UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

**Mapeamento por Cores
Baseado em Aspectos de
Percepção Visual**

Sandra Mara Tutida

Orientação: Profa. Dra. Maria Cristina Ferreira de Oliveira

*Dissertação apresentada ao Instituto de Ciências Matemáticas e de
Computação - USP, como parte dos requisitos para obtenção do título de Mestre em
Ciências - Área de Ciências de Computação e Matemática Computacional.*

USP – São Carlos
Abril de 1998

Aos meus pais, Malio e Tisuko

AGRADECIMENTOS

À Profa. Dra. Maria Cristina Ferreira de Oliveira, pelo apoio e paciência durante a realização deste trabalho.

À Profa. Dra. Rosane Minghim, pelas contribuições preciosas dadas a este trabalho.

À Profa. Dra. Agma Jucy Machado Traina, pelas contribuições e disponibilização de dados importantes para a realização deste trabalho.

Ao pessoal do LCAD: Onofre, Armando, Eldis, Paulo Renato e Magnus, pela ajuda e compreensão durante o desenvolvimento do projeto.

A minha família, pelo constante incentivo e apoio em todos os meus projetos.

Ao pessoal do CIP – UEL, pelo incentivo para ingressar no programa de mestrado deste Instituto.

Ao Luciano, pela ajuda, paciência e compreensão nos momentos mais difíceis durante a realização deste trabalho.

Aos funcionários do Instituto, pela ajuda.

À Marília, Laura, Beth, Andressa, Sandra e Adriana pelo auxílio e compreensão.

À Monão, Binha e Takeshi pela amizade e incentivo durante todos os momentos.

À Mirla e Veridiana, companheiras de república, pelo apoio e amizade.

Aos meus amigos: Jhuli, Mary, Fabiano, Silmara, Mário, Rudinei, Garga, Márcia, Karina, Ana, Jaque, Cris, Luís Carlos, Ernesto, Chandler, Simone, Paulo, Felipe, Andréia, Marisa, Mineiro pelos momentos de descontração e amizade.

À FAPESP e CNPq pelo apoio financeiro.

A Deus, por me acompanhar em todos os momentos.

RESUMO

Este trabalho aborda a utilização da técnica de mapeamento por cores no contexto da Visualização Científica. É apresentada uma revisão bibliográfica sobre percepção visual de cores e um estudo de abordagens que buscam apoiar o usuário no processo de geração de visualizações. A abordagem de visualização baseada em regras, que prevê a utilização de regras heurísticas que embutem conhecimentos sobre percepção visual de cores e informações relativas aos dados e aos objetivos do usuário ao gerar a visualização, é discutida. Nessa abordagem, as regras restringem o conjunto de mapeamentos gerados para uma visualização, de forma a garantir resultados mais efetivos, sendo que essa abordagem foi implementada na forma de um módulo de apoio à geração de tabelas de cores. Esse módulo restringe o número de cores que podem compor uma tabela e o comportamento das mesmas ao longo da tabela, a partir das informações obtidas sobre a frequência espacial, a natureza dos dados, e a tarefa de visualização, segundo uma taxonomia baseada em conhecimentos sobre percepção visual de cores proposta na literatura. Alguns resultados obtidos com a aplicação das diferentes tabelas de cores geradas pelo módulo são apresentados.

ABSTRACT

This work is concerned with the application of color mapping in the context of Scientific Visualization. It contains a study of color perception issues and a review of the visualization design process and of approaches discussed in the literature for supporting the users in the task of generating effective visualizations. The rule-based visualization approach, that relies on visual perception and information about the nature of the dataset and the user goals when generating visualizations is discussed. The rules restrict the set of attribute mappings available for a particular visualization to ensure the generation of effective visualizations. In this work, this approach was implemented for the particular case of the color mapping technique, in the form of a module to support the generation of color lookup tables. It restricts the set of colors in a table and their behavior according to information obtained about the data frequency distribution, data type and visualization task, according to a taxonomy based on color perception proposed in the literature.

ÍNDICE

1. INTRODUÇÃO	1
1.1 CONSIDERAÇÕES INICIAIS	1
1.2 MOTIVAÇÃO.....	2
1.3 ORGANIZAÇÃO DO TRABALHO	3
2. PERCEPÇÃO VISUAL	5
2.1 CONSIDERAÇÕES INICIAIS	5
2.2 O SISTEMA VISUAL HUMANO	5
2.3 PERCEPÇÃO DE COR	11
2.4 SISTEMAS DE CORES PRIMÁRIAS	15
<i>Modelo XYZ.....</i>	<i>18</i>
<i>Modelo RGB (Red, Green, Blue).....</i>	<i>21</i>
<i>Modelo HSV (Hue, Saturation, Value).....</i>	<i>22</i>
<i>Modelo HLS (Hue, Lightness, Saturation)</i>	<i>23</i>
2.5 CONSIDERAÇÕES FINAIS.....	24
3. UTILIZAÇÃO DE REGRAS DE PERCEPÇÃO VISUAL NO PROCESSO DE VISUALIZAÇÃO	25
3.1 CONSIDERAÇÕES INICIAIS	25
3.2 O PROCESSO DE VISUALIZAÇÃO.....	26
3.3 ABORDAGENS PARA AUXILIAR O PROCESSO DE VISUALIZAÇÃO	30
3.4 A ABORDAGEM BASEADA EM REGRAS – INCORPORAÇÃO DE ASPECTOS DE PERCEPÇÃO NO PROCESSO DE SELEÇÃO DE MAPAS DE CORES.....	37
<i>A Importância da Frequência Espacial.....</i>	<i>41</i>
3.5 CONSIDERAÇÕES FINAIS.....	43
4. IMPLEMENTAÇÃO DO MAPEAMENTO POR CORES BASEADO EM REGRAS.....	44
4.1 CONSIDERAÇÕES INICIAIS	44
4.2 O VISUALIZATION TOOLKIT – CONCEITOS BÁSICOS	44
4.3 MAPEAMENTO POR CORES	46
4.4 MAPEAMENTO POR CORES NO VTK - CLASSE vtkLOOKUPTABLE	48
4.5 MÓDULO DE IMPLEMENTAÇÃO DO MAPEAMENTO POR CORES BASEADO EM REGRAS – CLASSE vtkRULELOOKUPTABLE	50
4.6 MÓDULO DE IMPLEMENTAÇÃO DO CÁLCULO DA FREQUÊNCIA ESPACIAL – CLASSE SPATIALFREQUENCY	57
4.7 INCORPORAÇÃO DO CÓDIGO AO VTK.....	58
4.8 LIMITAÇÕES DA IMPLEMENTAÇÃO.....	59
4.9 CONSIDERAÇÕES FINAIS.....	60
5. TESTES	61
5.1 CONSIDERAÇÕES INICIAIS	61
5.2 CONVERSOR GIF-VTK	61
<i>O Formato dos Arquivos de Dados do VTK.....</i>	<i>62</i>
5.3 CASOS DE TESTE E RESULTADOS.....	64
<i>Caso 1 – Dados de Tomografia Computadorizada</i>	<i>64</i>
<i>Caso 2 – Dados de Sensoriamento Remoto.....</i>	<i>66</i>
<i>Caso 3 – Dados de Topografia.....</i>	<i>69</i>
<i>Caso 4 – Dados Marítimos.....</i>	<i>71</i>
<i>Caso 5 – Dados de Neurônios.....</i>	<i>72</i>
5.4 CONSIDERAÇÕES FINAIS.....	75
6. CONCLUSÕES	76
6.1 CONSIDERAÇÕES INICIAIS	76
6.2 CONTRIBUIÇÕES, LIMITAÇÕES E SUGESTÕES DE TRABALHOS FUTUROS	76
7. BIBLIOGRAFIA	80

ÍNDICE DE FIGURAS

FIGURA 2.1 - O OLHO HUMANO [FOR94]	6
FIGURA 2.2 - A ESTRUTURA DO SISTEMA VISUAL HUMANO [GRO94].....	7
FIGURA 2.3 - CONTRASTE SUCESSIVO [GRO94].....	10
FIGURA 2.4 - CONTRASTE SIMULTÂNEO [GRO94]	10
FIGURA 2.5 - ESPECTRO ELETROMAGNÉTICO [GRO94]	12
FIGURA 2.6 - DIAGRAMA DE UM MODELO DE VISÃO DE COR EM 2 PASSOS [GRO94].....	14
FIGURA 2.7 - ILUSTRAÇÃO DA MISTURA DE CORES ADITIVAS [FOR94]	17
FIGURA 2.8 - ILUSTRAÇÃO DA MISTURA DE CORES SUBTRATIVAS [FOR94]	17
FIGURA 2.9 - ILUSTRAÇÃO DA OBTENÇÃO DE <i>TINTS</i> , <i>SHADES</i> E <i>TONES</i>	18
FIGURA 2.10 - DIAGRAMA DE CROMATICIDADE DO CIE [HEA94].....	19
FIGURA 2.11 - REPRESENTAÇÃO DE ESCALAS DE COR NO DIAGRAMA DE CROMATICIDADE DO CIE [HEA94].....	20
FIGURA 2.12 - CUBO DO RGB [FOL96].....	21
FIGURA 2.13 - CONE HEXAGONAL DO HSV [FOL96].....	22
FIGURA 2.14 - CONE DUPLO DO HLS [FOL96].....	23
FIGURA 3.1 - O CENÁRIO DA VISUALIZAÇÃO CIENTÍFICA [ROB94].....	26
FIGURA 3.2 - O PROCESSO INTERATIVO E ITERATIVO DE VISUALIZAÇÃO [ROG93A].....	27
FIGURA 3.3 - A EVOLUÇÃO DOS SISTEMAS DE VISUALIZAÇÃO [ROG93A]	27
FIGURA 3.4 - ARQUITETURA DOS SISTEMAS DE VISUALIZAÇÃO MODERNOS [ROG93A].....	28
FIGURA 3.5 - ARQUITETURA DOS SISTEMAS DE VISUALIZAÇÃO MODERNOS COM A INCLUSÃO DE REGRAS [ROG93A]	31
FIGURA 3.6 - CATEGORIAS DE CODIFICAÇÃO DE VARIÁVEIS [SEN94]	35
FIGURA 3.7 - DADOS DE ALTA E BAIXA FREQUÊNCIAS ESPACIAIS [ROG96]	41
FIGURA 3.8 - TABELAS DE CORES PARA SEGMENTAÇÃO DE DADOS DE BAIXA E ALTA FREQUÊNCIAS ESPACIAIS [ROG96]	42
FIGURA 4.1 - O <i>PIPELINE</i> DE VISUALIZAÇÃO	45
FIGURA 4.2 - MAPEAMENTO DE VALORES ESCALARES ATRAVÉS DE TABELAS DE CORES	46
FIGURA 4.3 - APLICAÇÃO DE TABELAS DE CORES DIFERENTES SOBRE UM MESMO CONJUNTO DE DADOS [ROG96].....	47
FIGURA 4.4 - TABELAS DE CORES GERADAS PELO VTK.....	49
FIGURA 4.5 - ESQUEMA DA CLASSE <i>VTKRuleLookupTable</i> (COM OS PRINCIPAIS MÉTODOS E ATRIBUTOS)	50
FIGURA 4.6 - ILUSTRAÇÃO DO EFEITO DA UTILIZAÇÃO DE UM FILTRO SOBRE UMA MATRIZ	57
FIGURA 4.7 - ESQUEMA DA CLASSE <i>SpatialFrequency</i> (COM OS PRINCIPAIS MÉTODOS E ATRIBUTOS)	57
FIGURA 5.1 - EXEMPLOS DE TIPOS DE CÉLULAS [SCH96].....	63
FIGURA 5.2 - EXEMPLOS DE ATRIBUTOS DE DADOS [SCH96].....	63
FIGURA 5.3 - EXEMPLOS DE TIPOS DE DATASETS [SCH96]	63
FIGURA 5.4 - REPRODUÇÃO DA IMAGEM ORIGINAL UTILIZANDO UMA ESCALA DE CINZAS	65
FIGURA 5.5 - APLICAÇÃO DE DIFERENTES TABELAS DE CORES A DADOS DE TOMOGRAFIA COMPUTADORIZADA.....	65
FIGURA 5.6 - SEQUÊNCIA DE IMAGENS DE SENSORIAMENTO REMOTO.....	67
FIGURA 5.7 - APLICAÇÃO DE DIFERENTES TABELAS DE CORES A DADOS DE SENSORIAMENTO REMOTO	68
FIGURA 5.8- APLICAÇÃO DE DIFERENTES TABELAS DE CORES A DADOS TOPOGRÁFICOS	70
FIGURA 5.9 - APLICAÇÃO DE DIFERENTES TABELAS DE CORES A DADOS MARÍTIMOS	71
FIGURA 5.10 - VISUALIZAÇÃO DE NEURÔNIO UTILIZANDO UMA ESCALA DE CINZAS	73
FIGURA 5.11 - APLICAÇÃO DE DIFERENTES TABELAS DE CORES A DADOS DE NEURÔNIOS.....	74

ÍNDICE DE TABELAS

TABELA 3.1 - REGRAS PARA TAREFAS DE EXPLORAÇÃO E APESSENTAÇÃO [ROG93A]	32
TABELA 3.2 - CLASSIFICAÇÃO DE MAPAS DE CORES UTILIZADA NA FERRAMENTA PRAVDACOLOR [BER95]	40
TABELA 4.1 – CLASSIFICAÇÃO DE TABELAS PARA A TAREFA DE ISOMORFISMO	52
TABELA 4.2 – CLASSIFICAÇÃO DE TABELAS PARA A TAREFA DE SEGMENTAÇÃO	54
TABELA 4.3 – CLASSIFICAÇÃO DE TABELAS PARA A TAREFA DE ENFATIZAÇÃO	56

1. INTRODUÇÃO

1.1 Considerações Iniciais

A Visualização Científica (ViSC) é uma área que evoluiu a partir da Computação Gráfica, e que consiste do estudo de técnicas gráficas capazes de apoiar o processo de interpretação de grandes volumes de dados gerados através de dispositivos de detecção e de coleta de dados, ou a partir de simulações computacionais [Bro92]. Grinstein [Gri95] define ViSC como sendo o mapeamento de representações computacionais em representações perceptuais utilizando técnicas de codificação que maximizem a compreensão e a comunicação humana. Ou seja, os dados ou informações são transformados em imagens cuja compreensão envolve a utilização tanto da visão como da capacidade de processamento do cérebro.

Os sistemas de visualização oferecem técnicas para fazer o mapeamento de características dos dados em atributos visuais de objetos gráficos e permitem a interação direta com a imagem produzida. O mapeamento tem por objetivo ajudar cientistas e outros profissionais a capturar o significado dos seus dados e entender os fenômenos que geram esses dados. A interação direta permite a extração de informações contidas em descrições numéricas de fenômenos complexos. Tais sistemas são usados como ferramentas de análise de informação e também para comunicação de resultados com colegas, instituições financiadoras e o público em geral [CG87; Elv94].

O tratamento de dados complexos utilizando técnicas de visualização é considerado por pesquisadores de renome na área como um campo ainda em sua infância [Cra94]. Parte das deficiências das atuais técnicas, relacionadas à representação, apresentação e interpretação adequadas de dados, são consequência da complexidade dos dados envolvidos (particularmente no caso de dados multivariados), e da dificuldade de mapear graficamente esses dados para dispositivos bidimensionais (como a tela do computador), sem a decorrente perda de

informações [Kel93; Wol93]. O tratamento desses problemas requer pesquisas sobre mecanismos eficientes de comunicação de informações - considerando inclusive o uso de meios de representação alternativos, como o som - e também sobre aspectos relacionados à percepção visual e sonora pelos seres humanos [Min95].

Outro problema associado ao uso dos atuais sistemas e técnicas de visualização é a falta de orientação aos usuários quanto ao melhor mapeamento de atributos dos dados em representações visuais. A grande variedade de representações visuais existentes dificulta a criação de uma imagem que realmente ajude a analisar o fenômeno subjacente aos dados. Por isso, existe um grande interesse no desenvolvimento de ferramentas que ajudem a escolher uma representação visual de dados que seja realmente efetiva.

1.2 Motivação

A utilização de técnicas de codificação aliadas a um conhecimento sobre o processo de percepção visual humana pode possibilitar e facilitar a compreensão do comportamento dos fenômenos que geram os dados. Atualmente estão disponíveis inúmeras técnicas de visualização, em geral classificadas primariamente em relação ao tipo de dado que permitem visualizar (escalar, vetorial e tensorial) [Bro92].

Uma técnica de visualização clássica consiste na seleção de uma tabela de cores que define um mapeamento dos valores dos dados em cores. A escolha de mapeamentos de cores adequados é particularmente importante no apoio aos objetivos de visualização, assim como a necessidade de se fornecer mecanismos capazes de suportar a geração interativa e iterativa de visualizações alternativas. Uma tabela de cores adequada deve evidenciar as características relevantes do conjunto de dados a ser estudado, de modo a efetivamente ajudar o usuário a entender a estrutura dos dados, segmentar os dados de forma lógica e compreensível, e enfatizar características importantes dos mesmos [Ber95]. Em geral, os sistemas de visualização oferecem ao usuário uma tabela de cores *default* (normalmente a escala *rainbow*) e permitem que o usuário defina suas próprias tabelas, mas não oferecem ajuda sobre como escolher ou gerar uma tabela adequada. Por isso, é comum o uso de mapeamentos para cores que não levam em consideração aspectos de percepção visual e as características dos dados.

O *PowerVis* [Min97] é um sistema de visualização multi-modal que está sendo desenvolvido no ICMC-USP e que incorporará representações gráficas e sonoras dos dados. Esse sistema deve incorporar estratégias para facilitar a geração de visualizações e ajudar o usuário a selecionar a melhor representação para os seus dados. Essas estratégias levarão em consideração aspectos de percepção e análise da tarefa a ser desempenhada para orientar o usuário na exploração do seu conjunto de dados.

Este trabalho abrange um estudo de aspectos de percepção visual de cores, de como esses aspectos têm sido aplicados no contexto da visualização de dados e da implementação de uma abordagem em particular, a visualização baseada em regras. Essa implementação abrange uma técnica de visualização em particular, o mapeamento por cores, e utiliza aspectos de percepção visual de cores e informações sobre os dados (como o tipo e a frequência espacial) e sobre a tarefa de visualização, embutidas em regras heurísticas que auxiliam o usuário a gerar tabelas de cores adequadas aos dados e à tarefa em questão. As regras operam restringindo o número e o conjunto de cores que podem estar contidos em uma tabela. Foi considerada especificamente a visualização de dados escalares do tipo razão ou intervalo, definidos sobre uma malha regular bidimensional ou tridimensional. Nos testes realizados, foram utilizadas técnicas de visualização de superfícies e volumétricas combinadas com a definição de uma tabela de cores para o mapeamento dos valores dos dados (ou de conjuntos de valores de interesse) em cores.

1.3 Organização do Trabalho

Neste capítulo foram feitas algumas considerações gerais e introduzidas as bases que motivaram o desenvolvimento do presente trabalho.

No capítulo 2 são apresentados alguns aspectos da percepção visual humana relacionados à percepção de cores, incluindo definições de alguns termos relacionados à cor e discussões de alguns sistemas de representação de cor utilizados em Computação Gráfica e em Visualização.

No capítulo 3 é feita uma revisão bibliográfica sobre o processo de geração de visualizações e sobre abordagens descritas na literatura que tentam apoiar os usuários finais nesse processo. Esse capítulo descreve detalhadamente a abordagem baseada em regras para apoio à geração de visualizações, que serviu de base para o desenvolvimento do trabalho.

No capítulo 4 é apresentada uma descrição da implementação do módulo de mapeamento por cores baseado em regras e o processo de integração do mesmo ao *Visualization Toolkit* (VTK), que é a plataforma de software utilizada para o desenvolvimento deste trabalho.

No capítulo 5 são apresentados os resultados obtidos com a aplicação dos diferentes tipos de tabelas de cores geradas pelo módulo de mapeamento por cores baseado em regras a alguns conjuntos de dados.

O capítulo 6 apresenta as conclusões, as contribuições e sugestões para trabalhos futuros. Os protótipos das classes implementadas estão inseridos ao final desta dissertação, no Apêndice A.

2. PERCEPÇÃO VISUAL

2.1 Considerações Iniciais

O ato de ver parece uma ação que não requer esforço, mas essa impressão se deve ao fato de que não é preciso estar consciente de todas as tarefas que o sistema visual realiza para transformar as informações visuais vindas do mundo externo em imagens [Wal84]. Essas informações podem se manifestar através de formas, cores e padrões espaciais.

As imagens utilizam toda a capacidade de processamento do sistema visual humano. Elas produzem informações visuais que são captadas e transformadas em estímulos que permitem a reprodução da imagem original. A compreensão dos aspectos relacionados a esse processamento é essencial para a criação de uma visualização bem sucedida. Essas informações ajudam tanto na exploração das qualidades do sistema visual humano como na conscientização das limitações inerentes a ele.

Este capítulo apresenta uma descrição do sistema visual humano e do seu funcionamento na seção 2.2. Aspectos relacionados à percepção de cores são discutidos na seção 2.3. Alguns sistemas de representação de cores adotados em Computação Gráfica são apresentados na seção 2.4.

2.2 O Sistema Visual Humano

O conteúdo desta seção, que trata da fisiologia e características básicas do sistema visual humano, foi baseado principalmente no estudo das referências [Mur84], [Gro94], [Liv88b], e [Air85].

As informações visuais são obtidas a partir de imagens reais e podem estar distorcidas ou, até mesmo, ser irrelevantes. Por isso, essas informações são processadas e apenas aquelas que são úteis chegam ao cérebro [Wal84]. A visão é o processo de interpretação das informações pelo consciente e requer manipulações da imagem que vão desde a sua projeção na lente do olho até o processamento feito pelo cérebro [Pog78].

Do ponto de vista ótico, pode-se fazer uma analogia entre o olho e uma câmera escura na qual a luz incide sobre a lente, entra por um orifício anterior (pupila) e é focalizada na superfície interna posterior (retina). O olho humano é revestido por uma superfície escura muito vascularizada e pigmentada, denominada coróide, que absorve grande parte da luz que entra no olho, evitando assim que ela se espalhe muito.

Figura 2.1 - O olho humano [For94]

O sistema visual humano é formado pelo olho e pelas regiões do cérebro que sentem, transmitem ou interpretam as informações visuais que chegam ao olho. A figura 2.1 ilustra o olho, que tem como componentes principais a lente, a retina, o nervo ótico, a fóvea e a íris [For94; Fri91]. A lente tem a função de focalizar uma imagem sobre a retina, e essa focalização se dá através da alteração do seu formato (curvatura). O nervo ótico transmite os sinais da retina até o cérebro. A pupila e a íris ajudam a controlar a quantidade de luz que entra no olho. A fóvea é o ponto da retina onde as imagens são formadas com maior nitidez.

A retina é a parte sensível do olho que contém os foto-receptores, chamados de cones e bastonetes, responsáveis pela absorção da luz que incide sobre ela. Os bastonetes não são sensíveis a cores e detectam níveis de luz muito baixos, sendo responsáveis pela visão noturna. Os cones são sensíveis a cores e são responsáveis pela visão diurna. Existem 3 tipos de cones, sendo que um é mais sensível a comprimentos de onda curtos (luzes verde e azul); outro é mais

sensível a comprimentos de onda médios (luz verde) e o terceiro é mais sensível a comprimentos de onda longos (luzes amarela, alaranjada e vermelha).

