

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DE COMPUTAÇÃO

NATHALIE REY DA SILVA

**VISUALIZAÇÃO 3D DE DADOS
OCEANOGRÁFICOS SIMULADOS**

Porto Alegre
2006

VISUALIZAÇÃO 3D DE DADOS OCEANOGRÁFICOS SIMULADOS

NATHALIE REY DA SILVA

Dissertação apresentada como requisito para a obtenção do grau de Mestre, pelo programa de Pós-Graduação em Ciência da Computação da Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador. Dr. Dalcidio Moraes Claudio

Porto Alegre
2006



Dados Internacionais de Catalogação na Publicação (CIP)

S586v Silva, Nathalie Rey da
**Visualização 3D de dados oceanográficos simulados /
Nathalie Rey da Silva. – Porto Alegre, 2006.**

75 f. : il.

Dissertação (Mestrado) – Fac. de Informática, PUCRS, 2006.
Orientador: Prof. Dr. Dalcídio Moraes Claudio.

1. Oceanografia. 2. Visualização Científica. 3. Simulação
(Programação de Computadores). 4. Estrutura de Dados.
5. Informática na Oceanografia. 6. Computação Gráfica.
I. Claudio, Dalcídio Moraes. II. Título.

CDD 006.6

**Ficha Catalográfica elaborada pelo
Setor de Processamento Técnico da BC-PUCRS**

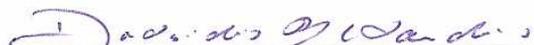
PUCRS

Campus Central
Av. Ipiranga, 6681 - prédio 16 - CEP 90619-900
Porto Alegre - RS - Brasil
Fone: +55 (51) 3320-3544 - Fax: +55 (51) 3320-3548
Email: bceadm@pucrs.br
www.pucrs.br/biblioteca



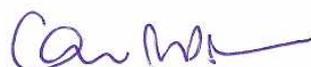
TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada “*Visualização 3D de Dados Oceanográficos Simulados*”, apresentada por Nathalie Rey da Silva, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Bioinformática e Modelagem Computacional, aprovada em 16/03/2006 pela Comissão Examinadora:


Prof. Dr. Dalcídio Moraes Claudio – PPGCC/PUCRS
Orientador


Prof. Dr. Márcio Serolli Pinho – PPGCC/PUCRS


Prof. Dr. Osmar Norberto de Souza – PPGCC/PUCRS


Profa. Dra. Carla Maria Dal Sasso Freitas – UFRGS

Homologada em...06/...11/...2006 conforme Ata No. 027... pela Comissão Coordenadora.


Prof. Dr. Fernando Luís Dotti
Coordenador.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família, pelo apoio em meus projetos pessoais e incentivo ao aprimoramento de meus estudos.

Aos meus amigos, que estiveram comigo e acompanharam todo o esforço dedicado nestes dois anos de estudos.

Aos meus colegas de curso, que me ajudaram nas dúvidas tanto em relação ao meu trabalho, como nas disciplinas do curso.

Agradeço também ao meu orientador, que me acompanhou e incentivou este trabalho.

Agradeço ao Dr. Angelo Sartori, pesquisador da área oceanográfica do CENPES, que me auxiliou durante o desenvolvimento deste projeto e forneceu os dados oceanográficos indispensáveis na pesquisa.

Em especial, agradeço a Dra. Isabel Mansour, que me acompanhou durante todo este trabalho e tornou possível o desenvolvimento desta dissertação.

Agradeço também o apoio financeiro da Capes e de todos professores da PUC.

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	MOTIVAÇÃO E OBJETIVOS.....	14
2	VISUALIZAÇÃO CIENTÍFICA.....	16
2.1	CLASSIFICAÇÃO DOS SISTEMAS DE VISUALIZAÇÃO.....	16
2.1.1	<i>Bibliotecas Gráficas e Pacotes de Apresentação Gráfica.....</i>	<i>16</i>
2.1.2	<i>Sistemas TurnKey.....</i>	<i>17</i>
2.1.3	<i>Construtores de Aplicação.....</i>	<i>17</i>
2.1.4	<i>Sistemas de Visualização Colaborativa.....</i>	<i>17</i>
2.2	CLASSIFICAÇÃO DE DADOS.....	18
3	DADOS OCEANOGRÁFICOS.....	20
3.1	COLETA DE DADOS.....	21
3.2	SIMULAÇÃO DE DADOS.....	22
3.2.1	<i>Modelo POM.....</i>	<i>23</i>
3.3	FORMATOS DE DADOS.....	26
3.3.1	<i>HDF.....</i>	<i>26</i>
3.3.2	<i>NetCDF.....</i>	<i>28</i>
4	TÉCNICAS DE VISUALIZAÇÃO DE DADOS OCEANOGRÁFICOS.....	31
4.1	VISUALIZAÇÃO DE DADOS VETORIAIS.....	32
4.1.1	<i>Hedgehog Display.....</i>	<i>32</i>
4.1.2	<i>Flow Lines.....</i>	<i>33</i>
4.1.3	<i>Colorwheel.....</i>	<i>35</i>
4.1.4	<i>Line Integral Convolution.....</i>	<i>35</i>
4.1.5	<i>Extraindo Características Topológicas.....</i>	<i>36</i>
4.1.6	<i>Técnicas de Visualização Volumétrica de Vetores.....</i>	<i>38</i>
4.2	COMPARAÇÃO DE MÉTODOS DE VISUALIZAÇÃO VETORIAL.....	38
4.3	VISUALIZAÇÃO DE DADOS ESCALARES.....	39
5	TRABALHOS RELACIONADOS.....	40
5.1	GeoZUI3D – GEOGRAPHIC ZOOMING USER INTERFACE.....	40
5.2	ISTV: INTERACTIVE STRUCTURED TIME-VARYING VISUALIZER.....	42
5.3	POPTX.....	44
5.4	Vis5D.....	45
5.5	ENVis/HUM.....	46
5.6	SISTEMA DE VISUALIZAÇÃO INTERATIVA.....	47
5.7	FERRET.....	47
5.8	ENSIGHT.....	49
6	DESCRIÇÃO DO PROTÓTIPO.....	50
6.1	AMBIENTE DE DESENVOLVIMENTO.....	51
6.2	ARQUITETURA DO SISTEMA.....	51
6.3	INTERFACE DO SISTEMA.....	58
6.4	TÉCNICAS DE VISUALIZAÇÃO E INTERAÇÃO.....	62
6.4.1	<i>Visualização de Dados Escalares.....</i>	<i>62</i>
6.4.2	<i>Visualização de Dados Vetoriais.....</i>	<i>63</i>
6.4.3	<i>Cortes Curvilíneos.....</i>	<i>65</i>
6.4.4	<i>Técnicas Básicas de Interação.....</i>	<i>68</i>
6.4.5	<i>Animação.....</i>	<i>68</i>
6.5	AValiação DO PROTÓTIPO.....	68
6.5.1	<i>Desempenho.....</i>	<i>69</i>
7	CONCLUSÃO E TRABALHOS FUTUROS.....	70

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional
3D	Tridimensional
API	<i>Application Program Interface</i>
AVS	<i>Advanced Visual Systems</i>
BOM	<i>Bergen Ocean Model</i>
CDF	<i>Common Data Format</i>
CEI	<i>Computational Engineering International</i>
CENPES	Centro de Pesquisa da Petrobras
COVISE	Collaborative Visualization and Simulation Environment
CSCV	<i>Computer-Supported Cooperative Visualization</i>
HDF	<i>Hierarchical Data Format</i>
IHC	<i>Interface Humano-Computador</i>
ISTV	<i>Interactive Structured Time-varying Visualizer</i>
LIC	<i>Line Integral Convolution</i>
MIT	<i>Massachusetts Institute of Technology</i>
MITgcm	<i>MIT General Circulation Model</i>
NCSA	<i>National Center for Supercomputing Applications</i>
NetCDF	<i>Network Common Data Form</i>
NSF	<i>National Science Foundation</i>
NOAA	<i>National Oceanic & Atmospheric Administration</i>
OCCAM	<i>Ocean Circulation and Climate Advanced Modeling</i>
POCM	<i>Parallel Ocean Circulation Model</i>
POM	<i>Princeton Ocean Model</i>
POP	<i>Parallel Ocean Program</i>
POSUM	<i>Parallel Oregon State University Model</i>
RGB	<i>Red-Green-Blue</i>
RGBA	<i>Red-Green-Blue-Alpha</i>
UML	<i>Unified Modeling Language</i>
UNESCO	<i>United Nations Educational, Scientific and Cultural Organization</i>
VTK	<i>Visualization Toolkit</i>

LISTA DE FIGURAS

Figura 3.1 - Sistema de coordenada sigma [Mellor, 2003]	24
Figura 3.2 - Grade retilínea	25
Figura 3.3 - Localização das variáveis na grade do modelo [Blumberg e Mellor, 1987].....	26
Figura 3.4 – Construtores do arquivo HDF [NCSA, 2005]	27
Figura 3.5 - Níveis de interação do HDF [Fonte: NCSA, 2005].....	28
Figura 3.6 - Exemplo da abstração NetCDF. Este conjunto de dados NetCDF contém três dimensões e 2 variáveis, cada uma com 3 atributos. [Fonte: Rew, 1990].....	29
Figura 4.1 – Aplicação da técnica <i>Straight Arrow Hedgehog Display</i> na visualização de fluídos [Fonte: Telea & Wijk, 1999].....	33
Figura 4.2 – Aplicação da técnica <i>Curved Arrow Hedgehog Display</i> na visualização de fluídos [Fonte: Telea & Wijk, 1999].....	33
Figura 4.3 – Exemplo de linhas de corrente [Fonte: Polito, 2004].....	34
Figura 4.4 – Exemplo de linha de partículas [Fonte: Polito, 2004].....	34
Figura 4.5 - Tabela circular de cores [Fonte: Moorhead et. al., 1999]	35
Figura 4.6 - Mar do Japão, com o uso da técnica <i>colorwheel</i> [Fonte: Moorhead et. al., 1999]	35
Figura 4.7 – (a) Ruído Branco, (b) campo vetorial 2D, (c) textura de ruído branco filtrada com o campo vetorial [Fonte: Mendonça 2001].....	36
Figura 4.8 – Processo de geração da imagem com a técnica <i>Line Integral Convolution</i> [Fonte: Universidade de Stuttgart, 2004]	36
Figura 4.9 - Técnica de mapeamento de cores [Rheingans, 2005].....	39
Figura 5.1 – Mapa digital de terreno [Fonte: GeoZui3D, 2004]	41
Figura 5.2 – Navegação no GeoZui3D [Fonte: GeoZui3D, 2004].....	41
Figura 5.3 – Temperatura do Golfo Pérsico com utilização de <i>color map</i> para valores de terreno [Fonte: ISTV, 2004]	43
Figura 5.4 – Iso-superfícies sobre o Oceano Pacífico utilizando o esquema de cores <i>color map</i> [Fonte: ISTV, 2004]	43
Figura 5.5 – Altura da superfície oceânica com a aplicação da técnica <i>hillshading</i> [Fonte: POPTEx, 2004]	44
Figura 5.6 – Altura da superfície oceânica: valores baixos em azul e valores altos em vermelho [McPherson et. al, 1998].....	45
Figura 5.7 – Temperatura em um período de tempo.....	46
Figura 5.8 – Batimetria [Fonte: Chupa, 2001]	47
Figura 5.9 – Topografia e batimetria	48
Figura 5.10 – Temperatura da superfície oceânica	49
Figura 6.1 - Diagrama de casos de uso	52
Figura 6.2 – Diagrama de classes para carregamento do volume	53
Figura 6.3 – Diagrama de classes para a técnica de mapeamento de cores	54
Figura 6.4 – Diagrama de classes para a técnica de <i>Hedgehogs</i>	55
Figura 6.5 – Diagrama de classes para a técnica de cortes curvilíneos	56
Figura 6.6 – Diagrama de classes para a geração da animação do volume.....	57
Figura 6.7 – Interface do protótipo	59
Figura 6.8 – Menu File.....	59
Figura 6.9 – Menu do Sistema.....	60
Figura 6.10 – Menu Tools	60
Figura 6.11 – Menu Movie.....	61

Figura 6.12 – Menu Window.....	61
Figura 6.13 – Menu Help	61
Figura 6.14 - Diagrama de fluxo de dados da técnica de mapeamento de cores.....	62
Figura 6.15 – Visualização da temperatura em Celsius	63
Figura 6.16 - Diagrama de fluxo de dados para a técnica de Hedgehogs.	64
Figura 6.17 - Visualização da velocidade das correntes do volume total (a) e aplicando zoom (b).	64
Figura 6.18 – Posição final da spline após sua manipulação	65
Figura 6.19– Sub-volume gerado pelo corte feito com a spline da figura 6.17	66
Figura 6. 20 – Identificadores das células do modelo estruturado	67

LISTA DE TABELAS

Tabela 2.1 – Classificação dos dados	18
Tabela 4.1 – Pontos críticos	37
Tabela 4.2 - Vantagens e desvantagem de métodos de visualização vetorial	38
Tabela 6.1 - Variáveis do modelo de dados	58

RESUMO

A Oceanografia é o estudo das características físicas e biológicas dos oceanos e dos mares. Diversos aspectos climáticos e meteorológicos são afetados pelo comportamento do oceano. Além disso, no fundo oceânico existem grandes depósitos de minerais, como petróleo e gás.

A simulação oceânica consiste em realizar previsões do seu comportamento, através do processamento de modelos numéricos. Com o avanço do poder computacional, grandes volumes de dados são gerados, obtendo com isso uma maior exatidão das informações. Assim, ferramentas para a visualização destes dados tornam-se imprescindíveis para auxiliar os oceanógrafos na análise e interpretação das informações.

O presente trabalho tem como objetivo apresentar o projeto e desenvolvimento de uma ferramenta para visualização 3D de dados oceânicos simulados. Os dados foram obtidos através do modelo *Princeton Ocean Model* (POM). Para observar as características físicas do oceano, foram desenvolvidas técnicas para a visualização de dados escalares e vetoriais, seleção de regiões de interesse utilizando cortes curvilíneos e animação do volume para observar a variação dos parâmetros ao longo do tempo, através da visualização tridimensional do oceano.

Para este trabalho, inicialmente foi realizado um estudo das técnicas e algoritmos de visualização de dados oceânicos. Também foram abordadas diversas características presentes em sistemas de visualização, que serviram como base para especificação das funcionalidades desenvolvidas na ferramenta.

Palavras-chave: Visualização de Dados Oceanográficos, Oceanografia, Visualização Científica, Interação.

ABSTRACT

Oceanography is the study of the physical and biological features of the oceans and seas. Several climatic and meteorological aspects are affected by the ocean behavior. Besides, great mineral deposits, such as petroleum and gas, exist deep in the ocean.

The simulation of oceanic data consists in forecasting oceanic behavior through the processing of numeric models. With the advance of computational power, great volumes of data are generated, obtaining as a result a better information accuracy. Thus, tools for the visualization of these data become essential to aid the oceanographers in the information analysis and interpretation.

The main goal of this work is to present the design and development of a tool for 3D visualization of simulated oceanic data. The data were obtained through the *Princeton Ocean Model* (POM). Scalar and vector visualization, selection of region of interest using curvilinear cuts, and animation of the volume were developed to observe the physics aspects of the ocean and the parameters variation along the time, through 3D visualization.

In this work, initially a study of techniques and visualization algorithms of oceanic data were done. Several features present in the visualization tools were also approached, which served as base for the specification of the necessary functionalities of the system.

Keywords: Oceanographical Data Visualization, Oceanography, Scientific Visualization, Interaction.

1 INTRODUÇÃO

A oceanografia, segundo a UNESCO, é uma ciência universal que tem por objetivo o estudo do meio marinho, sua flora, fauna e seus limites físicos com a terra firme e a atmosfera. Sendo uma ciência multidisciplinar e interdisciplinar, a oceanografia é convenientemente dividida em: Oceanografia Física, Oceanografia Biológica, Oceanografia Química e Oceanografia Geológica.

O oceano é um espaço 3D que varia temporalmente e é habitado por seres vivos de todas as escalas (de microrganismos a baleias), estando sujeito a uma grande variedade de funções de força, incluindo termodinâmica, sísmica, atmosférica, inercial e marítima [Rosenblum et al., 1989]. O estudo do oceano é importante pelos aspectos climáticos e meteorológicos. Além disso, a física, a química, a biologia e a geologia dos oceanos são fundamentais para o desenvolvimento e gerenciamento dos recursos vivos. Também existem no fundo do oceano grandes depósitos de minerais sólidos, como o diamante, e também minerais solúveis, como o petróleo e o gás.

Na oceanografia, muitos parâmetros oceanográficos estão envolvidos, tais como salinidade, temperatura, velocidades das correntes, densidade, pressão e pH. Estes dados podem ser obtidos através de simulação ou coleta de dados. Na simulação, modelos de dados numéricos, normalmente desenvolvidos em linguagens de programação como C ou Fortran, são utilizados para representar o comportamento do oceano. Estes modelos são processados e geram informações sobre a evolução das características oceânicas ao longo do tempo. Na coleta de dados, bóias e sondas captam dados reais do oceano em intervalos de tempo definidos.

A área de Visualização Científica engloba as técnicas e ferramentas necessárias para a exploração de dados e informações graficamente. Normalmente, os dados utilizados na Visualização Científica possuem um grande volume e são apresentados sob a forma numérica. O objetivo da Visualização Científica é propiciar uma maneira de analisar estes dados, procurando também definir relacionamentos entre as suas diferentes partes.

Os dados oceanográficos correspondem a dados científicos e são apresentados sob a forma numérica. Portanto, para visualizar e analisar estes dados, são utilizadas ferramentas de visualização científica. Através de técnicas de visualização apropriadas também se busca diminuir o tempo de análise.

Grandes esforços têm sido feitos no campo da visualização de dados oceanográficos. Em particular, o desafio tem sido encontrar o método apropriado ou o conjunto de métodos que melhor expressem o significado destes dados. A determinação de tais métodos depende de informações, como a estrutura dos dados, a escala do problema e as questões que devem ser respondidas [Gaither & Moorhead, 2002].

1.1 MOTIVAÇÃO E OBJETIVOS

Segundo Head [Head et al., 1997] é comprovado que as técnicas bidimensionais na análise de dados oceanográficos são suficientes para muitas tarefas de análise, revisão e planejamento. Entretanto, um melhor entendimento do conjunto de dados e uma redução no tempo de análise podem ser obtidos acrescentando técnicas de visualização tridimensional aos métodos tradicionais.

Atualmente existem algumas ferramentas gratuitas e comerciais que são utilizadas para a visualização e análise dos dados oceanográficos. Estas ferramentas disponibilizam diversas técnicas e métodos de interação. Porém, tais ferramentas deveriam ser adaptadas às necessidades de trabalho de cada instituição. Com isto, ferramentas gratuitas perdem neste quesito, pois, muitas vezes, não possuem um suporte eficiente ou os usuários não possuem habilidade para estender a ferramenta. Por outro lado, as ferramentas comerciais podem suprir todos estes problemas oferecendo um bom suporte, mas possuem um alto custo, já que o número de usuários é limitado.

Este trabalho tem como objetivo o projeto e desenvolvimento de uma ferramenta para a visualização de dados oceanográficos gerados através de simulação. Para isto, foi necessário pesquisar diversas técnicas de visualização e interação 3D, que possibilitassem agregar valor na visualização e análise dos fenômenos físicos envolvidos.

Como um estudo de caso, são utilizados os dados oceanográficos fornecidos pelo Centro de Pesquisa da Petrobras (CENPES), obtidos através de simulação e utilizando o modelo *Princeton Ocean Model* (POM). Os oceanógrafos do CENPES mostraram um grande interesse em contribuir para a pesquisa de visualização de dados oceanográficos gerados por simulação. Para o desenvolvimento da ferramenta, foi utilizada a linguagem de programação C++ [Mueller, 2002] e a biblioteca *Visualization Toolkit* [Schroeder, 2003].

1.2 ORGANIZAÇÃO DO TEXTO

O trabalho está dividido da seguinte maneira. O capítulo 2 apresenta uma introdução à área de visualização científica, abordando aspectos da visualização de dados oceanográficos. Os dados oceanográficos e os arquivos utilizados para o armazenamento destas informações são explicados no capítulo 3, com ênfase nos métodos de obtenção, que envolve a coleta de dados e a simulação. O modelo numérico de simulação *Princeton Ocean Model* foi detalhado neste capítulo, já que os dados utilizados neste trabalho foram gerados pelo processamento deste modelo.

O capítulo 4 apresenta um estudo das técnicas e algoritmos de visualização utilizados para dados oceanográficos, que agregam melhor valor na análise dos dados. O capítulo 5 apresenta diversas ferramentas oceanográficas, tanto as gratuitas como as comerciais. Neste capítulo são abordadas as técnicas de visualização, interação, modelo de dados, arquivos de saída e plataforma suportadas em cada ambiente de visualização.

O protótipo desenvolvido para este trabalho é apresentado no capítulo 6, incluindo uma descrição de suas funcionalidades, técnicas de visualização, projeto e implementação. Também são apresentados os resultados obtidos com o uso da ferramenta, além da descrição do próprio ambiente. Por fim, no capítulo 7, são apresentadas as conclusões deste trabalho, suas principais contribuições e possíveis extensões para futuros trabalhos.

2 VISUALIZAÇÃO CIENTÍFICA

A área de Visualização Científica abrange as técnicas e ferramentas utilizadas para a análise de dados e informações científicas graficamente [Earnshaw, 1992]. De maneira simplificada, consiste no processo de transformar dados numéricos em uma imagem para obter um melhor entendimento dos dados.

A Visualização Científica engloba técnicas de Computação Gráfica e de Processamento de Imagens. Nesta área, também são considerados aspectos envolvendo a percepção humana e as capacidades cognitivas, pois a forma como o usuário percebe e interage com a ferramenta de visualização pode influenciar o entendimento dos dados e a usabilidade do sistema. Interfaces visuais eficazes permitem observar, manipular, procurar, navegar, explorar, filtrar, descobrir, entender e interagir com grandes volumes de dados mais rapidamente e eficientemente [Gershow, 1997].

A diferença entre a Visualização Científica e uma simples representação gráfica é que a última se concentra primeiramente com a transmissão de informação e de resultados que já estão entendidos. Em Visualização Científica procura-se fornecer diferentes meios para entender os dados [Earnshaw & Wiseman, 1992].

2.1 CLASSIFICAÇÃO DOS SISTEMAS DE VISUALIZAÇÃO

Os sistemas de Visualização, segundo Gallagher [Gallagher, 1995], podem ser classificados em: bibliotecas gráficas e pacotes de apresentação gráfica, sistemas *turnkey*, construtores de aplicação e sistemas de visualização colaborativa. Exemplos de cada tipo de sistema, bem como as diferenças entre eles são apresentados nesta seção.

