(0,0) (PDS octet 11, PDS octet 12) with a user defined grid (grid=255) and it is a surface analysis

The second line is a further description of the parameter type

The third line describes timerange (PDS octet 21) P1 (PDS octet 19) P2 (PDS octet 20) TimeU (PDS octet 14) nx ny grid size as used by wgrib GDS grid (GDS octet 6) num$_i n_a ve$($PDSoctet$22 − 23)$numbermissing from average$($PDSoctet$24)$

The fourth line describes center (PDS octet 5) subcenter (PDS octet 26) process (PDS octet 6) parameter table version (PDS octet 4)

The fifth and sixth lines describe the grid type

The last line describes minimum and maximum values of the data the number of bits used to store the data the minimum value the decimal and binary scaling used

Most of the information within this description will only make sense if you have a copy of the GRIB definition as reference.

If you want to determine the contents of record N, try the command:

%wgrib land.grib -V -d N

This command also writes a binary dump of the record but it's quick. If you don't want a binary dump, try (on a UNIX machine),

%wgrib land.grib -V -d N -o /dev/null

*** Extracting Data ***

The second major function of wgrib is to extract data from a grib file. The output can be binary, IEEE (big endian), grib and text. All output formats except grib can be written with or without a header. See FORMATS.TXT for more information. The '-append' option appends the extracted data and the '-o [filename]' allows you to set the default output file which is normally "dump".

Note: binary format with a header is often compatible with fortran code. Note: IEEE output is "big-endian". Note: writing in binary is faster than writing ieee. Note: using a binary format is faster, more precise and uses less disk space than the text format.

Note: The standard NCEP convention is that the arrays are stored in fortran order starting from the north and 0E. The following data goes south and eastward.

*** How to select data to be extracted ***

1) by record number wgrib land.grib -d 1 (extract first record)

2) by position wgrib land.grib -p 0 (extract record starting at byte 0)

3) by (machine readable) inventory (UNIX/AMIGA/MS-DOS) wgrib land.grb | wgrib -i land.grb -o output.bin

The third method is the most powerful one. Suppose you have a grib file with many different fields. You want to extract all the zonal winds (UGRD in NCEP files), you could type at a Unix machine:

wgrib grib$_file|grep$" : $UGRD : wgribgrib_file - i$

Suppose you want to extract the 500 mb U winds, then you could type at a Unix machine:

wgrib grib$_file - s|grep$" : $UGRD : grep$" : 500$mb : wgrib - igrib_file$

For more information on how to write ieee, binary, text and grib files see the file FORMATS.TXT.

*** Some Output Formats ***

Binary with a f77-style header

Suppose you wish to convert all the 500 mb heights (HGT in NCEP files) to binary with a header. The following line would convert "infile"to "outfile".

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i infile -o outfile

The "outfile"is often compatible with the fortran compiler.

Binary with no header

Suppose you wish to convert all the 500 mb heights (HGT) to binary with a NO header. The following line would convert "infile"to "outfile".

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i -nh infile -o outfile

The "outfile"is often compatible with fortran direct-access I/O.

Text

Converting a grib file into a text file is slow (reading and writing), takes up much more disk space and can have less precision. Nevertheless it has its uses.

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i -text infile -o outfile

IEEE

Most workstations computers use big-endian IEEE as their binary format. For these machines, one should not use the -ieee option as it is slower and could lose some precision. However, the following line will create a big-endian IEEE with f77-style headers.

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i -ieee infile -o outfile

Without headers, one would use

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i -nh -ieee infile -o outfile

GRIB

Suppose you have a large file with every variable imaginable. But you are a simple person with limited means. You only want the 500 mb heights and you have limited disk space. The following will extract the 500 mb heights as a grib file.

% wgrib -s infile | grep ":HGT:500 mb: wgrib -i -grib infile -o outfile

Anexo B

Códigos das configurações iniciais

Listing 7.1: *Caso 1*

```
<Shape>
</Shape>
```

Listing 7.2: *Caso 2*

```
<Shape>
<Appearance>
<Material
diffuseColor='" + cor + "'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
<Coordinate point='-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.3: *Caso 3*

```
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
<Coordinate point='0.0 -1.0 1.0 1.0 -1.0 0.0 1.0 0.0 1.0'/>
</IndexedFaceSet>
</Shape>
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
```

```
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
<Coordinate point='-1.0 0.0 1.0 -1.0 1.0 0.0 0.0 1.0 1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.4: *Caso 4*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.5: *Caso 5*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.6: *Caso 6*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.7: *Caso 7*

```
<Shape>
<Appearance>
<Material
   diffuseColor='"+ cor +"'
   transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2 3' solid='false'>
<Coordinate point='1.0 0.0 1.0 1.0 0.0 -1.0 0.0 -1.0 -1.0 0.0 -1.0 1.0'/>
```

```
</IndexedFaceSet>
</Shape>
<Shape>
<Appearance>
<Material
   diffuseColor='"+ cor +"'
   transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2 3' solid='false'>
<Coordinate point='0.0 1.0 1.0 0.0 1.0 -1.0 -1.0 0.0 -1.0 -1.0 0.0 1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.8: *Caso 8*

```
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
<Coordinate point='-1.0 0.0 -1.0 1.0 0.0 1.0 1.0 0.0 -1.0'/>
</IndexedFaceSet>
</Shape>
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
<Coordinate point='-1.0 0.0 -1.0 -1.0 -1.0 0.0 1.0 0.0 1.0'/>
</IndexedFaceSet>
</Shape>
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2' solid='false'>
```

```
<Coordinate point='1.0 0.0 1.0 -1.0 -1.0 0.0 0.0 -1.0 1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.9: *Caso 9*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.10: *Caso 10*

```
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2 3' solid='false'>
<Coordinate point='-1.0 0.0 1.0 1.0 0.0 1.0 1.0 0.0 -1.0 -1.0 0.0 -1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.11: *Caso 11*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.12: *Caso 12*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.13: *Caso 13*

```
<Shape>
<Appearance>
<Material
diffuseColor='"+ cor +"'
transparency='" + transparencia + "'>
</Material>
</Appearance>
<IndexedFaceSet coordIndex='0 1 2 3' solid='false'>
<Coordinate point='0.0 -1.0 1.0 -1.0 0.0 1.0 -1.0 0.0 -1.0 0.0 -1.0 -1.0'/>
</IndexedFaceSet>
</Shape>
```

Listing 7.14: *Caso 14*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```

Listing 7.15: *Caso 15*

```
<Shape>
   <IndexedFaceSet coordIndex='0 1 2' solid='false'>
      <Coordinate point=
         '-1.0 -1.0 0.0 0.0 -1.0 1.0 -1.0 0.0 1.0'/>
   </IndexedFaceSet>
</Shape>
```