Antes de chegar ao cérebro, os sinais elétricos produzidos pelas células foto-receptoras são processados por neurônios e em seguida transmitidos para as células ganglionares, que formam uma camada mais interna de células retinais. Essa camada possui 2 tipos de células que estão misturadas, e que são diferenciadas pelo tamanho (existem células pequenas e grandes) e pelo modo com que processam as informações recebidas. As células grandes fazem a adição dos sinais recebidos sem identificar os tipos de cones que os enviaram. As células pequenas identificam os tipos de cones que enviaram os sinais e fazem a subtração dos mesmos, produzindo a informação de cor. Uma cor é identificada através da comparação dos sinais produzidos pelos 3 tipos de cones. Quando os sinais produzidos pelos cones vermelhos são mais fortes do que os sinais produzidos pelos cones verdes e azuis, tem-se a sensação de cor vermelha; quando os sinais produzidos pelos cones vermelhos e verdes possuem a mesma intensidade, tem-se a sensação de cor amarela. Uma célula ganglionar R-G, por exemplo, produz resposta apenas para sinais de luz vermelha, embora possa receber informações vindas de vários tipos de cones. Com isso, os sinais produzidos pelas células ganglionares pequenas possuem informações sobre cor que são mais expressivas do que os sinais produzidos pelos cones.

Figura 2.2 - A estrutura do sistema visual humano [Gro94]

A figura 2.2 ilustra a estrutura do sistema visual humano. Os sinais produzidos pelas células ganglionares são transmitidos do olho até o corpo geniculado lateral (CGL) através do nervo ótico. O CGL possui 2 sistemas espacialmente separados: o sistema parvo celular, formado por células pequenas, e o sistema magno celular, formado por células grandes. As células do

sistema parvo, que recebem informações das células ganglionares pequenas, fazem a distinção de cores independentemente do brilho relativo entre as mesmas. As células do sistema magno, que recebem informações das células ganglionares grandes, produzem informações relativas ao brilho de uma superfície.

Os sistemas magno e parvo são, posteriormente, reorganizados em 3 subdivisões: os chamados *interblobs* na Área Visual 1 do córtex visual, que recebem sinais do sistema parvo; a camada 4B também localizada na Área Visual 1, que recebe sinais do sistema magno; e os *blobs* também localizados na Área Visual 1, que recebem sinais dos dois sistemas. Os *blobs* contêm células altamente especializadas na percepção de cor ou brilho; os *interblobs* são especializados na percepção de orientação; e as células da camada 4B são especializadas na percepção de orientação e de movimento. Por exemplo, existem células *interblobs* que produzem resposta apenas para estímulos produzidos, por exemplo, por uma barra vertical, sem serem influenciadas pelo movimento ou pela cor da barra e não produzem sinais para barras em outras orientações. Existem células da camada 4B, por exemplo, que produzem resposta para barras horizontais que se movem para cima ou para barras verticais que se movem horizontalmente, mas não para ambas.

Após ser processada pelo CGL, a informação visual é transmitida para a Área Visual 1, que corresponde à superfície mais externa do cérebro. Os sinais produzidos pelo sistema magno são transmitidos para a parte superior dessa camada e os sinais produzidos pelo sistema parvo são transmitidos para a parte inferior. A Área Visual 1 processa e transmite as informações para uma região adjacente denominada Área Visual 2, que também está subdividida em 3 áreas: *pale stripes*, *thin stripes* e *thick stripes*. As *thin stripes* recebem informações dos *blobs* dando seqüência ao processamento de cor; as *pale stripes* recebem informações dos *interblobs* e, aparentemente, processam informações associadas à forma; e as *thick stripes* recebem informações do sistema magno e analisam a informação estereoscópica.

Segundo Livingstone [Liv88b], sabe-se que as *thick stripes* se estendem até a Área Temporal Média (MT), que está envolvida na detecção de movimento e estereopse (habilidade de julgar a sensação de profundidade tendo por base a diferença das imagens dos 2 olhos); que as *thin stripes* se estendem até a Área Visual 4, que está envolvida no processamento de cor; mas não se sabe até onde as *pale stripes* se estendem.

As funções das 3 subdivisões do sistema visual humano podem ser resumidas da seguinte forma [Liv88b]:

- O sistema definido pela sequência *parvo-interblob-pale-stripe* transmite informações de alta resolução a respeito de bordas definidas pelo contraste de matiz, mas não transmite informações relativas aos matizes que as definem. As bordas podem fornecer muitas informações sobre a forma de objetos e, por isso, esse sistema é considerado importante na percepção da forma de objetos. É provável que o tempo de resposta lento e a alta resolução desse sistema contribuam para a habilidade de distinguir detalhes em objetos estacionários.
- O sistema formado pela sequência *blob-thick-stripe-V4* processa informações sobre cores e tons de cinza. Esse sistema possui acuidade visual menor do que o sistema que inclui a subdivisão *interblob* e, portanto, apesar de identificar cores de objetos, não distingue detalhes de objetos coloridos.
- O sistema formado pela sequência *magno-4B-thick-stripe* transmite informações sobre movimento e profundidade estereoscópica. Os neurônios desse sistema produzem respostas rápidas, mas as respostas decaem rapidamente mesmo quando o estímulo é mantido.

Existe uma razão de brilho para a qual quaisquer pares de cores aparecem com a mesma tonalidade (*shade*) de cinza em uma foto em preto e branco. Da mesma forma, existe uma razão de brilho, por exemplo, entre as cores vermelha e verde, que faz com que essas cores pareçam idênticas ao sistema magno. Essa razão de brilho varia de pessoa para pessoa, assim como a sensibilidade de filmes em preto e branco varia para certas cores. Duas cores que possuem o mesmo brilho são ditas equiluminantes, e uma borda entre as mesmas é definida apenas pela diferença de matizes.

A sensibilidade visual, que corresponde à intensidade mínima necessária para excitar um foto-receptor, depende da iluminação do ambiente e do comprimento de onda da luz utilizada na iluminação. Essa intensidade mínima é mais alta no centro do espectro eletromagnético (ver figura 2.5 na seção 2.3); por isso, para que as cores verde ou amarela sejam percebidas elas devem ser mais intensas do que as cores azul e vermelha.

Os cones e os bastonetes ajustam a sua sensibilidade de acordo com o nível de iluminação global, e a mudança de um ambiente escuro para um ambiente claro requer um tempo de adaptação. A sensação de desconforto provocada por essa mudança repentina vai diminuindo até que ocorra a adaptação ao claro. A adaptação ao escuro ocorre quando a percepção de uma

pessoa atinge seu máximo num ambiente escuro, e esse processo é muito mais lento do que a adaptação ao claro.

A visão de objetos pelo sistema visual humano se dá através da detecção de bordas, que podem ser formadas pela variação das intensidades de cor, ou de brilho¹ (*brightness*), ou de ambos, nas regiões adjacentes. As bordas que são formadas apenas pela diferença de cor parecem ser imprecisas e desfocadas. Portanto, para que as bordas resultantes sejam bem definidas é necessário adicionar também diferença de brilho.

A percepção de objetos é muito influenciada pelas regiões de teste (*patches*), pelo tamanho do objeto e pelo contraste, que dá uma medida da diferença de luminância² relativa entre duas regiões adjacentes. A sensibilidade ao contraste e o processo de adaptação podem causar algumas ilusões de ótica, como as ilustradas nas figuras 2.3 e 2.4.

Figura 2.3 - Contraste sucessivo [Gro94]

Na figura 2.3, após olhar o objeto à esquerda por 30 segundos, e depois o objeto à direita, pode-se perceber a persistência de imagem, que desaparece de tempos em tempos. O efeito de persistência de imagem é causado por objetos muito contrastantes observados por um longo período de tempo. Isso ocorre porque as regiões da retina cobertas pelo objeto da esquerda estão mais sensíveis e adaptadas a um nível de luz mais baixo [Gro94].

Figura 2.4 - Contraste simultâneo [Gro94]

¹Brilho (*brightness*): termo usado para denotar a intensidade luminosa percebida por um observador, em um sentido psicológico [Fol96].

²Luminância: termo usado para indicar a quantidade de energia luminosa emitida por uma fonte ou refletida por uma superfície não emissora em um sentido físico. Intensidade luminosa é outro termo usado nesse contexto [Fol96].

A figura 2.4 mostra o efeito de contraste simultâneo, que é causado pelas cores das regiões (*patches*) adjacentes. Ao fixar os olhos na cruz que se encontra no centro da imagem, tem-se a impressão de que o disco cinza é mais brilhante com o fundo preto do que com o fundo branco. Isso ocorre porque a adaptação local em cada posição respectiva da retina é diferente.

Existe, ainda, o efeito denominado “*Mach Band*”, causado por duas regiões adjacentes com luminâncias diferentes (ver ilustração em Groß [Gro94], página 54). Esse efeito ocorre porque, embora a luminância seja preservada, o brilho percebido muda ao longo da superfície da região, fazendo com que as bordas pareçam ser mais escuras ou mais claras.

O contraste entre duas regiões pode ser reduzido pelo aumento de iluminação, causando um ofuscamento (*glare*) da imagem que se baseia nos fenômenos físicos de difração e de espalhamento. O ofuscamento pode ser descrito como uma superposição da luminância de adaptação com a luz espalhada, que faz com que partes da retina tenham um nível de adaptação maior.

A acuidade visual é a nitidez com que os objetos são vistos e pode ser medida pela capacidade de se distinguir linhas paralelas ou pontos muito próximos. Pode ser determinada por diversas técnicas, como a apresentação de círculos progressivamente menores ou pares de linhas progressivamente mais próximas, ou através do Anel de Landolt (ver Aires [Air85] e Groß [Gro94] para maiores detalhes). A acuidade visual é máxima para as imagens formadas na fóvea porque ela possui maior concentração de cones do que as outras regiões da retina e está exposta diretamente à luz; além disso, cada cone está ligado diretamente ao sistema nervoso central.

É preciso estar atento para a diferença entre sensibilidade visual e acuidade visual. A sensibilidade é maior para os bastonetes, tanto que esses são os responsáveis pela visão no escuro, quando a iluminação é baixa. Já a acuidade visual, capacidade de resolução espacial, é maior para os cones.

2.3 Percepção de Cor

A cor é o atributo da percepção visual que pode ser descrito através dos nomes usados para identificar as cores, como branco, cinza, preto, amarelo, etc., ou da combinação delas [Hun78]. As diferentes cores, ou espectros luminosos, que podem ser percebidos pelo

sistema visual humano correspondem a uma pequena faixa de frequências do espectro eletromagnético, que inclui as ondas de rádio, microondas, os raios infravermelhos e os raios X, como mostrado na figura 2.5.

Figura 2.5 – Espectro Eletromagnético [Gro94]

A frequência mais baixa do espectro visível corresponde à cor vermelha (4.3×10^{14} hertz) e a mais alta à cor violeta (7.5×10^{14} hertz). Os valores de frequência intermediários correspondem a cores que passam pelo alaranjado e amarelo e por todas as outras cores, até chegar nos verdes e azuis. As cores são ondas eletromagnéticas descritas pelo seu comprimento de onda (λ) e especificadas, tipicamente, em nanômetros (nm).

Os comprimentos de onda maiores possuem distâncias focais maiores e, conseqüentemente, requerem maior curvatura da lente do olho para serem focalizados, (a cor vermelha possui a maior distância focal e a azul, a menor). A utilização simultânea de cores localizadas em extremos opostos do espectro fazem com que a lente altere o seu formato constantemente, causando cansaço no olho. A cromostereopse (visão estereoscópica da cor) é um efeito relacionado ao processo de focalização da imagem na retina, que faz com que cores puras localizadas à mesma distância do olho pareçam estar a distâncias diferentes; por exemplo, a cor vermelha parece estar mais próxima e a cor azul parece estar mais distante.

Uma fonte de luz (como, por exemplo, o sol ou uma lâmpada) emite em todas as frequências do espectro visível, produzindo a luz branca que incide sobre um objeto. Parte dessa luz é absorvida e a outra parte é refletida, determinando a cor resultante do objeto. Quando há predominância das frequências baixas, diz-se que o objeto é vermelho, ou que a luz percebida possui uma frequência dominante (ou um comprimento de onda dominante) na frequência baixa do espectro. A frequência dominante também é chamada de matiz ou, simplesmente, de cor da luz.

O matiz é o atributo de uma sensação visual que faz com que uma área pareça ser similar a uma ou 2 das cores percebidas vermelha, amarela, laranja, azul, púrpura [Hun78]. A partir

dessa definição, pode-se diferenciar a cor cromática, que possui matiz, da acromática, que é desprovida de matiz.

As características da luz são descritas através de propriedades como o matiz e as sensações de brilho e saturação. O matiz é usado para dar um nome a uma cor, o brilho corresponde ao grau de luminância de uma cor em relação à luminância de outra ou em relação ao fundo, e a saturação é a pureza aparente de um matiz. Quanto maior o domínio de um comprimento de onda, maior é a sua saturação. As cores preta, branca e cinza possuem saturação uniforme em todos os comprimentos de onda e, por isso, são diferenciadas apenas pelo brilho. As propriedades de saturação e de matiz de uma cor são referenciadas como cromaticidade.

O Conceito da Trivariância, atribuído a Thomas Young, supõe que o olho possui sensibilidades diferentes para as 3 cores primárias para explicar o processo de percepção de cores. Young achava improvável que a retina fosse igualmente sensível a todas as partes do espectro visível e propôs a existência de 3 tipos de partículas sensíveis à luz. As cores resultariam então, de uma combinação de respostas das partículas sensíveis às cores vermelha, verde e azul, sendo que essas respostas seriam independentes umas das outras. Embora não exista nenhum conjunto de 3 cores que reproduza todas as cores possíveis, essa idéia ainda é fundamental para a teoria da cor. Herman von Helmholtz refinou essa idéia, o que resultou na descoberta dos 3 tipos receptores contidos no olho (Teoria Young-Helmholtz). O fato da percepção de cores estar fundamentada na combinação das respostas dos 3 tipos de receptores explica por que não é possível distinguir entre um comprimento de onda puro e uma cor resultante, por exemplo, da mistura de 2 cores primárias. Segundo a Teoria Young-Helmholtz, isso se deve ao fato de que os 2 estímulos provocam respostas idênticas nos 3 tipos de receptores e, por isso, não é possível distingui-las perceptualmente. Por exemplo, a mistura das luzes verde e vermelha é percebida como sendo uma luz amarela porque elas estimulam respostas dos cones verde e vermelho, que são iguais às respostas estimuladas por um amarelo espectral [Fri91].

Embora a Teoria Young-Helmholtz seja fundamental para a ciência da cor, ela não explica certos aspectos da percepção visual de cor, como o fato do contexto das cores ao redor de outra afetarem a aparência da cor, e a idéia de estabilidade da cor sob variados tipos de iluminação. Esses aspectos despertaram o interesse de Edwin Land, que propôs a Teoria Retinex. Essa teoria sugere que o sistema visual humano é capaz de determinar a reflectância, que é o grau de reflexão de uma energia luminosa de uma substância nos 3 tipos de receptores, para atribuir um valor de cor para um objeto. Essa teoria sugere que toda a imagem é processada

separadamente em 3 canais de reflectância, e comparada com as cores conhecidas. Edwin Land descobriu ainda que a resposta dos cones é desencadeada após a absorção da luz realizada pelos pigmentos presentes nos cones e que era possível criar imagens coloridas a partir de uma grande variedade de combinações de comprimentos de onda, pares e trios de comprimentos de onda diferentes dos tradicionais [Fri91; Gro94].

A sensação de cor produzida no cérebro resulta, então, de informações aferentes vindas de 3 tipos de cones existentes na retina [Air85]. Cada tipo de cone é sensível a uma cor e possui uma função de resposta característica. Essas funções podem ser interpretadas como canais visuais, dos quais um é independente do comprimento de onda e os outros dois sinalizam a cor [Gro94].

A Teoria das Cores Oponentes explica porque cada canal visual sinaliza apenas uma cor por vez, nunca ambas simultaneamente. Essa teoria sugere a existência de 3 componentes do processamento neural que manipulam a informação de cor, que são o canal B-W, que sinaliza branco ou preto; o R-G, que sinaliza vermelho ou verde; e o B-YE, que sinaliza azul ou amarelo. Um matiz alaranjado, por exemplo, resulta da sinalização da cor vermelha pelo componente R-G e da cor amarela pelo componente B-YE, ambas com a mesma intensidade. Através dessa teoria é possível explicar mecanismos como o de saturação, que depende da luz acromática e do matiz presente na cor. O CGL e o CV comparam o sinal acromático fornecido pelo componente B-W com o sinal fornecido pelos componentes R-G e B-YE para estimar o nível de saturação [Lan85].

Figura 2.6 - Diagrama de um modelo de visão de cor em 2 passos [Gro94]

A figura 2.6 mostra o esquema híbrido para codificação de cor que consiste dos sensores tricromáticos na retina e transmissão antagônica nos canais de cor. Os cones detectam valores de

R, G e B através de filtros característicos e distribuem o estímulo espectral a partir do qual são calculadas as diferenças entre as primárias percebidas. Essa operação linear resulta em 3 canais: 1 canal acromático (B-W) e 2 canais cromáticos (R-G e B-YE). O canal acromático produz a soma dos comprimentos curto, médio e longo das ondas, dando a informação de luminância. Um dos canais cromáticos, que corresponde ao canal vermelho-verde, produz a diferença entre os comprimentos médio e longo das ondas; e outro, que corresponde ao canal amarelo-azul, produz a diferença entre o canal acromático e a onda que possui comprimento curto.

Essa teoria também explica que não é possível produzir, por exemplo, uma combinação “verde avermelhada”, porque as cores vermelha e verde estão no mesmo canal, e cada canal sinaliza apenas uma das cores por vez e nunca as duas simultaneamente.

Algumas pessoas possuem uma anomalia, denominada daltonismo, que impede a distinção de uma ou mais cores. O daltonismo se deve a um defeito na constituição dos cones e está vinculada ao sexo. Ele atinge cerca de 8% dos homens, e apenas 0.5% das mulheres. Esse defeito pode se manifestar em 1, 2 ou nos 3 tipos de receptores. Os tricromatas são daltônicos que possuem os 3 sistemas de pigmentos, mas que utilizam os sistemas em proporções diferentes das pessoas normais e das pessoas com o mesmo defeito. Os dicromatas percebem as cores defeituosamente porque combinam apenas 2 sistemas. Os monocromatas percebem apenas graduações de claro e de escuro, pois sua estimulação visual se baseia em um único sistema cromático. Uma abordagem mais aprofundada sobre esse problema pode ser encontrada em Meyer [Mey88].

Um outro aspecto que também influencia na percepção de cores é o amarelamento das lentes do olho que ocorre com o passar dos anos, fazendo com que as pessoas se tornem menos sensíveis, por exemplo à cor amarela do que à cor azul.

2.4 Sistemas de Cores Primárias

O conteúdo desta seção, que descreve alguns sistemas de cores primárias, foi baseado principalmente em Hearn [Hea94], Capítulo 15.

Um sistema de cores é um método que explica as propriedades ou o comportamento das cores num contexto particular. Não existe um sistema que explique todos os aspectos relacionados à cor. Por isso, são utilizados sistemas diferentes para ajudar a descrever as

diferentes características da cor que são percebidas pelo ser humano. Existem vários sistemas de cores (ver Hearn [Hea94] e Foley [Fol96]), sendo que nesta seção serão apresentados apenas alguns dos principais: o XYZ, o RGB, o HSV e o HLS.

As cores primárias são as 2 ou 3 cores que um sistema utiliza para produzir outras cores. As cores podem ser produzidas a partir de uma combinação das primárias, ou então, da composição de 2 combinações. O universo de cores que podem ser reproduzidas por um sistema é chamado de espaço de cores (*color space* ou *color gamut*). Alternativamente, um espaço de cores (*color space*) pode ser definido como uma representação visual de um modelo de cores, como o cubo definido pelas componentes do modelo RGB, ou o cone definido pelo modelo HSV. Não existe um conjunto finito de cores primárias que produza todas as cores visíveis, mas sabe-se que uma grande parte delas pode ser produzida a partir de 3 primárias.

O estudo da utilização de 3 fontes de luz espectral para a geração de cores é chamado de colorimetria, e tem como um de seus objetivos determinar espaços de cor perceptualmente uniformes. Um espaço de cores (ou sistema de cores) perceptualmente uniforme é um no qual distâncias perceptuais iguais separam todas as cores. Por exemplo, a escala de cinzas do espaço deve transmitir uma transição suave entre o preto e o branco [Mey80]. A definição de um espaço de cores uniforme é feita através de medições empíricas obtidas sob condições experimentais rigidamente controladas - as condições do ambiente e outros parâmetros importantes devem ser mantidos constantes, como o tamanho das amostras de cores, o espaçamento entre as amostras, a luminância e cromaticidade do fundo e da luz ambiente. Apesar dessa limitação, os espaços perceptuais de cores fornecem ferramentas adequadas para a solução de problemas como a compressão de imagens (para decidir o nível de codificação da informação de cor) e pseudo-coloração³ (para mapear as cores da imagem em um conjunto com espaçamento perceptual máximo) [Gro94; Mey80; Mey86].

Os sistemas de cores podem ser aditivos ou subtrativos. Nos modelos aditivos (por exemplo, RGB e XYZ), as intensidades das cores primárias são adicionadas para produzir outras cores. A figura 2.7 ilustra a demonstração do funcionamento desses modelos através da sobreposição de círculos coloridos.

³A coloração “falsa” (“*pseudocoloring*”) ocorre quando as cores “verdadeiras” de uma imagem são mapeadas em outro conjunto de cores, ou quando uma imagem adquirida em tons de cinza é colorida.

Figura 2.7 - Ilustração da mistura de cores aditivas [For94]

Pode-se pensar que o branco é a mistura das intensidades máximas das 3 cores primárias aditivas (vermelha, verde e azul). Os matizes intermediários (amarelo, turquesa e magenta) são obtidos através da combinação das intensidades máximas de 2 cores.

Nos modelos subtrativos (por exemplo, o CMY [Hea94; Fol96]), as cores são geradas subtraindo-se o comprimento da onda dominante da luz branca, por isso, a cor resultante corresponde à luz que é refletida. A figura 2.8 ilustra a demonstração do funcionamento desses modelos através da sobreposição de círculos coloridos.

Figura 2.8 - Ilustração da mistura de cores subtrativas [For94]

Pode-se pensar que o preto é a combinação das 3 cores subtrativas (turquesa, magenta e amarela). A quantidade de preto numa cor é indicada pela diferença entre o branco e a intensidade máxima das 3 cores primárias aditivas. E, da mesma forma, a quantidade de branco numa cor é indicada pela diferença entre o preto e a intensidade mínima das 3 cores primárias aditivas.

As cores puras e saturadas não representam toda a classe de cores possíveis, existem ainda os *tints*, *shades* e *tones* que correspondem, respectivamente, às cores obtidas através da adição de branco, preto e cinza às cores saturadas, causando uma alteração no efeito da cor. A figura 2.9 mostra a ilustração⁴ de *tints*, *shades* e *tones* obtidos a partir da cor vermelha.