2.1.1 Bibliotecas Gráficas e Pacotes de Apresentação Gráfica

Esta categoria é a mais antiga. Neste caso, as bibliotecas e pacotes fornecem uma API (*Application Program Interface*), ou seja, um conjunto de rotinas, para o desenvolvimento de sistemas de visualização. Como é necessário implementar a aplicação propriamente dita, ou

seja, o programa principal, a interface com o usuário, o mapeamento geométrico e o gerenciamento de dados, esta classe de sistema é voltada para o programador.

Atualmente, existem diversas bibliotecas de visualização. As bibliotecas OpenGL [Hearn e Baker, 2004] e *Visualization Toolkit* [Avila et al., 2003] são as mais conhecidas e estão disponíveis gratuitamente para o uso.

2.1.2 Sistemas *TurnKey*

Sistemas *turnkey* são sistemas que provêm uma série de funcionalidades que podem ser acessadas através de uma *interface* gráfica, dispensando a necessidade de programação, uma vez que consistem em sistemas “prontos para usar”. Apesar desta facilidade, em geral, tais sistemas são limitados, pois podem oferecer apenas algumas funcionalidades ou uma parte da solução que o usuário necessita.

Muitos produtos desta categoria são aplicações específicas, como, por exemplo, para a exploração de petróleo, modelagem molecular e modelagem arquitetônica [Earnshaw e Wiseman, 1992]. Exemplos de sistemas desta categoria incluem o Ensign (comercializado pela empresa CEI - *Computational Engineering International*), o IDL (desenvolvido pelo RSI - *Research Systems*), o MatLab e o Maple.

2.1.3 Construtores de Aplicação

Esta categoria de software é uma combinação de sistemas *turnkey* e bibliotecas gráficas. Estes sistemas provêm uma série de módulos pré-definidos, mas o usuário também pode adicionar ou substituir por módulos próprios. Cada módulo provê uma funcionalidade específica e possui uma entrada e uma saída. Através da combinação de diferentes módulos é possível construir diferentes aplicações.

Alguns sistemas desta categoria são AVS (*Advanced Visual Systems*), *Data Explorer* (desenvolvido pela IBM), *IRIS Explorer* (desenvolvido pela Silicon Graphics), VisiQuest (antigamente conhecido como kronosPro, atualmente comercializado pela empresa AccuSoft).

2.1.4 Sistemas de Visualização Colaborativa

São sistemas que permitem a colaboração entre diversos usuários que estão fisicamente distantes, permitindo a inserção de anotações e debates em tempo real. Esta categoria de sistemas foi impulsionada pela expansão da Internet. Basicamente, estes sistemas possuem característica de sistemas *turnkey*, contendo ferramentas de colaboração, para troca de informações entre os usuários. Estes sistemas são chamados de *Computer-Supported Cooperative Visualization (CSCV)*.

O COVISE (*Collaborative Visualization and Simulation Environment*) é um sistema projetado para um ambiente distribuído, integrando simulações, pós-processamento e funcionalidades de visualização, utilizado em diversas áreas de visualização científica.

2.2 CLASSIFICAÇÃO DE DADOS

Segundo Freitas [Freitas, 1994], os dados científicos podem ser classificados de acordo com a classe de informação que representam, os tipos de valores que podem assumir, a dimensão e a natureza do domínio. A tabela 2.1 apresenta um esquema desta classificação.

Tabela 2.1 – Classificação dos dados

Critério	Classes	Significado
Classe de informação	Característica	Característica Isolada, Independente
	Escalar	Grandeza Escalar, amostrada de uma função
	Vetor	Grandeza Vetorial, amostrada de uma função
	Tensor	Grandeza Física
	Agregação	Coleção de atributos
Tipo dos valores	Alfanumérico	Valores de Identificação
	Numérico	Valores ordinais, discretos ou contínuos
	Símbolo	Sub-atributo
Natureza do Domínio	Discreto	Enumeração, conjunto finito ou infinito
	Contínuo	Todos os pontos no espaço 1D, 2D, 3D, nD
	Contínuo-Discretizado	Regiões no espaço 1D, 2D, 3D, nD
Dimensão do Domínio	1D	Dado definido no espaço 1D
	2D	Valor associado ao espaço 2D
	3D	Valor associado à posição no espaço 3D
	nD	Valores no espaço n-dimensional

Fonte: [Freitas, 1994].

De acordo com a tabela 2.1, a classe de informação que um dado representa pode ser:

- **Característica:** característica isolada de uma entidade, sem função associada. Por exemplo, o comprimento e altura de um objeto.
- **Escalar:** conjunto de valores amostrados de uma relação definida no espaço da entidade. Por exemplo, a temperatura de uma cidade.
- **Vetor:** representa grandezas que possuem magnitude, direção e sentido (e.g. curso de um rio).
- **Tensor:** representa grandezas físicas, usado em aplicações como dinâmica de fluidos computacionais e análise de elementos finitos.

- Agregação: representa uma coleção de dados. Como exemplo, a cor de um *pixel* no modelo RGB, formada pela combinação de três cores (vermelho, verde e azul).

Os tipos de valores, que estão relacionados com a natureza do dado, podem ser numéricos, alfanuméricos e simbólicos. A natureza de um domínio indica se um dado pode estar definido num domínio discreto, restrito a um conjunto finito ou infinito de elementos, num domínio contínuo ou num domínio contínuo-discretizado. Em um domínio contínuo, os valores podem estar definidos para todos os pontos (continuamente) ou para regiões. Considerando uma região geográfica como o objeto em estudo, o dado que designa a altura do terreno é definido para todos os pontos do domínio contínuo, enquanto outro que indica a densidade populacional em sub-áreas é definido por regiões, sobre o mesmo domínio contínuo [Manssour, 1998].

Considerando a dimensão, um dado pode ser:

- Unidimensional (1D): Por exemplo, a distância entre dois pontos;
- Bidimensional (2D): Por exemplo, mapas geográficos;
- Tridimensional (3D): Por exemplo, áreas geográficas, contendo a posição e altura de entidades geográficas;
- N-Dimensional (nD): Obtidos de aplicações que geram dados multi-variados, como por exemplo, dados estatísticos.

3 DADOS OCEANOGRÁFICOS

A água do mar possui grande variedade de sais e minerais em solução. Os recursos minerais do fundo oceânico, ao contrário dos contidos na água, costumam concentrar-se na superfície e no interior de rochas e sedimentos do solo submarino. O movimento das águas oceânicas profundas é regido principalmente por variações de temperatura e salinidade. A água fria é mais pesada do que a quente porque se condensa ao esfriar. O sal também aumenta o peso da água [Scofano e Nunes, 2001].

As propriedades físicas da água do mar, importante para o estudo da dinâmica dos fluidos, dependem da temperatura, salinidade e pressão [Scofano e Nunes, 2001]. A salinidade é a massa em gramas das substâncias sólidas contidas em um quilo de água do mar. As variações de salinidade dos oceanos são causadas pelos processos físicos de congelamento, precipitação ou evaporação, cujo efeito é retirar ou adicionar água doce à solução. Geralmente a salinidade das águas oceânicas varia de 34 a 36‰. A temperatura é a propriedade conservativa da água do mar a menos da camada mais superficial, submetida a grandes trocas de calor com a atmosfera. A temperatura em mar aberto limita-se entre -2 graus Celsius (controlado pelo ponto de congelamento da água salgada) e 30 graus Celsius (controlado pela capacidade de mistura e condutividade térmica da água do mar). A pressão possui como unidade de medida o decibar e nos oceanos a pressão aumenta com a profundidade.

As velocidades das correntes marinhas mais comum encontram-se em torno de 1.2 nós¹. As causas principais da velocidade são agrupadas em duas categorias: as intrínsecas à água do mar e as extrínsecas. As causas intrínsecas são representadas pela temperatura e salinidade, fatores estes que fazem alterar a densidade da água, tornando-a mais pesada ou mais leve. Por outro lado, a flutuação de salinidade é consequência de causas extrínsecas, como o vento e a chuva [Scofano e Nunes, 2001].

¹ Um nó equivale a uma velocidade de 0.5144 m/s

Os dados oceanográficos podem ser obtidos através da coleta de dados ou simulação. A coleta de dados permite obter dados reais sobre as características oceânicas. A simulação utiliza modelos numéricos sobre o oceano, que são processados, obtendo-se, assim, a previsão do seu comportamento. Após a simulação, os dados são enviados para arquivos específicos para posterior visualização.

3.1 COLETA DE DADOS

A coleta de dados oceanográficos pode ser realizada através de embarcações de pesquisas, bóias e sondas, com ou sem transmissores de satélite, ou através de sensoriamento remoto. Em pesquisas oceânicas, muitas vezes é necessário ir até a área do oceano para a realização do estudo. Os navios são equipados com diversos equipamentos como guincho e pequenos barcos, além de equipamentos de satélite para transmissão dos dados. Estes navios também possuem um laboratório para o processamento e análise dos dados.

As bóias e as sondas coletam diferentes tipos de dados, tais como salinidade, velocidade da água, luz, temperatura, pH, geografia do fundo do oceano e som. Algumas delas são equipadas com transmissores de satélites. Atualmente, o satélite Argos é o único satélite dedicado para o monitoramento do meio-ambiente. Neste caso, as bóias, equipadas com equipamentos transmissores, enviam os dados coletados para o satélite. O sistema de monitoramento Argos é mais econômico, poupando tempo de navegação e pesquisadores no local da coleta.

O sensoriamento remoto não se limita à geração e interpretação de dados na forma de imagens. Aliado aos recursos computacionais, o sensoriamento remoto apresenta novas perspectivas para o entendimento do oceano. Na área oceanográfica, é possível utilizar ondas ultra-sônicas, que se propagam por grandes distâncias submarinas. Com isso é possível medir a batimetria², realizar inspeções submarinas, caçar minas submersas e detectar cardumes.

A quantidade de parâmetros oceanográficos que podem ser medidos e monitorados por sensoriamento remoto é bem ampla. As aplicações dos dados orbitais são tão diversas que podemos considerar este meio de aquisição de informações para a oceanografia como um todo – biológica, química, geológica e física – tão eficaz como as informações obtidas por meios convencionais [Kampel, 2002].

² Profundidade do oceano

Recentemente, também foi desenvolvido no Instituto de Oceanografia Woods Hole, em Massachusetts, o primeiro robô para exploração autônoma. Ele foi programado para se mover sem piloto, já que o monitoramento de uma área se prolonga por um longo período de tempo.

3.2 SIMULAÇÃO DE DADOS

A simulação de dados é um método que permite entender e prever o comportamento do oceano, através da modelagem de dados. Estes modelos de dados são implementados em linguagens de baixo nível, como Fortran e C, e geralmente o processamento é multi-processado, para melhorar o desempenho. A pesquisa sobre modelagem oceânica começou em 1960 e até hoje tem sido limitada pelo computador. O aumento da velocidade das máquinas permite que grandes problemas possam ser resolvidos em curtos espaços de tempo.

O sistema oceânico é dividido por massas de terra, variando na densidade e profundidade. Forças externas, como vento, influenciam o seu comportamento. O modelo oceânico é uma solução computacional, que utiliza leis físicas de conservação de massa, movimento, calor e diversas outras, além de estimação de campos de força [Peng et al., 2002]. Sendo assim, é possível prever correntes oceânicas e outras propriedades, como temperatura, salinidade e parâmetros biológicos ou químicos.

Muitos modelos oceânicos foram desenvolvidos durante a última metade do século. Alguns são computacionalmente simples e prevêm um limitado número de variáveis oceanográficas, como modelos relativos à maré ou modelos climáticos de ondas. Outros, como modelos de circulação geral do oceano baseados em equações primitivas³, são computacionalmente caros e resolvem diversas equações para prever correntes oceânicas tridimensionais e Temperatura-Salinidade (T-S) [Peng et al., 2002].

A modelagem de regiões costeiras auxilia na recreação, navegação, operações de resgate, derramamento de óleo, entre outras. Em largas escalas, as correntes oceânicas afetam o clima, já que carregam calor ao redor do globo e também são capazes de absorver e redistribuir gases, tal como o dióxido de carbono.

³ São equações utilizadas para expressar as leis físicas, com equações de movimento (segunda lei de Newton), primeira lei de termodinâmica, equações de continuidade de massa.

Os modelos oceânicos utilizam um sistema de grades. Dependendo do espaçamento da grade do modelo, maior o tempo necessário de simulação. Por exemplo, 1/10 de grau de longitude e latitude corresponde a 11 Km de espaçamento entre as grades do modelo. Quanto menor for este espaçamento, maior é o tempo de processamento, pois o número de informações é maior. Tipicamente, os modelos oceânicos são processados por 10 ou mais anos simulados.

Uma aplicação que permita visualizar o modelo oceânico resulta em uma importante ferramenta que ajuda a acrescentar realismo ao modelo, identificar e diagnosticar problemas e aprender mais sobre o oceano e seu comportamento [Chupa et al., 2004].

A geração de dados através de massivas simulações, executadas por processamento paralelo, tem ultrapassado nossa habilidade de visualizar grandes conjuntos de dados. O tamanho dos modelos gerados por simulações distribuídas torna difícil coletar os resultados em um único computador para visualização. Com isso, simulações armazenam menos dados para poucos períodos de tempo, e visualizações são geradas menos frequentemente [Law, 2001].

Alguns modelos de dados utilizados em simulações oceanográficas são: OCCAM (Ocean Circulation and Climate Advanced Modelling), POCM (Parallel Ocean Circulation Model), MITgcm (MIT General Circulation Model), BOM (Bergen Ocean Model) e POSUM (Parallel Oregon State University Model) e POM (Princeton Ocean Model) descrito na seção 3.2.1. Após o processamento, os resultados são enviados para arquivos de saída para posterior visualização.

3.2.1 Modelo POM

Princeton Ocean Model é um modelo oceânico de circulação geral⁴ baseado na integração numérica das equações primitivas que regem a dinâmica da circulação oceânica, utilizando o método de diferenças finitas⁵. Este modelo foi desenvolvido por Blumberg e Mellor [Blumberg e Mellor 1987] e aplicado a problemas oceanográficos no programa de Ciências Atmosféricas e Oceânicas da universidade de Princeton, no laboratório de Fluidos Dinâmicos Geofísicos do NOAA e Análises Dinâmicas de Princeton. A idéia era que o POM representasse a física do oceano do modo mais realístico possível, através de um modelo tridimensional da costa oceânica.

⁴ Modelo computacional da circulação oceânica da Terra, utilizado para a simulações climáticas.

⁵ O método das diferenças finitas é um método de resolução de equações diferenciais que se baseia na aproximação de derivadas por diferenças finitas. A fórmula de aproximação obtém-se da série de Taylor da função derivada.

O POM é um modelo de coordenada sigma, onde o sigma substitui a coordenada z utilizada na direção vertical. A coordenada sigma é literalmente definida como uma porcentagem da coluna d'água, variando entre 0 na superfície e -1 no fundo. A coordenada sigma está definida na equação 3.1, na qual z representa a coordenada cartesiana z e d representa a profundidade da coluna d'água, com σ variando de 0 a -1 (possuindo 21 níveis sigma). Desta forma, a coluna d'água sempre possui 21 níveis, conforme ilustra a figura 3.1, onde η é a elevação da superfície, z é a coordena cartesiana z , h é a altura da coluna d'água e σ corresponde ao nível sigma.

$$\sigma = z/d \quad (3.1)$$

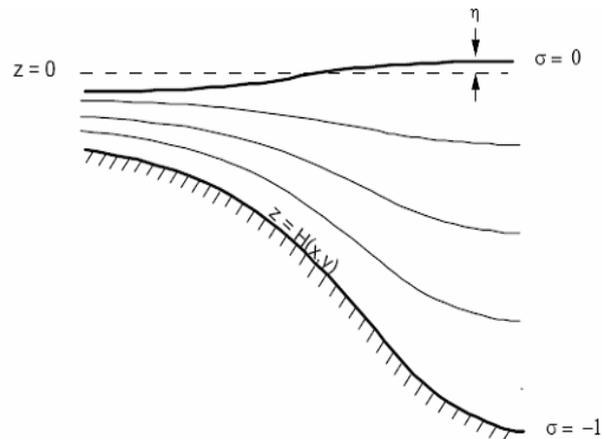


Figura 3.1 - Sistema de coordenada sigma [Mellor, 2003]

Este sistema de coordenada sigma é um importante atributo para lidar com a variabilidade topográfica [Mellor 2003]. Ao invés de trabalharmos com distâncias verticais, a espessura da camada depende da profundidade da coluna d'água, obtendo o mesmo número de camadas em todos os pontos.

Os primeiros modelos oceânicos 3D eram multi-níveis, nos quais os modelos de coordenadas z eram aplicados aos oceanos, possuindo forma retangular e profundidade uniforme. Os oceanos reais possuem uma geometria complicada, com grandes mudanças de profundidade, que variam de alguns metros perto da costa a alguns quilômetros nas bacias profundas. É difícil resolver igualmente a coluna da água e ser também eficiente em regiões rasas e profundas de uma bacia. Como o número de níveis verticais em um modelo do oceano é regulado, principalmente por economia de processamento, o fundo do oceano ou as regiões próximas da superfície têm que ser sacrificadas. Assim, para variações fortes de topografia,

modelos de coordenadas z são inerentemente desvantajosos. Modelos tridimensionais de coordenadas z são mais adequados para aplicações de bacias oceânicas profundas. Aplicações deste modelo em regiões litorâneas rasas ou bacias geralmente excluem as regiões rasas. Em um sistema de coordenada sigma, o número de níveis verticais da coluna de água é o mesmo em qualquer lugar.

A grade horizontal do modelo é uma grade retilínea, conforme é mostrado na figura 3.2. Nesta grade, a distância entre as células pode variar, ou seja, os elementos não são idênticos, mas são alinhados aos eixos. Em um volume, os dados desta grade são compostos por células hexaédricas, dispostas topologicamente, que podem ser representadas como uma matriz de índices (i, j, k) . Uma característica desta representação é que ela apresenta uma estrutura de conectividade implícita entre os elementos, permitindo que todos os vizinhos de uma célula sejam obtidos pelo incremento (ou decremento) do índice da célula atual. Esta característica define o que é chamado de malha estruturada [Speray e Kennon 1990].

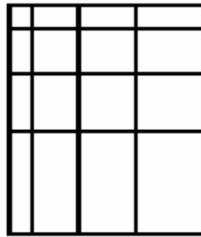


Figura 3.2 - Grade retilínea

Além da profundidade da coluna d'água, é possível visualizar também a elevação da superfície, que é processada conforme ilustra a equação 3.2, na qual H é a altura da coluna d'água, η é a elevação da superfície e z é a coordena cartesiana z que corresponde ao nível sigma.

$$\sigma = \frac{z - \eta}{H + \eta} \quad (3.2)$$

As variáveis de prognóstico do modelo são os três componentes da velocidade, temperatura, salinidade e turbulência. A figura 3.3 apresenta a localização das variáveis na grade, onde u , v e w são os componentes da velocidade, T corresponde à variável temperatura, S corresponde à salinidade, Δx e Δy correspondem ao espaçamento da grade em x e y , respectivamente, e $\Delta \sigma$ é o incremento vertical.

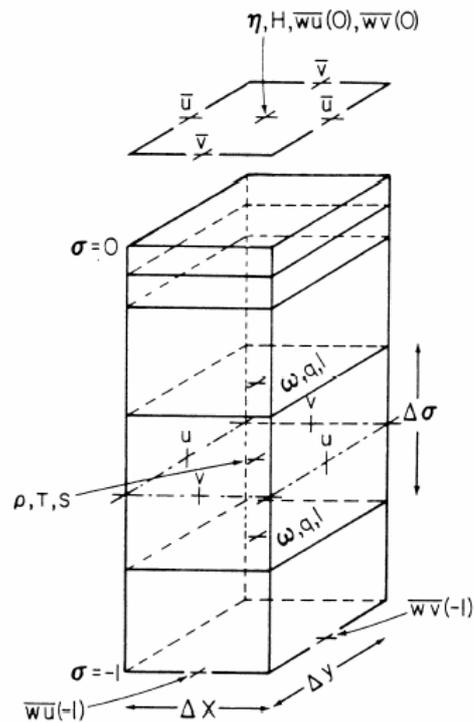


Figura 3.3 - Localização das variáveis na grade do modelo [Blumberg e Mellor, 1987]

3.3 FORMATOS DE DADOS

Diversos formatos de dados são utilizados para armazenar os dados numéricos oceanográficos. Os dados gerados em uma simulação são transferidos para arquivos de saída, para a visualização e análise. As aplicações da área de oceanografia geralmente trabalham com dados nos formatos HDF, NetCDF, CDF, entre outros.

3.3.1 HDF

Hierarchical Data Format (HDF) é um formato de arquivo multi-objeto utilizado para transferência de dados científicos gráficos e pontos flutuantes em um ambiente distribuído com diferentes plataformas de *hardware* [HDF 2005]. Foi desenvolvido pelo *National Center for Supercomputing Application* (NCSA).

As estruturas de dados HDF consistem em construtores primários para armazenamento dos dados. Os construtores utilizados no arquivo HDF são: *include raster image*, *palette*, *scientific data set*, *annotation*, *vdata*, and *vgroup*, conforme mostra a figura 3.4.

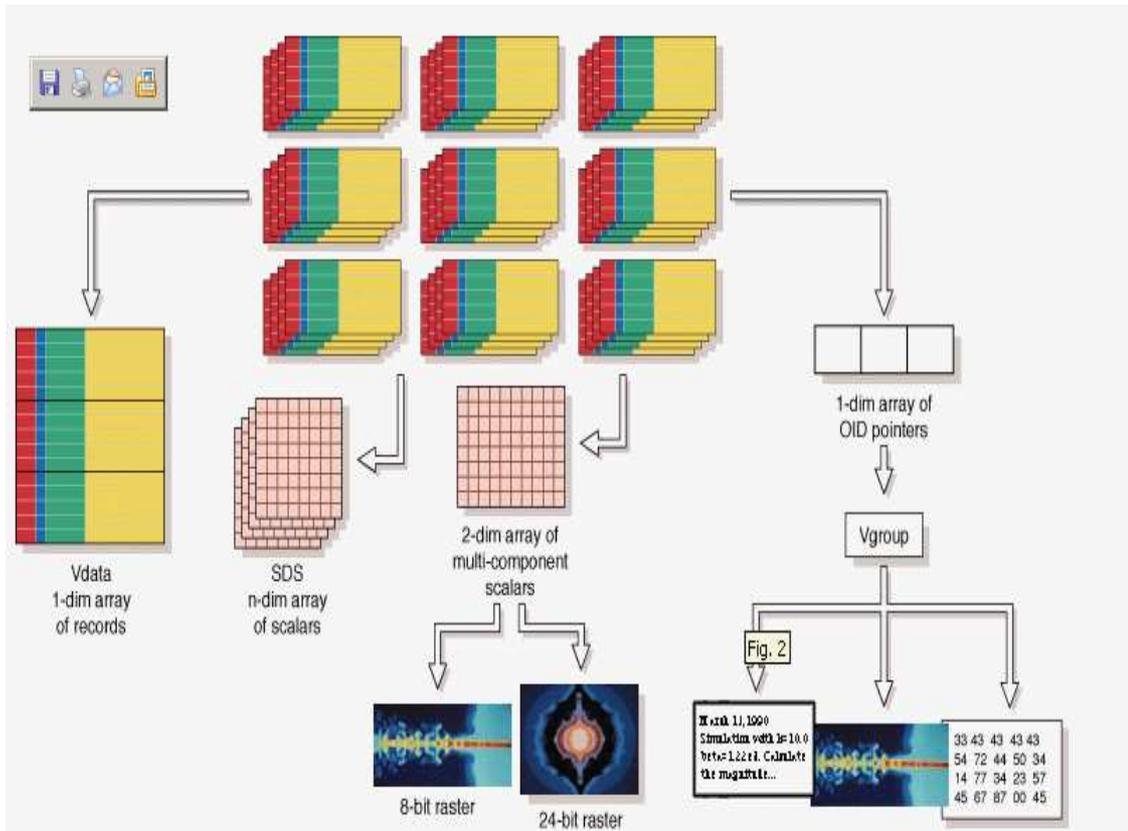


Figura 3.4 – Construtores do arquivo HDF [NCSA, 2005]

O arquivo HDF é auto-descritivo, possuindo metadados que apresentam uma informação compreensiva sobre os dados e sua localização no arquivo.