⁴ reproduzida de <http://www.contrib.andrew.cmu.edu:8001/usr/dw4c/color/tints.html>

Figura 2.9 - Ilustração da obtenção de *tints*, *shades* e *tones*

A adição de branco clareia uma cor e cria um *tint* (por exemplo, adicionando branco ao vermelho para obter a cor *pink*). A adição de preto escurece uma cor e cria um *shade* (por exemplo, adicionando preto ao vermelho para obter um castanho-avermelhado (*maroon*)). Além disso, a adição de cinza reduz o brilho de uma cor e cria um *tone*. Uma composição monocromática é formada inteiramente de *tints*, *shades* e *tones* da mesma cor.

Modelo XYZ

O sistema XYZ de cores primárias da CIE (Comissão Internacional de Iluminação) é um sistema aditivo que descreve as cores através de 3 cores primárias virtuais X, Y e Z. Esse sistema foi criado devido à inexistência de um conjunto finito de cores primárias que produza todas as cores visíveis possíveis. Nesse sistema, as cores C_λ podem ser expressas pela seguinte equação:

$$C_\lambda = x\mathbf{X} + y\mathbf{Y} + z\mathbf{Z}$$

em que x, y e z especificam as quantidades das primárias padrão necessárias para descrever uma cor espectral. A normalização dessa quantidade em relação à luminância ($X + Y + Z$) possibilita a caracterização de qualquer cor. As cores desse sistema podem ser expressas pelas quantidades normalizadas abaixo:

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z}$$

com $x + y + z = 1$. Assim, qualquer cor pode ser definida através de X e Y ^z que, por dependerem apenas do matiz e da saturação, são chamadas de coordenadas de cromaticidade. Essas coordenadas descrevem a direção do raio produzido por uma fonte de luz real e podem ser aplicadas a qualquer sistema de 3 estímulos. Esse raio deve estar contido no cone de cores reais

e, por isso, deve intersectar o plano unitário $X + Y + Z = 1$. A descrição completa de uma cor é dada pelas coordenadas de cromaticidade e pelo valor de um dos 3 estímulos originais, normalmente do Y, que contém a informação de luminância. Essa descrição possibilita a obtenção das quantidades de X e Z através das equações abaixo:

$$X = \frac{x}{y} Y \quad Z = \frac{z}{y} Y$$

onde $z = 1 - x - y$.

O sistema XYZ é formado por cores imaginárias que são definidas matematicamente. Nesse sistema, as combinações de valores negativos e outros problemas relacionados à seleção de um conjunto de primárias reais são eliminados. Através das coordenadas de cromaticidade x e y é possível representar todas as cores num gráfico bidimensional. O traçado das quantidades de x e y normalizadas para as cores no espectro visível resulta na curva ilustrada na figura 2.10.

Figura 2.10 - Diagrama de cromaticidade do CIE [Hea94]

Os pontos que representam as cores puras no espectro eletromagnético são rotulados de acordo com os seus comprimentos de onda e estão localizados ao longo da curva que vai da extremidade correspondente à cor vermelha até a extremidade correspondente à cor violeta. A linha reta que une os pontos espectrais vermelho e violeta é chamada linha púrpura, e não faz parte do espectro. Os pontos internos correspondem a todas as combinações possíveis de cores visíveis, e o ponto C corresponde à posição da luz branca.

Devido à normalização, o diagrama de cromaticidade não representa os valores de luminância. Por isso, as cores com luminâncias diferentes e cromaticidades iguais são mapeadas no mesmo ponto. Através desse diagrama, é possível determinar e comparar os espaços de cores

dos diferentes conjuntos de primárias, identificar as cores complementares (2 cores que, somadas, produzem a cor branca) e determinar o comprimento de onda dominante e a saturação de uma cor.

Figura 2.11 - Representação de escalas de cor no diagrama de cromaticidade do CIE [Hea94]

Os espaços de cor são representados no diagrama, ilustrado na figura 2.11a, através de linhas retas ou de polígonos. Todas as cores ao longo da linha que une os pontos C_1 e C_2 na figura 2.11a podem ser obtidas através da mistura de quantidades apropriadas das cores correspondentes a esses pontos. A escala de cores para 3 pontos (por exemplo, C_3 , C_4 e C_5 na figura 2.11a) é representada por um triângulo cujos vértices são definidos pelas cores correspondentes às 3 posições e inclui cores contidas no interior e nas margens fronteiriças desse triângulo.

Através do diagrama é possível perceber que nenhum conjunto formado por 3 primárias pode gerar todas as cores, pois nenhum triângulo contido no diagrama abrange todas as cores possíveis. As cores complementares são identificadas por 2 pontos localizados em lados opostos do ponto C e conectados por uma linha reta. Por exemplo, misturando quantidades apropriadas de 2 cores C_1 e C_2 (figura 2.11b), obtém-se a luz branca.

A determinação do comprimento de onda dominante de uma cor pode ser feita interpretando-se a escala de cores entre 2 primárias. O comprimento de onda dominante da cor C_1 , representada na figura 2.11c, é determinado traçando-se uma linha reta que parte do ponto C passando pelo ponto C_1 e intersectando a curva espectral no ponto C_s . A cor C_1 , corresponde então, à combinação da luz branca com a cor espectral C_s , pois C_s é o comprimento de onda dominante de C_1 . O comprimento de onda dominante das cores que estão entre o ponto C e a linha púrpura é determinado de outra forma. Traça-se uma linha a partir do ponto C (figura 2.11c) passando pelo ponto C_2 e intersectando a linha púrpura no ponto C_p . Como esse ponto não pertence ao espectro visível, o ponto C_2 é referenciado como sendo uma cor não

espectral e o seu comprimento de onda dominante é obtido através do prolongamento da reta até que ela intercepte a curva espectral, no ponto C_{sp} . As cores não espectrais estão entre púrpura e magenta, e são geradas através da subtração do comprimento da onda dominante (como, por exemplo, o C_{sp}) da luz branca.

A pureza de uma cor (por exemplo, de C_1 na figura 2.11c) é determinada através da distância relativa do ponto C_1 , que corresponde à linha reta que vai do ponto C até o ponto C_s . Pode-se calcular a pureza do ponto C_1 através da relação d_{c1}/d_{cs} , onde d_{c1} representa a distância entre C e C_1 e d_{cs} representa a distância entre C e C_s . A cor C_1 é cerca de 25% pura porque está situada a aproximadamente $\frac{1}{4}$ da distância total entre C e C_s .

Modelo RGB (Red, Green, Blue)

O sistema RGB de cores primárias também é aditivo e está baseado na teoria dos 3 estímulos (*Tristimulus Color Theory*) proposta por Young-Helmholtz e discutida na seção 2.3. Segundo essa teoria, o olho humano percebe a cor através da estimulação dos 3 pigmentos visuais presentes nos cones da retina, que possuem picos de sensibilidade aproximada nos seguintes comprimentos de onda: 630 nm (Vermelho-Red), 530 nm (Verde-Green) e 450 nm (Azul-Blue). Esse sistema pode ser representado graficamente através do cubo unitário definido sobre os eixos R, G e B, como ilustrado na figura 2.12.

Figura 2.12 - Cubo do RGB [Fol96]

A origem representa a cor preta, o vértice de coordenadas (1,1,1) representa a cor branca, os vértices que estão sobre os eixos representam as cores primárias e os demais vértices representam o complemento de cada cor primária. Cada ponto no interior do cubo corresponde a uma cor que pode ser representada pela tripla (R,G,B), com os valores R, G e B variando de 0 a 1. Os tons de cinza são representados ao longo da diagonal principal do cubo, que vai da origem

(ponto correspondente a cor preta) até o vértice que corresponde à cor branca. Cada tom ao longo dessa diagonal é formado por contribuições iguais de cada primária. Logo, um tom de cinza médio entre o branco e o preto é representado por (0.5,0.5,0.5). As cores C_λ desse sistema podem ser expressas na forma:

$$C_\lambda = rR + gG + bB$$

A resposta do olho aos estímulos espectrais não é linear e, por isso, algumas cores não podem ser reproduzidas pela sobreposição das 3 primárias. Isso significa que algumas cores existentes na natureza não podem ser mostradas nesse sistema. Por isso, um fenômeno natural colorido como, por exemplo, de formação de rochas, não pode ser reproduzido com precisão (conforme discutido em Wolf [Wol93]).

Modelo HSV (Hue, Saturation, Value)

O sistema HSV utiliza descrições de cor que são mais intuitivas do que combinações de um conjunto de cores primárias e, por isso, é mais adequado para ser usado na especificação de cores em nível de interface com o usuário. A cor é especificada através de uma cor espectral e das quantidades de branco e preto que serão adicionadas para a obtenção de *shades*, *tints* e *tones* diferentes. A representação gráfica tridimensional do sistema HSV é um cone de 6 lados derivado do cubo RGB, mostrado na figura 2.13.

Figura 2.13 - Cone hexagonal do HSV [Fol96]

Os parâmetros de cor utilizados nesse sistema são o matiz (*hue*), a saturação (*saturation*) e a luminância (*value*). Os vários matizes estão representados na parte superior do cone, a saturação é medida ao longo do eixo horizontal e a luminância é medida ao longo do eixo vertical, que passa pelo centro do cone. O matiz, que corresponde às arestas ao redor do eixo

vertical, varia de 0° (vermelho) a 360° , e o ângulo entre os vértices é de 60° . A saturação varia de 0 a 1 e é representada como sendo a razão entre a pureza de um determinado matiz e a sua pureza máxima ($S=1$). Um determinado matiz possui $\frac{1}{4}$ de pureza em $S=0.25$. Quando $S=0$ tem-se a escala de cinzas. A luminância varia de 0 (no pico do cone), que representa a cor preta, a 1 (na base), onde as intensidades das cores são máximas.

Modelo HLS (Hue, Lightness⁵, Saturation)

O sistema HLS também é baseado em parâmetros mais intuitivos para a descrição de cores. Os 3 parâmetros de cor utilizados são o matiz (*hue*), a luminosidade (*lightness*⁶) e a saturação (*saturation*).

Figura 2.14 - Cone duplo do HLS [Fol96]

A representação gráfica tridimensional desse sistema é um cone duplo, como mostra a figura 2.14. Os 3 parâmetros de cor utilizados são o matiz (*hue*), a luminosidade (*lightness*⁷) e a saturação (*saturation*). O ângulo em relação ao eixo vertical varia de 0° (matiz azul) a 360° em intervalos de 60° e especifica um matiz. O eixo vertical corresponde à luminosidade e varia de 0 (preto) a 1 (branco) e é onde se encontra a escala de cinzas. A saturação varia de 0 a 1, e os

⁵ Outros autores denominam esse sistema como *Hue, Luminance, Saturation* [Rog93a].

⁷ Luminosidade (*lightness*): Segundo Foley [Fol96], esse termo está relacionado à noção acromática da intensidade percebida de um objeto que reflete luz. Brilho (*brightness*) é usado no lugar de luminosidade para denotar a intensidade percebida de um objeto que emite luz.

matizes puros são encontrados no plano onde a luminosidade é igual a 0.5 e a saturação é igual a 1. Quanto menor o valor da saturação menor é a pureza do matiz; e quando a saturação é igual a 0, tem-se a escala de cinzas.

Os sistemas HLS e HSV permitem que se pense em termos de cores mais “claras” e mais “escuras”. As cores são especificadas através de um ângulo, e os diversos *shades*, *tints*, e *tones* de cada cor são obtidos através do ajuste do brilho ou luminosidade e da saturação. As cores mais claras são obtidas através do aumento do brilho ou da luminosidade e as cores mais escuras pela diminuição dos mesmos. As cores intermediárias são obtidas através da diminuição da saturação.

2.5 Considerações Finais

A sensação visual de cor é produzida fundamentalmente pela energia eletromagnética que entra no olho e atinge a retina. Essa sensibilidade é muito influenciada por estímulos como, por exemplo, a quantidade de energia do comprimento de onda, a interação entre os elementos presentes no campo de visão, a iluminação do ambiente e a variação temporal.

O processo de visão engloba a realização de processos físicos que vão desde a absorção da luz pelo olho até chegar ao cérebro; e de processos cognitivos, que correspondem à percepção de informações visuais e o reconhecimento de objetos do mundo real. Grande parte do processo de visão é atribuído ao processamento pré-consciente, que consiste de mecanismos que não podem ser controlados voluntariamente e não estão relacionados com a experiência consciente. A parte do processo de visão atribuído ao processamento consciente está relacionada à percepção visual, que envolve o pensamento e a análise da cena que está sendo observada.

Este capítulo apresentou alguns conceitos básicos relacionados aos processos fisiológicos que são realizados pelo sistema visual humano, bem como alguns aspectos relacionados à percepção de cores. Foram descritos alguns dos principais sistemas de representação de cor, como o XYZ, RGB, HSV e HLS, utilizados para descrever e representar cores no espectro visível. O sistema XYZ é um conjunto padrão para expressar cores, construído a partir da transformação linear de informações visuais contidas em outros sistemas de cores.

3. UTILIZAÇÃO DE REGRAS DE PERCEPÇÃO VISUAL NO PROCESSO DE VISUALIZAÇÃO

3.1 Considerações Iniciais

A representação visual é a única forma viável de se trabalhar com grandes conjuntos de dados. As capacidades do sistema visual humano de analisar imagens bidimensionais e de mentalizar um objeto ou um gráfico tridimensional através de sensações visuais requerem um processo de visualização interativo que permita a geração e manipulação de visões diferentes de um mesmo conjunto de dados [Bro92].

O objetivo dos softwares de visualização é apresentar uma imagem que ajude o usuário a estabelecer hipóteses sobre os fenômenos em estudo e a identificar tendências, características e padrões nos dados. Entretanto, a grande variedade de mapeamentos, manipulações e renderizações dos conjuntos de dados oferecidos pelos sistemas torna difícil e lento o processo de gerar visualizações efetivas. A inclusão de procedimentos que orientem o usuário na seleção de ferramentas e técnicas adequadas pode simplificar e tornar mais rápida a produção de visualizações eficazes, sendo que várias abordagens nessa linha são descritas na literatura [Mac86; Fol88; Rog93a; Rog93b; Sen96]. O processo de visualização pode ser melhorado, por exemplo, se aspectos da percepção visual forem levados em consideração. A observação visual de detalhes contidos nos dados e a fidelidade dos resultados podem ser aumentadas, por exemplo, com a utilização de uma escala de cores adequada e de um contraste de brilho apropriado entre regiões adjacentes.

Este capítulo apresenta o processo de geração de visualizações e algumas abordagens adotadas pelos sistemas atuais para apoiar o usuário nesse processo. A seção 3.2 descreve o processo de visualização. Na seção 3.3 são apresentadas algumas abordagens utilizadas para a inclusão de algum tipo de orientação ao usuário no processo de visualização, propostas por vários autores. A abordagem baseada em regras, utilizada para o desenvolvimento deste trabalho é descrita detalhadamente na seção 3.4.

3.2 O Processo de Visualização

A visualização é uma ferramenta que utiliza as habilidades naturais do sistema visual humano para a exploração e a análise dos enormes volumes de informação atualmente disponíveis [Sch96]. A exploração e a análise de informações requerem a realização de julgamentos quantitativos e qualitativos a respeito da apresentação que dependem da percepção humana. É preciso, então, considerar o fato de que a efetividade de uma apresentação depende de estímulos que propiciem a percepção espontânea das estruturas contidas nos dados. Por exemplo, um gráfico de pontos esparsos pode ser efetivo em certos contextos porque estimula a capacidade natural do sistema visual de agrupar pontos no espaço, segundo uma das proposições da Psicologia Gestalt [Lan85]. Além disso, a efetividade da apresentação também depende de fatores que podem ou não ser controlados pelo usuário, como a fisiologia, as experiências anteriores e o que se procura nos dados [Gri90; Reu90].

O processo de visualização pode ser generalizado como sendo uma seqüência de transformações que convertem em uma imagem um conjunto de dados numéricos adquiridos, por exemplo, através de uma simulação computacional ou de medições empíricas. Essas transformações compreendem a manipulação de dados, o mapeamento dos objetos de interesse em objetos gráficos e a renderização. A manipulação consiste na disposição dos dados de entrada em uma estrutura organizada (em geral, uma grade), o que pode exigir a interpolação dos mesmos. O mapeamento corresponde à definição de um relacionamento entre os dados manipulados e as primitivas de visualização (por exemplo, cor e textura). Esse relacionamento tem o objetivo de identificar um conjunto de primitivas que transmitam efetivamente o conteúdo dos dados. Após o mapeamento, os dados são renderizados em uma imagem utilizando operações como a projeção, a tonalização e a remoção de superfícies ocultas [Sen94; Sch96].

Figura 3.1 - O cenário da visualização científica [Rob94]

A figura 3.1 mostra o cenário típico do uso das técnicas de visualização científica em um sistema de visualização. Esses sistemas podem embutir implicitamente algum conhecimento sobre as opções de mapeamento mas podem, também, criar apresentações ambíguas ou confusas [Rob94].

Figura 3.2 - O processo interativo e iterativo de visualização [Rog93a]

A figura 3.2 ilustra o processo de visualização embutido no contexto da investigação científica do qual faz parte. O usuário pode interagir com a imagem e, baseado em sua interpretação, modificar parâmetros e associar novos mapeamentos a fim de compreender melhor a sua estrutura, ou então de criar uma visualização que enfatize uma característica em particular. Nessa abordagem, o usuário controla o processo e os demais componentes contribuem para a interpretação e a análise dos dados [Rog93a]. O processo é altamente iterativo, principalmente quando o objetivo é explorar características dos dados. Quanto mais eficiente é o processo de iteração, mais bem sucedido será o processo de visualização.

Figura 3.3 - A evolução dos sistemas de visualização [Rog93a]

A figura 3.3 mostra a evolução dos sistemas de visualização, incluindo a visão que Rogowitz e outros autores [Rog93a] têm do que seria a tendência para a próxima geração de

sistemas. Nos sistemas tradicionais, as imagens são geradas com a utilização de ferramentas de baixo nível. Os usuários precisam escrever programas para gerar visualizações, dificultando o processo de visualização iterativo. Já os sistemas de visualização modernos possuem uma interface de nível mais alto e uma variedade de ferramentas que tornam o processo iterativo mais rápido. Uma abordagem usual consiste em disponibilizar uma biblioteca de técnicas de visualização que pode ser acessada através de paradigmas de programação visual. Nessa classe, estão incluídos sistemas como o *apE (Animation Production Environment)*, o *AVS (Application Visualization System)* e o *IBM Data Explorer*, que fornecem alguns mecanismos básicos para o mapeamento, mas não fornecem muita orientação aos usuários sobre como selecionar técnicas e mapeamentos capazes de melhorar a visualização resultante. Por exemplo, os diagramas de fluxo de dados que esses sistemas disponibilizam para os usuários especificarem as técnicas de visualização são fáceis de usar mas, por não considerarem qualquer aspecto relacionado à efetividade da visualização, podem produzir tanto resultados bons como ruins [Sen94].

Figura 3.4 - Arquitetura dos sistemas de visualização modernos [Rog93a]

A figura 3.4 ilustra o processo de visualização suportado por muitos sistemas de visualização atuais. Esses sistemas disponibilizam muitas ferramentas de processamento (como a aplicação de mapas de cores e filtros), de mapeamento (geração de estruturas geométricas, como contornos ou *streamlines*) e de renderização (como a geração da imagem por técnicas de *shading* ou *ray-tracing*). Em geral, eles dão ao usuário a liberdade de definir praticamente qualquer seqüência de utilização dessas ferramentas, gerando uma enorme variedade de operações que, na verdade, oferecem mais do que o usuário pode aplicar efetivamente.

A escolha de uma representação visual apropriada é o ponto chave para a compreensão e avaliação crítica dos dados que podem trazer benefícios para a análise, o processamento e a

tomada de decisões. Existem diversas representações possíveis para características de interesse nos dados e, por enfatizar aspectos diferentes dos mesmos, cada uma pode revelar características completamente diferentes [Rob91]. Por isso, é necessário conhecer quais informações são relevantes na visualização e como elas contribuem para a informação como um todo. A efetividade e a relevância de qualquer característica perceptual devem ser determinadas pelo contexto da tarefa a ser desempenhada, seja ela de identificação, comparação, categorização ou segmentação [Mar93].

Outros autores identificaram diferentes tarefas de visualização. Beshers e Feiner [Bes94] descrevem as tarefas em termos de operadores (de exploração, busca e comparação) e de seleções (que definem o foco da visualização como sendo em pontos, regiões, grupos ou global), enquanto que Senay e Ignatius [Sen94] falam em tarefas do tipo busca ou computação (estabelecer correlações e comparações) e no foco da tarefa (novamente, em pontos, regiões, grupos ou global). Já segundo Rogowitz e outros [Rog93a; Rog93b] existem 2 tipos de tarefas, que são exploração e análise ou apresentação de dados. No primeiro caso, o usuário tem uma idéia sobre o que está procurando. No segundo, já criou um entendimento que deseja compartilhar com outros. As tarefas de exploração e análise requerem que a estrutura seja representada de forma fiel na visualização, daí a necessidade de minimizar artefatos perceptuais. Por exemplo, o uso de mapas de cores baseados unicamente em variação de matiz pode levar ao mascaramento e ocultamento de características espaciais verdadeiras dos dados. Já a tarefa de apresentação requer que as características sejam transmitidas de forma não ambígua e que as representações visuais chamem a atenção para características importantes dos dados. Para atingir esse propósito, podem ser usadas regras que transformem a cor, a escala, o brilho ou a geometria, enfatizando o atributo de interesse que irá determinar a interpretação. Embora essa taxonomia de regras de visualização (isomorfismo X ênfase) e de tarefas de visualização (exploração X apresentação) seja simplista, os autores acreditam que ela sirva de ponto de partida para a reflexão sobre os objetivos e processos de visualização e ajude na definição das informações que devem ser fornecidas para as regras que seriam usadas como apoio ao processo de gerar visualizações.

O objetivo de muitas tarefas de visualização é a criação de uma representação fiel e isomórfica dos dados. Em um mapeamento isomórfico, por exemplo, se o intervalo de variação dos dados for duplicado, a dimensão visual associada a essa variável também deve transmitir esse aumento com a mesma intensidade. Em alguns casos, pode haver a necessidade de se

caracterizar os dados de várias maneiras. Por exemplo, uma superfície pode ser colorida para ilustrar valores de interesse de uma variável e “contornada” (*contoured*) de acordo com os valores de outra. Nesses casos, é importante que a percepção de uma variável não afete a percepção da outra. Por exemplo, o brilho relativo das cores pode facilitar a compreensão de um mapa de contornos no qual a cor indica a elevação (nesse caso, as elevações maiores poderiam ser representadas por cores claras e as elevações menores poderiam ser representadas por cores mais escuras). Outros problemas ocorrem porque as dimensões perceptuais disponíveis em um sistema de visualização podem, na verdade, variar ao longo de mais de uma dimensão perceptual. Um exemplo típico é a tabela de cores que obedece a variação do espectro luminoso (*rainbow colormap*). Nesse caso, além da variação no matiz ocorre também variação no brilho (a menos que isso seja explicitamente evitado). Azuis e vermelhos, por exemplo, são em geral menos brilhantes que turquesas e amarelos. Isso pode gerar mensagens conflitantes para quem observa a imagem final [Liv88a].

Os sistemas de visualização podem reduzir consideravelmente o esforço necessário para a criação de visualizações. Além disso, eles podem permitir o entendimento de conjuntos de dados cada vez maiores e mais complexos através do aumento da interatividade e da inclusão de considerações a respeito de requisitos científicos e analíticos. A aceitação desses sistemas depende da facilidade de uso, que pode ser propiciada através da simplicidade de criação de visualizações. Por exemplo, o sistema deve permitir que o cientista seja capaz de criar novos mapeamentos associando interativamente os valores dos dados às visualizações [Fol94]. Entretanto, os sistemas atuais não atendem essa meta, e não podem ser considerados “fáceis de usar” por usuários não especialistas em visualização.

3.3 Abordagens para Auxiliar o Processo de Visualização

A criação de visualizações é uma arte que requer conhecimentos em várias áreas, como gerenciamento de dados, computação gráfica, psicologia da percepção, e artes gráficas e *design* visual, os quais a maioria dos cientistas que trabalham com visualização não possui. Por isso, vários autores acreditam que a incorporação desses conhecimentos nos sistemas de visualização pode melhorar tanto o processo de visualização como os resultados obtidos [Ger92; Sen94]. Basicamente 3 classes de abordagens para auxiliar a criação de visualizações podem ser identificadas na literatura:

1. Auxílio baseado em regras que restringem o universo de opções de mapeamentos possíveis através da incorporação de conhecimento sobre aspectos de percepção visual;
2. Auxílio através da geração semi-automatizada de visualizações, com colaboração e interferência do usuário;
3. Auxílio através da geração totalmente automatizada de visualizações.

Adotando a primeira abordagem, Rogowitz e outros [Rog93a] propõem que a ajuda seja incluída como conhecimento explícito representado na forma de regras. As regras incorporariam princípios conhecidos resultantes de estudos realizados por especialistas em percepção visual, processos cognitivos e teoria da cor. Esses autores apresentam uma arquitetura baseada em regras, denominada PRAVDA (*Perceptual Rule-Based Architecture for Visualizing Data Accurately*), para ajudar o usuário na seleção dos parâmetros de visualização. Nessa arquitetura, cada operação de visualização pode ser associada a regras que restringem o conjunto de opções que serão oferecidas ao usuário. Dessa forma, os sistemas incorporariam alguma capacidade de orientar os usuários na geração de visualizações sem, entretanto, gerar as visualizações de forma automatizada, e deixando o controle do processo com o usuário. Nessa abordagem, os sistemas tentariam reduzir a dificuldade e o tédio dos usuários ajudando na escolha da estratégia de visualização e da manipulação de representações.

Figura 3.5 - Arquitetura dos sistemas de visualização modernos com a inclusão de regras [Rog93a]

A figura 3.5 mostra a adição de uma camada complementar (ilustrada em cinza) aos sistemas atuais para auxiliar o usuário na aplicação das ferramentas disponíveis. Essa camada corresponde à inclusão de metadados (informações adicionais sobre o conjunto de dados) e do conjunto de regras adotadas pelo sistema. Com a inclusão dessa camada complementar, que não

altera o processo de geração das visualizações, os metadados são transmitidos para as regras de visualização, que restringem o conjunto de operações disponibilizadas ao usuário. Assim, a partir de um conjunto de opções mais restritas, porém mais adequadas, o usuário pode selecionar uma ferramenta considerada efetiva para visualizar seus dados. A representação dos dados em termos de abstrações de alto nível pode fornecer uma interface mais intuitiva e eficiente para o processo de visualização porque o usuário interage com ferramentas que manipulam as descrições dos dados (metadados), e essa interação pode ser mediada por regras baseadas em visão humana, cognição e teoria da cor, que orientam o usuário na seleção de operações de visualização apropriadas.

As regras propostas em Rogowitz e outros [Rog93a] são divididas em 2 categorias, sendo que a primeira está relacionada à representação dos dados, e a segunda está relacionada à tarefa de visualização a ser desempenhada pelo usuário. Na primeira categoria, as regras podem ser divididas em 2 subclasses: uma que procura preservar a estrutura e a invariância dos dados (isomorfismo de representação), e outra que procura enfatizar as características dos dados (*highlighting*). Na segunda categoria, as regras são classificadas de acordo com as tarefas de visualização: de exploração ou análise dos dados, e de apresentação.

Tabela 3.1 – Regras para tarefas de exploração e apresentação [Rog93a]

Regras/Tarefas	Análise ou Exploração	Apresentação
1:1 Representação Verídica (Isomórfica)	<ul style="list-style-type: none"> ■ Imagens médicas ■ Sensoriamento remoto 	<ul style="list-style-type: none"> ■ <i>Medical Demographics</i> (correlação entre parâmetros) ■ Comparação do modelo de previsão com os dados
Enfatização (<i>Highlighting</i>)	<ul style="list-style-type: none"> ■ Reconhecimento (segmentação de características e identificação) ■ Segmentação de campo ■ <i>Model-based enhancement</i> 	<ul style="list-style-type: none"> ■ Apresentação do fenômeno para entidades financiadoras ■ Suporte à decisão ■ Publicação científica ■ Ensino

A tabela 3.1 mostra alguns domínios de aplicação das regras nas tarefas de análise ou exploração e de apresentação. Por exemplo, a aplicação de ênfase a imagens da Terra obtidas por sensoriamento remoto pode ajudar na identificação de regiões de cultivo de diferentes culturas. Essas regiões precisam ser mostradas preservando o contexto dos dados. Essa abordagem será discutida mais detalhadamente na seção 3.4.

Um exemplo de abordagem da segunda classe, de geração semi-automatizada de visualizações, é adotada no sistema IVE [Bes94; Koc91] que é provavelmente o primeiro

trabalho nessa linha. Esse sistema introduz uma abordagem para a geração de visualizações denominada projeto cooperativo auxiliado por computador (CCAD - *Cooperative Computer-Aided Design*). Nesse sistema, as decisões tomadas pelo usuário são alternadas com as decisões tomadas pelo computador. Essa abordagem reconhece a dificuldade de gerar um projeto de visualização (*design*) totalmente automatizado e a conseqüente importância de permitir que o usuário forneça um *feedback* durante o processo, dadas as limitações das bases de conhecimento de projetos. Em cada etapa do processo de geração de uma visualização, o usuário seleciona um conjunto de regras que o sistema aplica para criar um conjunto de projetos alternativos. O usuário pode selecionar uma ou mais alternativas de projetos a serem elaborados pelo sistema. As alternativas elaboradas são mostradas em uma superfície na qual o usuário pode aplicar um *zoom* ou *pan* para examinar ou selecionar as que considera melhores.

As demais abordagens aqui apresentadas se enquadram na terceira classe, de geração totalmente automatizada de visualizações. Muitos desses sistemas “inteligentes” de visualização são baseados na escolha e combinação de membros de um conjunto pré-definido de técnicas gráficas. Vários são baseados nos conceitos introduzidos no sistema APT [Mac86; Bes94; Hyp96].

O APT (*A Presentation Tool*) foi o primeiro sistema gráfico a utilizar o paradigma de Geração-e-Teste (*Generate-and-Test*), que requer a definição de um conjunto de primitivas gráficas, de regras para a geração de visualizações a partir do conjunto de primitivas e de critérios de teste para cada visualização gerada. As primitivas gráficas utilizadas nesse sistema incluem áreas (por exemplo, círculos), linhas, sinais (por exemplo, +), e atributos visuais como cor, tamanho e orientação. Essas primitivas definem linguagens gráficas como: eixo horizontal, eixo vertical, gráfico de linha, gráfico de barra, pontos esparsos, cor, forma, tamanho, saturação, textura, orientação, árvore e rede. Existem 3 regras de composição para a geração de gráficos completos: composição de 2 eixos, composição de 1 eixo e composição de sinal. A composição de eixos é feita através da sobreposição de 2 conjuntos de dados que compartilham eixos horizontal e vertical comuns. A composição de 1 eixo é feita através da justaposição de 2 conjuntos de dados que compartilham um eixo comum. A composição de sinal é feita através da utilização de 2 atributos diferentes de uma área ou de um sinal para codificar 2 variáveis diferentes. Os critérios de teste utilizados nesse sistema foram obtidos a partir de uma combinação de estudos experimentais e levam em consideração a efetividade e a expressividade de uma linguagem. Uma linguagem é expressiva para um conjunto de fatos quando ela codifica

todos os fatos e não acrescenta dados incorretos. Uma apresentação é efetiva, por exemplo, quando pode ser interpretada com precisão ou rapidez. Para cada apresentação gerada, o sistema filtra as primitivas, gera uma lista de projetos a partir do critério de expressividade e ordena os projetos de acordo com o critério de efetividade de forma que o projeto mais efetivo seja escolhido primeiro.

O sistema SAGE (*A System for Automatic and Graphic Explanation*) [Rot91; Bes94] é uma extensão do trabalho de Mackinlay [Mac86] que adiciona descrições semânticas dos dados e dos relacionamentos entre eles. O sistema representa, por exemplo, relações algébricas na forma: $\text{custo} = \text{material} + \text{mão-de-obra}$. Essa representação permite que as informações sejam discriminadas e apresentadas separadamente, por exemplo, em um gráfico de barras segmentado em custo material e custo de mão-de-obra.

O sistema BOZ [Cas91, Bes94] também utiliza o paradigma de Geração-e-Teste, incluindo uma abordagem diferente que enfoca principalmente o objetivo ou a tarefa que o usuário deseja realizar. O sistema gera diferentes visualizações e, para cada uma delas, a sequência de busca e das operações de comparação que o usuário deve efetuar para atingir o objetivo estabelecido. O critério de teste que determina a alternativa a ser aceita é essencialmente o tempo que o usuário precisa para realizar a sequência de operações. Em seus experimentos, Casner [Cas91, Bes94] constatou que a utilização do sistema BOZ melhorava o desempenho dos usuários em relação à utilização de sistemas menos elaborados, e que o tempo previsto para a realização das operações tinha uma boa correlação com os tempos despendidos pelos usuários.

Senay e Ignatius [Sen94] propõem a incorporação de um sistema baseado em conhecimento aos sistemas de visualização. Nessa abordagem, o conhecimento embutido no sistema na forma de um “sistema especialista” seria utilizado para “controlar” o processo de gerar a visualização que poderia ser (paralelamente) automatizado. O VISTA (*Visualization Tool Assistant*) [Sen94] é um sistema especialista que ajuda usuários a escolher uma técnica de visualização apropriada para um conjunto de dados, e permite que eles modifiquem interativamente o projeto da visualização. O usuário especifica o conjunto de dados a ser visualizado e o sistema gera uma estratégia de visualização apropriada para esse conjunto de dados e renderiza a imagem resultante. O sistema gera as visualizações com base na composição de técnicas de visualização primitivas. As técnicas primitivas são capazes de codificar entre uma a quatro variáveis independentes, e pertencem a 3 categorias (posicionais, temporais e retiniais), como indicado na figura 3.6. O sistema baseado em conhecimento utiliza regras heurísticas para

combinar essas técnicas primitivas de forma a gerar visualizações efetivas para conjuntos de dados multidimensionais.

Figura 3.6 - Categorias de codificação de variáveis [Sen94]

As regras embutidas no sistema foram obtidas da literatura e a partir de discussões com especialistas em visualização. O sistema permite que os usuários modifiquem as técnicas de visualização interagindo diretamente com o processo de geração ou manipulando as imagens geradas pelo sistema.

Através da análise das técnicas de visualização existentes, os autores identificaram 5 categorias de conhecimentos necessários para a visualização de dados científicos, que são: as características dos dados; o vocabulário de visualização; as técnicas primitivas de visualização; as regras de composição; e as regras de percepção visual. Entre as características dos dados que afetam a seleção de primitivas de visualização estão incluídos o tipo da variável, as dependências entre as variáveis, o espaçamento entre os pontos amostrais, as unidades de medida, o sistema de coordenadas e a continuidade dos dados. O vocabulário de visualização identifica os elementos de construção das técnicas, entre eles o símbolo (*mark*), que é o componente mais primitivo para a codificação de uma informação, e pode ser simples ou composto. Os sinais compostos são coleções de sinais simples que formam uma unidade perceptual (por exemplo, as linhas de contorno, as malhas de arame, os *glyphs*, as linhas de fluxo e as partículas). As técnicas primitivas codificam 1 variável dependente e até 4 independentes que, através da aplicação de regras de composição, podem gerar as técnicas compostas. As regras de percepção visual são heurísticas relacionadas à expressividade e efetividade das primitivas de visualização, ao

escalonamento, à manipulação de peculiaridades dos dados, etc. Os autores listam várias regras a serem seguidas e que devem ser respeitadas na aplicação de cada técnica primitiva.

Em outro exemplo de abordagem baseada na automatização do processo da visualização, Beshers e Feiner [Bes93] propõem a geração automatizada de “mundos virtuais” para a visualização de relações multivariadas. O *AutoVisual* é um sistema que apoia o processo de gerar visualizações para o sistema de visualização *n-Vision*, que usa um paradigma denominado “*worlds within worlds*”. Esse paradigma suporta um método interativo para a representação de uma hierarquia de sistemas de coordenadas heterogêneos (“*worlds*”) aninhados. Cada “*world*” pode conter um grafo que codifica um subconjunto da relação codificada pelo seu “*world*” pai. O usuário pode “segurar” cada “*world*” usando um *DataGlove* e movê-lo pelo espaço definido por seu “pai”, permitindo assim a exploração interativa do espaço de dados.

Foley e outros [Fol94] também propõem a automatização do processo de geração de visualizações. O sistema *Glyphmaker* permite que o usuário construa e organize *glyphs* para representar dados multivariados. Os *Glyphs* são objetos gráficos cujos atributos visuais (como posição, tamanho, forma, cor, orientação, etc.) são associados a atributos dos dados. O sistema oferece uma biblioteca de *glyphs*, e permite que os usuários definam os seus próprios *glyphs* interativamente. O sistema permite também que o usuário associe interativamente atributos de variáveis de interesse a atributos do *glyph* para apoiar a exploração dos padrões e relações contidos nos dados.

Outros autores propõem abordagens alternativas para o problema de apoiar os usuários no processo de gerar visualizações. Rheingans e outros [Rhe90] propõem o controle dinâmico do mapeamento de cores para a exploração de dados. A ferramenta descrita permite, por exemplo, que os valores de interesse sejam isolados e que os intervalos de dados sejam representados por uma cor específica.

Grinstein e Smith [Gri90] propõem uma abordagem iconográfica para geração de visualizações de dados multidimensionais. Segundo esses autores, o requisito crítico para uma visualização efetiva é que ela seja capaz de estimular a percepção espontânea das estruturas contidas nos dados - como é o caso dos “*scatter plots*” que estimulam a capacidade natural de identificar agrupamentos de pontos no espaço. A abordagem adotada por esses autores estende o conceito dos “*scatter plots*”, e explora outra capacidade perceptual espontânea: a habilidade de identificar e discriminar textura em figuras geradas a partir do traçado de linhas. Para estimular a

percepção visual de textura, muitos elementos idênticos mas discrimináveis devem ser desenhados sobre uma área pequena. Na abordagem proposta, cada amostra de dados é representada por uma pequena unidade gráfica denominada “ícone”. Os atributos de um ícone, como os tamanhos e posições de seus componentes, são definidos segundo os dados. Muitos ícones desenhados densamente sobre uma região geram uma imagem iconográfica na qual pode-se identificar texturas. Os ícones podem conter, além de atributos visuais, atributos sonoros.

Existe, ainda, a abordagem baseada no paradigma da cena natural, proposta por Robertson [Rob91], que procura representar as variáveis através de propriedades de objetos que podem ser imediatamente reconhecidas. Segundo Robertson, é importante que o observador consiga construir um modelo mental a partir da visualização, cujos atributos visuais representem atributos dos dados de uma maneira bem definida. Ele sugere o “paradigma da cena natural”, que usa modelos claros e facilmente compreensíveis como estruturas ou cenas 3D; representa variáveis de dados através de propriedades reconhecíveis dos objetos ou cenas; e usa técnicas gráficas de simulação de cenas para induzir o reconhecimento de propriedades pelo observador. A idéia é explorar a habilidade dos seres humanos de olhar para uma cena “real” e ganhar um entendimento imediato da estrutura de sua superfície 3D, o que recobre a superfície e a condição ou estado dessa cobertura. Esse paradigma, estudado por diferentes autores, é proposto por Robertson como base para uma metodologia capaz de apoiar a geração de visualizações adequadas para conjuntos de dados.

3.4 A Abordagem Baseada em Regras – Incorporação de Aspectos de Percepção no Processo de Seleção de Mapas de Cores

O mapeamento de valores em uma escala de cores é uma das operações mais comuns em visualização. Uma escala de cores é uma representação pictórica de valores numéricos, sendo que cada valor é associado a sua própria cor [Lev92]. Apesar da importância dessa operação, que inclui a criação e seleção de mapas de cores, os sistemas atuais não fornecem suporte adequado para a definição de tabelas de cores. Normalmente, esses sistemas oferecem um mapa de cores padrão e uma ferramenta para a criação de mapas de cores personalizados, mas não ajudam o usuário a selecionar mapas que propiciem a compreensão da estrutura dos dados, ou que garantam uma segmentação significativa dos dados em regiões, ou a ênfase de características importantes. A utilização do mapa de cores do arco-íris (*rainbow*), que é o padrão

mais comum, muitas vezes dá a impressão errônea de que os dados estão organizados em regiões discretas, porque cada região é representada por uma cor diferente. Isso pode levar o usuário a inferir estruturas inexistentes e a perder detalhes agrupados numa região mapeada em uma única cor. Alguns sistemas permitem que o usuário crie qualquer mapa de cores, mas são difíceis de usar por criarem mapas de cores complexos [Ber95]. Além disso, como muitas das cores em uma escala podem não ser percebidas como cores distintas, para obter uma representação informativa é importante maximizar o número de cores distintas percebidas ao longo da escala. Esse número é denominado número de JNDs (*Just Noticeable Differences*) [Lev92]. Levkowitz e Herman [Lev92] introduzem o conceito de uma escala de cores ótima, que maximiza o número de JNDs mantendo a “ordem natural” entre as cores, e propõem uma estratégia para gerar escalas ótimas.

A inclusão de regras de percepção nos sistemas tem o objetivo de assegurar a representação fiel das estruturas contidas nos dados e de garantir que os artefatos perceptuais não sejam interpretados como características dos dados. Bergeron e outros [Ber95; Ber96] implementaram um módulo de apoio à seleção de mapas de cores denominado PRAVDAColor. Esse módulo segue os princípios idealizados para a arquitetura PRAVDA, proposta em [Rog93a; Rog93b] e descrita na seção 3.3.

O PRAVDAColor é um módulo baseado em regras que apresenta ao usuário um conjunto de mapas de cores que são apropriados tanto para o tipo de dado a ser visualizado como para a tarefa de visualização a ser realizada pelo usuário. Para isso, foram construídas bibliotecas de mapas de cores e um conjunto de regras perceptuais que são usadas para selecionar os mapas de cores mais apropriados. A arquitetura do módulo também estabelece uma associação entre as diferentes regras que controlam as várias operações de visualização, sendo que a escolha de parâmetros para uma operação pode restringir o conjunto de opções das outras. Bergeron e outros [Ber95; Ber96] usam o modelo HLS (*Hue, Luminance, Saturation*, ou Matiz, Luminância, Saturação) para a especificação de cores.

As regras são divididas em 2 classes, sendo que uma procura assegurar que as estruturas sejam fielmente representadas na visualização e a outra enfatiza certas características em particular. Essas regras são baseadas em metadados relativos ao tipo e à frequência espacial dos dados, e na tarefa de visualização que o usuário deseja realizar. São considerados 2 tipos de dados: tipo razão, que é caracterizado pela presença de um valor de referência denominado zero; e tipo intervalo, cujos valores representam incrementos em relação a um valor anterior, a partir

de um valor inicial. Assim, incrementos iguais nos valores dos dados deveriam aparecer como incrementos com magnitudes percebidas idênticas na representação visual gerada. Adicionalmente, a dimensão visual escolhida para representar os dados deve parecer contínua para o usuário da visualização. Já dados do tipo razão contém valores que aumentam/diminuem monotonicamente em torno de um valor de referência. Portanto, os valores indicam uma escala de variação em relação ao valor de referência que deve ser preservado na representação dos dados. O valor de referência corresponde ao valor 0 quando o conjunto de dados possui valores positivos e negativos; ou pode ser outro valor que representa um limiar de referência para o usuário.

Também é considerada a frequência espacial dos dados, que podem ser classificados como sendo de alta ou baixa frequência. Dados de alta frequência geram imagens com detalhes finos ou bordas, enquanto que os de baixa frequência geram imagens com variações suaves. A importância de considerar a frequência espacial dos dados está no fato de que o sistema visual humano possui sensibilidades espaciais diferentes para cada componente da cor. Enquanto a luminância (variação entre claro e escuro) é fundamental para transmitir variações nos dados de alta frequência espacial, a saturação e o matiz são fundamentais para a transmissão de variações nos dados de baixa frequência espacial. Assim, se os metadados indicam que os dados, quando exibidos na tela, produziram altas variações na frequência espacial, as regras oferecem ao usuário mapas de cores com um componente de luminância variando monotonicamente. Se os dados contêm predominantemente baixas frequências espaciais, as regras oferecem ao usuário mapas de cores com variações na saturação dentro de um único matiz ou variações de saturação entre 2 matizes, como os pares de cores oponentes azul-amarelo ou verde-vermelho (azul ou verde saturado, até cinza isoluminante, até amarelo ou vermelho saturado).

Na implementação realizada pelos autores, os metadados tipo e *range* (intervalo de variação) dos dados são obtidos a partir do próprio conjunto de dados; a tarefa de visualização é fornecida pelo usuário; e a frequência espacial é determinada através de um cálculo aproximado.

As tarefas de visualização consideradas na arquitetura são isomorfismo, segmentação e ênfase. A tabela 3.2 mostra a taxonomia utilizada para a geração das tabelas de cores.

Tabela 3.2 - Classificação de mapas de cores utilizada na ferramenta PRAVDAColor [Ber95]

Tipo de Dado	Frequência Espacial	Tarefa de Representação		
		Isomorfismo	Segmentação	Enfatização
Razão	Baixa	L: uniforme M: pares oponentes ou complementares S: aumentando monotonicamente a partir do cinza	- Número par de segmentos - Muitos segmentos ok	- Variação maior para as características enfatizadas
	Alta	L: aumentando monotonicamente M: pares oponentes ou complementares S: aumentando monotonicamente a partir do cinza	- Número par de segmentos - Menos segmentos	- Variação menor para as características enfatizadas
Intervalo	Baixa	L: uniforme M: pares oponentes S: aumentando monotonicamente a partir do cinza	- Muitos segmentos ok	- Variação maior para as características enfatizadas
	Alta	L: aumentando monotonicamente M: variação pequena ou uniforme de matiz S: diminuindo monotonicamente	- Menos segmentos	- Variação menor para as características enfatizadas

A tarefa de isomorfismo busca uma representação que reflita as estruturas contidas nos dados, eliminando os artefatos perceptuais. As restrições para a geração de tabelas são os pares de cores que a compõem e os comportamentos dos parâmetros de saturação e luminância. As tabelas são compostas por 1 matiz ou por 2 matizes que definem pares de cores oponentes ou complementares, e a variação dos parâmetros saturação e luminância é monotônica.