HDF pode ser visto como diversos níveis de interação, conforme mostra a figura 3.5. No nível mais baixo, HDF é um arquivo em formato físico para armazenamento de dados científicos. No nível mais elevado, HDF é uma coleção de utilitários e aplicações para manipulação, visualização e análise de dados em arquivos HDF. Entre estes níveis, HDF é uma biblioteca de software que provê APIs para níveis mais altos e uma *interface* para dados em níveis mais baixos. [NCSA, 2005]. A maioria das rotinas de *interface* está disponível na linguagem FORTRAN-77 e C.

Algumas características do arquivo HDF são listadas a seguir [NCSA, 2005]:

- Provê o mecanismo para que os programas obtenham informações sobre o dado (metadados) em um arquivo do próprio HDF;
- Permite ao usuário armazenar dados de diferentes fontes em apenas um arquivo, assim como armazenar os dados e informações relacionadas em arquivos separados, mesmo quando os arquivos são processados pelo mesmo programa;

- Padroniza os formatos e descrições de muitos tipos de conjuntos de dados comumente usados, como por exemplo, imagens raster e conjuntos de dados multidimensionais;
- Encoraja o uso de um formato de dados comum por todas as máquinas e programas que produzem arquivos contendo dados específicos;
- Pode ser adaptado para acomodar virtualmente qualquer tipo de dado que possa ser armazenado digitalmente, independente da sua origem ou tamanho.



Figura 3.5 - Níveis de interação do HDF [Fonte: NCSA, 2005]

Para a visualização científica e análise de dados existem softwares desenvolvidos pela própria NCSA, softwares para desenvolvimento de ferramentas e alguns pacotes comerciais. Existem também utilitários para operar diretamente nos arquivos HDFs através de linhas de comandos. Através de um *prompt* de comando, o usuário executa operações nos arquivos, ao invés de escrever um programa.

A biblioteca HDF e utilitários, usada para a implementação de softwares para a leitura e criação de arquivos HDF, suportam diferentes máquinas e sistemas operacionais, tais como Cray (UNICOS), PC (Windows, Solaris86, Linux, FreeBSD), Macintosh (MacOS), Sun Sun4 (SunOS, Solaris), H/P HP9000 (HPUX), DEC VAX (OpenVMS) e DEC Alpha (Digital Unix, OpenVMS).

3.3.2 NetCDF

Network Common Data Form (NetCDF) foi desenvolvido pela Unidata, um grupo financiado pelo *US National Science Foundation* localizado nos Estados Unidos, responsável pela pesquisa e educação da ciência atmosférica.

NetCDF é uma abstração de dados para armazenamento e recuperação de dados multidimensionais. NetCDF é distribuído como uma biblioteca de software que provê uma

implementação concreta da abstração. A implementação provê um formato independente de máquina para representar dados científicos. Juntos, a abstração, biblioteca e formato de dados suportam a criação, acesso e distribuição de dados científicos [Rew, 1990]. Diversos tipos de dados podem ser representados com NetCDF: dados escalares e vetores de bytes, caracteres, inteiros e números de ponto-flutuante.

Muitas organizações já adotaram o NetCDF como padrão para a representação de dados científicos, sendo hoje utilizado em diversos sistemas para dados oceanográficos, hidrográficos, meteorológicos, geológicos e atmosféricos.

3.3.2.1 Estrutura do Arquivo

NetCDF é auto-descritivo, já que inclui informações sobre os dados que contém. Cada arquivo NetCDF possui três componentes: dimensões, variáveis e atributos (figura 3.6). Na dimensão é especificada a forma de uma ou mais variáveis do arquivo, como, por exemplo, tempo, latitude, longitude ou nível atmosférico. A variável é um vetor de valores do mesmo tipo, usada para armazenamento de dados. A variável possui nome, tipo de dados e forma (lista de dimensões). Os atributos podem estar associados às variáveis, para representar alguma informação sobre a variável. O atributo pode ser um valor único ou um vetor de valores.

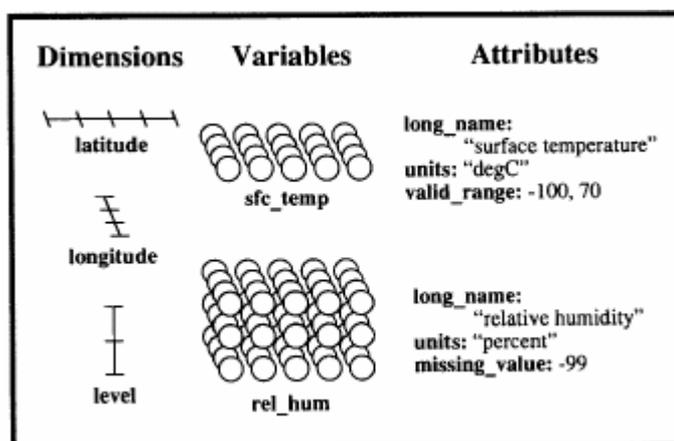


Figura 3.6 - Exemplo da abstração NetCDF. Este conjunto de dados NetCDF contém três dimensões e 2 variáveis, cada uma com 3 atributos. [Fonte: Rew, 1990]

3.3.2.2 Acesso Direto e Independência de Máquina

A interface de acesso usa nome de variáveis e vetores para acesso aos dados, ao invés de leituras e escritas.

NetCDF é independente de máquina, ou seja, dados escritos em uma máquina podem ser lidos por outra, sem conversão explícita.

Para garantir a independência de máquina, é utilizado o XDR (*eXternal Data Representation*), que consiste em um padrão para a codificação. A biblioteca XDR foi implementada em diversas máquinas, como Workstations, PCs, mainframes e supercomputadores.

O XDR é usado para a representação de conjuntos de bytes, inteiros curtos de 16 bits, inteiros longos de 32 bits e números de ponto-flutuante de 32 e 64 bits do padrão IEEE.

A biblioteca XDR foi compilada e testada em diversos ambientes computacionais, como Suns, VAXs, DECstations, Apple Macintoshes, IBM PCs e Crays.

4 TÉCNICAS DE VISUALIZAÇÃO DE DADOS OCEANOGRÁFICOS

A visualização tridimensional, ou visualização volumétrica, é o processo de gerar informação visual e com significado em um plano de imagem bidimensional a partir de um conjunto de dados tridimensional. Existem duas abordagens principais para a visualização volumétrica: Visualização através de Superfície e Visualização Direta de Volume.

A Visualização através de Superfície é uma técnica onde polígonos que representam a superfície do objeto são extraídos de um volume de dados. A partir disto, a visualização é realizada com a utilização de técnicas tradicionais de Computação Gráfica.

Na Visualização Direta de Volume, o objeto de interesse é representado por blocos cúbicos ou tetraedros chamados de *voxels*. Cada *voxel* tem associado a ele um ou mais valores, quantificando alguma medição ou uma propriedade calculada sobre o objeto original, como transparência, luminosidade, densidade ou velocidade do fluido. A cor final do *pixel* da imagem é obtida através da soma das contribuições de cada *voxel*. Como as faces podem ser consideradas semi-transparentes, os raios de luz passam de *voxel* a *voxel*, ao invés de parar na superfície.

Para a modelagem oceânica, técnicas de visualização direta de volume melhoram a análise da estrutura interna do oceano. Estas técnicas permitem a visualização de informação sobre a temperatura do oceano, salinidade, velocidade e volume, provendo mais informação que um conjunto de cortes e mapas de plano [Coelho et al., 2004].

Apesar de existirem técnicas que permitem tornar a superfície transparente ou translúcida, existem ainda muitos fenômenos que não podem ser simulados usando apenas técnicas de renderização de superfície (e.g., *scattering* ou emissão de luz) [Schroeder, 2003].

Feita a visualização, técnicas de interação podem ser aplicadas. Por exemplo, a extração de planos de corte horizontal e vertical servem para visualizar apenas uma determinada fatia do volume. Neste caso, existe uma perda da dimensão de profundidade quando o dado tridimensional é projetado em uma superfície bidimensional, mas torna-se

possível analisar estruturas ou informações que muitas vezes passam despercebidas na visualização volumétrica.

As variáveis envolvidas na visualização podem ser escalares, vetoriais ou tensoriais. No espaço tridimensional cartesiano, os escalares são representados por um número, os vetores são representados por dois números e os tensores por três números. Polito [Polito 2004] salienta que raramente precisamos descrever variáveis oceanográficas como tensores. A disposição geométrica e a escala espaço-temporal do fluxo geofísico também simplificam de forma muito oportuna o tratamento de variáveis tensoriais (stress, gradiente de velocidade, etc.). Direções e superfícies privilegiadas (e.g. interface ar-mar) e também simetrias devido ao tratamento macroscópico (e.g. homogeneidade), reduzem muito o número de componentes interessantes dos tensores. Assim sendo, para a vasta maioria dos casos se lida com vetores e escalares. Nas próximas seções são apresentadas algumas técnicas de visualização específicas para dados oceanográficos.

4.1 VISUALIZAÇÃO DE DADOS VETORIAIS

A visualização de dados vetoriais é geralmente mais complicada que a visualização de dados escalares devido a uma maior quantidade de informação existente neste tipo de dado [Bajaj, 1999]. Durante os últimos vinte anos um número de poderosas técnicas foram desenvolvidas para a visualização de dados vetoriais. Por exemplo, *Streamlines* e *Stream Tubes* mostram o caminho que uma partícula segue em um fluido. Métodos baseados em ícones, como *Hedgehogs*, mostram a direção e a magnitude de um campo vetorial em pontos selecionados. Métodos como *spot noise* e *line integral convolution (LIC)* usam textura para mostrar a direção do fluido sobre superfícies [Crossno et. al., 2001].

Nesta seção serão abordadas as técnicas de visualização de dados vetoriais utilizadas em oceanografia.

4.1.1 Hedgehog Display

A técnica *Straight Arrow Hedgehog Display* mostra a informação vetorial como um ícone de seta linear. A magnitude é o comprimento da seta e a direção é a própria direção para qual a seta aponta [Gaither & Moorhead, 2002]. Com este método, setas que possuem grandes comprimentos podem deixar a imagem confusa. Um exemplo desta técnica é apresentado na figura 4.1, com a visualização de fluídos.

A técnica *Curved Arrow Hedgehog Display* mostra a informação vetorial com o uso de ícones de setas curvilíneas, nas quais a magnitude é dada pelo tamanho da seta curva e a direção é dada pela direção na qual a seta está aponta [Gaither & Moorhead, 2002]. Esta técnica abrevia os problemas associados com a falta de clareza da imagem. Um exemplo desta técnica é apresentado na figura 4.2, com a visualização de fluídos.

Estas técnicas podem ser utilizadas para a visualização de fluídos em 2D e 3D, mas a adição de dimensão ao vetor, muitas vezes, torna a imagem mais difícil de ser interpretada.

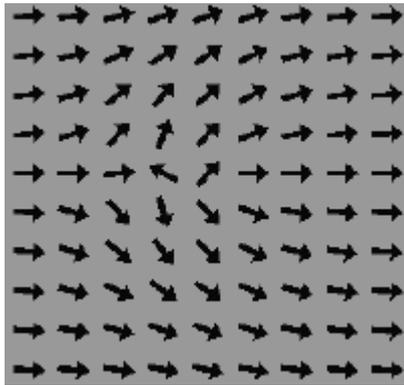


Figura 4.1 – Aplicação da técnica *Straight Arrow Hedgehog Display* na visualização de fluídos [Fonte: Telea & Wijk, 1999]

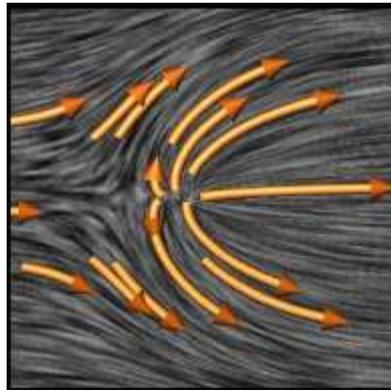


Figura 4.2 – Aplicação da técnica *Curved Arrow Hedgehog Display* na visualização de fluídos [Fonte: Telea & Wijk, 1999]

4.1.2 Flow Lines

Nesta seção, são abordadas as técnicas de visualização do movimento das correntes. Na visualização de fluidos, tornou-se comum o uso de linhas de corrente (*Streamlines*), de trajetória (*Pathline* ou *Particle Path*) e de emissão (*Streakline*). Linhas de trajetória e de emissão podem ser simuladas pela injeção de partículas no fluxo.

4.1.2.1 Linhas de Corrente

As linhas de corrente são curvas instantâneas que são tangentes à direção da velocidade de um fluxo em um fluido, considerando que o fluxo contém um vetor velocidade a cada momento e a cada ponto do espaço. O fluido é o líquido e o fluxo é o deslocamento. A figura 4.3 apresenta um exemplo de linhas de corrente.

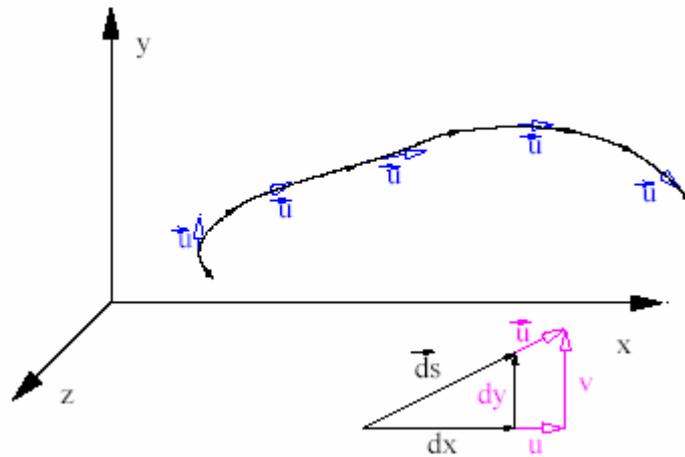


Figura 4.3 – Exemplo de linhas de corrente [Fonte: Polito, 2004]

4.1.2.2 Linha de partículas ou Linha de Emissão

Linha de partículas é aquela que conecta uma seqüência de partículas individuais existentes no fluxo. É marcada no espaço a posição de várias partículas em um dado tempo.

A figura 4.4 mostra um exemplo desta técnica. Esta imagem apresenta a posição das partículas em quatro momentos diferentes, em uma mesma região, conforme os tempos indicados. Para marcar a posição das partículas no fluxo foram utilizadas bóias, que são instrumentos de coleta de dados. A cada hora é liberada uma bóia numerada. A posição de cada bóia é marcada a cada hora (0:30, 1:30, 2:30, 3:30). A linha contínua que une as quatro bóias no último mapa é a linha de partículas.

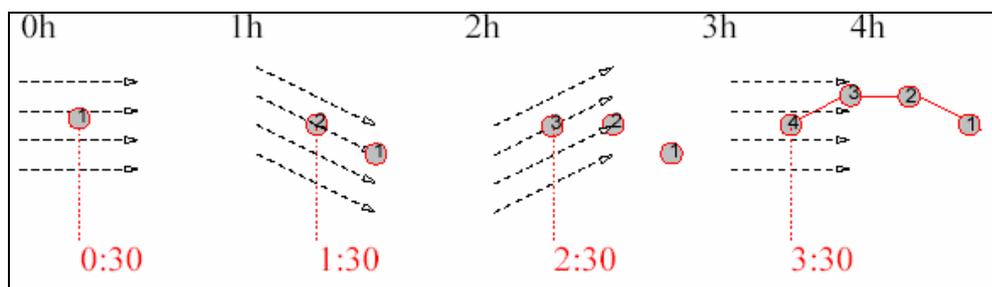


Figura 4.4 – Exemplo de linha de partículas [Fonte: Polito, 2004]

4.1.2.3 Trajetória

É o caminho percorrido por uma partícula individual durante um determinado tempo. É marcada no espaço a posição ocupada pela partícula em vários tempos.

4.1.3 Colorwheel

Este método, ao invés de usar setas, aplica cores a uma camada de dados, representando a magnitude e a direção. É usada uma cor individual para cada direção 2D, implementada por uma tabela circular, onde o ângulo da direção é computado, acessando uma posição dentro da tabela e uma cor representativa é encontrada. Este método é muito útil em estruturas 2D, mas não em 3D. O exemplo da figura 4.5 mostra uma tabela circular de cores. A figura 4.6 apresenta a aplicação do método *Colorwheel* na visualização do mar do Japão.

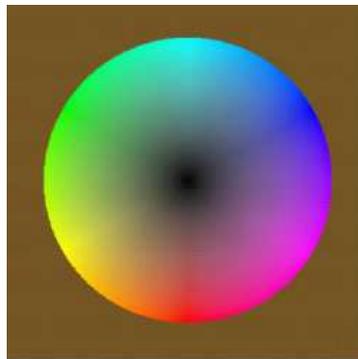


Figura 4.5 - Tabela circular de cores [Fonte: Moorhead et. al., 1999]

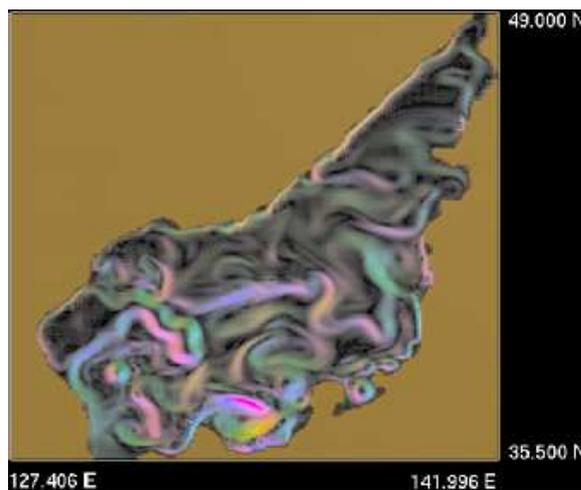


Figura 4.6 - Mar do Japão, com o uso da técnica *colorwheel* [Fonte: Moorhead et. al., 1999]

4.1.4 Line Integral Convolution

O método *Line Integral Convolution* é uma técnica de textura que pode ser utilizada para visualizar dados de campos vetoriais, como densos fluídos, com visualização de curvas

integrais. Tipicamente, uma textura de alta frequência (figura 4.7 (a)) é filtrada de acordo com um campo vetorial (figura 4.7 (b)) produzindo uma visão global do fluxo inteiro (figura 4.7 (c)).

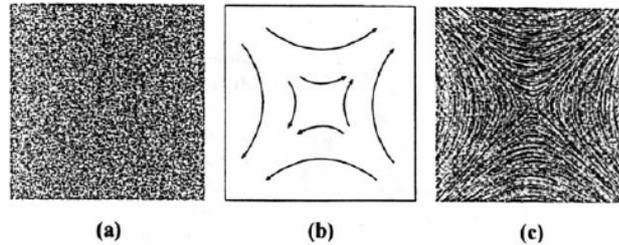


Figura 4.7 – (a) Ruído Branco, (b) campo vetorial 2D, (c) textura de ruído branco filtrada com o campo vetorial [Fonte: Mendonça 2001]

Com a utilização de um campo vetorial e uma textura, geralmente uma imagem de ruído branco como entrada, o algoritmo produz uma imagem bidimensional de uma fatia selecionada desse conjunto de dados, interpolando a textura ruído segundo o campo vetorial a ser visualizado, conforme mostra a figura 4.8. Existem diversas técnicas de interpolação que são utilizadas com esta técnica. Este método é aplicado apenas em superfícies ou campos vetoriais bidimensionais.

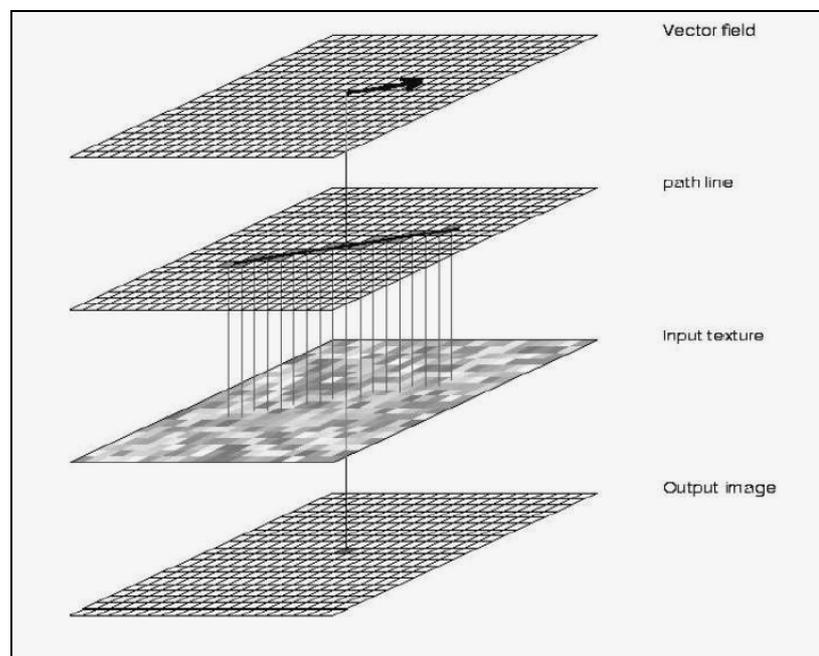


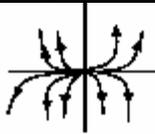
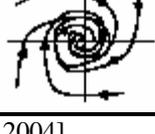
Figura 4.8 – Processo de geração da imagem com a técnica *Line Integral Convolution* [Fonte: Universidade de Stuttgart, 2004]

4.1.5 Extraindo Características Topológicas

A extração é utilizada para identificar pontos críticos nos dados vetoriais. Estes pontos críticos correspondem à área do fluxo na qual a velocidade ou a magnitude da velocidade torna-se zero [Gaither & Moorhead, 2002].