Na tarefa de segmentação, o objetivo é dividir a imagem em regiões de interesse distintas, sendo que a principal restrição considerada para a geração de tabelas é a quantidade de segmentos a serem gerados. Para dados de alta frequência, essa quantidade deve ser menor uma vez que, se a imagem possuir detalhes pequenos e a tabela possuir muitos segmentos, a imagem resultante pode ser segmentada em muitas regiões pequenas, dificultando a identificação de detalhes. Além disso, para dados do tipo razão, o número de segmentos gerados deve ser par; de forma que o número de segmentos associados a valores acima e abaixo do valor de referência seja igual.

Na tarefa de ênfase o objetivo é chamar a atenção do usuário para características de interesse. A restrição para a geração de tabelas para essa tarefa é o tamanho do intervalo de variação de interesse ou seja, o número de posições da tabela a serem ocupadas pela mesma cor. O intervalo deve ser menor para dados de alta frequência e maior para dados de baixa frequência. Um aspecto fundamental para a criação de tabelas de cores que enfatizam um conjunto de dados é que não deve haver interferências entre os diferentes componentes da cor utilizados numa representação.

A Importância da Frequência Espacial

Um fato importante em visualização é a existência de sensibilidades espaciais diferentes para cada componente da cor. A luminância (claro e escuro) é fundamental para transmitir variações em dados de alta frequência espacial para mostrar detalhes pequenos. Da mesma forma, a saturação e o matiz são fundamentais para transmitir variações nos dados de baixa frequência espacial. Nesse caso, uma tabela de cores com variação apenas na luminância não pode transmitir adequadamente as alterações graduais na estrutura espacial dos dados.

A frequência espacial dos dados define, então, o equilíbrio entre a variação da luminância e da saturação numa tabela de cores isomórfica. Assim, as variações dos dados do tipo intervalo e de alta frequência espacial são melhor transmitidas com a utilização de uma escala monotônica com variação da luminância; e as variações de dados do tipo intervalo e de baixa frequência espacial são melhor transmitidas com a utilização de uma escala monotônica com variação na saturação.

Figura 3.7 - Dados de alta e baixa frequências espaciais [Rog96]

A figura 3.7 mostra uma tabela de cores com variação na luminância (lado esquerdo) e uma tabela de cores com variação na saturação (lado direito) aplicadas a dados de baixa frequência espacial (parte superior) e a dados de alta frequência espacial (parte inferior). Em todos os casos, os dados contínuos foram mapeados com tabelas de cores isomórficas, eliminando artefatos perceptuais como o de inclusão de contornos artificiais [Rog96].

Os dados mostrados na parte superior foram obtidos através de um modelo de previsão do tempo que calcula, entre outras coisas, a variação na umidade relativa do ar sobre uma região geográfica. A tabela com variação no matiz e na saturação é mais informativa, principalmente nas regiões geográficas onde há alteração pequena na umidade, como ocorre nas proximidades da região central, com umidade abaixo de 65% [Rog96].

As imagens mostradas na parte inferior mostram uma varredura de um sensor de radar de previsão do tempo, medindo variações de alta frequência espacial da intensidade refletida (por exemplo, de nuvens pesadas). A tabela com variação na luminância fornece uma representação com detalhes da estrutura dos dados e revela, ainda, artefatos amostrais que são introduzidos pelo sensor. A tabela com variação no matiz dá um aspecto borrado aos detalhes pequenos e, pelo fato de mapear alguns valores com um matiz diferente, enfatiza regiões de modo não apropriado [Rog96].

Figura 3.8 - Tabelas de cores para segmentação de dados de baixa e alta frequências espaciais [Rog96]

A figura 3.8 apresenta tabelas de cores com 5 níveis de segmentação (lado esquerdo) e 10 níveis de segmentação (lado direito) aplicados a dados de baixa frequência espacial (parte superior) e a dados de alta frequência espacial (parte inferior). A tabela com mais níveis de segmentação fornece informações adicionais para os dados de baixa frequência espacial porque revela características adicionais do campo magnético da Terra no hemisfério Sul. Por exemplo, o gradiente no pólo magnético do Sul fica mais nítido na imagem à direita. Por outro lado, o aumento do número de segmentos não acrescenta informações para os dados de alta frequência espacial. Ao contrário, os fragmentos de nuvem ficam com uma aparência “borrada” [Rog96].

3.5 Considerações Finais

O mapeamento de dados em primitivas gráficas que transmitam efetivamente o conteúdo informativo dos mesmos constitui um dos pontos mais importantes na visualização de dados científicos. Porém, a existência de vários tipos de mapeamentos requer que o usuário tenha criatividade e conhecimento consideráveis para selecionar a alternativa mais adequada a cada situação. A inclusão de aspectos da percepção visual pode aprimorar o processo de visualização e, conseqüentemente, o seu resultado. Esse aprimoramento pode se dar através do aumento da visibilidade de detalhes dos dados, da fidelidade dos resultados, da rapidez e do dinamismo do processo [Ger94]. A inclusão pode se dar na forma de conhecimentos e princípios que são adquiridos com a experiência e a partir de pesquisas em percepção visual, e incorporados aos sistemas de visualização na forma de regras heurísticas.

Este capítulo apresentou a descrição do processo de visualização considerando a motivação para a incorporação de aspectos de percepção ao processo. Foram apresentadas, ainda, várias abordagens propostas na literatura com esse objetivo. A abordagem baseada em regras adotada por Rogowitz e outros [Rog93a; Rog93b], que serviu de base para o desenvolvimento deste trabalho, foi descrita em maiores detalhes.

4. IMPLEMENTAÇÃO DO MAPEAMENTO POR CORES BASEADO EM REGRAS

4.1 Considerações Iniciais

O módulo de mapeamento por cores baseado em regras desenvolvido ao longo deste projeto reproduz a abordagem apresentada em Rogowitz e outros [Rog93a; Rog93b] e descrita na seção 3.4. Os procedimentos para a geração de tabelas, segundo a taxonomia apresentada na tabela 3.2, foram adicionados ao *Visualization Toolkit* (VTK) na forma de uma nova classe do VTK, e serão posteriormente incorporados ao sistema de visualização *PowerVis* [Min97].

Este capítulo apresenta o desenvolvimento do módulo de mapeamento por cores baseado em regras e descreve o processo de integração do mesmo ao VTK. A seção 4.2 apresenta uma breve descrição sobre o *Visualization Toolkit*, que é a plataforma de software utilizada para o desenvolvimento deste trabalho. Na seção 4.3, é descrita a técnica de mapeamento por cores. A classe *vtkLookupTable*, que implementa essa técnica no *Visualization Toolkit*, é descrita na seção 4.4. A classe *vtkRuleLookupTable*, que implementa o mapeamento por cores baseado em regras, é descrita em detalhes na seção 4.5. A seção 4.6 descreve a classe *SpatialFrequency*, responsável pelo cálculo da frequência espacial. A seção 4.7 descreve os passos necessários para a incorporação da nova funcionalidade ao *Visualization Toolkit*. A seção 4.8 descreve as limitações desta implementação.

4.2 O *Visualization Toolkit* – Conceitos Básicos

O VTK (*Visualization Toolkit*) [Sch96; Sch97] é um software multiplataforma facilmente disponível que consiste de algoritmos e estruturas de dados para visualização, compatível com plataformas UNIX, Windows NT e Windows 95. O software foi desenvolvido em C++ de acordo com o paradigma de orientação a objetos. O *toolkit* é extensível, permitindo o desenvolvimento

de novas aplicações com uma interface gráfica própria, a alteração das técnicas já implementadas e a inclusão de novas técnicas de visualização, pois o seu código fonte está disponível.

A geração de visualizações no VTK é feita através da criação de *pipelines* de visualização, que correspondem a uma seqüência de transformações de dados de entrada até a renderização na tela, como mostra a figura 4.1.

Figura 4.1 - O *pipeline* de visualização

O VTK possui um conjunto de classes com funcionalidades específicas, como as classes dos tipos fontes, filtros, mapeadores e gráficos. Os objetos da classe fonte iniciam o *pipeline* de visualização, fazendo a leitura de arquivos de dados externos ou criando dados a partir de variáveis de instância, e gerando um ou mais conjuntos de dados de saída. Os objetos da classe filtro recebem pelo menos um conjunto de dados de entrada e geram um ou mais conjuntos de dados de saída. Os mapeadores finalizam o *pipeline* de visualização, recebendo um ou mais objetos de entrada, e fazendo o mapeamento de dados para um determinado dispositivo, ou para uma biblioteca gráfica. Os objetos gráficos representam a funcionalidade do núcleo de renderização. Esses objetos permitem a manipulação de fontes de luz, câmeras e atores, das propriedades dos atores e dos atributos de renderização (como, por exemplo, a cor de fundo do renderizador). Os 8 objetos básicos do VTK utilizados para renderizar uma cena são:

- *vtkRenderMaster*: Coordena os métodos independentes do dispositivo e cria uma janela de renderização;
- *vtkRenderWindow*: Gerencia uma janela no monitor e permite que um ou mais renderizadores desenhem (*draw*) em uma instância dessa classe;
- *vtkRenderer*: Coordena o processo de renderização envolvendo fontes de luz, câmeras e atores;
- *vtkActor*: Representa um objeto renderizado em uma cena, com as suas propriedades e a sua posição no sistema de coordenadas do usuário;
- *vtkProperty*: Define as propriedades da aparência de um ator, como a cor e a transparência; o tipo de iluminação, como a especular e a difusa; e os tipos de representação, como *wireframe* ou superfície sólida;
- *vtkLight*: Representa uma fonte de luz para iluminar uma cena;
- *vtkCamera*: Define a posição de observação de uma cena;

- *vtkMapper*: Define a representação geométrica de um ator.

4.3 Mapeamento por Cores

A técnica de mapeamento por cores é bastante usada para visualizar conjuntos de dados escalares 1D, 2D, ou 3D, e constitui uma maneira simples de aumentar a quantidade de informações transmitidas em uma visualização. O ponto chave para a utilização de mapas de cores na visualização de dados escalares é a escolha adequada das cores que compõem uma tabela. Como já foi citado em capítulos anteriores, o uso cauteloso de cores pode realçar características importantes dentro do conjunto de dados, ao passo que uma tabela mal escolhida pode enfatizar detalhes que não são importantes ou criar artefatos visuais causados pela interação entre as cores.

A implementação dessa técnica consiste na utilização de valores escalares como índices para uma tabela de cores. A tabela mantém um vetor de cores que podem ser definidas em termos dos seus componentes vermelho, verde e azul (modelo RGB), ou através de outros modelos de cor. Em geral, cada entrada na tabela também inclui um componente de opacidade. Os valores de dados mínimo e máximo (que definem o *scalar range*) são mapeados nas posições inicial e final da tabela, e os valores intermediários são mapeados linearmente (em geral) nas posições intermediárias. Assim, para cada valor escalar S_i , o índice i em uma tabela com n entradas e sem *offsets* é obtido como ilustrado pela figura 4.2.

Figura 4.2 - Mapeamento de valores escalares através de tabelas de cores [Sch96]

A maioria dos sistemas de visualização atuais fornece uma tabela de cores padrão (sendo a mais comum a tabela de cores do arco-íris, que mapeia o menor valor em azul, o maior em vermelho, e os valores intermediários linearmente nas cores do espectro luminoso) e uma ferramenta para a criação de tabelas de cores personalizadas. As tabelas geradas nesses sistemas

podem levar o usuário a inferir estruturas inexistentes nos dados e a perder detalhes agrupados em uma região mapeada em uma única cor.

Figura 4.3 - Aplicação de tabelas de cores diferentes sobre um mesmo conjunto de dados [Rog96]

A figura 4.3 ilustra quatro representações de uma imagem da cabeça humana, obtida através de ressonância magnética. Através dessa imagem, pode-se verificar como a utilização de diferentes tipos de tabelas de cores sobre um mesmo conjunto de dados produz resultados bem diferentes e pode influenciar na interpretação feita pelo usuário. Foram utilizadas 3 tabelas de cores, uma isomórfica, uma de segmentação e outra de ênfase, as quais são comparadas com a tabela *default*. Essa imagem deixa claro que uma variação no método de representação dos dados pode influenciar de modo significativo tanto a percepção como a interpretação dos mesmos [Rog96].

Se o objetivo é representar fielmente a estrutura dos dados, a tabela *default* é inadequada, pois o que se vê são faixas de cores, e não um aumento gradual ao longo do intervalo de variação, mascarando as variações sutis de intensidades geradas pela ressonância magnética. Já a tabela isomórfica, apesar de chamar menos a atenção, garante que a luminância seja mantida constante e que a saturação varie monotonicamente segundo os valores dos dados, produzindo um crescimento monotônico da magnitude percebida ao longo do intervalo de variação.

Se o objetivo é segmentar a imagem segundo regiões de valores de interesse, a tabela de segmentação garante que os valores de interesse serão mapeados nas cores desejadas, que podem ser escolhidas de forma a atrair a atenção do usuário para intervalos críticos – por exemplo as

regiões mapeadas em vermelho e amarelo chamam mais a atenção do que as mapeadas em verde ou preto.

Já a tabela de ênfase é utilizada para enfatizar regiões de interesse. No exemplo, a tabela foi projetada para atrair a atenção para áreas cujos valores estão perto da mediana do intervalo de variação dos dados.

4.4 Mapeamento por cores no VTK - Classe *vtkLookupTable*

A classe *vtkLookupTable* do VTK define um objeto gráfico que permite estabelecer um mapeamento de valores escalares para cores ou de cores para valores escalares, gerando uma tabela de cores através da criação de instâncias da classe. As tabelas de cores são definidas através da especificação de um número de entradas e de parâmetros de variação das cores. Os parâmetros podem ser definidos através dos modelos HSV (*hue*-matiz, *saturation*-saturação, *value*-luminância) ou RGB (*red*-vermelho, *green*-verde, *blue*-azul). Além disso, pode-se especificar ainda um valor de transparência (*alpha*), que varia de 0 a 1. Para gerar uma tabela utilizando o modelo HSV deve-se especificar o intervalo de variação de cada parâmetro, que pode variar de 0 a 1 ao longo de toda a tabela. As cores da tabela são definidas através de uma função linear aplicada aos valores iniciais e finais dos componentes HSV e de transparência especificados. A geração de uma tabela de cores *default* no VTK, utilizando a linguagem C++, pode ser feita como exemplificado no trecho de código a seguir.

```
vtkLookupTable *lut = new
vtkLookupTable;
    lut->SetHueRange(0.0, 0.6667);
    lut->SetSaturationRange(1.0, 1.0);
    lut->SetValueRange(1.0, 1.0);
    lut->SetAlphaRange(1.0, 1.0);
    lut->SetNumberOfColors(256);
    lut->Build();
```

```
vtkLookupTable *lut = new vtkLookupTable;
    lut->SetNumberOfColors(256);
    lut->Build();
```

O trecho de código à esquerda, no qual são utilizados os valores *default* para os parâmetros matiz, saturação, luminância, transparência e para o tamanho da tabela, é equivalente ao trecho à direita. A tabela gerada por esse código é mostrada na figura 4.4a.

A geração de tabelas usando as primitivas do modelo RGB é feita especificando os componentes R, G, B de cada entrada da tabela individualmente. É preciso especificar o número

de entradas da tabela, construí-la e, então, inserir os componentes R, G e B e a transparência de cada cor. Desse modo, é possível gerar tabelas personalizadas que não podem ser expressas como funções lineares de valores HSV. O exemplo a seguir mostra a criação de uma tabela com 3 entradas, correspondendo às cores vermelha, verde e azul, na linguagem C++. A tabela gerada por esse trecho de código é ilustrada na figura 4.4b.

```
vtkLookupTable *lut = new vtkLookupTable;
lut->SetNumberOfColors(3);
lut->Build();
lut->SetTableValue(0, 1.0, 0.0, 0.0, 1.0);
lut->SetTableValue(1, 0.0, 1.0, 0.0, 1.0);
lut->SetTableValue(2, 0.0, 0.0, 1.0, 1.0);
```

Figura 4.4 – Tabelas de cores geradas pelo VTK

As *lookup tables* do VTK são associadas a mapeadores gráficos que criam uma tabela de cores padrão, que vai do vermelho ao azul, caso não seja definida uma tabela específica. Para aplicar uma tabela de cores específica, deve-se usar a operação *mapper*→*SetLookupTable(lut)*, em que *mapper* pode ser uma instância da classe *vtkMapper* ou uma de suas subclasses. Para evitar que o objeto seja colorido de acordo com seus valores escalares, pode-se usar o método *mapper*→*ScalarsVisibleOff()* para desligar o mapeamento de cores. Os valores mínimo e máximo da tabela de cores (*scalar range*) são especificados no mapeador através do método *mapper*→*SetScalarRange(min, max)*.

Os mapeadores usam sua própria *lookup table* para mapear os valores escalares em cores. Caso não seja especificado um conjunto de dados escalares, os mapeadores não controlam o mapeamento da cor do objeto. Nesse caso, o controle é feito pelo objeto *vtkProperty* associado à classe *vtkActor*. É preciso usar o método *actor*→*GetProperty()*→*SetColor(r, g, b)*, em que *r*, *g* e *b* são valores do tipo *float* que especificam uma cor no modelo RGB.

4.5 Módulo de Implementação do Mapeamento por Cores Baseado em Regras

– Classe *vtkRuleLookupTable*

A *vtkRuleLookupTable* foi implementada no VTK como uma classe derivada da *vtkLookupTable* e que possui métodos para gerar as tabelas de cores definidas segundo a taxonomia proposta por Rogowitz e outros [Rog93a; Rog93b], descrita na seção 3.4. A figura 4.5 ilustra os principais métodos e atributos da classe, utilizando a notação do OMT (*Object Modeling Technique*) [Rum91].

Figura 4.5 – Esquema da classe *vtkRuleLookupTable* (com os principais métodos e atributos)

As regras são implementadas pelos métodos *IsomorphicRules*, *SegmentationRules* e *HighlightingRules* e a geração das tabelas é feita pelos métodos *DefineIsomorphicTable*, *DefineSegmentationTable* e *DefineHighlightingTable*.

Existem outros métodos de apoio associados a essa classe, como um método de conversão de parâmetros de cor do modelo HSV para o modelo RGB; métodos que permitem a gravação e recuperação de tabelas de cores; e um método de mapeamento de valores de dados em índices da tabela. O método *HSVToRGB()*, baseado no algoritmo apresentado em Foley [Fol96], faz a conversão dos parâmetros H, S e V associados a cada entrada da tabela para os parâmetros R, G e B correspondentes. O método de conversão de parâmetros HSV para RGB da classe *vtkLookupTable* faz a conversão desses parâmetros para intervalos de variação de matiz, ou seja, apenas para seqüências de matizes e não para entradas individuais. O método *WriteRuleLookupTable(arquivo)* imprime em um arquivo as informações sobre o conjunto de dados que gerou a tabela e os parâmetros RGB de cada entrada da tabela. A tabela armazenada

em arquivo pode ser recuperada através do método *ReadRuleLookupTable(arquivo)* e aplicada a outros conjuntos de dados. O método *MapDataValueToIndex(valor)* verifica o intervalo de variação dos dados e estima a posição na qual o valor do dado será mapeado na tabela de cores do VTK. Esse método é utilizado na criação de tabelas de cores de segmentação e de ênfase, nas quais os segmentos ou intervalos são especificados em termos dos valores de dados que os delimitam. Foram implementados ainda outros métodos para validação de parâmetros fornecidos pelo usuário, para a criação e a verificação das tabelas, descritos no Apêndice deste trabalho.

Os possíveis matizes das cores nas tabelas são especificados através de códigos numéricos definidos pela classe, que variam de 1 a 6, e foram atribuídos de acordo com a ordem em que as cores são representadas em um círculo de matizes. Nesse círculo, no qual cada cor corresponde a um ângulo que varia de 0° a 360°, a cor vermelha corresponde a 0° ou 360° (código 1), amarela a 60° (código 2), verde a 120° (código 3), turquesa a 180° (código 4), azul a 240° (código 5) e magenta a 300° (código 6). Os códigos dos pares de cores opostas ou complementares são compostos pelos códigos de cada cor e não possuem uma ordem definida. Um par composto pelas cores verde e vermelha, por exemplo, é especificado pelo código 31, em que o 3 corresponde ao verde e o 1 ao vermelho.

Os metadados “tarefa de visualização” e tipo do dado também são especificados pelo programador através de códigos numéricos. Os códigos definidos para a especificação da tarefa são 1 para indicar tarefa de isomorfismo, 2 para tarefa de segmentação e 3 para tarefa de ênfase. A classe permite ainda a geração da tabela de cores do arco-íris (código 4) e a tabela com a escala de cinzas (código 5). O tipo do dado é definido pelos códigos 1 e 2, sendo que 1 especifica que os dados são do tipo razão e 2 especifica que os dados são do tipo intervalo. O intervalo de variação (*range*) de dados é obtido do próprio conjunto de dados, através da verificação do menor e maior valores no mesmo, e a frequência espacial dos dados é determinada pelos métodos da classe *SpatialFrequency*, descrita na seção 4.6.

Sempre que o usuário especifica um código incorreto, o programa envia uma mensagem de erro e assume um valor *default*. Os valores *default* para matiz são vermelho para tabelas formadas por um único matiz, e o par vermelho-verde para tabelas formadas por 2 matizes. O valor *default* para tarefa de visualização é 1 (tarefa de isomorfismo) e o *default* para o tipo do dado é 1 (tipo razão).

As tabelas de cores para a tarefa de isomorfismo são projetadas para produzir uma representação fiel das estruturas contidas nos dados – ou seja, incrementos iguais nos valores dos dados correspondem a incrementos perceptuais iguais na escala de cores.

O método privado *IsomorphicRules()* implementa as regras para a geração de tabelas de cores adequadas para a tarefa de isomorfismo, e o método privado *DefineIsomorphicTable()* constrói as tabelas (no modelo HSV) para essa tarefa. O usuário deve definir a tarefa de visualização através do método *SetVisualizationTask(tarefa)*, o número de entradas da tabela através do método *SetTableSize(num_entradas)*, o tipo do dado através do método *SetType(tipo)*, o matiz ou par de matizes através do método *SetTableColor(codigo)*, o nome do arquivo de dados a ser visualizado através do método *DataCharacteristics(arquivo)*, aplicar as regras através do método *ApplyRules()*, e em seguida construir a tabela (no modelo RGB) através do método *Build()*. O usuário pode definir valores de opacidade para o início e o fim da tabela através do método *SetTableOpacity(opc_ini, opc_fim)*. Esses valores variam entre 0 e 1 e são incrementados ou decrementados monotonicamente nas posições intermediárias da tabela. Um exemplo que ilustra a criação de uma tabela de cores para a tarefa de isomorfismo é fornecido a seguir:

```
vtkRuleLookupTable *lut = new vtkRuleLookupTable;  
lut->SetVisualizationTask(1);           //tarefa de isomorfismo  
lut->SetTableSize(50);                  //tamanho da tabela = 50  
lut->SetType(1);                        //tipo do dado = razão  
lut->SetTableColor(31);                  //cores da tabela = turquesa e vermelho  
lut->DataCharacteristics(arquivo)        //nome do arquivo visualizado  
lut->ApplyRules();  
lut->Build();
```

A tabela 4.1 ilustra algumas tabelas geradas pela classe *vtkRuleLookupTable* para a tarefa de isomorfismo, de acordo com o tipo e a frequência espacial dos dados.

Tabela 4.1 – Classificação de tabelas para a tarefa de isomorfismo

Tipo do Dado	Razão		Intervalo	
Frequência Espacial	Baixa	Alta	Baixa	Alta
Tabela Resultante	<div><div>n-1</div> <div>0</div></div>	<div><div>n-1</div> <div>0</div></div>	<div><div>n-1</div> <div>0</div></div>	<div><div>n-1</div> <div>0</div></div>

As tabelas para dados do tipo razão são formadas por 2 matizes que compõem pares de cores oponentes ou complementares (vermelho-verde, azul-amarelo, turquesa-vermelho, magenta-verde). Nesta implementação, essas tabelas são divididas em 2 porções de tamanhos iguais, cada uma associada a um matiz. Para dados de baixa frequência espacial, a luminância é mantida constante e a saturação possui valor máximo (1.0) nos extremos da tabela e valor mínimo (0.0) no meio da tabela. Para dados de alta frequência espacial, a saturação e a luminância possuem valor máximo (1.0) nos extremos da tabela e valor mínimo (0.0) no meio da tabela. As tabelas para dados do tipo intervalo são formadas por 1 único matiz. Para dados de baixa frequência espacial, a luminância é mantida constante (1.0) e a saturação varia monotonicamente (a partir de 0.0). Para dados de alta frequência espacial, a saturação é mantida constante (1.0) e a luminância varia monotonicamente (a partir de 0.0).

As tabelas para a tarefa de segmentação são projetadas para delinear regiões visualmente e, portanto, são compostas por segmentos de cores com características diferenciadas que devem corresponder a diferentes intervalos de interesse nos dados. A quantidade de segmentos que a tabela pode conter é definida de acordo com o tipo dos dados e a frequência espacial. Nesta implementação o número máximo de segmentos permitidos em qualquer caso foi fixado (arbitrariamente) em 6, pois considerou-se que um número maior poderia dificultar a distinção de segmentos na imagem resultante. Para cada segmento deve ser especificado o seu início; e os parâmetros matiz, saturação e luminância de cada cor. Nesta implementação, o início de cada segmento é especificado em termos dos valores de dados correspondentes ao início de um intervalo de interesse, sendo que o início do primeiro segmento deve corresponder ao menor valor do conjunto de dados. O final de cada segmento coincide com o início do próximo, sendo que o último vai até o maior valor do conjunto de dados.

O método privado *SegmentationRules()* define o número máximo de segmentos que a tabela de cores pode conter, de acordo com o tipo e a frequência espacial dos dados, e o método privado *DefineSegmentationTable()* constrói as tabelas (no modelo HSV) para essa tarefa. O usuário deve especificar a tarefa de visualização através do método *SetVisualizationTask(tarefa)*, o número de entradas da tabela através do método *SetTableSize(num_entradas)*, o nome do arquivo a ser visualizado através do método *DataCharacteristics(arquivo)*, o tipo do dado através do método *SetType(tipo)*, o número de segmentos através do método *SetNumberOfSegments(numero)*; o início de cada segmento (ou seja, o valor do dado a ser mapeado no início desse segmento da tabela), bem como os seus parâmetros de matiz, saturação

e luminância através do método *SetSegmentationTable(ini, codigo, sat, lum)*, aplicar as regras através do método *ApplyRules()* e construir a tabela (no modelo RGB) através do método *Build()*. Se o usuário especificar um número de segmentos incorreto, o programa envia uma mensagem de erro e assume um número de segmentos *default* que depende do tipo e da frequência espacial dos dados. Se o usuário não especificar um código de cor para algum segmento, ou se especificar o mesmo matiz mais de uma vez ou valores dos segmentos fora do intervalo de variação dos dados o programa envia uma mensagem de erro e assume a tabela de segmentação *default*. Caso o usuário especifique um valor de saturação ou de luminância inválido, o programa envia uma mensagem de erro e assume o valor 1 para ambos. O método *SetSegmentationTable(ini, codigo, sat, lum)* assume que o fim de um segmento coincide com o início do próximo. Através do método *SetSegmentationTable(ini, codigo, sat, lum, opc)*, o usuário pode definir também um valor de opacidade para cada segmento que varia entre 0 e 1. As informações sobre cada segmento são lidas pelo método privado *DefineSegmentationTable()* que constrói a tabela de cores. Para um conjunto de dados escalares inteiros, uma possível tabela de cores com 3 segmentos pode ser criada da seguinte forma:

```
vtkRuleLookupTable *lut = new vtkRuleLookupTable;
lut->SetVisualizationTask(2);           //tarefa de segmentação
lut->SetTableSize(50);                  //tamanho da tabela = 50
lut->SetType(2)                          //tipo do dado = intervalo
lut->SetNumberOfSegments(3);             //número de segmentos = 3
lut->DataCharacteristics(arquivo)         //nome do arquivo visualizado
lut->SetSegmentationTable(0,1,1,1); //início=0 cor vermelha, saturação e luminância=1
lut->SetSegmentationTable(4,2,1,1); //início=4, cor amarela, saturação e luminância=1
lut->SetSegmentationTable(8,5,1,1); //início=8, cor azul, saturação e luminância=1
lut->ApplyRules();
lut->Build();
```

A tabela 4.2 ilustra algumas tabelas de segmentação geradas pela classe *vtkRuleLookupTable* para a tarefa de segmentação, de acordo com o tipo e a frequência espacial dos dados.

Tabela 4.2 – Classificação de tabelas para a tarefa de segmentação

Tipo do Dado	Razão		Intervalo	
Frequência Espacial	Baixa	Alta	Baixa	Alta
Tabela Resultante

Se os dados são do tipo razão as tabelas são formadas por um número par de segmentos. Para dados de baixa frequência espacial pode ser aplicada uma tabela com vários segmentos, e para dados de alta frequência recomenda-se a utilização de tabelas com poucos segmentos.

As tabelas para a tarefa de ênfase são projetadas para atrair a atenção do usuário para regiões da imagem que apresentam certas características de interesse. No caso, o usuário deve identificar intervalos de valores dos dados a serem enfatizados perceptualmente na representação visual.

Um fator essencial para a criação de tabelas de cores que enfatizam um conjunto de dados é a não existência de interferência entre os diferentes componentes da cor utilizados numa representação. Se, por exemplo, o matiz for utilizado para diferenciar os elementos de um conjunto de dados, a variação de luminância ao longo de um intervalo faz com que este seja enfatizado sem que haja detrimento de outra informação. As tabelas para a tarefa de ênfase são compostas por 2 matizes, sendo que os parâmetros de saturação e luminância permanecem constantes ao longo da tabela. A taxonomia não faz restrições quanto ao tipo de dado para a criação de tabelas para essa tarefa, apenas sugere que o intervalo de ênfase seja menor para dados de alta frequência e maior para dados de baixa frequência. Nesta implementação, o usuário define um intervalo de valores a seu critério, ou seja, ele pode enfatizar uma única região de interesse e não se verifica que o intervalo é adequado em relação à frequência espacial dos dados.

O método privado *HighlightingRules()* define o número de matizes que a tabela de cores pode conter, que nesta implementação foi fixado (arbitrariamente) em 2, e o método privado *DefineHighlightingTable()* constrói as tabelas (no modelo HSV) para essa tarefa. O usuário deve especificar a tarefa de visualização através do método *SetVisualizationTask(tarefa)*, o número de entradas da tabela através do método *SetTableSize(tamanho)*, o nome do arquivo a ser visualizado através do método *DataCharacteristics(arquivo)*, o par de matizes que irá compor a tabela de cores através do método *SetTableColor(codigo)*, o intervalo de valores a ser enfatizado através do método *SetHighlightingInterval(inicio, fim)*, aplicar as regras através do método *ApplyRules()* e construir a tabela (no modelo RGB) através do método *Build()*. O intervalo de valores deve estar dentro do intervalo de variação do conjunto de dados, caso contrário o programa envia uma mensagem de erro e assume a tabela de ênfase *default*. A tabela *default*, que pode ser especificada através do método *SetDefaultHighlightingTable()*, é formada pelas cores azul e amarela e enfatiza uma faixa correspondente aos 10 valores centrais do

conjunto de dados. O usuário pode definir valores de opacidade para o intervalo enfatizado e para as outras entradas da tabela através do método *SetIntervalOpacity(intervalo, opc)*. A definição de uma tabela de cores para a tarefa de ênfase, para um conjunto de dados escalares inteiros com intervalo de variação de 0 a 100 e intervalo de valores a serem enfatizados entre 45 e 55, pode ser feita como mostra o exemplo a seguir:

```
vtkRuleLookupTable *lut = new vtkRuleLookupTable;
lut->SetVisualizationTask(3);           //tarefa de ênfase
lut->SetTableSize(50);                  //tamanho da tabela = 50
lut->SetType(2)                          //tipo do dado = intervalo
lut->SetHighlightingInterval(45,55);    //valores de dados enfatizados = 45 a 55
lut->SetTableColor(25);                  //cores da tabela = amarelo e azul
lut->DataCharacteristics(arquivo)        //nome do arquivo visualizado
lut->ApplyRules();
lut->Build();
```

A tabela 4.3 ilustra a tabela de ênfase gerada pela classe *vtkRuleLookupTable*.

Tabela 4.3 – Classificação de tabelas para a tarefa de ênfase

Tipo do Dado	Razão	Intervalo
Frequência Espacial	Baixa/Alta	
Tabela Resultante

Além da criação de tabelas de cores para as diferentes tarefas de visualização, a classe *vtkRuleLookupTable* permite que o usuário crie uma biblioteca de tabelas de cores através do método *WriteRuleLookupTable(arquivo)* que armazena em arquivo informações como o tamanho da tabela, o tipo e a frequência espacial dos dados que a geraram, e os valores das entradas na tabela de cores. Posteriormente, o usuário pode recuperar essas informações através do método *ReadRuleLookupTable(arquivo)* e reutilizar a tabela criada.

4.6 Módulo de Implementação do Cálculo da Frequência Espacial – Classe *SpatialFrequency*

O cálculo da frequência espacial dos dados é feito de forma aproximada. Aplica-se um filtro passa-baixa convencional [Gon78] generalizado para trabalhar com conjuntos de dados 2D e 3D. Essa operação de filtragem resulta em uma suavização das altas variações de frequência contidas no conjunto de dados. O filtro passa-baixa utiliza uma máscara de dimensão 3 e cria uma matriz composta pela média dos vizinhos da matriz original, como ilustra a figura 4.6.

Figura 4.6 – Ilustração do efeito da utilização de um filtro sobre uma matriz

A classe *SpatialFrequency* é independente da classe *vtkRuleLookupTable* e implementa métodos de leitura de arquivos de dados no formato VTK e de manipulação de matrizes de dados 2D e 3D. Essa classe consiste, basicamente, nos métodos de leitura do conjunto de dados, de filtragem e de análise da frequência espacial desenvolvidos na linguagem C++. A figura 4.7 ilustra os principais métodos e atributos da classe, utilizando a notação do OMT (*Object Modeling Technique*) [Rum91].

Figura 4.7 – Esquema da classe *SpatialFrequency* (com os principais métodos e atributos)

O método privado *ReadDataSet()* faz a leitura do arquivo de dados, armazena os valores lidos em uma matriz (2D ou 3D, de acordo com as dimensões especificadas no cabeçalho do arquivo) e obtém os valores mínimo e máximo do conjunto (intervalo de variação dos dados). A

operação de filtragem, implementada pelos métodos privados *SurfFilter()*, para conjuntos de dados bidimensionais, e *VolumeFilter()*, para conjuntos de dados tridimensionais, resulta na eliminação das frequências altas do conjunto de dados e o efeito dessa operação sobre a imagem resultante da visualização seria a suavização das bordas, que correspondem aos dados de alta frequência. O método público de análise da frequência, denominado *FrequencyAnalysis(stream)*, implementa uma operação que calcula a diferença entre a matriz que contém os dados originais e a matriz que contém os dados filtrados, e a seguir calcula o desvio padrão normalizado dessa diferença, segundo o intervalo de variação dos dados. O valor do desvio padrão será maior quando o conjunto de dados original contiver informação de alta frequência; e menor, caso contrário. Caso o resultado seja menor ou igual a 0.1, os dados são considerados como sendo de baixa frequência espacial; do contrário são considerados de alta frequência espacial, como sugerido por Bergman e outros [Ber95; Ber96]. Essa é uma forma aproximada de resolver o problema, que evita o cálculo de transformadas de Fourier. A análise seria mais precisa se realizada através de uma análise de Fourier, mas o custo computacional associado seria bastante alto.

4.7 Incorporação do código ao VTK

A classe *vtkRuleLookupTable* foi incorporada ao VTK de forma a torná-la disponível como parte do *toolkit*. Essa incorporação pode ser feita de 2 maneiras no ambiente do Visual C++: gerando bibliotecas de ligação dinâmica e bibliotecas de ligação estática que implementam as novas classes e acessando-as a partir de programas C++; ou recompilando todo o código do VTK com os novos fontes e bibliotecas (*.lib* e *.dll*) incluídas.

Para gerar as bibliotecas, as novas classes devem ser escritas na linguagem C++ e inseridas na hierarquia de classes do VTK. Cria-se um projeto do tipo MFC AppWizard (DLL), e adiciona-se todos os arquivos da nova classe, além da *vtkdll.lib*, de outras bibliotecas e outros fontes do VTK que são requisitadas pelo compilador. Após a compilação, são geradas uma biblioteca de ligação estática (*.lib*) e uma biblioteca de ligação dinâmica (*.dll*), que podem ser utilizadas por quaisquer outros programas desenvolvidos no ambiente Visual C++. Para isso, a biblioteca de ligação dinâmica deve ser inserida no diretório que contém o executável do programa de aplicação e a biblioteca de ligação estática deve ser adicionada ao projeto, juntamente com os fontes da aplicação.

4.8 Limitações da Implementação

Rogowitz e outros [Rog93a; Rog93b] definem o comportamento dos parâmetros de cor ao longo das tabelas de cores para cada tarefa de visualização. Alguns aspectos, como o tamanho dos intervalos nas tabelas de segmentação e de ênfase, e a especificação de valores de interesse para a tarefa de isomorfismo, não são abordados pelos autores. Além disso, a classificação dos tipos de dados apresentada é bastante restrita. Nesse sentido, há várias questões em aberto em relação a taxonomia apresentada. Neste trabalho, essas questões não foram abordadas, entretanto, devido às limitações próprias de um projeto de curto prazo. São ressaltadas a seguir algumas limitações adicionais introduzidas por esta implementação, que podem ser superadas em implementações posteriores.

- O tipo do dado deve ser especificado pelo usuário, pois considerou-se que não havia subsídios confiáveis para decidir por um tipo em particular analisando o conjunto.
- As tabelas de cores para a tarefa de isomorfismo e para dados do tipo razão possuem como limiar o valor central da tabela, que corresponderia ao zero. O ideal seria que o usuário escolhesse o limiar.
- O número de segmentos para a tarefa de segmentação foi fixado, arbitrariamente em 6 porque considerou-se que um número maior de segmentos poderia dificultar a identificação de segmentos na imagem gerada. Adicionalmente os parâmetros de saturação e luminância são mantidos fixos em 1.0. Entretanto, essas escolhas não são ideais para todos os casos.
- É permitida a especificação de apenas um intervalo de ênfase e não é feita uma verificação do tamanho desse intervalo em relação à frequência espacial do conjunto de dados, como sugerido na taxonomia da tabela 3.2.
- O número máximo de entradas da tabela foi fixado em 256. Essa restrição se deve à constatação de que o VTK trabalha de fato com tabelas de 256 entradas. Quando é especificado um número maior de entradas, o VTK faz o mapeamento linear das entradas em um intervalo de 256 entradas.

Várias dessas limitações poderiam ser superadas se o usuário pudesse alterar as tabelas através de uma interface gráfica interativa, por exemplo redefinindo o valor intermediário para tarefas de isomorfismo, ou alterando os segmentos nas tarefas de segmentação e ênfase de

acordo com suas necessidades. Essa interface é essencial para que o usuário possa “experimentalmente” tabelas distintas.

4.9 Considerações Finais

Este capítulo descreveu a implementação da abordagem baseada em regras de Rogowitz e outros [Rog93a; Rog93b] como uma nova classe do VTK. A utilização de diferentes tipos de tabelas de cores sobre um mesmo conjunto de dados produz resultados bem diferentes, e pode influenciar na interpretação feita pelo usuário. O módulo de mapeamento por cores baseado em regras tem o objetivo de facilitar o processo de geração de tabelas de cores adequadas aos dados.

5. TESTES

5.1 Considerações Iniciais

Alguns testes foram realizados com o objetivo de avaliar a implementação e verificar se a utilização das tabelas de cores específicas para as tarefas de isomorfismo, segmentação e ênfase pode produzir resultados mais significativos do que a utilização da tabela de cores *default* do VTK. Foram utilizados conjuntos de dados variados com o objetivo de ilustrar aplicações dos diferentes tipos de “tarefas de visualização”. Alguns dos conjuntos de dados foram obtidos a partir de imagens no formato GIF, as quais foram “transformadas” em um arquivo do tipo malha regular do VTK através de um conversor.

Este capítulo apresenta os resultados obtidos com a aplicação de diferentes tipos de tabelas de cores geradas pela classe *vtkRuleLookupTable* a alguns conjuntos de dados. Na seção 5.2 é descrito o conversor desenvolvido que, a partir de imagens do formato GIF gera conjuntos de dados no formato malha estruturada do VTK, e é apresentada também uma descrição sucinta sobre o formato de arquivos do VTK. A seção 5.3 apresenta uma descrição de cada conjunto de dados utilizados nos testes e os resultados obtidos em cada caso.

5.2 Conversor GIF-VTK

O conversor foi desenvolvido a partir do código fonte obtido da rede Internet¹, com o objetivo de aumentar o conjunto de dados de teste disponíveis para a avaliação do módulo implementado. Basicamente, esse conversor faz a leitura de imagens GIF e gera arquivos de dados no formato malha estruturada do VTK (*vtkStructuredPoints*).

¹ <http://www.dcs.ed.ac.uk/~mxr/gfx/utils-lo.html>

Um arquivo GIF é composto pelas seguintes seções: descritor da tela do monitor (largura e altura da tela, etc.), descritor da imagem (largura e altura da imagem, etc.), tabelas de cores, dados de varredura e, opcionalmente, outras informações específicas de uma aplicação. As informações utilizadas para a criação do arquivo de dados VTK são os dados do descritor da imagem e os dados de varredura. Os dados do descritor (que já estão no formato decimal) são lidos do arquivo e os dados de varredura, que estão comprimidos usando o algoritmo de codificação LZW, são descomprimidos e armazenados em arquivo. A compressão LZW é feita da seguinte forma: toda vez que um padrão (que pode ser um valor ou uma sequência de valores) é lido pela primeira vez, o codificador o associa a um código e os insere (o padrão e o código) em uma lista. Quando um padrão é lido pela primeira vez, ele é impresso em arquivo sem ser codificado. Dessa forma, o decodificador constrói uma lista de códigos, a partir dos primeiros valores lidos em arquivo, que será idêntica à lista criada pelo codificador. Essas listas serão idênticas porque o conjunto de valores que compõem o conjunto é definido na seção de dados de varredura [Bro95].

A adaptação do código original consistiu na correção de alguns erros de compilação e na inclusão de uma rotina para conversão dos dados de varredura que estavam no formato hexadecimal para valores no formato decimal. Os valores dos pixels contidos nas imagens representadas em arquivos GIF são usados como índices para uma tabela de cores especificada no modelo RGB. Na conversão para um arquivo de dados, esses índices são tratados como valores escalares inteiros definidos sobre uma malha regular, e mapeados utilizando as tabelas de cores geradas pela classe *vtkRuleLookupTable*.

O Formato dos Arquivos de Dados do VTK

Um objeto fonte do VTK faz a leitura dos dados de entrada e os disponibiliza para os demais objetos no *pipeline* de visualização. Esses dados, que podem ser lidos de arquivos, possuem características fundamentais para a escolha do melhor método de representação gráfica. Essas características são a natureza discreta do conjunto de dados, a sua regularidade e a sua dimensão topológica dada pelo número de parâmetros necessários para endereçar o domínio de um objeto.

O conjunto de dados pode ser regular (estruturado), quando existe uma relação inerente entre os seus pontos; ou irregular (não estruturado). O armazenamento de dados em malhas regulares requer menos memória porque são armazenados apenas a posição inicial do intervalo,

o espaçamento entre os pontos e o número total de pontos. Não há necessidade de armazenar as coordenadas dos pontos, pois elas são implicitamente conhecidas. As malhas irregulares permitem a representação de dados que não possuem padrões regulares de organização e oferecem mais liberdade na representação dos dados.

As células são os elementos de construção fundamentais dos sistemas de visualização. No VTK, as células são definidas através da especificação de um tipo e de uma lista ordenada de pontos (também chamada de lista de conectividade) que define implicitamente a topologia das mesmas. A figura 5.1 ilustra alguns tipos de células.

Figura 5.1 – Exemplos de tipos de células [Sch96]

Os atributos dos dados são informações (por exemplo, temperatura ou velocidade em um ponto) que devem ser representadas em uma imagem. A figura 5.2 mostra alguns exemplos de atributos de dados utilizados pelo VTK. Os escalares, que são os dados mais simples e a forma mais usual de dados a serem visualizados, associam uma informação a cada ponto da imagem. Exemplos de dados escalares são valores que correspondem à temperatura, densidade ou pressão.

Figura 5.2 – Exemplos de atributos de dados [Sch96]

Os objetos que representam os dados no *pipeline* de visualização são chamados de *datasets* e consistem de uma estrutura de organização dos dados e de atributos de dados associados a essa estrutura. A estrutura descreve a topologia, que corresponde ao elemento de construção fundamental de imagens; e a geometria, que especifica as coordenadas de cada ponto da célula. A figura 5.3 apresenta alguns exemplos de tipos de *datasets* do VTK.

Figura 5.3 – Exemplos de tipos de *datasets* [Sch96]

O *dataset* do tipo *structured points*, que é o tipo mais comum em visualização, é formado por células cujas linhas, colunas e planos são paralelos ao sistema de coordenadas globais x, y e z. O formato de arquivo de dados definido pelo VTK consiste de 5 partes dispostas na seguinte ordem: versão do arquivo e um identificador, cabeçalho de 256 caracteres, formato do arquivo, estrutura do *dataset* e atributos dos dados (escalares, vetores de tensores associados aos vértices de cada célula).

5.3 Casos de Teste e Resultados

Nesta seção são apresentados 5 casos de teste aos quais foram aplicadas as tabelas geradas pela classe *vtkRuleLookupTable* e os resultados obtidos. Desses, 3 consistem de conjuntos de dados bidimensionais e 2 de conjuntos de dados tridimensionais, todos definidos em malhas regulares. Todos os volumes de dados contém atributos do tipo escalar. Em cada caso, uma comparação é feita entre as visualizações obtidas utilizando as tabelas oferecidas pela classe *vtkRuleLookupTable* e a visualização obtida com a tabela de cores *default* do VTK (a escala de cores do arco-íris), que é a tabela disponível para uso imediato e cuja utilização não requer esforço adicional por parte do usuário. Em alguns casos, também são exibidas, para efeito de comparação, as imagens obtidas com tabelas de cores definidas originalmente por usuários, sendo que nesse caso, em geral, não se conhece o esforço necessário para gerar essas tabelas e nem o grau de satisfação dos usuários com as mesmas.