O processo de detecção de pontos críticos dá-se da seguinte maneira: Células candidatas, que contêm pontos críticos, podem ser encontradas examinando o sinal dos componentes u e v do vetor em cada nodo da célula. Se existe uma mudança de sinal no componente, um ponto crítico pode existir naquela célula. Por exemplo, uma célula candidata poderia ter a seguinte combinação de sinais: $(+,+)$ $(+,+)$ $(+,-)$, $(-, -)$ $(+,+)$ $(-,+)$, mas não poderia ter $(+,+)$ $(+,+)$ $(+,+)$ $(+,+)$. Todas as células candidatas devem ser testadas para determinar se um ponto crítico realmente existe dentro da célula. Equações bilineares simultâneas podem ser resolvidas para determinar a localização x e y do ponto crítico. Se o x e y estiverem do lado de fora da célula, não é um ponto crítico. Quando os pontos críticos forem determinados, eles podem, então, ser classificados e apresentados. Uma derivada parcial de primeira ordem, com relação à posição, determina o comportamento do campo vetorial. Se ambas raízes são reais e positivas, o ponto crítico é uma fonte. Se ambas são reais e possuem sinais opostos, o ponto crítico é um ponto de sela. Se as raízes são conjugados complexos ($K - Li$, $K + Li$) e K é positivo, o ponto crítico é uma fonte espiral. Se K é negativo, o ponto crítico é um sorvedor espiral. A tabela 4.1 apresenta a classificação dos pontos críticos.

Tabela 4.1 – Pontos críticos

Nome	Aparência
Fonte	
Sela	
Sorvedor Espiral	
Espiral	

Fonte: [Rodrigues, 2004]

Este método pode ser utilizado em visualização 2D e 3D, sendo mais útil para a visualização oceânica em pequena escala, devido ao grande número de pontos críticos que são detectados em aplicações de médias e largas escalas.

4.1.6 Técnicas de Visualização Volumétrica de Vetores

A visualização volumétrica de vetores se baseia no fato de o campo vetorial estar contido em um volume e ser, então, tratado como tridimensional. Diversas abordagens são utilizadas com a visualização volumétrica de vetores, tais como a extração de características e a visualização direta de volume vetorial [Crawfis & Max, 1992].

O oceano é bastante complexo e é preciso explorar características de interesse nos dados. O método de Planos de Corte é uma técnica de extração de características, na qual planos traçados arbitrariamente interseccionam o volume em uma dada localidade. A informação do volume, que está no plano de corte, é projetada em um plano para visualização. Esta técnica reduz a informação tridimensional para 2D, perdendo com isso significado.

Outra técnica é a detecção de características (*feature detection*). É usada, por exemplo, na detecção de remoinhos. Um típico remoinho em um fluido possui um padrão de circulação fechado ou semi-fechado [Gaither & Moorhead, 2002].

4.2 COMPARAÇÃO DE MÉTODOS DE VISUALIZAÇÃO VETORIAL

Não existe certo ou errado quando se tenta escolher métodos para visualizar a informação vetorial. O usuário deve decidir qual é a técnica que melhor retrata o significado dos dados. Vantagens e desvantagens dos métodos para visualização vetorial são apresentadas na tabela 4.2.

Tabela 4.2 - Vantagens e desvantagem de métodos de visualização vetorial

Método	Vantagem	Desvantagem
<i>Straight Arrow Hedgehog</i>	Interativa	Visualização Confusa
<i>Curved Arrow Hedgehog</i>	Interativa	Nenhum detalhe local
<i>Flow Lines</i>	Permite obter informação detalhada	Precisão Numérica/ Velocidade
<i>Color Wheel</i>	Interativa Extraí características do fluido	Limitado em 2D
<i>Line Integral Convolution</i>	Extraí bem características de pequenas escalas	Limitado em 3D ou superfícies
Topologia	Bom para visualização em pequenas escalas	Visualização confusa em grandes escalas
Planos de Corte	Reduz a desordem na visualização	Perda de informação
Detecção de Características	Boa detecção de remoinho	Computação em modo <i>batch</i>

Fonte: [Gaither & Moorhead, 2002].

4.3 VISUALIZAÇÃO DE DADOS ESCALARES

Existem diversos algoritmos aplicados à visualização de dados escalares, por ser um dado bastante simples. O mapeamento de cores é uma técnica muito comum e consiste em associar uma tabela de cores usando os valores escalares como índice da tabela. A tabela possui um valor mínimo e máximo e cada valor possui uma cor associada. Um exemplo desta técnica é apresentado na figura 4.9.

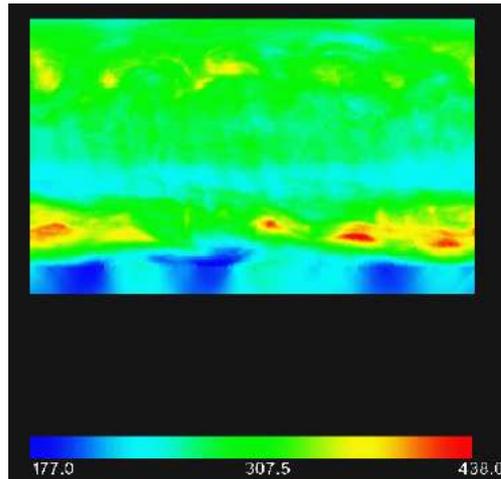


Figura 4.9 - Técnica de mapeamento de cores [Rheingans, 2005]

A técnica de isosuperfície consiste em identificar um conjunto de valores constantes no volume de dados para, a partir deles, traçar uma superfície volumétrica, onde os parâmetros desta técnica são valores de interesse. Em geral, técnicas de interpolação são utilizadas para auxiliar na formação de isosuperfícies.

5 TRABALHOS RELACIONADOS

A visualização de resultados de modelos oceânicos é uma importante ferramenta que ajuda a avaliar a conformidade do modelo com a realidade, identificar e diagnosticar problemas e aprender mais sobre o oceano e seu comportamento. Neste capítulo serão abordadas as características mais importantes de algumas ferramentas oceanográficas existentes.

5.1 GEOZUI3D – GEOGRAPHIC ZOOMING USER INTERFACE

GeoZui3D é um sistema de visualização 3D interativo, desenvolvido para interpretar múltiplas fontes de dados 3D. Esta ferramenta foi desenvolvida na Universidade de Hampshire e suporta visualização simultânea de conjuntos de dados heterogêneos de uma variedade de tipos de sensores, que coletam dados oceanográficos. O objetivo é integrar dados de vários sensores em tempo real e unir com dados já obtidos anteriormente. Além disso, também possui como intenção criar interfaces para Sistemas de Informações Geográficas 3D, para a visualização e interpretação de dados multidisciplinares.

As principais funcionalidades são [Ware et al., 2001]:

- Visualização de mapas de terrenos;
- Visualização de imagens no formato jpeg (2D) como imagens georeferenciadas em uma cena 3D;
- Utilização de tubos para representar caminhos 3D;
- Geração de planos de corte;
- Uso de polígonos, linhas, polilinhas e pontos para constuir estruturas 3D;
- Disponibilidade de sons dinâmicos que corresponde à visualização em tempo real de sons;
- Métodos para mostrar as posições e informações de escala utilizando coordenadas e grades.

A figura 5.1 mostra um exemplo de visualização no GeoZui3D, de um mapa digital de terreno.

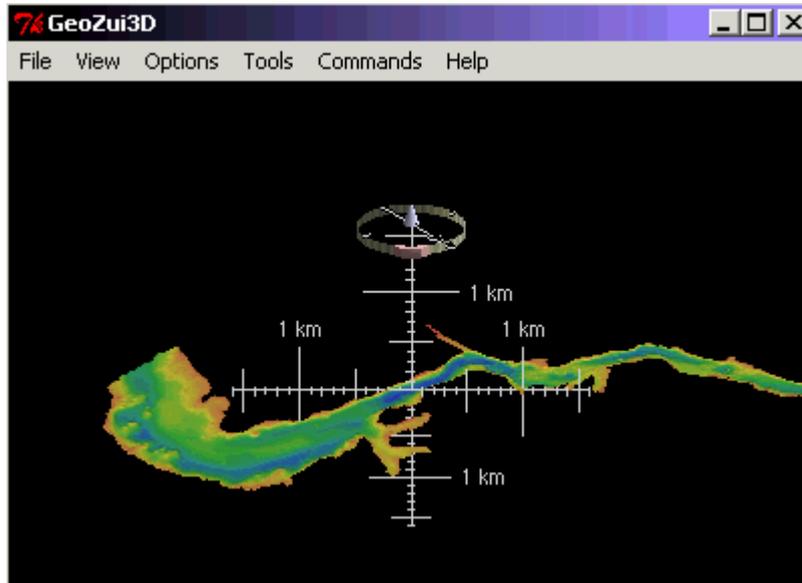


Figura 5.1 – Mapa digital de terreno [Fonte: GeoZui3D, 2004]

Possuindo técnicas de interação em 3D, o GeoZui3D possibilita mudar o ponto de vista para melhor entender as relações entre as características 3D, utilizar o *zoom* e também visualizar os dados com o uso de estereoscopia. A figura 5.2 mostra a navegação no GeoZui3D, onde é possível realizar *zoom in* e *zoom out*, mover pontos de interesse para o centro do *workspace* e rotacionar o centro do *workspace*. O *zoom* é realizado com o clique do mouse. Para um *zoom* horizontal, é utilizada a linha horizontal (*horizontal axis*). O controle *tilt* permite inclinar a cena e o *heading Control* permite rotacionar a cena.

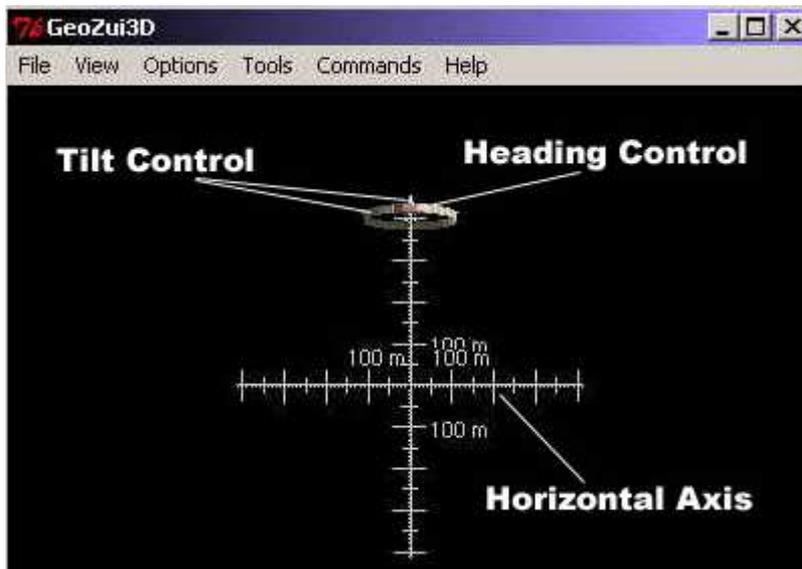


Figura 5.2 – Navegação no GeoZui3D [Fonte: GeoZui3D, 2004]

O download do GeoZui3D pode ser realizado gratuitamente no seu web site [GeoZui3D, 2004].

5.2 ISTV: INTERACTIVE STRUCTURED TIME-VARYING VISUALIZER

ISTV é um sistema de visualização desenvolvido pelo NFS Engineering Research Center e criado para visualização de grandes modelos oceanográficos [Moorhead et al., 1999]. Esta ferramenta foi implementada na linguagem C, com a utilização da biblioteca OpenGL para as funcionalidades gráficas e X/Motif para a elaboração da *interface* gráfica com o usuário. O sistema está disponível para *download* gratuitamente no próprio site [ISTV 2005], podendo ser utilizado em qualquer plataforma *Unix*.

O ISTV suporta a visualização simultânea de diversos conjuntos de dados com variação temporal e resoluções espaciais em uma única visualização de volume. O ISTV trata cada conjunto de dados como uma grade lógica de quatro dimensões: 3 dimensões espaciais e mais o tempo associado, podendo conter um número arbitrário de valores escalares e vetoriais definidos em cada ponto da grade.

Como o ISTV lida com grandes conjuntos de dados, possui diversos algoritmos que garantem a interatividade do sistema. Além disso, é possível selecionar um subconjunto do espaço 3D e animar este subconjunto para um período específico de tempo.

Na visualização de campos escalares, dois objetos geométricos básicos podem ser apresentados: camadas e isosuperfícies. Uma camada é uma superfície definida ao longo de um índice constante da grade. Uma isosuperfície é definida por uma constante contendo um campo escalar definido por uma variável. As representações gráficas disponíveis para cada objeto geométrico são: *wireframe*, superfície sólida, linhas de contorno e mini-cubos. Além disso, três esquemas de colorações estão disponíveis para cada objeto: cor sólida, mapeamento de cor e *color wheel*. Em cor sólida, apenas uma cor é dada ao objeto. Com mapeamento de cor, o objeto é colorido baseado em variáveis independentes. *Color Wheel* usa o espaço de cores HSV [Watt, 2000] para colorir os objetos, com base em duas variáveis independentes.

Para a visualização de campos vetoriais, são utilizadas as representações gráficas: *tufts*, *streamlines*, *streaklines* e *pathlines* (ver capítulo 4 para maiores detalhes sobre estas técnicas).

O ISTV suporta a inserção de textos no espaço 3D do modelo ou em um local específico da tela.

A figura 5.3 mostra um exemplo de imagem gerada pelo ISTV, utilizando o esquema de cores *color map*. Nesta imagem, é mostrada a temperatura do Golfo Pérsico.

A figura 5.4 mostra isosuperfícies sobre Oceano Pacífico, que utiliza o esquema de cores *color map*.

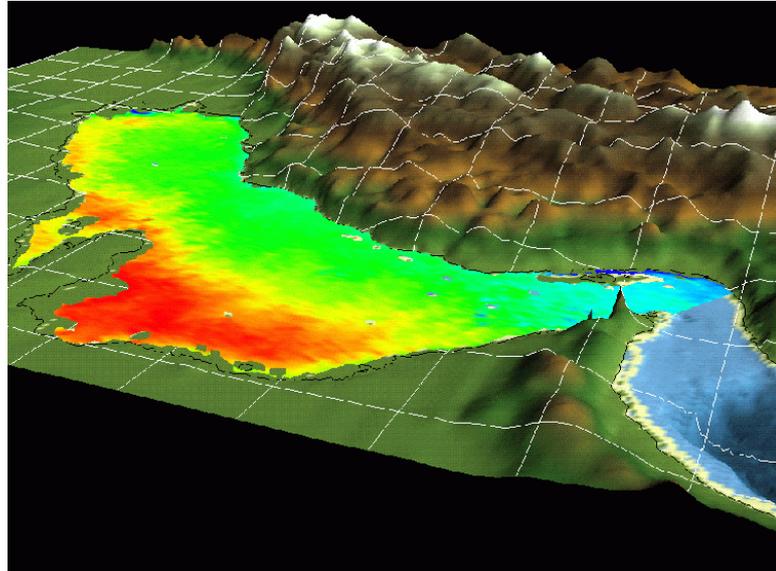


Figura 5.3 – Temperatura do Golfo Pérsico com utilização de *color map* para valores de terreno [Fonte: ISTV, 2004]

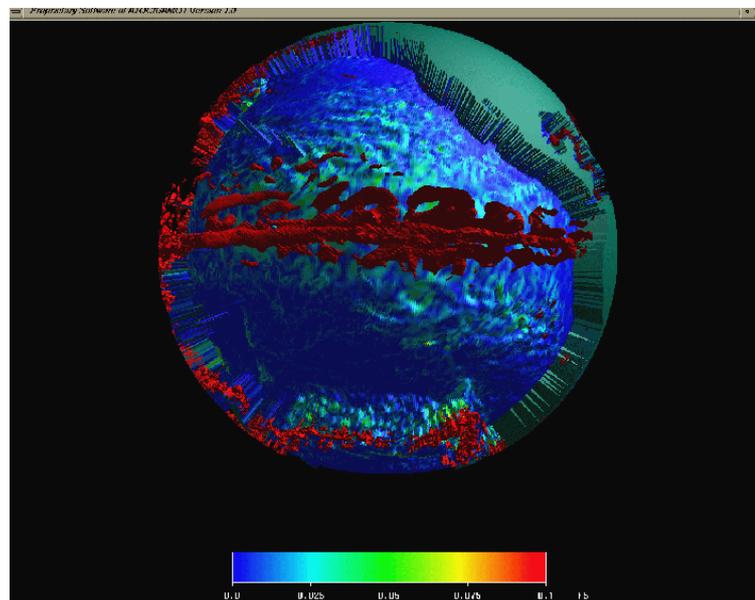


Figura 5.4 – Iso-superfícies sobre o Oceano Pacífico utilizando o esquema de cores *color map* [Fonte: ISTV, 2004]

O ISTV é frequentemente utilizado com o modelo oceânico *NLOM* (*Navy Layered Ocean Model*) [Wallcraft, 1991]. Suporta grades retilíneas e curvilíneas com a utilização do formato de dados HDF.

5.3 POPTX

POPTX [McPherson et al., 1998] é um sistema de visualização de dados oceânicos simulados, desenvolvido pelo *Los Alamos National Laboratory*. Este sistema é utilizado para visualizar os dados simulados do modelo oceânico *Parallel Ocean Program (POP)*, desenvolvido também pelos cientistas de *Los Alamos*.

A ferramenta foi implementada na linguagem C++, utilizando a biblioteca *OpenGL*. Este sistema foi desenvolvido para executar nos computadores *Connection Machine* e *Silicon Graphics Origin 2000*.

Os dados de cada variável computada durante a simulação (e.g. salinidade, temperatura) são armazenados em arquivos individuais. A partir de cada arquivo, são geradas várias imagens, que são colocadas em movimento, para a produção de um vídeo, visualizando, assim, o progresso da simulação. Esta ferramenta é dinâmica, interativa e flexível, ou seja, é possível modificar estas imagens, sem precisar criar um novo vídeo e interagir com as mesmas, aplicando *zoom* ou utilizando um novo mapeamento de cores, por exemplo. Para obter esta flexibilidade, são combinados poderosos *hardwares* gráficos de alto desempenho, como, por exemplo, *hardware* de mapeamento de textura e também uma grande quantidade de memória RAM, cerca de 16GB.

Com relação às técnicas de visualização aplicadas, o POPTX além de trabalhar com mapeamento de cores, também utiliza outras técnicas de mapeamento, como a técnica *hillshading* [Horn, 1981]. A figura 5.5 apresenta esta técnica, utilizada para representar a altura da superfície oceânica, no leste do Atlântico, com a elevação do terreno sombreada.

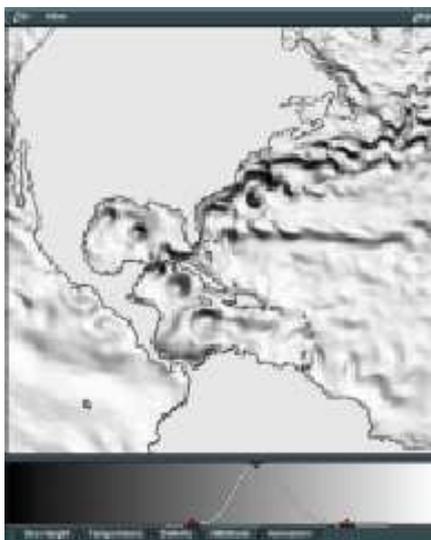


Figura 5.5 – Altura da superfície oceânica com a aplicação da técnica *hillshading* [Fonte: POPTX, 2004]

A figura 5.6 mostra um exemplo de imagem de um vídeo gerado pelo POPTEX.

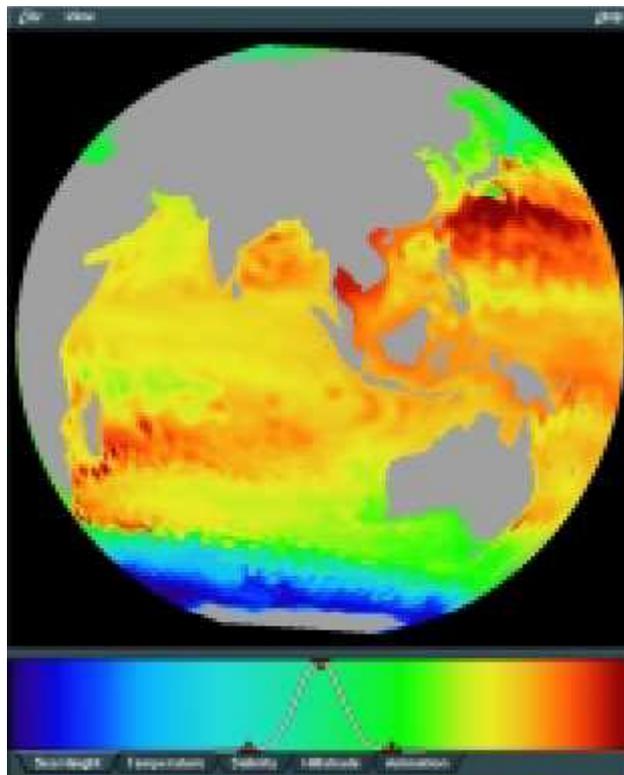


Figura 5.6 – Altura da superfície oceânica: valores baixos em azul e valores altos em vermelho [McPherson et. al, 1998]

5.4 VIS5D

Vis5D é um sistema para visualização interativa de grandes conjuntos de dados em grades 5D, como as produzidas por modelos numéricos climáticos. O Vis5D foi desenvolvido inicialmente pelo MIT e atualmente é mantido pelo NOAA Forecast Systems Lab. A ferramenta foi implementada na linguagem OpenGL e está disponível gratuitamente para download [Vis5D, 2004]. Suporta diversas plataformas operacionais, tais como IRIX, AIX, Sun, HP, OS/2, Linux e Windows.

O Vis5D provê uma visualização interativa 3D de arquivos de formato próprio. É utilizado para visualizar os dados de saída gerados pelo processamento de modelos da superfície atmosférica e de oceanos, mas também pode ser utilizado para visualizar qualquer dado 3D disposto em uma grade regular. Os arquivos Vis5D possuem cinco dimensões. As dimensões são: linha, coluna, nível, tempo e um índice que se refere a um conjunto de campos físicos (temperatura, pressão, umidade, salinidade, velocidade das correntes etc.). Linha, coluna e nível podem corresponder à latitude, longitude e altitude, mas também podem ser abstratas ou estar em uma projeção de mapa não-cartesiano.

Entre outras funcionalidades, o Vis5D suporta a extração de isosuperfícies, linhas de contorno, fatias coloridas, renderização de volume de dados em grade 3D, também rotacionando e animando as imagens em tempo real.