Caso 1 – Dados de Tomografia Computadorizada

Esse conjunto de dados médicos obtidos através de tomografia computadorizada e foi cedido pelo MPIG (*Medical Image Processing Group*), do Departamento de Radiologia, Universidade da Pensilvânia, para o grupo de Computação Gráfica e Processamento de Imagens do ICMC-USP. Os dados foram disponibilizados na forma de imagens GIF, sendo destinadas em princípio à reconstrução tridimensional [Pra96], e não à análise de informações de interesse. Algumas imagens foram selecionadas para os testes, e os resultados obtidos com uma delas são apresentados a seguir.

Figura 5.4 – Reprodução da imagem original utilizando uma escala de cinzas

A figura 5.4 reproduz a imagem original utilizando uma escala de cinzas, que é a tabela mais utilizada para representar imagens nesse domínio de aplicação específico. A imagem GIF original foi convertida para um conjunto de dados do tipo malha regular 2D do VTK, utilizando o conversor descrito na seção 5.2.

Figura 5.5 – Aplicação de diferentes tabelas de cores a dados de tomografia computadorizada

As figuras 5.5a a 5.5d ilustram os resultados obtidos com a aplicação das tabelas *default*, isomórfica, de segmentação e de ênfase a esse conjunto de dados, sendo que a única técnica de visualização utilizada foi a de mapeamento por cores dos valores escalares. Esse conjunto de dados contém valores escalares do tipo inteiro variando de 1 a 101 e foi classificado como sendo

de alta frequência espacial, do tipo intervalo. Foram utilizadas tabelas de cores com 101 entradas, de forma que cada valor escalar fosse mapeado em uma posição diferente da tabela. Os segmentos e o intervalo de ênfase foram definidos aleatoriamente para os testes, e não indicam necessariamente valores significativos do ponto de vista do usuário.

A imagem ilustrada na figura 5.5a foi gerada com a aplicação da tabela de cores *default* do VTK. A utilização dessa tabela cria regiões na imagem, dando a impressão de que existe uma diferença significativa entre os valores dos dados em regiões adjacentes, o que nem sempre é o caso, como por exemplo na vizinhança entre o amarelo e o verde, ou o verde e o azul. A introdução desse tipo de artefato visual é comum quando a tabela do arco-íris é usada sem uma análise de sua adequação à natureza dos dados.

A imagem gerada pela tabela isomórfica (ilustrada na figura 5.5b) apresenta variações suaves, e é semelhante à imagem gerada com a utilização de uma escala de cinzas (ilustrada na figura 5.4). Essa tabela possui variação constante na luminância do amarelo, sendo que os valores próximos de 1 estão representados pela cor amarela com valores de luminância próximos de 0 e os valores próximos de 101 estão representados pela cor amarela com valores de luminância próximos de 1.

A tabela de segmentação permite a distinção de intervalos de interesse que correspondem a valores específicos dos dados que aparecem em regiões claramente definidas na imagem. Na imagem ilustrada na figura 5.5c, a cor turquesa representa valores entre 1 e 34; a cor vermelha representa valores entre 35 e 69; a cor amarela representa valores entre 70 e 99; e a cor azul representa os valores 100 e 101, que correspondem ao fundo.

A tabela de ênfase permite que um intervalo de valores de interesse nos dados seja enfatizado na imagem. Na imagem ilustrada na figura 5.5d o intervalo de valores enfatizados varia de 70 a 99, e está representado na cor amarela. A cor azul representa os demais valores de dados.

Caso 2 – Dados de Sensoriamento Remoto

Os conjuntos de dados foram gerados a partir de 3 imagens GIF que mostram uma área de plantação de soja de uma região do Estado do Mato Grosso do Sul. Essas imagens fazem parte de uma sequência de 4 imagens que mostram as diferentes fases do cultivo de soja (que

incluem a preparação do solo, cultivos irrigados na fase adulta e a colheita) durante o período de 15 de Novembro de 1990 a 26 de Maio de 1991. As imagens foram adquiridas por satélite, e pertencem ao INPE (Instituto de Pesquisas Espaciais), tendo sido obtido via Internet².

As imagens GIF originais estão ilustradas nas figuras 5.6a, 5.6b e 5.6c. Essas imagens foram convertidas para conjuntos de dados do tipo malha regular 2D do VTK, e os dados visualizadas através de mapeamento por cores. Esse conjunto de dados foi classificado como sendo de alta frequência espacial, do tipo intervalo e contém valores escalares do tipo inteiro variando de 0 a 255.

Figura 5.6 – Sequência de imagens de sensoriamento remoto

A figura 5.6a mostra uma imagem da área de plantação de soja adquirida em 15 de Novembro de 1990. Nessa imagem, a cor azul representa as áreas com solo preparado para cultivo, a cor verde representa as áreas de cultivo irrigado na fase adulta, e a cor rosa representa áreas de pastagem.

A figura 5.6b mostra uma imagem da mesma região ilustrada na figura 5.6a, adquirida em 19 de Fevereiro de 1991. Nessa imagem, as áreas representadas pela cor verde (indicando cultivo na fase adulta), correspondem às áreas representadas pela cor azul (solo preparado para cultivo) ilustradas na figura 5.6a. A cor magenta representa áreas nas quais o cultivo de soja já foi colhido, e as áreas de pastagem.

A figura 5.6c mostra uma imagem da mesma área adquirida em 26 de Maio de 1991. A cor marrom representa as áreas de cultivo de soja após a colheita.

² As imagens não são de domínio público e foram disponibilizadas exclusivamente como exemplo de imagens obtidas por satélites. É vedada a sua reprodução, venda ou distribuição dessa imagem, sob qualquer meio digital ou fotográfico, bem como a sua divulgação por outros meios eletrônicos (tais como BBS) e seu uso para fins comerciais. O site é <http://www.dgi.inpe.br/html/imagens.htm>

Figura 5.7 – Aplicação de diferentes tabelas de cores a dados de sensoriamento remoto

A figura 5.7a1 mostra o resultado obtido com a aplicação da tabela de cores *default* ao conjunto de dados obtido a partir da imagem ilustrada na figura 5.6a. A aplicação dessa tabela faz com que mais de uma informação seja representada pela mesma cor, ou que a mesma informação seja representada por mais de uma cor. Nessa imagem, as áreas de pastagem e de cultivo irrigado estão representadas pela cor amarela; e as áreas de cultivo irrigado estão representadas pelas cores amarela e verde. A figura 5.7a2 ilustra o resultado obtido com a aplicação de uma tabela com 3 segmentos, representados pelas cores verde, azul e amarela. A cor verde representa valores entre 0 e 100 e corresponde às áreas com cultivo irrigado na fase adulta;

a cor azul representa valores entre 101 e 200 e corresponde às áreas de solo preparado para cultivo; a cor amarela representa valores entre 201 e 255 e corresponde às áreas de pastagem. Nesse caso, a distinção em relação à imagem original está unicamente na escolha das cores: a imagem 5.7a2 contém a mesma informação da imagem 5.6a, mapeada através de outras cores. Esse também é o caso das figuras 5.7b2 e 5.6b.

A figura 5.7b1 ilustra os resultados obtidos com a aplicação da tabela de cores *default* ao conjunto de dados obtido a partir da imagem na figura 5.6b. A aplicação dessa tabela dá a impressão de que existe uma diferença grande de valores em uma mesma região. Nessa imagem, as regiões correspondentes às áreas de cultivo irrigado, por exemplo, estão representadas pelas cores amarela e vermelha. A figura 5.7b2 ilustra a imagem obtida com a aplicação de uma tabela com 3 segmentos, representados pelas cores verde, azul e amarela. A cor verde representa valores entre 0 e 100 e corresponde às áreas com cultivo irrigado na fase adulta; a cor azul representa valores entre 101 e 170 e corresponde às áreas de solo preparado para cultivo; a cor amarela representa valores entre 171 e 255 e corresponde às áreas onde o cultivo já foi colhido e as áreas de pastagem.

A figura 5.7c1 mostra o resultado produzido com a aplicação da tabela de cores *default* ao conjunto de dados visualizado na figura 5.6c. A aplicação dessa tabela faz com que a mesma informação seja mapeada em mais de uma cor. Por exemplo, as áreas em que houve colheita estão representadas pelas cores turquesa e azul. A figura 5.7c2 mostra o resultado da aplicação de uma tabela de ênfase. A cor vermelha representa os valores entre 0 e 50 e entre 180 e 255, e corresponde às áreas de cultivo após a colheita. Nesse caso, a tabela foi escolhida para enfatizar as regiões em que houve colheita, sendo que a informação enfatizada foi mapeada na cor vermelha.

Caso 3 – Dados de Topografia

O terceiro caso de teste consiste de dados topográficos do continente africano. Os dados foram coletados, interpolados em uma malha regular e visualizados através de mapeamentos por cores. Esse conjunto de dados bidimensional foi obtido da rede Internet³ e convertidos para o formato de arquivos do VTK.

³ <http://cres.anu.edu.au/software/africatxt.html>

Figura 5.8– Aplicação de diferentes tabelas de cores a dados topográficos

A figura 5.8 ilustra os resultados obtidos com a aplicação de diferentes tabelas cores a esse conjunto de dados. Os valores escalares originais foram mapeados em cores utilizando diferentes tabelas. Os dados foram classificados como de alta frequência, do tipo intervalo e contém valores escalares do tipo *float* variando de 0.01 a 0.9. Foram utilizadas tabelas com 20 entradas.

A figura 5.8a ilustra o resultado obtido com a aplicação da tabela de cores *default* do VTK. A utilização dessa tabela mostra regiões na imagem, dando a impressão de que existe uma diferença significativa de valores em regiões de cores muito diferentes.

A figura 5.8b mostra a imagem obtida com a aplicação da tabela isomórfica. Nessa imagem, a cor verde com valores de luminância menores representa alturas menores; e a cor verde com valores de luminância maiores (próximos a 1.0) indica alturas maiores. A figura 5.8c mostra o resultado obtido com a aplicação de uma tabela com 3 segmentos, em que o objetivo é

identificar as regiões mais altas. A cor azul representa valores entre 0.01 e 0.39 e corresponde ao fundo; a cor amarela representa valores entre 0.40 e 0.69; e a cor vermelha representa valores entre 0.70 e 0.90.

Caso 4 – Dados Marítimos

O quarto caso de teste consiste de um conjunto de dados que contém diferentes níveis de concentração da substância química *Dimethyl Sulphide* (DMS) nas águas do Mar do Norte ao longo de um período, obtidas através de medições [Day97].

Figura 5.9 – Aplicação de diferentes tabelas de cores a dados marítimos

A figura 5.9 ilustra os resultados da aplicação de diferentes tabelas de cores a esse conjunto de dados, que foi classificado como de baixa frequência espacial, do tipo intervalo e contém valores escalares do tipo *float* variando de 0 a 31. Os dados foram coletados, interpolados e visualizados através de renderização volumétrica direta. Foram utilizadas tabelas de cores com 32 entradas.

A figura 5.9a mostra o resultado obtido com a aplicação da tabela de cores *default* do VTK. O conjunto de dados possui valores muito próximos que são mapeados em uma única região ou em regiões próximas da tabela, dificultando a análise do comportamento do DMS. A análise tem o objetivo de detectar alterações temporais e posicionais na distribuição dessa substância nas águas do Mar do Norte.

A figura 5.9b mostra o resultado obtido com a aplicação de uma tabela com 6 segmentos representados pelas cores turquesa, azul, verde, amarela, magenta e vermelha. A cor turquesa foi associada a um valor de opacidade 0.0 e representa valores de 0.0 a 0.9 que foram renderizados de forma transparente; a cor azul possui valor de opacidade 0.1 e representa valores de 1.0 a 2.9; a cor verde possui valor de opacidade 0.2 e representa valores de 3.0 a 4.9; a cor amarela possui valor de opacidade 0.3 e representa valores de 5.0 a 7.9; a cor magenta possui valor de opacidade 0.4 e representa valores de 8.0 a 15.9; e a cor vermelha possui opacidade 0.5 e representa valores de 16.0 a 31.0.

A figura 5.9c mostra o resultado obtido com a aplicação da tabela de ênfase. O intervalo enfatizado corresponde os valores de 8.0 a 16.0 (em magenta na imagem representada na figura 5.9b) e foi representado na cor turquesa, com valor de opacidade 0.5. Os outros valores foram mapeados na cor vermelha com valor de opacidade 0.0 (tornando-os transparentes).

Caso 5 – Dados de Neurônios

O quinto caso de teste consiste de um conjunto de dados obtidos a partir da síntese da forma de estruturas neurais. A síntese é realizada através de gramáticas estocásticas que operam sobre uma série de regras de produção bem definidas, derivadas a partir de caracterizações estatísticas e fractais de cada tipo de neurônio [Ces97]. Esse conjunto de dados tridimensional foi cedido pelo grupo de Visão Cibernética do IFSC-USP (Instituto de Física de São Carlos da Universidade de São Paulo).

O conjunto de dados original consiste em uma descrição do neurônio na forma de uma poligonal bi ou tridimensional definida em um espaço discreto. Nesse espaço discreto, valores 0 ou 1 são colocados para indicar a presença ou ausência do neurônio, respectivamente. Os dados foram processados através da aplicação de uma Gaussiana, gerando valores escalares entre 0 e 1. Para manipulação no VTK, foi gerada uma malha regular, sendo que nos pontos introduzidos para tornar a malha regular foi inserido um valor inválido, no caso 2.

Figura 5.10 – Visualização de neurônio utilizando uma escala de cinzas

A figura 5.10 mostra a imagem resultante da aplicação de uma escala de cinzas a um desses conjuntos de dados.

A figura 5.11 mostra os resultados obtidos com a aplicação de diferentes tabelas de cores aos dados. Essas imagens foram geradas através de renderização volumétrica direta dos dados. O conjunto de dados foi considerado de alta frequência espacial, do tipo intervalo e contém valores escalares do tipo *float* variando de 0 a 2. Foram utilizadas tabelas de cores com 20 entradas.

Figura 5.11 – Aplicação de diferentes tabelas de cores a dados de neurônios

A imagem da figura 5.11a mostra o resultado obtido com a aplicação da tabela de cores isomórfica. Foi definida uma tabela com a cor turquesa e valores de opacidade decrescentes variando de 0.8 a 0.0 (para o valor 2.0, que nesse caso corresponde ao “fundo” e não constitui informação relevante). Os valores próximos de 0 estão representados pela cor turquesa, com valores de luminância próximos ou iguais a 0.0, e valores de opacidade próximos ou iguais a 0.8. Os valores próximos de 2 estão representados pela cor turquesa com valores de luminância próximos ou iguais a 1.0 e valores de opacidade próximos ou iguais a 0.0.

A figura 5.11b mostra a imagem gerada com a aplicação de uma tabela de segmentação com as cores vermelha, amarela, turquesa e azul. A cor vermelha possui valor de opacidade 0.6 e representa valores entre 0.0 e 0.79; a cor amarela possui valor de opacidade 0.3 e representa valores entre 0.80 e 0.89; a cor turquesa possui valor de opacidade 0.2 e representa valores entre 0.90 e 0.99; e a cor azul possui valor de opacidade 0.0 e representa valores entre 1.0 e 2.0.

A figura 5.11c mostra a imagem resultante da aplicação de uma tabela de ênfase. O intervalo de valores enfatizado varia de 0.90 a 1.0 e está representado pela cor vermelha com valor de opacidade 0.3. Os demais valores da tabela foram mapeados na cor turquesa com valor de opacidade 0.0, tornando-os transparentes.

5.4 Considerações Finais

A utilização de tabelas de cores para diferentes “tarefas de visualização” pode facilitar o processo de criação de visualizações. A aplicação da tabela isomórfica produz imagens que permitem a verificação do comportamento de variação dos dados. A aplicação de tabelas de segmentação permite a identificação de regiões de interesse nos dados e sua separação em regiões bem delimitadas na imagem resultante. Já a tabela de ênfase permite que o usuário destaque um intervalo de valores específicos dos dados na imagem resultante. Observa-se que a definição do número de entradas da tabela pode influenciar de modo significativo a imagem resultante. A partir da verificação do intervalo de variação e do tipo dos dados, é possível maximizar a utilização das entradas de uma tabela.

Os resultados obtidos não são conclusivos, devido à própria natureza dos testes e ao número limitado de casos de teste. Acredita-se que os resultados seriam mais significativos se as tabelas de cores tivessem sido definidas por usuários do domínio de cada aplicação, e se as imagens resultantes tivessem sido analisadas pelos mesmos. Apesar disso, os testes serviram para detectar falhas na implementação e para ilustrar a potencialidade da abordagem, pois indicam que as tabelas geradas podem ser mais efetivas que a tabela *default* ou tabelas geradas de forma *ad hoc*. Adicionalmente, a geração dessas tabelas oferecem um ponto de partida ao usuário, ainda que não sejam ideais, e podem ser geradas com pouco esforço por parte do usuário.

6. CONCLUSÕES

6.1 Considerações Iniciais

O uso de cores em Visualização influencia de modo significativo a interpretação de uma imagem. Uma tabela de cores mal projetada pode impossibilitar a identificação de estruturas contidas na imagem resultante, ao passo que uma tabela bem projetada pode permitir a identificação e a análise dessas estruturas. Entre os fatores que dificultam a definição de uma tabela apropriada estão incluídos o número de entradas, a variação dos parâmetros de cor, o intervalo de valores de dados mapeados, e a relação entre o número de entradas da tabela e a quantidade de valores distintos contidos no conjunto de dados. A maioria dos sistemas atuais não oferece ajuda sobre como gerar mapas de cores apropriados para a visualização de um conjunto de dados e as ferramentas que esses sistemas disponibilizam podem gerar tabelas de cores mal projetadas, apesar de existirem diversas abordagens que tentam incluir ajuda aos usuários na definição de mapeamentos para a geração de visualizações adequadas.

6.2 Contribuições, Limitações e Sugestões de Trabalhos Futuros

Neste trabalho, foi implementada uma abordagem baseada em regras para apoiar a geração de tabelas de cores para visualização de dados escalares. A taxonomia de regras utilizada foi obtida da literatura consultada. Entre as vantagens dessa abordagem pode-se citar a sua generalidade, pois não se restringe a uma classe de técnicas ou de domínios de aplicação específicos, e o fato de deixar o controle da escolha de mapeamentos a cargo prioritariamente do usuário, e não do sistema. Adicionalmente, a implementação é relativamente simples, desde que se tenha o conhecimento e o embasamento necessários para a definição das regras, e é modular, pois pode ser feita de forma relativamente independente para os diferentes tipos de

mapeamentos. Uma análise da interação entre os diferentes mapeamentos se faz necessária, entretanto, no caso da utilização de múltiplos mapeamentos em uma visualização. Vale também observar que a definição de regras adequadas não é uma tarefa simples.

A implementação da classe *vtkRuleLookupTable* garantiu a incorporação ao VTK de um tipo de apoio que o software não oferece. As tabelas geradas, ainda que não sejam ideais em qualquer situação, oferecem uma opção inicial melhor do que a tabela *default* para que o usuário possa chegar em um mapeamento que considere adequado, podendo diminuir o tempo necessário para tal.

Nesse sentido, uma extensão essencial para esse trabalho é o desenvolvimento de uma interface gráfica interativa para a definição e alteração das tabelas. Essa interface permitirá a imediata aplicação das tabelas aos dados, seguida de uma avaliação de resultados e redefinição de parâmetros, se necessário. Ao redefinir parâmetros, o usuário poderia ser alertado caso viole a taxonomia das regras. Outra extensão seria gerar e deixar disponíveis, eventualmente através da mesma interface gráfica, uma biblioteca de tabelas de cores que o usuário poderia selecionar para aplicar aos seus dados, como sugerido por Bergman e outros [Ber95]. As tabelas da biblioteca também poderiam ser editadas ou alteradas interativamente.

Como contribuições adicionais, foram implementados métodos para salvar e recuperar tabelas de cores em arquivos, permitindo a sua utilização e a criação de uma “biblioteca” de tabelas que pode ficar disponível para os usuários do VTK, além da classe que analisa de forma aproximada a distribuição de frequência espacial dos dados.

Além da ausência de uma interface gráfica o módulo desenvolvido possui outras limitações, entre as quais pode-se citar: o universo de cores disponíveis está restrito ao espaço de cores do modelo HSV; o número de segmentos em uma tabela de segmentação possui um valor máximo fixado pela implementação; as tabelas de ênfase permitem enfatizar um único intervalo de dados; há um número limitado e restrito de pares de cores disponibilizadas para as tabelas de isomorfismo e de ênfase, o que poderia ser flexibilizado (por exemplo para considerar aspectos relacionados ao domínio da aplicação).

A utilização do modelo de cor HSV permite que as características de saturação e de luminância de uma cor sejam definidas separadamente, mas impossibilita a especificação de cores não espectrais como o marrom e o violeta, por exemplo. As limitações no número máximo de segmentos em tabelas de segmentação e no número de intervalos enfatizados em uma tabela

de ênfase são muito rígidas, podendo existir casos em que seria interessante oferecer maior flexibilidade. A utilização de pares de cores opostas ou complementares nas tabelas associadas às tarefas de ênfase e de isomorfismo garante a utilização de cores perceptualmente distintas, por outro lado, restringe o conjunto de pares de cores que podem ser utilizados.

A flexibilização na escolha de pares de cores possibilitaria a utilização de cores específicas de um domínio de aplicação, e poderia ajudar no reconhecimento das estruturas caso o usuário já tenha observado outras imagens relacionadas ao fenômeno em estudo em que outras escolhas de cores tenham sido adotadas. Nesse caso, uma solução mais adequada seria manter a cor de referência e buscar como par uma cor a uma distância perceptual suficientemente grande. Uma solução desse tipo se beneficiaria de estudos sobre espaços de cores perceptualmente uniformes e o seu uso em visualização. O conjunto de cores poderia ser definido e alterado através da utilização de uma interface gráfica que permitisse que o usuário interaja com os parâmetros de cor e com as tabelas geradas. Pode-se também elaborar estratégias para identificar valores de dados que correspondem ao fundo de uma imagem e evitar que esses valores sejam mapeados em uma cor que deveria ser utilizada para representar informações de interesse na imagem resultante da visualização. Assim, pode-se considerar como as regras poderiam ser estendidas para incorporar estes e outros aspectos relacionados ao domínio da aplicação.

A variação apenas do matiz nas tabelas de segmentação e de ênfase é uma maneira de garantir que os intervalos de valores de interesse sejam mapeados em cores que não sejam parecidas (perceptualmente próximas), mas pode ser interessante permitir também a variação dos parâmetros saturação e luminância dentro de cada intervalo. Com isso, o usuário poderia enfatizar um intervalo de valores e representar a variação de valores desse intervalo.

Finalmente, a análise da distribuição de frequência dos dados é bastante simplificada. Seria interessante analisar o seu comportamento para verificar se a solução empírica adotada produz de fato resultados adequados na prática, como afirmado em Bergman e outros [Ber95]. Desenvolvimentos futuros poderiam considerar a implementação de uma abordagem mais precisa.

Os testes iniciais realizados neste trabalho não avaliaram extensivamente a abordagem adotada e nem o conjunto de regras implementadas, mas serviram para mostrar a potencialidade da abordagem. Os resultados obtidos com a aplicação das diferentes tabelas de cores geradas pelo módulo de mapeamento baseado em regras comprovam que diferentes tipos de tabelas

influenciam de modo significativo a imagem resultante. Futuramente, testes adicionais com usuários de um domínio de aplicação com o objetivo de avaliar a abordagem em contextos reais podem ser realizados.