A figura 5.7 mostra um exemplo de imagem de uma animação gerada pelo Vis5D, que mostra a variação da temperatura e a altura ao longo do tempo.

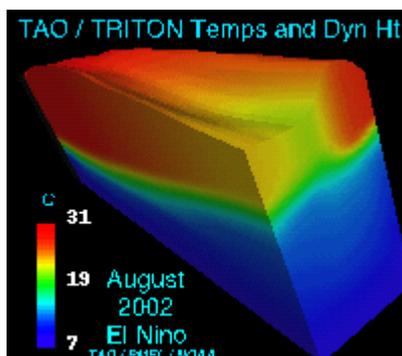


Figura 5.7 – Temperatura em um período de tempo

5.5 ENVIS/HUM

A partir de um conjunto de dados de entrada, gerado pela simulação de um modelo oceanográfico, o EnVis cria imagens RGB de alta resolução utilizando uma variedade de técnicas de sombreamento de superfície. Assim, permite que características em pequena escala tornem-se muito mais aparente para pesquisadores do que o convencional [Chupa et al., 2004].

Hum cria uma animação das imagens geradas empregando uma arquitetura *multithreaded*, que implementa *pipelines* compostos de carregamento de imagem, decodificação, *caching* e visualização.

Algumas das técnicas implementadas na ferramenta, além de técnicas de sombreamento, consistem no uso de tabelas de mapeamento para a resolução utilizada, colorindo campos escalares e obtendo diferenças visuais mesmo em regiões que contenham poucas variações. Além disso, permite a composição de imagens, como, por exemplo, um terreno sobre porções de um modelo de imagem. Também possui glifos para dados vetoriais.

Algumas técnicas de interação da ferramenta são o *zoom* e a rotação. Além disso, as imagens podem ser visualizadas com o uso de estereoscopia, criando uma sensação de profundidade da imagem.

A figura 5.8 é um exemplo de imagem criada pelo Envis, mostrando a batimetria, ou seja, a medida de profundidade da massa de água. Este exemplo utiliza os dados simulados do

modelo oceânico *Navy Layered* desenvolvido pelo *Naval Research Laboratory*, localizado nos Estados Unidos.

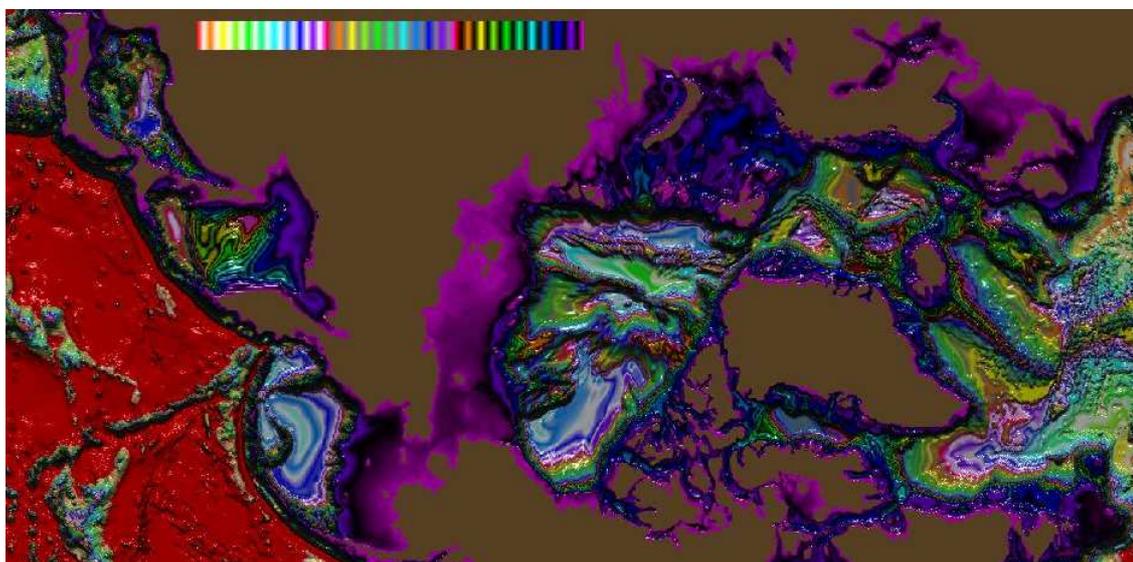


Figura 5.8 – Batimetria [Fonte: Chupa, 2001]

5.6 SISTEMA DE VISUALIZAÇÃO INTERATIVA

A visualização de dados oceanográficos gerados através de simulação é tradicionalmente um passo de pós-processamento. Com a visualização interativa, a solução computada pode ser vista em tempo real e o usuário pode mudar os parâmetros do modelo oceanográfico e imediatamente observar o impacto na solução.

Um sistema de visualização interativa [Nations et al., 1996] foi desenvolvido pelo *NSF Engineering Research Center*, da Universidade do Estado do Mississippi. Esta ferramenta foi desenvolvida para visualizar o modelo oceânico SJS, que é trabalhado pelo *Naval Research Laboratory*.

5.7 FERRET

Ferret é uma ferramenta interativa de visualização e análise de grandes e complexos conjuntos de dados. Este *software* é gratuito e foi desenvolvido como parte do projeto *Thermal Modeling and Analysis Project (TMAP)* [PMEL, 2005].

Este sistema analisa a saída gerada pelos modelos numéricos (através de simulação), possuindo cálculos matemáticos, onde variáveis podem ser definidas interativamente como expressões matemáticas envolvendo um conjunto de dados.

O *software* oferece uma excelente flexibilização na parte matemática e possui um ótimo gerenciamento de memória, necessário para grandes cálculos, além de processamento simétrico em 4 dimensões.

A figura 5.9 apresenta um exemplo de visualização gerada pelo Ferret, onde é apresentada a topografia e batimetria.

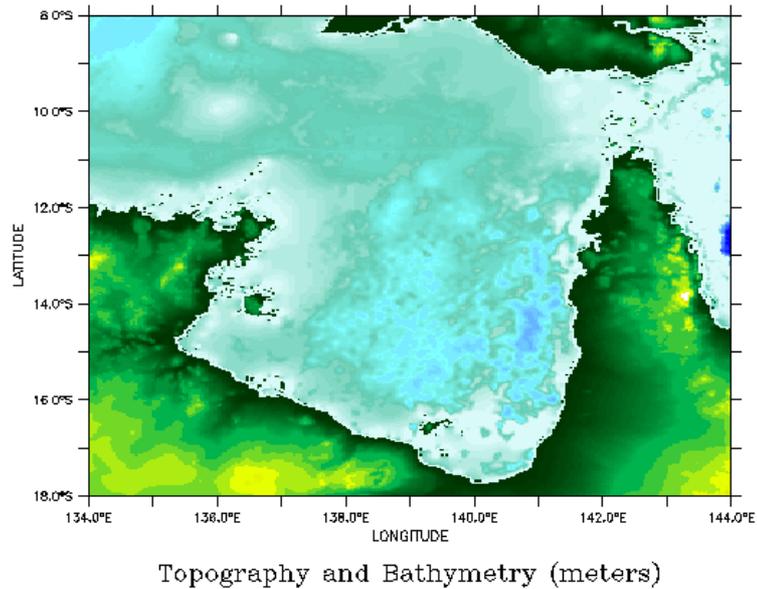


Figura 5.9 – Topografia e batimetria

A figura 5.10 apresenta visualização da temperatura da superfície oceânica, utilizando linhas de contorno.

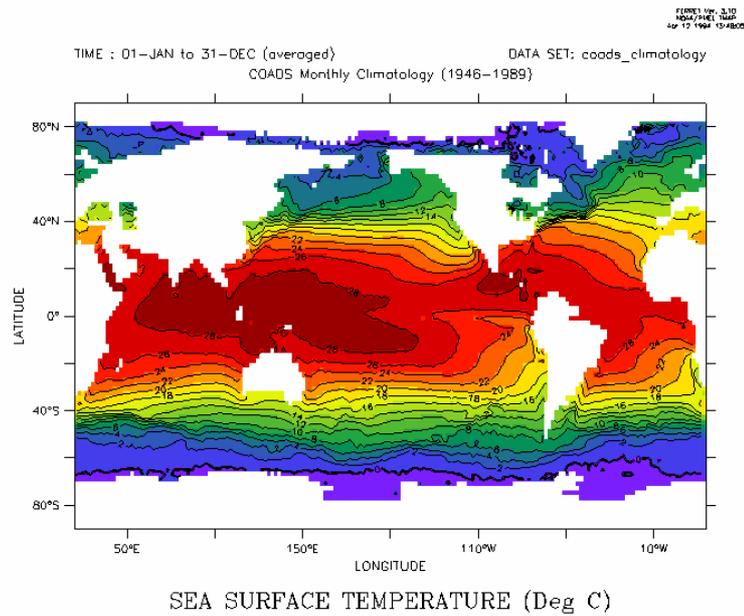


Figura 5.10 – Temperatura da superfície oceânica

5.8 ENSIGHT

O Ensign [Ensign 2005] é um exemplo de ferramenta de visualização científica comercial que também pode ser utilizada com dados oceanográficos. Esta ferramenta suporta diversas plataformas operacionais, tais como IRIX, AIX, Sun Solaris, HP-UX, Tru64, Linux e Windows.

Diversos formatos de arquivos podem ser utilizados para a leitura dos dados (por exemplo, NetCDF, HDF), além de um formato próprio da ferramenta.

O *software* oferece elementos geométricos 1D, 2D, 3D na forma linear ou quadrática. A geometria pode variar com o tempo. Possui diversas técnicas de visualização como contornos e isosuperfícies, caminho de partículas, ícones de setas para dados vetoriais, superfícies elevadas e planos de corte. Além disso, é possível desenvolver animações.

6 DESCRIÇÃO DO PROTÓTIPO

O protótipo desenvolvido neste trabalho permite a visualização tridimensional do oceano, possuindo técnicas para a visualização de dados escalares e vetoriais. Estas funcionalidades permitem obter mais informações sobre o oceano do que um conjunto de cortes e mapas de plano.

Técnicas de interação que possibilitam ao usuário obter um melhor entendimento de dados volumétricos também estão presentes na aplicação. É possível transladar, rotacionar e aplicar *zoom in* e *zoom out* no volume. Além disso, o usuário pode realizar cortes para permitir uma melhor análise das informações. Neste caso, o usuário pode selecionar uma região de interesse no volume. Esta seleção é feita através da manipulação de uma *spline* [Bartels, 1987] em um plano xy para depois, a partir desta curva, ser gerado um sub-volume, como se a curva fosse sendo arrastada no eixo z .

Além da visualização do volume do oceano, a ferramenta também permite gerar uma animação. Os dados simulados são obtidos em tomadas de tempo que representam diversos dias simulados. Para analisar esta variação temporal, foi elaborado um método que permite atualizar a posição e o estado de cada variável ao longo do tempo. A animação é realizada através da leitura de uma das variáveis oceanográficas em todas as tomadas de tempo e da subsequente aplicação de uma técnica de visualização adequada para a variável. Com isso, é possível observar as mudanças no comportamento desta variável oceânica.

Como um estudo de caso, e com o objetivo de testar as funcionalidades implementadas, estas técnicas foram aplicadas para dados oceânicos como temperatura, salinidade e velocidades das correntes marítimas. Estes dados, que foram gerados pelo modelo POM (seção 3.2.1) e fornecidos pelo CENPES, correspondem a uma simulação de dados de 10 dias, realizada sobre a Bacia de Campos e estão armazenados no formato NetCDF.

6.1 AMBIENTE DE DESENVOLVIMENTO

A ferramenta proposta neste trabalho foi desenvolvida para a plataforma *Windows* utilizando a linguagem de programação C++ e a biblioteca *Visualization ToolKit* (VTK) versão 5.0. Para a leitura dos dados oceanográficos, foi utilizada uma biblioteca desenvolvida pelo Unidata [Unidata, 2005], para o formato NetCDF.

O VTK [VTK 2005] possui código aberto, é totalmente orientado a objetos e tem sido bastante utilizado para o desenvolvimento de sistemas de Computação Gráfica, Visualização e Processamento de Imagens. O código-fonte da biblioteca VTK foi compilado para o *Visual Studio 7 .net*, que foi o ambiente de desenvolvimento utilizado. A *interface* da ferramenta foi implementada neste ambiente utilizando as classes MFC (*Microsoft Foundation Classes*).

A biblioteca VTK possui diversas classes pré-definidas que foram utilizadas para o desenvolvimento da ferramenta. Além disso, algumas de suas classes foram estendidas para que fosse possível implementar as funcionalidades propostas.

6.2 ARQUITETURA DO SISTEMA

O sistema foi desenvolvido utilizando o paradigma de programação orientado a objetos, que, aliado a documentação do código, permite facilmente a extensão da aplicação. Sua implementação utiliza o padrão de desenvolvimento de *software Document-View* [Buschmann et al. 1997]. O *Document-View* é uma variante do padrão MVC (*Model-View-Controller*). Um *Document* é um objeto de dados que é manipulado pelo usuário em uma seção de edição. Uma *View* é um objeto janela que permite que o usuário interaja com o documento ao qual ela está associada. Neste padrão, os eventos (mouse, teclado) são manipulados pela classe *View*. Uma das vantagens de usar este padrão é a possibilidade de alterar a interface sem a necessidade de modificar o núcleo do sistema. Com isso, novas funcionalidades podem ser adicionadas mais facilmente.

No núcleo funcional do sistema, é utilizado outro padrão de arquitetura de *software*, chamado Fases e Filtros [Buschmann et al. 1997]. Este padrão provê uma estrutura para sistemas que processam fluxo de dados. Cada passo do processamento é encapsulado em um componente de filtro e os dados são passados entre as fases através de filtros adjacentes.

Este padrão também é utilizando no modelo de visualização do VTK, que é baseado no paradigma de fluxo de dados, adotado por muitos sistemas comerciais, como, por exemplo, o *Data Explorer* (desenvolvido pela IBM) e o *IRIS Explorer* (desenvolvido pela Silicon Graphics). Os diagramas de fluxo de dados possuem dois tipos de objetos: processos e dados

[Schroeder et al. 1996]. Os processos são classificados em fontes de dados, filtragem e mapeamento. Os objetos de dados representam informação e permitem a realização de operações sobre os dados.

O protótipo do sistema foi modelado utilizando a linguagem UML [Larman 2004]. Para a modelagem foram utilizados diagramas de casos de uso e diagramas de classes. A figura 6.1 apresenta o diagrama de casos de uso contendo as funcionalidades de visualização e interação do protótipo.

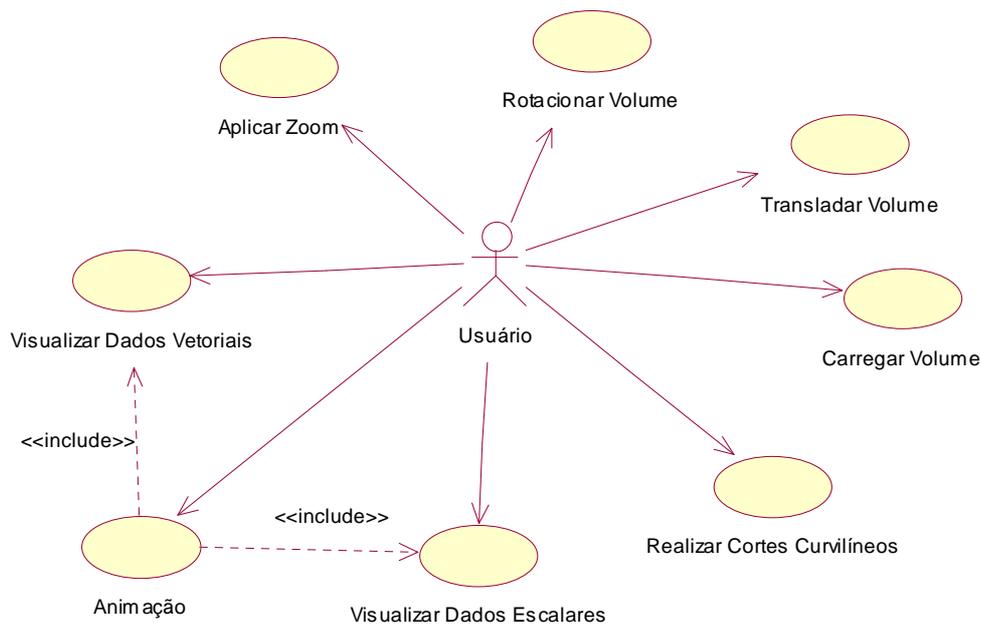


Figura 6.1 - Diagrama de casos de uso

Para as principais funcionalidades do diagrama de casos de uso, são apresentados diagramas de classes, de maneira a incluir na documentação todas as classes utilizadas no sistema. Durante o desenvolvimento, foi utilizado um padrão para nomenclatura das classes, na qual as iniciadas por *vtk* fazem parte do *Visualization ToolKit*. Com a utilização do padrão *Document-View*, as classes terminadas por *View* são visões e as classes terminadas por *Doc* são documentos. As classes terminadas por *Dlg* estão associadas às caixas de diálogo da *interface* do sistema, através da qual o usuário entra com os parâmetros desejados.

O diagrama de classes da figura 6.2 apresenta a modelagem para o carregamento do volume. A classe *COpenDoc* contém as operações para o carregamento do arquivo NetCDF, utilizando uma instância da classe *CReaderNetCDF* para a leitura dos dados oceânicos simulados no formato NetCDF. Após a leitura, os dados são armazenados em um objeto da classe *vtkStructuredGrid*, utilizado para a criação da grade 3D.

A classe *CPrincipalView* é instanciada para a visualização do volume na tela do computador. Para a geração da imagem a ser visualizada, é utilizada uma instância da classe *vtkDataSetMapper*, que faz o mapeamento de estruturas de dados em primitivas gráficas, e da classe *vtkActor*, para representar o objeto que faz parte da cena. Vinculada à classe *CPrincipalView*, estão outras 3 classes do VTK: *vtkRenderWindow*, *vtkRender*, *vtkRenderWindowInteractor*. A primeira é responsável pelo gerenciamento da janela do dispositivo de visualização. A classe *vtkRender* é responsável pelo gerenciamento das luzes, câmeras e atores de uma cena, com a finalidade de gerar uma imagem. A classe *vtkRenderWindowInteractor* é responsável pela captura de eventos para um objeto *vtkRender* em uma janela de exibição (*vtkRenderWindow*).

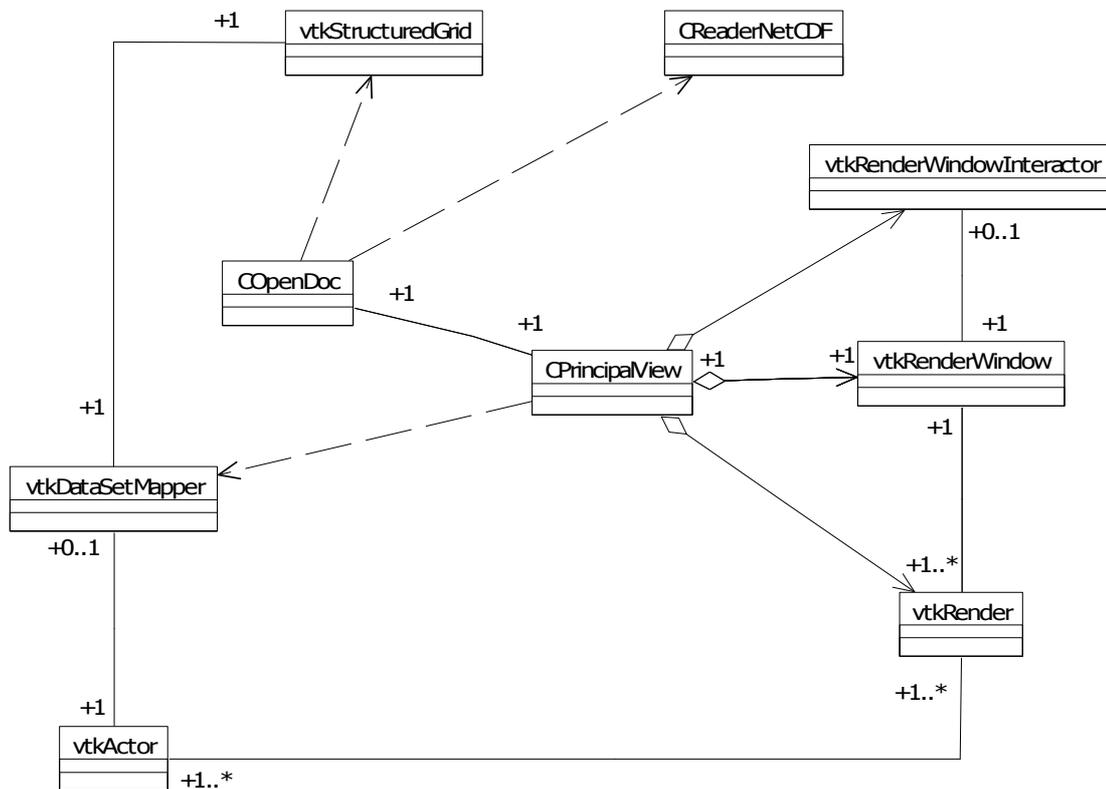


Figura 6.2 – Diagrama de classes para carregamento do volume

O diagrama da figura 6.3 apresenta a modelagem realizada para a técnica de mapeamento de cores. É necessário criar uma instância da classe *COpenDoc* para a aplicação desta técnica, carregando o arquivo de visualização. Depois, então, o usuário pode especificar os parâmetros da técnica através dos métodos disponíveis na classe *CMapcolorDlg*, tais como a escolha da variável oceanográfica a ser visualizada e seus valores mínimo e máximo. A

classe *CMapcolorView* é responsável pela aplicação da técnica e visualização, utilizando instâncias das classes *vtkLookupTable* para mapear valores escalares em cores RGBA (*red-green-blue-alpha transparency*), e *vtkScalarBarActor* para a apresentação da barra de cores na janela de visualização do protótipo.

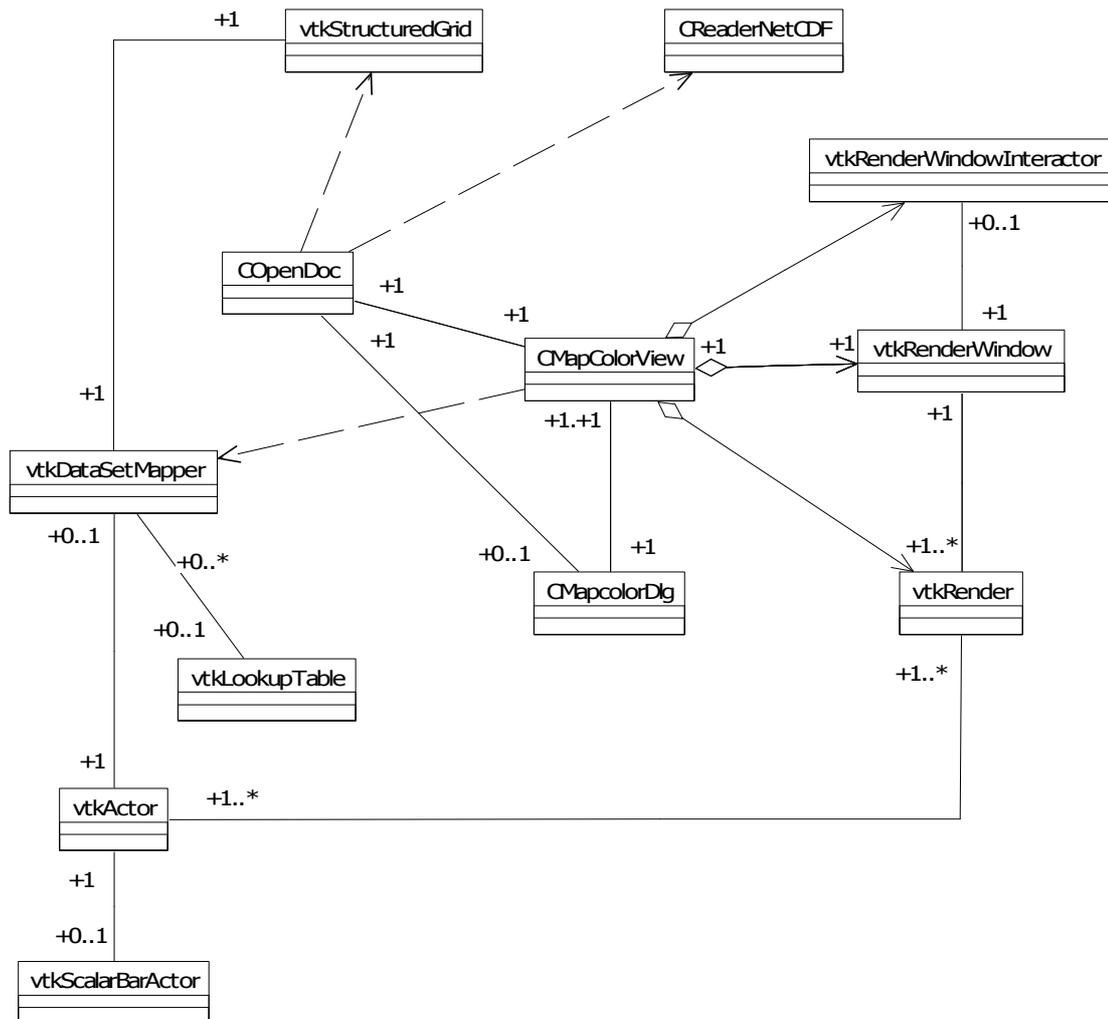


Figura 6.3 – Diagrama de classes para a técnica de mapeamento de cores

O diagrama de classes modelado para a visualização vetorial utilizando a técnica *Hedgehogs* é apresentado na figura 6.4. Para esta técnica, a classe *COpenDoc* também deve ter sido instanciada. Após, o usuário pode especificar os parâmetros da técnica através de uma instância da classe *CHedgehogsDlg*, que permite escolher a variável oceanográfica e os valores mínimo e máximo a serem visualizados. A classe *CHedgehogsView* é responsável pela aplicação da técnica e visualização, utilizando instâncias das classes: *vtkLookupTable* para a aplicação de cores *Hue-Saturation-Value-Alpha*; *vtkGlyphSource2D* para servir como

uma fonte para glifos, utilizando a geometria de setas, na qual diversos parâmetros podem ser definidos; e *vtkGlyph3D*, que é um filtro que copia a representação geométrica dos glifos para cada ponto na estrutura de dados de entrada. O glifo é definido através de dados polinomiais obtidos através da fonte de entrada. Para a visualização das setas, é utilizada uma instância da classe *vtkPolyDataMapper* que faz o mapeamento dos dados polinomiais das setas em primitivas gráficas, e da classe *vtkActor* para a representação das setas na cena.

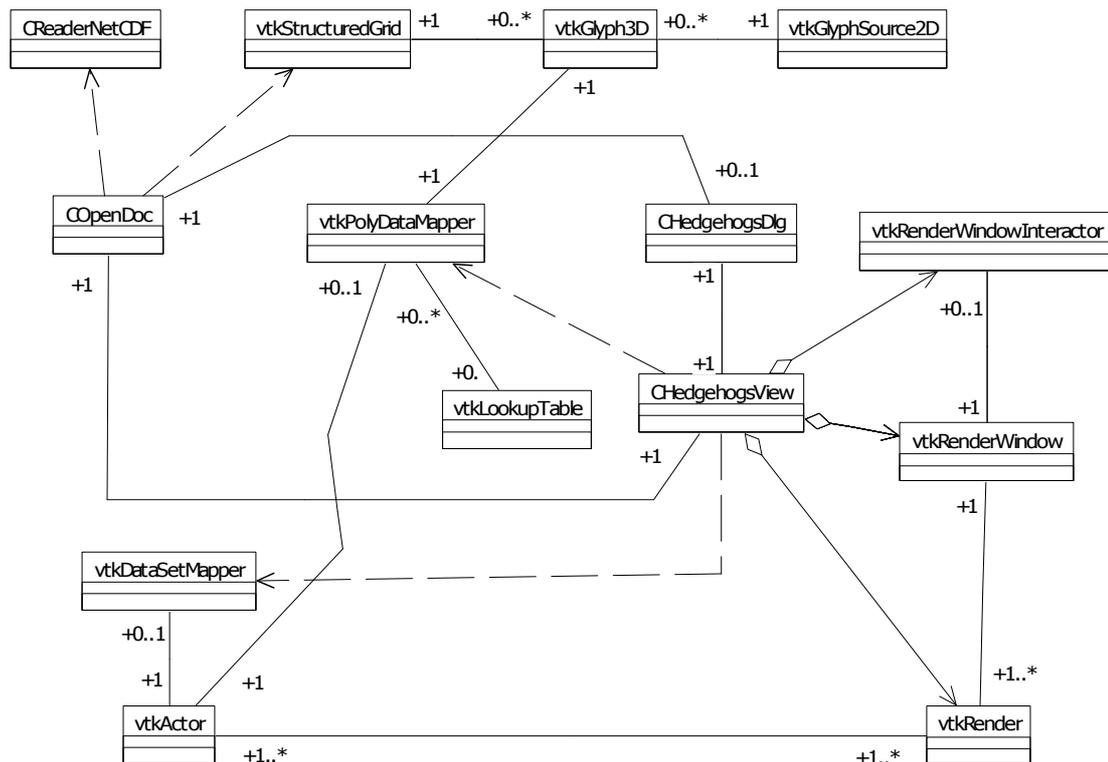


Figura 6.4 – Diagrama de classes para a técnica de *Hedgehogs*

A figura 6.5 apresenta o diagrama de classes modelado para a funcionalidade de cortes curvilíneos. Para esta técnica, a classe *COpenDoc* deve ter sido instanciada e uma técnica de visualização deve ter sido aplicada. Quando o usuário seleciona a ferramenta de corte, a classe *CSpline* é instanciada, e a *spline* é posicionada no plano *xy* da superfície do volume. A janela de visualização consiste em uma instância da classe *View*, e é nesta janela que o usuário pode manipular a *spline* e realizar o corte através de comandos. A classe *vtkCommand* é responsável pela operações de monitoramento dos eventos do usuário de pressionar as teclas e também a interação com a *spline*, realizada através do mouse. Portanto, ao interagir com a *spline*, uma instância da classe *CSplineInteractor* também é criada. Uma instância da classe *CCallback* entra em execução sempre que o usuário pressionar uma tecla, já que o corte

curvilíneo só é realizado quando o usuário pressiona a tecla “c”. A classe *CCallback* é uma subclasse da classe *vtkCommand*, responsável por mostrar mensagens de erro e verificar as teclas pressionadas pelo usuário.

Para esta técnica, é preciso instanciar a classe *vtkCellLocator* para obter todas as células que contêm ou estão próximas dos pontos que compõe a *spline*. Após, um algoritmo é executado para definir todas as células do volume que fazem parte do corte, ou seja, que estão dentro da região determinada pela *spline*. O sub-volume é apresentado na janela de visualização, na qual as células que não fazem parte do corte são definidas como invisíveis e instancias das classes *vtkStructuredGridGeometryFilter* e *vtkPolyDataMapper* são usadas, respectivamente, como os objetos de filtro e mapeamento do *pipeline* de visualização.

A classe *CSplineInteractor* é uma subclasse da classe *vtkCommand* e possui como atributo os pontos da *spline*, que são definidos sempre na interação. Com isso, a classe *CCallback* obtém os pontos da *spline* sempre atualizados, à medida que o usuário manipula a *spline*.

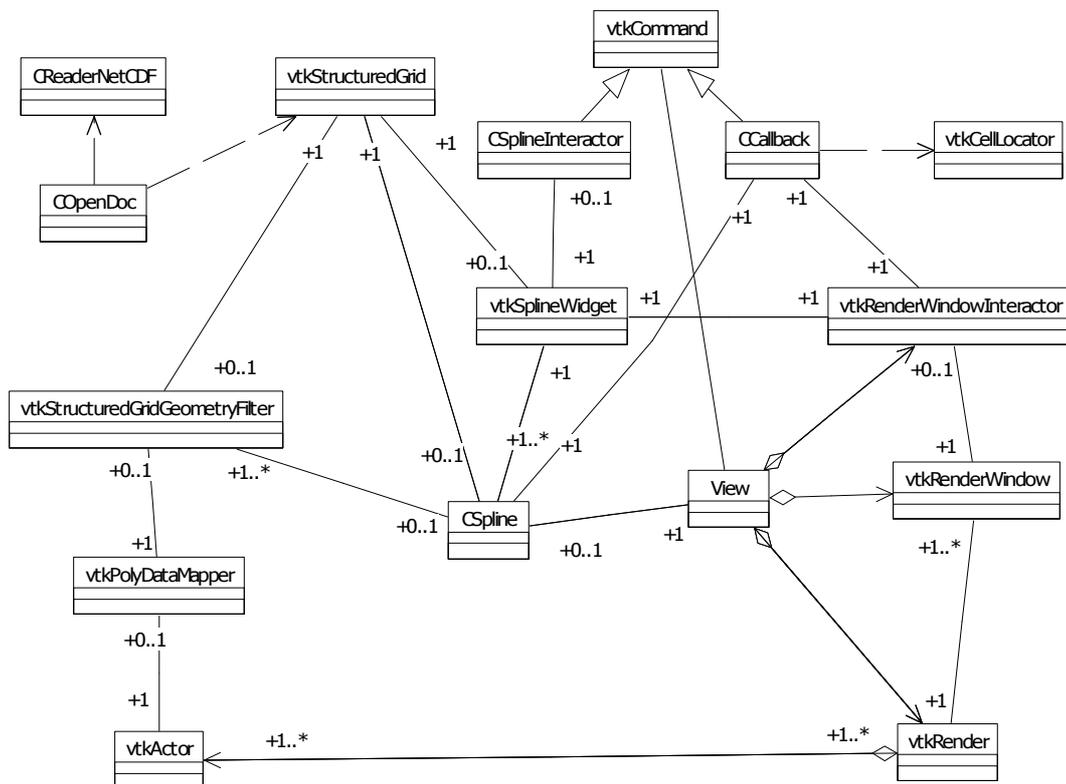


Figura 6.5 – Diagrama de classes para a técnica de cortes curvilíneos

A figura 6.6 apresenta o diagrama de classes para a animação do volume. Para gerar esta animação, a classe *COpenDoc* também deve ter sido instanciada e alguma técnica de

visualização deve ter sido aplicada. Através das funcionalidades disponíveis na classe *CAnimationDlg*, o usuário pode especificar os parâmetros da animação, selecionando a tomada de tempo inicial e final. A classe *CAnimation* é responsável geração da animação.

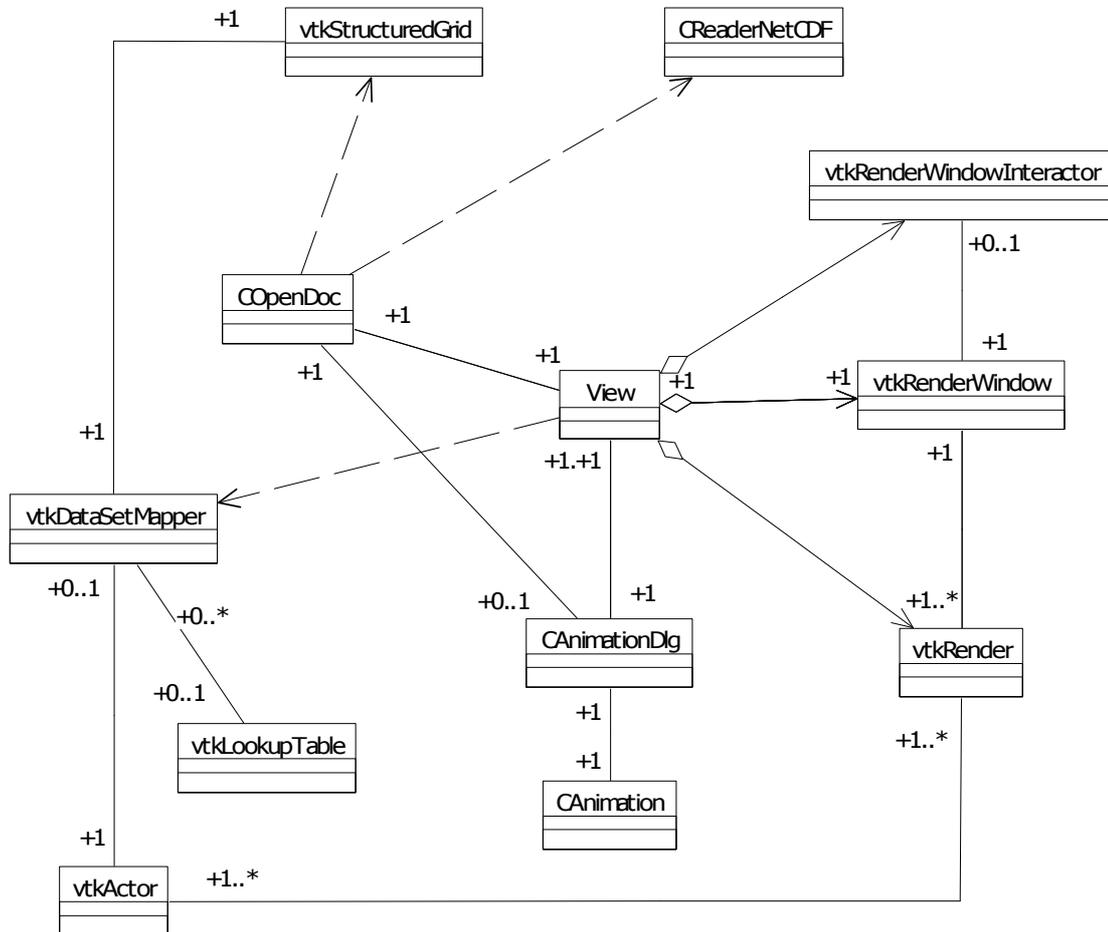


Figura 6.6 – Diagrama de classes para a geração da animação do volume

Através da leitura dos dados no formato NetCDF, foi possível obter o valor das variáveis geradas ao longo do tempo. Além das variáveis puramente oceanográficas, existem outras necessárias para a apresentação do volume de dados. As variáveis utilizadas no desenvolvimento são apresentadas na tabela 6.1.

Para a visualização da temperatura, salinidade e velocidade, são lidos os valores correspondentes a uma determinada tomada de tempo dos dados simulados. Estas informações são armazenadas em uma estrutura de dados própria do VTK, o *vtkStructuredGrid*, que é uma grade de dados estruturados na qual os dados são regulares na topologia e irregulares na geometria. Esta estrutura é composta de células hexaedras (3D) e

quadriláteras (2D). Esta grade estruturada é criada com a especificação das dimensões da grade (para definir a topologia) e os pontos de coordenadas x , y e z (para definir a geometria).

Tabela 6.1 - Variáveis do modelo de dados

Variável	Nome da Variável	Unidade
Δx	Espaçamento da grade em x	Metros
Δy	Espaçamento da grade em y	Metros
Z	Variável sigma, da face da célula	Nível sigma
Zz	Variável sigma, do centro da célula	Nível sigma
H	Altura da coluna d'água, em x e y	Metros
T	Temperatura	Celsius
S	Salinidade	PSS
U	Componente da velocidade em x	m/s
V	Componente da velocidade em y	m/s
W	Componente da velocidade em z	m/s
Time	Tomadas de tempo, onde é gerado o dado simulado	Cada tomada corresponde a 6 horas do dia

Nesta estrutura, os valores de x e y foram obtidos com o incremento do espaçamento de grade definidos pelas variáveis Δx e Δy , apresentadas na tabela 6.1. O valor de z foi obtido aplicando-se a equação 3.1. Para isto, é realizada a leitura dos valores sigma (variável z) e h (altura da coluna d'água). Considerando os valores numéricos do espaçamento horizontal, algo em torno de 5000 metros na grade, e o fato da altura possuir valores ínfimos, iniciando perto da costa em 1 metro até 3000 metros após a quebra da plataforma oceânica, os valores da altura foram alterados multiplicando seu valor numérico por 100. Só assim foi possível observar a altura em relação ao volume na visualização.

Após a geração do volume, pode ser aplicada uma forma de visualização para uma determinada variável oceanográfica. Os dados são, então, mapeados geometricamente e apresentados na tela. A partir disso, o usuário pode interagir com o volume e/ou animá-lo ao longo do tempo para observar as mudanças de valores da variável.

6.3 INTERFACE DO SISTEMA

A *interface* do sistema é composta por menus de acesso, uma barra de ferramentas e uma barra de *status*. A figura 6.7 apresenta a tela inicial do protótipo desenvolvido neste trabalho, na qual apenas o menu *File*, *View* e *Help* estão habilitados.



Figura 6.7 – Interface do protótipo

O menu *File* é utilizado para abrir e fechar arquivos de dados, acesso rápidos aos 4 últimos arquivos e imprimir a tela de visualização. A figura 6.8 apresenta as opções de submenu.

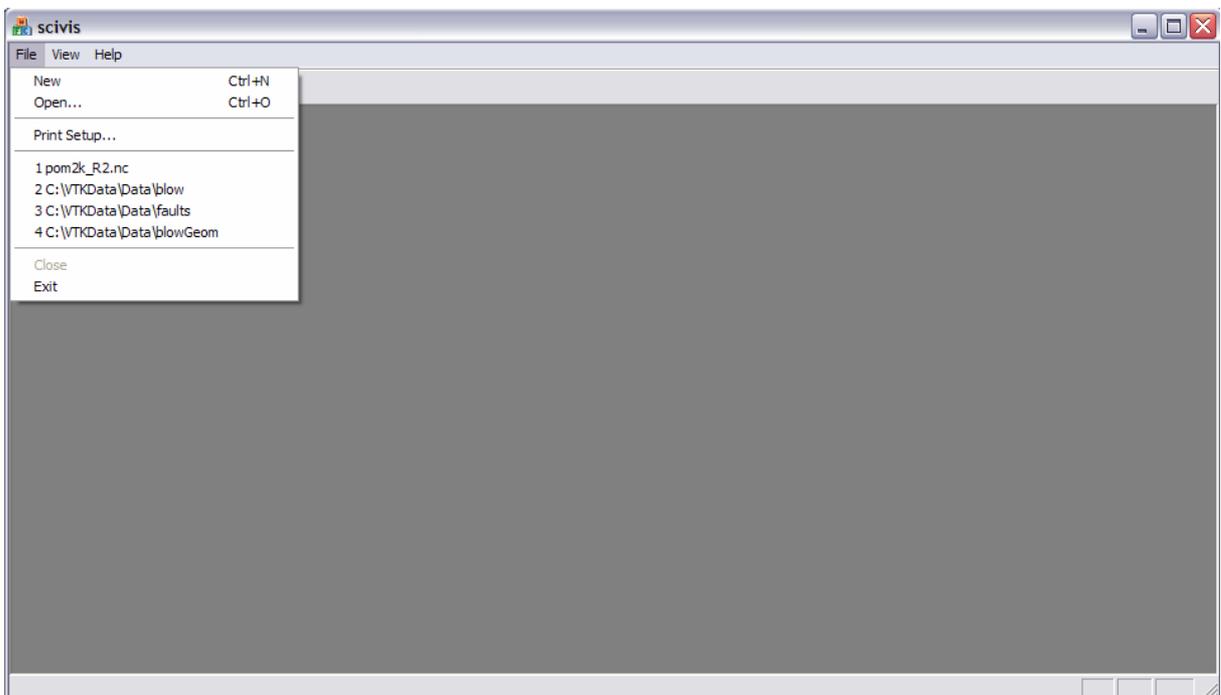


Figura 6.8 – Menu File

Após a abertura do arquivo os dados são carregados para a estrutura de dados e uma visualização *default* do volume é apresentada ao usuário. Nesta visualização é possível apenas ter uma idéia da dimensão do volume. Com isso, as outras opções de menu ficam habilitadas para uso. A figura 6.9 apresenta o menu do sistema após o carregamento do arquivo de dados.

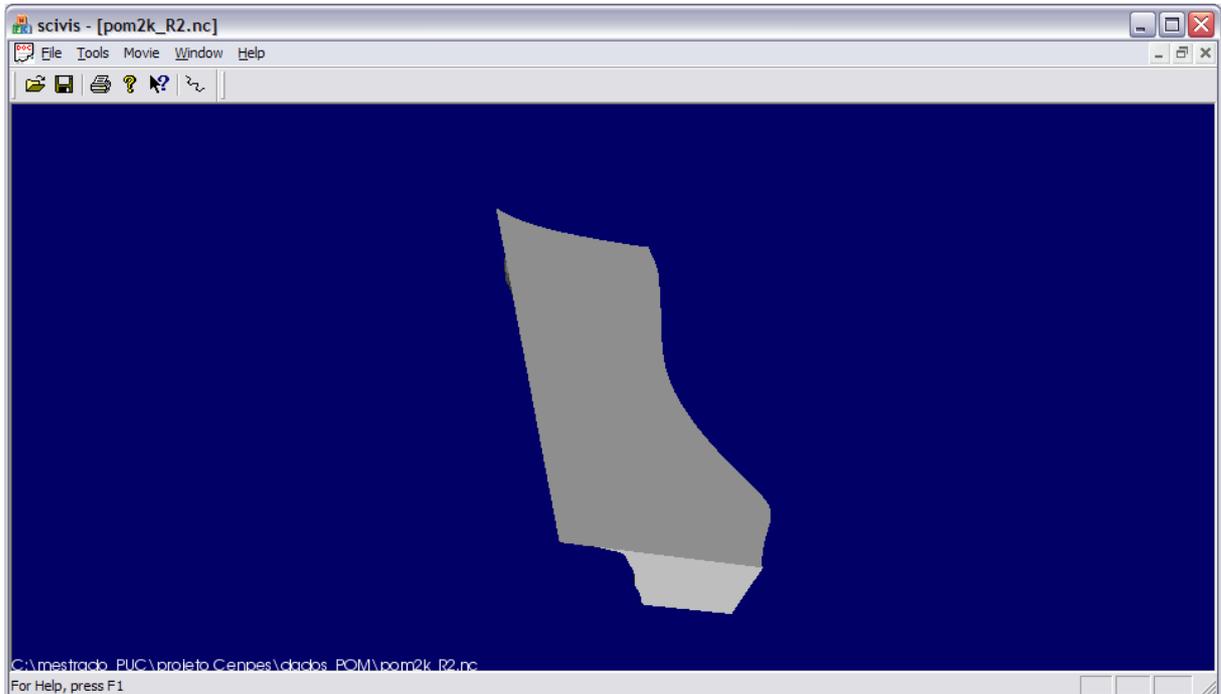


Figura 6.9 – Menu do Sistema

O menu *Tools* provê acesso às técnicas de visualização escalares e vetoriais, além das técnicas de interação. Neste menu também é possível ativar/desativar a barra de ferramentas e a barra de *status*. A figura 6.10 mostra as opções de submenu.

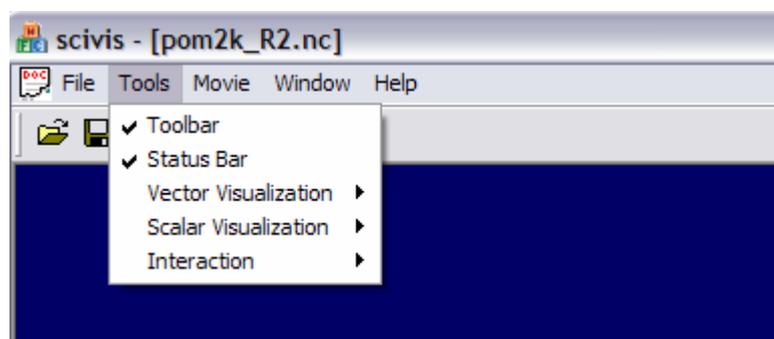


Figura 6.10 – Menu Tools

No menu *Movie* o usuário pode gerar uma animação a partir de um arquivo de dados já carregado na ferramenta, selecionando o período de tempo. A figura 6.11 apresenta as opções de submenu.



Figura 6.11 – Menu Movie

No menu *Window* o usuário pode manipular as janelas da ferramenta. A figura 6.12 apresenta as opções de submenu.

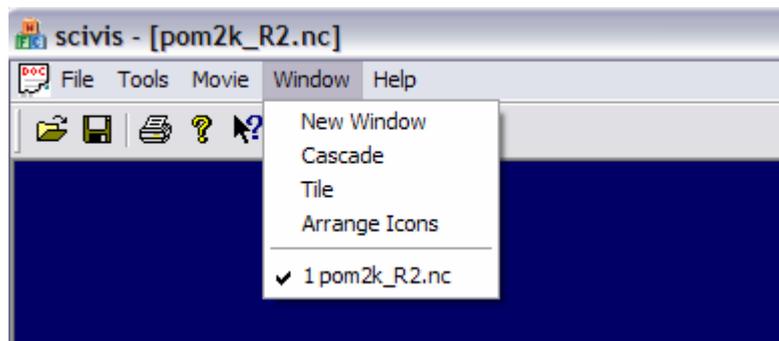


Figura 6.12 – Menu Window

O menu *Help* contém arquivos de ajuda ao usuário e o usuário pode obter informação sobre a versão da ferramenta. A figura 6.13 apresenta as opções de submenu.

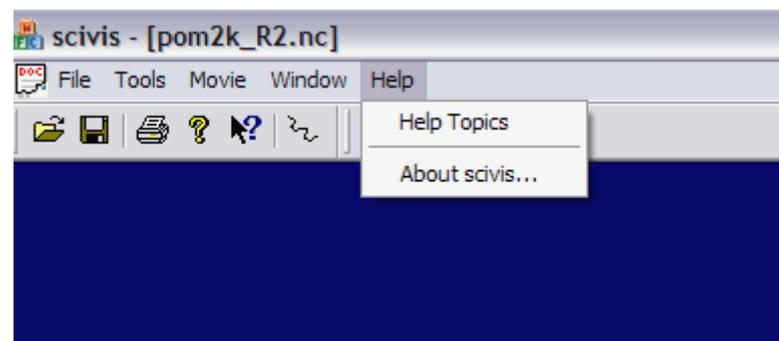


Figura 6.13 – Menu Help

6.4 TÉCNICAS DE VISUALIZAÇÃO E INTERAÇÃO

6.4.1 Visualização de Dados Escalares

Para os dados escalares como temperatura e salinidade, foi implementada a técnica de mapeamento de cores (ver seção 4.3). O diagrama de fluxo de dados da figura 6.14, utilizando a notação Gane-Sarson [Gane e Sarson, 1990], mostra o processamento dos dados para a aplicação da técnica. Fontes e depósitos de dados são representados como retângulos abertos no lado direito. Processos são representados como círculos.

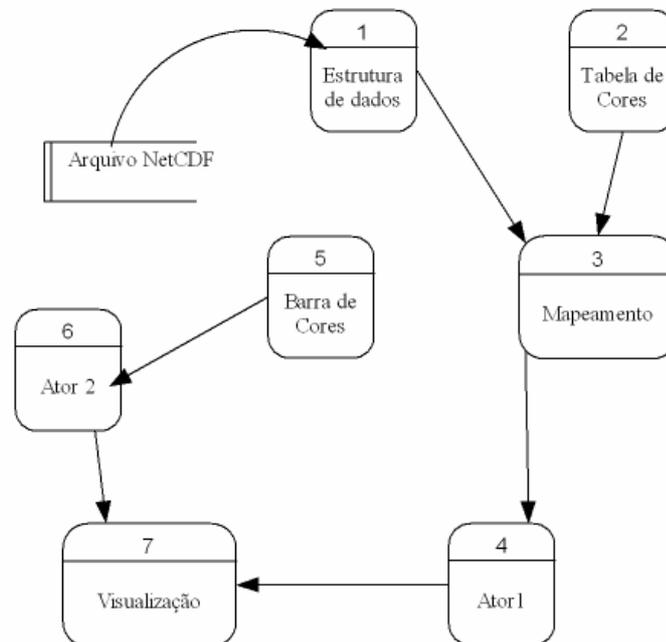


Figura 6.14 - Diagrama de fluxo de dados da técnica de mapeamento de cores

Conforme ilustra o diagrama da figura 6.14, os dados são lidos do arquivo NetCDF. Após a leitura, é criada uma estrutura de grade 3D, contendo a variável escalar. Esta estrutura é mapeada para primitivas gráficas, onde então é transformada em um ator⁶ para a visualização na tela. Na figura 6.15 é apresentada a visualização gerada para a variável temperatura em grau *Celsius*, utilizando o mapeamento de cores RGB. Neste volume, os valores de temperatura são mapeados em cinco intervalos entre 0 e 30 graus, com uma cor correspondendo a cada intervalo de temperatura. Os dados utilizados foram fornecidos pelo CENPES (seção 6).

⁶ Um Ator é utilizado para representar uma entidade em uma cena de renderização. Faz parte do *pipeline* de visualização do VTK.

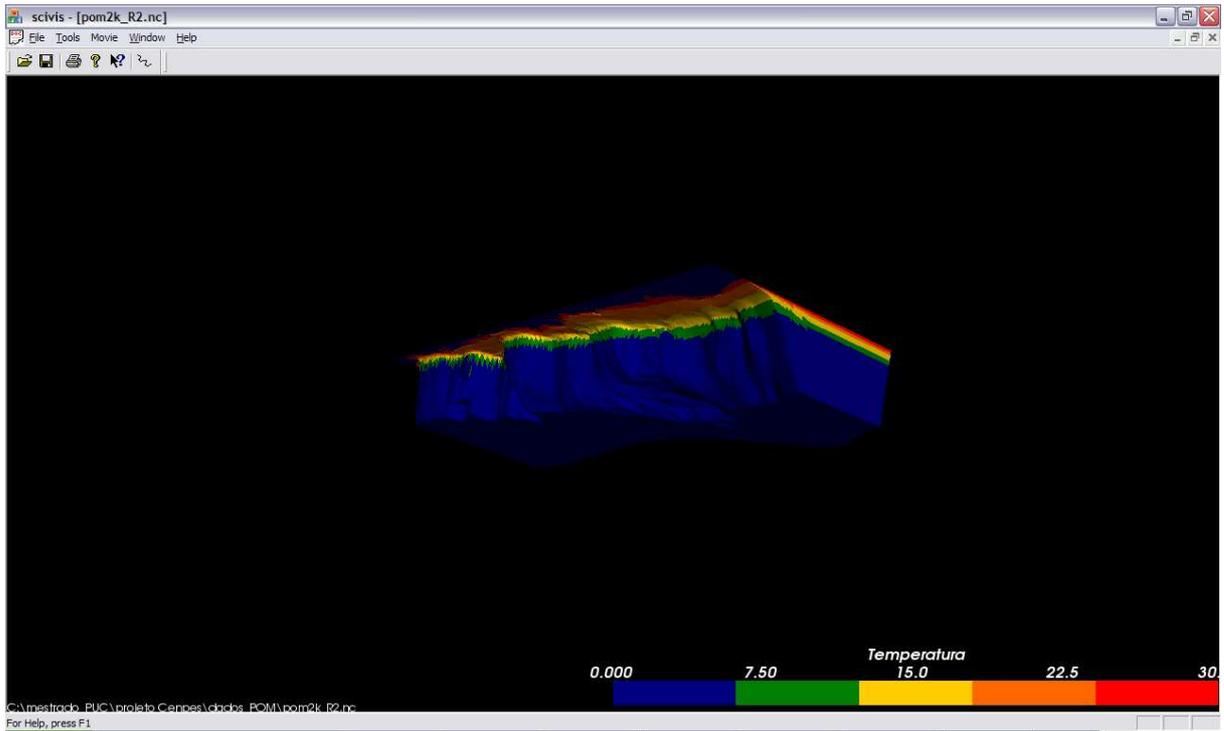


Figura 6.15 – Visualização da temperatura em Celsius

6.4.2 Visualização de Dados Vetoriais

Para a visualização da velocidade, a técnica de *Hedgehogs* descrita na seção 4.11 foi aplicada ao modelo. As setas 3D utilizadas indicam a direção e o sentido, e a magnitude é dada pelo tamanho da seta e mapeamento de cores.

Com uma grade contendo 630000 pontos, é necessário aplicar um filtro para que apenas os pontos com velocidades relevantes sejam apresentados, evitando, assim, prejudicar a visualização do modelo, pois algumas setas podem ficar sobrepostas devido ao tamanho da grade em algumas regiões. A figura 6.16 apresenta o diagrama de fluxo de dados utilizado. Neste diagrama, os dados são lidos do arquivo NetCDF. Após a leitura, é criada uma estrutura de dados 3D, que é então mapeada em primitivas gráficas e transformada em um ator, para a geração da visualização. Também a partir do processo de estrutura de dados, é aplicado um filtro para a seleção dos pontos de interesse. A partir destes dados, são criadas as setas para a representação da variável oceanográfica, sendo então mapeadas em primitivas gráficas e transformadas em um ator, para a visualização na tela.

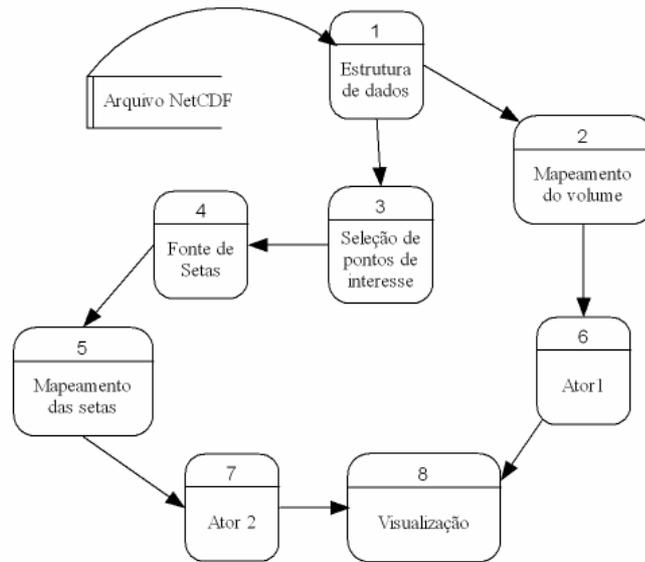


Figura 6.16 - Diagrama de fluxo de dados para a técnica de Hedgehogs.

A figura 6.17 ilustra o resultado da aplicação da técnica de *Hedgehogs*. Nesta imagem, a magnitude da velocidade é dada pelo comprimento da seta e mapeamento de cores. A seta é orientada conforme a direção da velocidade das correntes.

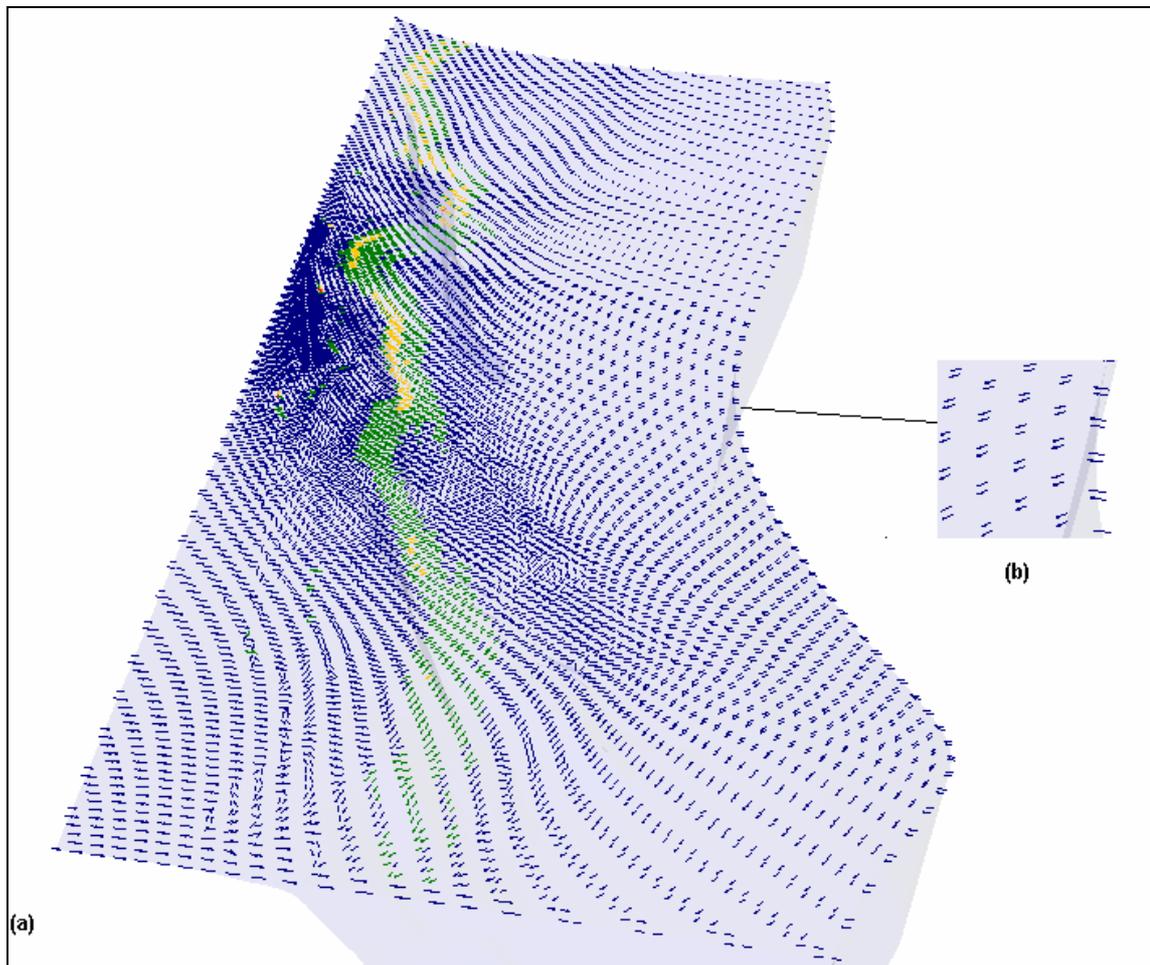


Figura 6.17 - Visualização da velocidade das correntes do volume total (a) e aplicando zoom (b).

6.4.3 Cortes Curvilíneos

Neste trabalho foi desenvolvida uma técnica de interação que consiste na geração de cortes curvilíneos para a seleção da região de interesse. Com esta técnica, o usuário interage com uma *spline*⁷, movendo os pontos de controle dos diversos segmentos de linhas que a compõe. A *spline* utiliza a interpolação Kochanek [Kochanek e Bartels, 1984], que possibilita um maior controle sobre a forma da curva da spline.

Muitos pesquisadores da área oceanográfica utilizam apenas visualizações bidimensionais, observando as camadas do oceano (superfície x, y) para cada ponto de profundidade (z) da grade. O corte facilita a análise tridimensional dos dados, observando-se a variação dos fenômenos físicos em todas as camadas ao mesmo tempo, através de um corte mais preciso, já que é curvilíneo.

Para a geração do corte, a *spline* é manipulada em uma superfície que corresponde ao plano *xy*. A *spline* é utilizada para definir uma região de interesse, da seguinte maneira: após a sua manipulação no plano *xy*, ela é arrastada no eixo *z*. Assim, depois da seleção da área, é gerado um sub-volume, com adição da profundidade (coordenada *z*). Com esta técnica, é possível também utilizar este sub-volume para a animação, com variação dos parâmetros ao longo do tempo. A figura 6.18 ilustra a *spline* após sua manipulação no volume e a figura 6.19 apresenta o sub-volume gerado a partir do corte feito com a *spline*.

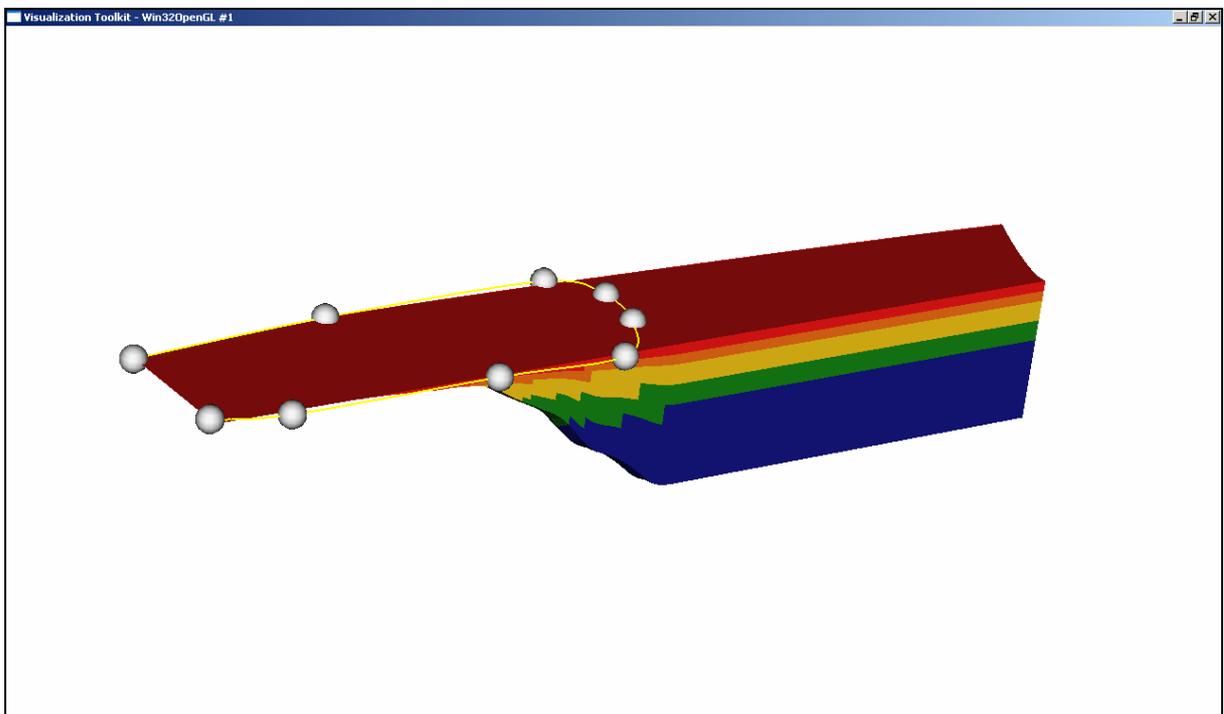


Figura 6.18 – Posição final da spline após sua manipulação

⁷ *Spline* é uma curva definida matematicamente por dois ou mais pontos de controle.

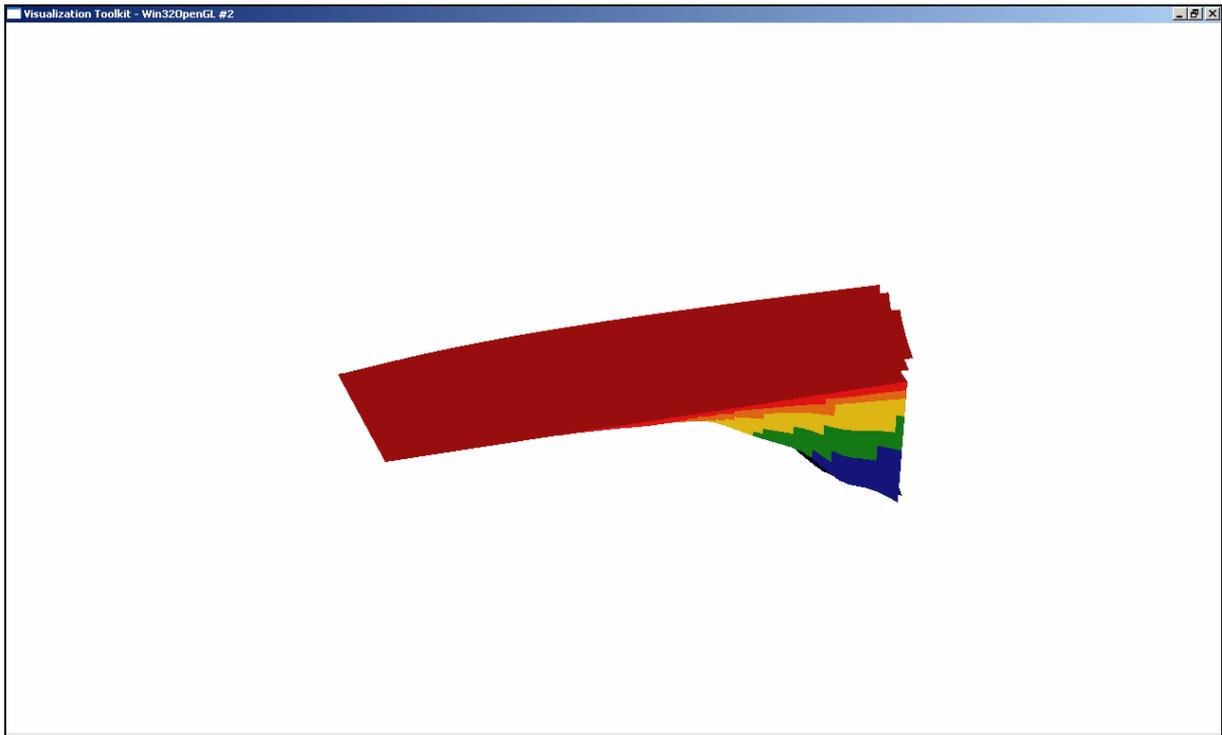


Figura 6.19– Sub-volume gerado pelo corte feito com a spline da figura 6.17

Para a geração do corte, os pontos de controle das pontas da *spline* devem estar posicionados na borda superior. O corte é realizado com a seleção das células do interior do corte e aquelas que colidem com a *spline*. A detecção das células do volume que colidem com os pontos definidos na *spline* é o primeiro passo a ser realizado.

Uma das alternativas é procurar em todas as células qual contém o ponto p procurado. Para isto, funções de interpolação da célula são resolvidas para as coordenadas paramétricas. Se estas coordenadas estão dentro da célula, então p está na célula. Para garantir um melhor desempenho, devem ser utilizadas estruturas de dados que suportem busca espacial. Dentre as estruturas, estão mapas MIP, *octrees*, *Kd-trees* e *binary sphere trees*.

No VTK existe a classe *vtkCellLocator* que adiciona performance às buscas espaciais, utilizando uma subdivisão *octree*. Esta classe funciona da seguinte maneira: dado um ponto p_1 (x, y, z), encontra-se o ponto p mais próximo (identificador). Este ponto encontrado não necessariamente faz parte dos vértices. Assim, obtém-se também o identificador da célula que contém o ponto p . A partir do identificador da célula, a variável temperatura ou salinidade é obtida.

Assim, após obter as células da superfície que colidem com a *spline*, é utilizada uma outra estrutura de lista, que irá armazenar todas as células que fazem parte do corte no plano xy .

Os identificadores das células da grade estão ordenados em ordem crescente da esquerda para a direita, de baixo para cima e são atribuídos na criação da estrutura de grade, como números naturais. Esta atribuição é realizada internamente pela biblioteca VTK na geração da estrutura de dados. A figura 6.20 ilustra os identificadores das células.

12	13	14
9	10	11
6	7	8
3	4	5
0	1	2

Figura 6. 20 – Identificadores das células do modelo estruturado

O corte é realizado ao longo do eixo *y*. Para a realização do corte, as pontas da *spline* devem estar posicionadas em uma das bordas do volume deste eixo, na borda esquerda ou direita, conforme indica a figura 6.20. Primeiramente são obtidos os identificadores das células da superfície que colidem com a *spline*, armazenando em uma lista. Este algoritmo é executado até que todos os identificadores de célula da lista sejam lidos.

O algoritmo de corte funciona da seguinte maneira:

1. Leitura de um valor da lista de células.
2. A partir do identificador de célula da lista, testa primeiramente se a célula logo abaixo (coluna) faz parte da borda e se é menor que o maior valor de identificador da lista. Em caso positivo, este identificador faz parte do corte. Um novo identificador é lido da lista (passo 1). Caso falso, segue o passo 3.
3. A partir do identificador de célula da lista, testa se a célula logo abaixo (coluna) faz parte da lista, ou se o vizinho desta célula abaixo (vizinho esquerdo no caso da borda esquerda ou vizinho direito no caso da borda direita) faz parte da lista. Caso seja falso, este identificador faz parte do corte. Um novo valor da lista é lido (passo 1).

A partir dos identificadores da superfície que fazem parte do corte, o sub-volume é gerado, obtendo as outras células do volume que se encontram em razão da profundidade.

6.4.4 Técnicas Básicas de Interação

A ferramenta de visualização contém algumas técnicas básicas de interação com o volume. Estas técnicas incluem a aplicação de *zoom in* e *out*, rotação e translação do volume, aplicação de *wireframe* e cor sólida.

Estas técnicas básicas são providas pelo VTK como *default*, e fazem parte da classe *vtkRenderWindowInteractor*. Esta classe manipula as interações do mouse e teclado realizadas pelo usuário.

6.4.5 Animação

A animação consiste em apresentar a visualização de todas as tomadas de tempo da simulação, permitindo, assim, analisar a variação das características oceanográficas, como temperatura, salinidade e velocidade das correntes oceânicas. Sendo assim, o volume é animado com a escolha de uma variável oceanográfica, uma técnica de visualização e tomada de tempo inicial e final. Obtendo o valor da variável em todas as tomadas de tempo, é realizada uma atualização do valor na grade do volume para cada tomada de tempo, gerando uma animação, contendo um *timer* para a apresentação em seqüência de cada tomada de tempo.

6.5 AVALIAÇÃO DO PROTÓTIPO

Nesta seção serão detalhadas as funcionalidades presentes nas ferramentas estudadas e um comparativo com o protótipo desenvolvido neste trabalho. A interatividade ainda é o ponto-chave na maioria das ferramentas, com exceção do ISTV e Ensign. Nenhuma ferramenta possui o corte curvilíneo.

A ferramenta GeoZui3D possui uma interação básica, incluindo *zoom*, rotação e translação. Utiliza mapas de terreno e imagens georeferenciadas. Possui linhas, polígonos e pontos para a construção de estruturas 3D. Como comparativo, este sistema não possui um leitor para dados simulados, é preciso incluí-lo na ferramenta. Além disso, não realiza animação. Já a ferramenta ISTV é extremamente rica e gratuita. Possui técnicas de mapeamento de cores e *color Wheel* para a visualização de dados escalares e técnicas de tubos, *streamlines*, *streaklines* e *pathlines* para dados vetoriais. Possui também animação e boa interatividade do sistema. Em comparação com o protótipo, esta ferramenta é utilizada apenas em plataforma *Unix*. Isto limita o número de usuários da ferramenta.

O POPTEX é uma ferramenta proprietária e suporta apenas os dados simulados POP. Possui geração de imagens em altíssima resolução, produzindo assim um vídeo. Porém, não possui outras técnicas de interação. Visualiza as variações oceanográficas apenas com a animação. O Envis/Hum também é outra ferramenta onde a visualização das mudanças do comportamento do oceano é realizada através da geração de imagens em alta resolução. Possui apenas técnicas básicas de interação como *zoom* e rotação.

O Vis5D é uma ferramenta gratuita, para a visualização de grande conjuntos de dados. No entanto, possui apenas leitor para o formato próprio da ferramenta. Novos leitores devem ser desenvolvidos e incluídos. O Ferret também é uma ferramenta gratuita que roda na maioria dos sistemas Unix. No Windows é necessário utilizar um emulador. Possui uma excelente análise matemática, e é similar ao Matlab. A interface e acesso a funções é através de uma *shell* de comandos e por não possuir uma *interface* gráfica, torna mais difícil sua utilização sem um grande estudo da ferramenta.

O Ensign é uma ferramenta bastante rica, incluindo técnicas de visualização para dados vetoriais e escalares, animação, interação e suporta diversos leitores de dados. Porém, o leitor de NetCDF para os dados do modelo POM não pode ser utilizado e precisou ser realizada uma adaptação nos dados, gerando um formato intermediário de leitura. Além disso, é uma ferramenta comercial, com código proprietário.

6.5.1 Desempenho

Foram realizados testes de performance da ferramenta, utilizando o modelo POM, através da leitura do arquivo em formato NetCDF. Este modelo possui uma grade de 100x300x21, possuindo 630000 células. A máquina utilizada possui processador Pentium 4 3.33 GHz, com 448 MB de memória RAM.

Para o carregamento dos dados e geração do volume, sem a aplicação de técnica de visualização, levou cerca de 1 segundo. Para a aplicação da técnica de dados escalares, o volume foi gerado em cerca de 3 segundos. Para a técnica de visualização vetorial, o volume foi gerado em cerca de 4 segundos.

Para o corte curvilíneo, o tempo de execução do corte e geração de visualização levou cerca de 6 segundos.

7 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi projetado e desenvolvido um sistema gratuito destinado à visualização 3D de dados oceanográficos. Esta ferramenta tem como objetivo permitir a visualização de dados do oceano, obtidos através de simulação, contendo a variação temporal dos diversos parâmetros físicos. Para este trabalho, foi realizado um grande estudo para entender os dados oceanográficos e modelo de dados, além da biblioteca VTK utilizada na implementação do protótipo. A ferramenta foi desenvolvida utilizando o paradigma de orientação a objetos e pode ser facilmente estendida para a inclusão de novas funcionalidades.

Como estudo de caso, foi utilizado o modelo de dados simulados *Princeton Ocean Model* (POM), armazenado no formato de dados NetCDF. Os dados oceanográficos foram fornecidos pelo CENPES, através de uma visita à sede no Rio de Janeiro. No CENPES, é realizada apenas a visualização bidimensional dos dados, e este tipo de visualização ainda é muito utilizada pelos oceanógrafos. Portanto, a visualização tridimensional é muito nova nesta área.

Com base nas características físicas do oceano, foram desenvolvidas técnicas de mapeamento de cores e *Hedgehogs* para a visualização de dados escalares e vetoriais. Além das técnicas básicas, tais como rotação, translação e zoom, foi desenvolvida uma técnica de interação através de cortes curvilíneos. Esta técnica tem como base a utilização e manipulação de *splines* para selecionar a região de interesse. A partir da *spline*, é gerado um sub-volume. Considerando a variação temporal das características oceanográficas, foi desenvolvida uma animação, através da visualização dos dias de dados simulados. Desta forma, a análise é realizada de maneira mais fácil e rápida.

Para este trabalho, foram realizados diversos estudos de técnicas e ferramentas destinadas à visualização de dados oceanográficos, tais como a ISTV e o Ensign, que são extremamente ricas no que se refere à aplicação de técnicas de visualização, leitura de dados e interação. Entretanto, estas ferramentas carecem de funcionalidades úteis aos oceanógrafos. Portanto, como contribuição ao estado da arte, foi desenvolvida a técnica de interação

utilizando *splines*. Com isso, é possível selecionar a região de interesse através de um corte mais preciso utilizando curvas e através de visualização tridimensional. Este tipo de corte não foi encontrado nas ferramentas estudadas e permite ao usuário visualizar o oceano em sua profundidade de maneira mais fácil e simples, já que as variações do oceano são mais buscadas na profundidade. Esta técnica foi desenvolvida para suprir também as necessidades de visualização indicadas pelo CENPES.

São poucas as ferramentas gratuitas de visualização oceânica e este trabalho se propõe a ser uma primeira etapa no desenvolvimento deste tipo de sistema. Existem ferramentas gratuitas não específicas para a visualização oceânica, mas devem ser sempre adaptadas pelo usuário. A ferramenta não se restringe apenas a visualização de dados simulados, podendo visualizar qualquer dado oceânico no formato NetCDF. Outros formatos podem ser incorporados à ferramenta, com a adição de novos leitores de dados.

Como trabalhos futuros, novos modelos de dados e outros formatos de arquivos podem ser acrescentados à ferramenta, através do desenvolvimento de um leitor apropriado. Além disso, novas técnicas de visualização podem ser incluídas à ferramenta, enriquecendo assim a análise, como, por exemplo, a implementação da técnica de isosuperfícies e *streamlines*.

A técnica de corte utilizando *splines* também pode ser enriquecida, com a realização do corte em outros planos, além do *xy*. Outras técnicas de interação também podem ser implementadas e incluídas na ferramenta, de maneira a possibilitar alterações qualitativas e quantitativas da imagem, como a alteração da tabela de cores, a seleção de pontos de interesse e a utilização de funções de cortes. O uso de técnicas de visualização tridimensional, associadas à técnicas bidimensionais, permite também obter um melhor entendimento dos dados.

Como trabalho futuro, fica também a sugestão de desenvolvimento de técnicas de Realidade Virtual. A implementação destas técnicas é facilitada com o uso da biblioteca VTK, que é baseada em *OpenGL*. Atualmente, existe um grande investimento nesta área, que permite adicionar maior realismo à visualização científica. Além disso, técnicas de IHC devem ser incorporadas à ferramenta, já que estas técnicas diminuem o tempo de análise dos dados.

REFERÊNCIAS BIBLIOGRÁFICAS

- AVILA, L. S. et al. **The VTK user's guide** : updated for VTK version 4.2. Upper Saddle River, NJ : Kitware, 2003. 324 p.
- BAJAJ, C. **Data visualization techniques**. Chichester: John Wiley & Sons, 1999. 144 p.
- BARTELS, R. H. **An introduction to Splines for use in Computer Graphics and geometric modeling**. Los Altos, CA : M. Kaufmann, 1987. 476 p.
- BLUMBERG, A. F.; MELLOR, G. L. A description of a three-dimensional coastal ocean circulation model. In: **THREE-DIMENSIONAL COASTAL OCEAN MODELS, 1987. Proceedings...** Washington D.C : N. Heaps, 1987. p. 208-233.
- BUSCHMANN, F. et al. **A system of patterns: pattern-oriented, software architecture**. Chichester : J. Wiley, 1997. 467 p.
- CHUPA, M. A. **Notes for MSU ERC slides**. Disponível em: http://www.erc.msstate.edu/vail/projects/hpvci/NPSPIPS/200101_notes/notes.html. Recuperado em novembro de 2004.
- CHUPA, M. A. et al. EnVis/Hum: High-resolution Ocean Model Visualization and Display. In: **IEEE OCEANS, 2003. Proceedings...** Los Alamitos: IEEE Computer Society Press, 2003. p. 1030-1031.
- COELHO, A. et al. Parallel Volume Rendering for Ocean Visualization in a cluster of PCs. In: **GEOINFO: SIMPÓSIO BRASILEIRO DE GEOINFORMÁTICA, 4., 2004, Campos do Jordão, São Paulo. Anais...** [S.l.: s.n.], 2004. p. 291-304.
- CRAWFIS, R.; MAX, N. Direct volume visualization of three dimensional vector fields. In: **ACM WORKSHOP ON VOLUME VISUALIZATION, 1992. Proceedings...** ACM Press, 1992. p. 55-60.
- CROSSNO, P.; ANGEL, E.; MUNICH, D. Case Study: Visualizing Ocean Currents With Color and Dithering. In: **IEEE SYMPOSIUM ON PARALLEL AND LARGE-DATA VISUALIZATION AND GRAPHICS, 2001. Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p. 37-40.
- EARNSHAW, R. A.; WISEMAN, N. **An Introductory Guide to Scientific Visualization**. Berlin: Springer-Verlag, 1992. 156 p.
- ENSIGHT. **Computacional Engineering Internacional - CEI**. Disponível em: <http://www.mscsoftware.com.au/products/software/cei/ensight/>. Recuperado em setembro de 2005.
- FREITAS, C. M. D. S. **Uma Abordagem Unificada para Análise Exploratória e Simulação Interativa Visual**. 1994. 143 f. Tese (Doutorado em Ciência da Computação)-Faculdade de Informática, UFRGS, Porto Alegre, 1994.

GAITHER, K.; MOORHEAD, R. J. Visualizing Vector Information in Ocean Environments. In: IEEE VISUALIZATION, 2002. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2002. p. 529 – 532.

GALLAGHER, R. S. **Computer visualization** : graphics techniques for scientific and engineering analysis. Boca Raton: Editora CRC, 1995. 312 p.

GANE, C.; SARSON, T. **Análise estruturada de sistemas**. Rio de Janeiro: Editora LTC, 1990. 257 p.

GEOZUI3D. **Geographic Zooming User Interface**. Disponível em: <http://www.ccom.unh.edu/vislab/projects/GeoZui3D.html>. Recuperado em outubro de 2004.

GERSHON, N.; EICK, S. G. Information Visualization. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 17 , n. 4 , p. 29-31, Jul./Ago. 1997.

HDF. **Hierarchical Data Format**. Disponível em: <http://hdf.ncsa.uiuc.edu>. Recuperado em setembro de 2005.

HEAD, M. E. M. et al. Applications of 3-D visualizations of oceanographic data bases. In: IEEE OCEANS, 1997. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1997. p. 1210-1215.

HEARN, D.; BAKER, M. P. **Computer graphics with OpenGL**. 3. ed. Upper Saddle River, NJ : Pearson Education, 2004. 857 p.

HORN, B. K. P. Hillshading and the reflectance map. **Geo-Processing**, v. 2, n. 1, p. 65-146, Out. 1982.

ISTV. **Interactive Structured Time-varying Visualizer**. Disponível em: <http://www.erc.msstate.edu/vail/projects/ISTV/>. Recuperado em outubro de 2004.

KAMPEL, M. **Sensoriamento Remoto Aplicado à Oceanografia**. Instituto Nacional de Pesquisas Espaciais, 2002.

KOCHANNEK, D.; BARTELS, R. Interpolating Splines with Local Tension, Continuity, and Bias Control. **ACM Computer Graphics**, v. 18, n. 3, p. 33-41, Jul. 1984.

LARMAN, C. **Utilizando UML e padrões**. 2. ed. Porto Alegre : Bookman, 2004. 607 p.

LAW, C. C.; HENDERSON, A.; AHRENS, J. An Application Architecture for Large Data Visualization: A Case Study. In: IEEE SYMPOSIUM ON PARALLEL AND LARGE-DATA VISUALIZATION AND GRAPHICS, 2001. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p. 125-128.

MANSSOUR, I. H. **Visualização Colaborativa de Dados Científicos com Ênfase na Área Médica**. 1998. 121 f. Exame de Qualificação (Doutorado em Ciência da Computação)-Faculdade de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 1998.

MANSSOUR, I. H. **Visualização de Estruturas Internas em Volumes de Dados Multimodais**. 2002. 135 f. Tese (Doutorado em Ciência da Computação)–Faculdade de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), PortoAlegre, 2002.

MCPHERSON, A.; MALTRUD, M. POPTX: Interactive Ocean Model Visualization Using Texture Mapping Hardware. In: IEEE VISUALIZATION, 1998. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p. 471-570.

MELLOR, G. L. **Users guide for a three-dimensional, primitive equation, numerical ocean model**. 2003. 53 p. Relatório Técnico-Programa de Ciências Atmosféricas e Oceânicas, Universidade de Princeton, Junho de 2003.

MENDONÇA, M. B. **Aplicação de texturas em visualização científica**. 2001. 92 f. Dissertação (Mestrado em Ciência da Computação)-Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Paulo, 2001.

MOORHEAD, R. J. et al. ISTV: Interactive Structured Time-varying Visualizer. In: IEEE OCEANS ,1999. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999.

MOORE, C. W. et al. Exploring 3-Dimensional Oceanographic Data Sets On The Web Using Virtual Reality Modeling Language. In: IEEE OCEANS, 1999. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p. 1501-1503.

MUELLER, J. P. **Visual C++ .NET developer's guide**. New York, NY: Osborne McGraw-Hill, 2002. 732 p.

NATIONS, S. et al. Interactive Visualization of Ocean Circulation Models. In: IEEE VISUALIZATION, 1996. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1996. p. 429-432.

NCSA. **National Center for Supercomputing Applications**. Disponível em: <http://www.ncsa.uiuc.edu/>. Recuperado em dezembro de 2005.

PENG, G.; LESLIE, L. M.; SHAO, Y. **Environmental modelling and prediction**. Berlin : Springer, 2002. 480 p.

PMEL. **Pacific Marine Environmental Laboratory**. Disponível em: <http://www.pmel.noaa.gov>. Recuperado em dezembro de 2005.

POLITO, P. S. **Oceanografia Dinâmica**. 2004. 71 p. Apostila. Departamento de Oceanografia Física, Instituto Oceanográfico da Universidade de São Paulo, São Paulo, Outubro de 2004.

POPTX. **Projeto do Sistema de Visualização POPTX**. Disponível em: <http://www.ccs.lanl.gov/ccs1/projects/Viz/pr-poptx.html>. Recuperado em outubro de 2004.

REW, R.; DAVIS, G. NetCDF: an interface for scientific data access. **IEEE Computer Graphics and Applications**, Los Alamitos, v. 10 , n. 4 , p. 76–82, Jul. 1990.

RODRIGUES, P. S. S. **Uma metodologia para a detecção de padrões em imagens de escoamento**. 2004. Dissertação (Mestrado em Ciência da Computação)-Instituto de Ciências Exatas, Universidade Federal de Minas Gerais, Minas Gerais, 2004.

ROSENBLUM, L. J. **Visualization of Experimental Data at the Naval Research Laboratory**. US Naval Res. Lab., Washington, DC, 1989.

SCHROEDER, W. J.; MARTIN, K. M.; LORENSEN, W. E. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In: IEEE VISUALIZATION, 1996. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1996. p. 93-100.

SCHROEDER, W. **The visualization toolkit** : an object-oriented approach to 3D graphics. 3. ed. Upper Saddle River, NJ : Prentice-Hall, 2003. 474 p.

SCOFANO, A. M.; NUNES, L. M. P. **Noções Básicas de Oceanografia**. Apostila publicada pela UERJ, fevereiro de 2001, 83 p.

SPERAY, D.; KENNON, S. Volume Probes: Interactive Data Exploration on Arbitrary Grids. In: ACM WORKSHOP ON VOLUME VISUALIZATION, 1990, San Diego. **Proceedings...** ACM Press, 1990. p. 5-12.

TELEA, A.; WIJK, J. J. V. Simplified Representation of Vector Fields. In: IEEE VISUALIZATION, 1999. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p. 35-42.

UNIDATA. **Comunidade Unidata**. Disponível em: <http://my.unidata.ucar.edu/content/software/netcdf/index.html>. Recuperado em junho de 2005.

UNIVERSIDADE DE STUTTGART. Instituto para Visualização e Sistemas Interativos, Universidade de Stuttgart. Disponível em: <http://www.vis.uni-stuttgart.de/eng/research/proj/ito/materials/>

VIS5D. **Sistema de Visualização Vis5D**. Disponível em: <http://www.ssec.wisc.edu/~billh/vis5d.html>. Recuperado em setembro de 2005.

VTK. **Visualization ToolKit**. Disponível em: <http://www.vtk.org/>. Recuperado em janeiro de 2005.

WALLCRAFT, A. J. **The Navy Layered Ocean Model User's Guide**. OARL Report 35, Naval Research Laboratory, Stennis Space Center, MS, 1991.

WARE, C. et al. GeoZui3D: Data Fusion for Interpreting Oceanographic Data. In: IEEE OCEANS, 2001, Hawaii. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p. 1960-1964.

WATT, A. **3D Computer Graphics**. 3. ed. Harlow: Addison-Wesley, 2000. 570 p.