A escolha do VTK como plataforma de software de desenvolvimento garantiu não apenas que a implementação pudesse se beneficiar do conjunto de métodos já disponíveis, como também garantiu uma implementação modular, baseada no paradigma de orientação a objetos, que pode ser alterada e modificada com relativa facilidade. Adicionalmente, torna o módulo disponível para um grande conjunto de usuários, pois o VTK é um software de baixíssimo custo e bastante acessível.

Assim, acredita-se que o módulo desenvolvido oferece um ponto de partida para que o grupo de Computação Gráfica e Processamento de Imagens do ICMC-USP possa continuar as pesquisas em andamento sobre aspectos de percepção visual e sua influência em visualização.

7. BIBLIOGRAFIA

- [Air85] Aires, A. M. A. et al; "Sistema Sensoriais" - in *Fisiologia Básica*, (cap. 4), Guanabara Koogan, 1985.
- [Ber89] Bergeron, R. D.; Grinstein, G. G. – "A Reference Model for the Visualization of Multi-Dimensional Data", in *Eurographics Association*, Hansmann et al (Eds), 1989, pp. 393-399.
- [Ber95] Bergman, L. D.; Rogowitz, B. E.; Treinish, L. A. - "A Rule-based Tool for Assisting Colormap Selection", in *Proceedings of the IEEE Computer Society Visualization '95 Conference*, Maio, 1995, pp. 118-125.
- [Ber96] Bergman, L. D.; Rogowitz, B. E.; Treinish, L. A. - "A Rule-based Tool for Assisting Colormap Selection", 1996, pp. 1-11, at <http://www.almaden.ibm.com/dx/vis96/proceedings/PRAVDA/index.htm>.
- [Bes93] Beshers, C.; Feiner, S. - "AutoVisual: Rule-Based Design of Interactive Multivariate Visualizations", *IEEE Computer Graphics & Applications*, Vol. 13, No. 4, Julho, 1993, pp. 41-49.
- [Bes94] Beshers, C. G.; Feiner, S. K. - "Automated Design of Data Visualizations", in *Scientific Visualization - Advances and Challenges*, (cap. 6), Academic Press, 1994, pp. 88-102.
- [Bro88] Brooks Jr., F. P. - "Grasping Reality Through Illusion- Interactive Graphics Serving Science", in *Proceedings of ACM CHI'88*, 1988, pp. 1-11.
- [Bro92] Brodlie et al (Eds) - *Scientific Visualization – Techniques and Applications*, Springer-Verlag, 1992.
- [Bro95] Brown, C. W.; Shepherd, B. J. – *Graphics File Formats*, Prentice-Hall, 1995.
- [Cas91] Casner, S. - "A Task-Analytic Approach to the Automated Design of Graphic Presentations", in *ACM Transactions on Graphics*, Vol. 10, No. 2, 1991, pp. 111-151.
- [Ces97] Cesar Jr., R. M.; Costa, L. F. - "Semi-automated Dendogram Generation for Neural Shape Analysis", in *Proceedings of Sibgraph 97*, IEEE CS Press, 1997, pp. 147-154.

- [CG87] CM SIGGRAPH Computer Graphics, *Special Issue on Visualization in Scientific Computing*, Vol. 21, No. 6, Novembro, 1987.
- [Cra94] Crawfis, R. et al - "Advanced Techniques for Scientific Visualization", in *ACM SIGGRAPH'94* (Notas de Curso).
- [Day97] Day, A. M. et al - "Visualization and Sonification of Marine Survey Data", in *Visualization & Modeling*, Earnshaw et al (Eds), Academic Press, 1997, pp. 87 a 97.
- [Elv94] Elvins, T. T. - "A Survey of Algorithms for Volume Visualization", *SIGGRAPH'94*, 1994.
- [Fol88] Foley, J. D.; Grimes, J. - "Using Color in Computer Graphics", in *IEEE Computer Graphics & Applications*, Vol. 8, No. 5, Setembro, 1988, pp. 25-27.
- [Fol94] Foley, J.; Ribarsky, B. - "Next-Generation Data Visualization Tools", in *Scientific Visualization - Advances and Challenges*, (cap. 7), Academic Press, 1994, pp. 103-127.
- [Fol96] Foley, J. D. et al - *Computer Graphics: Principles and Practice*, Addison Wesley, Segunda Edição em C, 1996.
- [For94] Forrest, A. R.- *Notes on Colour for Computer Graphics*, CMP-2E21, Dezembro, 1994, pp. 1-22 (Notas de Curso).
- [Fri91] Friedhoff, R. M.; Benzon, W. - *The Second Computer Revolution*, W. H. Freeman and Company, 1991, pp. 10-45.
- [Gal4] Gallop, J. - "Underlying Data Models and Structures for Visualization", in *Scientific Visualization - Advances and Challenges*, (cap. 15), Academic Press, 1994, pp. 240-250.
- [Ger92] Gershon, N. D. - "From Perception to Visualization", in *Panel on SIGGRAPH'92*, Vol. 26, No. 2, Julho, 1992, pp. 26-31.
- [Ger94] Gershon, N., - "From Perception to Visualization", in *Scientific Visualization - Advances and Challenges*, (cap. 8), Academic Press, 1994, pp. 129-139.
- [Gon78] Gonzalez, R. C.; Woods, R. E. - *Digital Image Processing*, Addison Wesley Publishing Company, 1978.
- [Gri90] Grinstein, G.; Smith, S. - "The Perceptualization of Scientific Data", in *Proceedings SPIE - Extract Meaning from Complex Data: Processing, Display, Interaction*, Vol. 1259, Fevereiro, 1990, pp. 190-199.
- [Gri95] Grinstein, G.; Kienegger-Domik, G.; Owen, G. S. - "Guidelines for Teaching a Course in Visualization", *Proceedings of the IEEE Computer Society*

Visualization'95 Conference, Outubro 19 - Novembro 3, 1995 (Notas de Tutorial).

- [Gro94] Groß, M. - *Visual Computing*, Springer Verlag, 1994.
- [Hab90] Haber, R. B.; MacNabb, D. A. - "Visualization Idioms: A Conceptual Model for Scientific Visualization Systems", in Nielson, G. M. (ed) - in *Visualization in Scientific Computing*, ed. Nielson G. M. et al, IEEE Computer Graphic Press, 1990, pp. 74-93.
- [Hea94] Hearn, D.; Baker, M. P. - *Computer Graphics*, Prentice-Hall International Inc., Segunda Edição, 1994.
- [Hoa90] Hoadley, E. D. - "Investigating the Effects of Color", in *Communications of the ACM*, Vol. 33, No. 2, Fevereiro, 1990, pp. 120-139.
- [Hyp96] in <http://www.education.siggraph.org/materials/hypervis/concepts/concepts.htm>.
- [Hun78] Hunt, R. W. G. - "Colour Terminology", in *Color Research and Application*, Vol. 3, No. 2, Verão, 1978, pp. 79-87.
- [Kau90] Kaufmann, (R) A.- "3D Volume Visualization", in *Eurographics'90*, Eurographics Technical Report Series, 1990, pp. 34 (Notas de Tutorial).
- [Kau93] Kaufman, A.; Cohen, D.; Yagel, R. - "Volume Graphics", in *IEEE Computer Graphics & Applications*, Julho, 1993, pp. 51-64.
- [Kel93] Keller, P. K.; Keller, M. M.- "Visual Cues. Practical Data Visualization", in *IEEE Computer Society Press e IEEE Press*, 1993.
- [Koc91] Kochhar, S.; Friedell, M.; LaPolla, M.- "Cooperative, Computer-Aided Design of Scientific Visualizations", in *Proceedings of the IEEE Visualization'91*, Outubro, 1991, pp. 306-313.
- [Lan85] Lansdown, J. - "Visual Perception and Computer Graphics", in Earnshaw, R. A. (ed) - in *Fundamental Algorithms for Computer Graphics*, The Proceedings of the NATO Advanced Study Instituto on "Fundamental Algorithms for computer Graphics", 1985, Springer-Verlag, pp. 1005-1026.
- [Lev92] Levkowitz, H.; Herman, G. T. - "Color Scales for Image Data", in *IEEE Computer Graphics & Applications*, Vol. 12, No. 1, Janeiro, 1992, pp. 72-80.
- [Liv88a] Livingstone, M.; Hubel, D. - "Segregation of Form, Color, Movement and Depth: Anatomy, Physiology, and Perception", in *Science*, Vol. 240, Maio, 1988, pp. 740-749.
- [Liv88b] Livingstone, M. S - "Art, Illusion and the Visual System", in *Scientific American*, Vol. 256, 1988, pp. 68-75.

- [Mar93] Marchak, F. M. et al - Panel: "The Psychology of Visualization", in *Proceedings of the IEEE Computer Society Visualization'93 Conference*, 1993, pp. 351-354.
- [Mac86] Mackinlay, J. - "Automating the Design of Graphical Presentations of Relational Information", in *ACM Transactions on Graphics*, Vol. 5, No. 2, pp. 110-141.
- [Mey80] Meyer, G. W.; Greenberg, D. P. - "Perceptual Color Spaces for Computer Graphics", *ACM SIGGRAPH Computer Graphics*, 1980, pp. 254-261.
- [Mey86] Meyer, G. W. - "Tutorial on Color Science", in *The Visual Computer*, Springer-Verlag, 1986, pp. 278-290.
- [Mey88] Meyer, G. M.; Greenberg, D. P. - "Color-Defective Vision and Computer Graphics Displays", in *IEEE Computer Graphics & Applications*, Setembro, 1988, pp. 28-40.
- [Min95] Minghim, R. - *On Sound Support for Visualization*, University of East Anglia, Norwich, UK, Março, 1995, pp. 273 (Tese de Doutorado).
- [Min97] Minghim, R.; Oliveira, M. C. F. - "PowerVis: Empowering the User with a Multi-Modal Visualization System", in *Proceedings of the II Workshop on Cybernetic Vision*, IFSC-USP, Dezembro, 1997.
- [Mur84] Murch, G. - "Physiological Principles for the Effective Use of Color", in *IEEE Computer Graphics & Applications*, Novembro, 1984, pp. 49-54.
- [Pog78] Poggio, T.; Reichardt, W. - "Vision by Man and Machine", in *Theoretical Approaches in Neurobiology: Based on a Work Session Neuroscience Research Program*, Cambridge MIT Press, Maio, 1978, pp. 68-78.
- [Pra96] Prado, A. H. M. A., - *Interpolação de Imagens Tomográficas através de Matching*, Novembro, 1996 (Dissertação de Mestrado).
- [Reu90] Reuter, L. H. et al - Panel: "Human Perception and Visualization", in *Proceedings of the IEEE Computer Society Visualization'90 Conference*, 1990, pp. 401-406.
- [Rhe90] Rheingans, P.; Tebbs, B. - "A Tool for Dynamic Explorations of Color Mappings", in *Communications of the ACM*, Vol. 24, No. 2, 1990, pp. 145-146.
- [Rob88] Robertson, P. K. - "Visualizing Color Gamuts: A User Interface for The Effective Use of Perceptual Color Spaces in Data Displays", in *IEEE Computer Graphics & Applications*, Vol. 8, No. 8, Setembro, 1988, pp. 50-64.
- [Rob91] Robertson, P. K. - "A Methodology for Choosing Data Representations", in *IEEE Computer Graphics & Applications*, Vol. 11, No. 3, Maio, 1991, pp. 56-66.

- [Rob94] Robertson, P. K.; De Ferrari, L. - "Systematic Approaches to Visualization: Is a Reference Model Needed?", in *Scientific Visualization - Advances and Challenges*, Academic Press, 1994, pp. 288-305.
- [Rog93a] Rogowitz, B. E.; Treinish, L. A. - "Data Structures and Perceptual Structures", in *Proceedings of the SPIE/SPSE Symposium on Electronic Imaging*, Vol. 1913, Fevereiro, 1993, pp. 600-612.
- [Rog93b] Rogowitz, B. E.; Treinish, L. A. - "An Architecture for Rule Based Visualization", in *Proceedings of the IEEE Computer Society Visualization'93 Conference*, Outubro, 1993, pp. 236-243.
- [Rog96] Rogowitz, B. E.; Treinish, L. A. - "How NOT to Lie with Visualization", 1996, pp. 1-11, at <http://www.almaden.ibm.com/dx/vis96/proceedings/PRAVDA/index.htm>.
- [Rot90] Roth, S.; Mattis, J. - "Data Characterization for Intelligent Graphics Presentations", in *Proceedings of the CHI'90*, Abril, 1990, pp. 193-200.
- [Rum91] Rumbaugh, J. et al - *Object-Oriented Modeling and Design*, Prentice-Hall, 1991.
- [Sch96] Schroeder, W. J.; Martin, K.; Lorensen, W. - *The Visualization Toolkit - An Object-Oriented Approach to 3D Graphics*, Prentice-Hall, 1996.
- [Sch97] Schroeder, W. J.; Martin, K.; Lorensen, W. - *The Visualization Toolkit - An Object-Oriented Approach to 3D Graphics*, Prentice-Hall, 1997.
- [Sen94] Senay, H.; Ignatius, E. - "A Knowledge-Based System for Visualization Design", in *IEEE Computer Graphics & Applications*, Vol. 14, No. 6, Novembro, 1994, pp. 36-47.
- [Sen96] Senay, H.; Ignatius, E. - *Rules and Principles of Scientific Data Visualization*, Julho, 1996, pp. 1-19.
- [Sil94] Silver, D.; Zabusky, N. J. - "Scientific Visualization and Computer Vision", in *IEEE Computer Graphics & Applications*, Abril, 1994, pp. 55-61.
- [Sta90] Stark, L. W. et al - Panel: "Making a Picture Fit the Eye: Human Engineering for Computer Graphics", in *Proceedings of IEEE Computer Society Visualization'90*, 1990, pp. 411-420.
- [Tom90] Tominaga, S. - "Color Classification of Color Images Based on Uniform Color Spaces", in *Proceedings of SPIE Conference on Human Vision and Electronic Imaging: Models, Methods and Applications*, Vol. 1249, 1990, pp. 356-365.
- [Tom94] Tomé, M. F.; McKee, S. - "GENSMAC: A Computational Marker-and-Cell Method for Free Surface Flows in General Domains", in *Journal of Computational Physics*, Vol. 110, pp. 171-186.

- [Tom96] Tomé, M. F. et al - "Numerical Simulation of Axysimmetric Free Surface Flows", ICMSC-USP, Julho, 1996 (Relatório Interno).
- [Wal84] Waldrop, M. M. - "Computer Vision", in *Science*, Junho, 1984, pp. 1225-1227.
- [War88] Ware, C. - "Color Sequences for Univariate Maps: Theory, Experiments, and Principles", in *IEEE Computer Graphics & Applications*, Vol. 8, No. 5, Setembro, 1988, pp. 41-49.
- [Wol93] Wolf, R. S.; Yaeger, L. - *Visualization of Natural Phenomena* (ed), Springer-Verlag New York, 1993.
- [Yeo94] Yeo, B. L.; Liou, S. P. - "From Visualization to Perceptual Organization", in *Proceedings of the IEEE Workshop on Visualization and Machine Vision*, IEEE Computer Society Press, 1994, pp. 62-73.

Apêndice A

Este apêndice apresenta os atributos e os protótipos dos métodos associados às classes criadas durante o desenvolvimento deste trabalho.

Classe vtkRuleLookupTable

Atributos definidos pelo usuário

Atributos	Descrição	Valores
NumberOfColors	Número de elementos da tabela de cores	Valor entre 2 e 256 Valor <i>default</i> = 256
VisualizationTask	Tipo de tabela que o usuário deseja especificar	1 – isomorfismo 2 – segmentação 3 – ênfase 4 – <i>rainbow</i> (ROYGCB) 5 – <i>grayscale</i> (tons de cinza) Valor <i>default</i> = isomorfismo
DataType	Tipo de dado	1 – razão 2 – intervalo Valor <i>default</i> = razão
NumberOfSegments	Número de segmentos da tabela de ênfase	Valor entre 1 e 6 (número máximo especificado nesta implementação) Valores <i>default</i> : EVEN = 6 ODD = 5
End	Valor de dado correspondente ao final do intervalo de ênfase	Valor pertencente ao intervalo de variação dos dados Valor <i>default</i> = (valor médio do <i>range</i> dos dados) + 5
Init	Valor de dado correspondente ao início do intervalo de ênfase	Valor pertencente ao intervalo de variação do conjunto de dados Valor <i>default</i> = (valor médio do <i>range</i> dos dados) – 5
TableOpacity[2]	Valores inicial e final de opacidade para as tabelas isomórfica, <i>default</i> e de escala de cinzas	Valor <i>float</i> entre 0.0 e 1.0 Valores <i>default</i> = 1.0 e 1.0

ColorCode	Código da cor ou do par de cores da tabela	<p>1 – vermelho; 2 – amarelo; 3 – verde; 4 – turquesa; 5 – azul; 6 – magenta; 25 – amarelo-azul 52 – azul-amarelo 36 – verde-magenta 63 – magenta-verde 14 – vermelho-turquesa 41 – turquesa-vermelho 13 – vermelho-verde 31 – verde-vermelho Valores <i>default</i>: 25(enfatização) 13(isomórfica-intervalo) 1(isomórfica-razão) 1, 2, 3, 4, 5, 6 (segmentação)</p>
-----------	--	--

Atributos definidos pela classe

Atributos	Descrição	Valores
ColorType	Tipo de cor/par de cores da tabela	PAIR – matizes opoentes ou complementares HUEVARIATION – 1 matiz
*Color	Código das cores para as tabelas de isomorfismo e ênfatização	Valores inteiros
NumberOfHues	Número de elementos de *Color para tabelas de segmentação e isomorfismo	Valor maior do que 0
DataFrequency	Frequência espacial dos dados	LOW – baixa HIGH – alta
SegmentationType	Quantidade de segmentos que uma tabela para a tarefa de segmentação pode conter	EVEN – valor par EVENODD – valor par ou ímpar
rgba[4]	Valores RGB obtidos a partir de valores HSV	Valores entre 0 e 1
Sat	Valor da saturação	Valor entre 0 e 1
Value	Valor da luminância	Valor entre 0 e 1
SatInc	Valor do incremento para a saturação	Valor entre 0 e 1
ValueInc	Valor do incremento para a luminância	Valor entre 0 e 1
MinVal	Menor valor do conjunto de dados	Sem restrição
MaxVal	Maior valor do conjunto de dados	Sem restrição
DefaultTable	Flag para indicar se foi especificada uma tabela de segmentação ou de ênfatização	0 – Tabela especificada 1 – Tabela não especificada
*Frequency	Instância da classe <i>SpatialFrequency</i>	
*Opacity	Valor de opacidade de cada entrada da tabela	Valores entre 0 e 1
*RGB	Parâmetros de cor no modelo RGB	fr(vermelho) – Valor entre 0 e 255 fg(verde) – Valor entre 0 e 255 fb(azul) – Valor entre 0 e 255 fo(opacidade) – Valor entre 0 e 255

*HSV	Parâmetros de cor no modelo HSV	fh(matiz) – Valor entre 1 e 6 fs(saturação) – Valor entre 0 e 1 fv(luminância) – Valor entre 0 e 1
*Segments	Parâmetros para a criação da tabela de segmentação	init_seg – Valor dentro do intervalo de variação dos dados color_hue – Valor entre 1 e 6 color_sat – Valor entre 0 e 1 color_val – Valor entre 0 e 1 color_opc – Valor entre 0 e 1

Métodos Protegidos

Método	Descrição
CreateTable()	Faz alocação de memória para as tabelas de cores intermediárias (HSV e RGB) e a chamada para a função que faz alocação de uma tabela de cores no VTK
CreateColorArray()	Faz alocação de memória para o vetor de código(s) da(s) cor(es) que forma(m) uma tabela de cores
CreateOpacityArray()	Faz alocação de memória para o vetor de valores de opacidade de cada entrada da tabela de cores
CreateSegmentationArray()	Faz alocação de memória para o vetor de informações sobre os segmentos
IsomorphicRules()	Especifica valores iniciais e incrementos dos parâmetros HSV, e o número de matizes (1 ou 2) para a tarefa de isomorfismo
SegmentationRules()	Especifica a quantidade de segmentos para a tarefa de segmentação
HighlightingRules()	Especifica o número de matizes para a tarefa de ênfase
DefineIsomorphicTable()	Insere os valores H, S e V em uma tabela para a tarefa de isomorfismo
DefineSegmentationTable()	Insere os valores H, S e V em uma tabela para a tarefa de segmentação
DefineHighlightingTable()	Insere os valores H, S e V em uma tabela para a tarefa de ênfase
DefineGrayScale()	Insere os valores H, S e V da tabela de tons de cinza
DefineRainbowTable()	Insere os valores H, S e V da tabela <i>default</i> do VTK
DefineTableOpacity()	Insere os valores de opacidade no vetor de opacidade
DefineColorArray()	Insere o código de uma cor ou de um par de cores no vetor de código de cores
HSVToRGB()	Faz a conversão de parâmetros de cor no modelo HSV para parâmetros de cor no modelo RGB das tabelas de isomorfismo, segmentação, ênfase e grayscale
HSVToRGBDefault()	Faz a conversão de parâmetros de cor no modelo HSV para parâmetros de cor no modelo RGB da tabela <i>default</i> do VTK
ValidateDataType()	Faz a validação do tipo de dado, atribuindo o tipo de dado <i>default</i> (tipo razão)

ValidateInterval()	Faz a validação do intervalo de ênfatização, definindo a tabela de ênfatização <i>default</i> (ênfatisa os 10 valores intermediários do conjunto de dados)
ValidateColorCode()	Faz a validação de pares de cores para a tarefa de isomorfismo, atribuindo o par de cores <i>default</i> (vermelho para dados do tipo intervalo; e vermelho-verde para dados do tipo razão)
ValidateSegmentationArray()	Faz a validação da definição dos segmentos, definindo a tabela de segmentação <i>default</i> (6 segmentos nas cores vermelha, amarela, verde, turquesa, azul e magenta)
ValidateTableSize()	Faz a validação do número de elementos da tabela, atribuindo o número máximo permitido (256 entradas)
ValidateNumberOfSegments()	Faz a validação do número de segmentos, definindo a tabela de segmentação <i>default</i> (6 segmentos)
ValidateOpacityArray()	Faz a validação dos valores de opacidade, atribuindo o valor de opacidade <i>default</i> (1.0)
ValidateTableRange()	Faz a validação do intervalo de variação dos dados, atribuindo o menor e o maior valores do conjunto de dados
ValidateVisualizationTask()	Faz a validação da tarefa de visualização, atribuindo a tarefa de visualização <i>default</i> (isomorfismo)
InitializeColorArray()	Inicializa o vetor de código de matizes da tabela
InitializeOpacityArray()	Inicializa o vetor de valores de opacidade de uma tabela de cores
InitializeSegmentationArray()	Inicializa o vetor de informações sobre os segmentos
DefaultSegmentationTable()	Define valores <i>default</i> para os parâmetros de cada segmento de uma tabela
DefaultHighlightingTable()	Define valores <i>default</i> para os parâmetros de cor e de intervalo ênfatizado de uma tabela
DefaultNumberOfSegments()	Define o número de segmentos para a tabela <i>default</i> da tarefa de segmentação
GetIntervalOpacity()	Faz a leitura de valores de opacidade da tabela de ênfatização armazenados no vetor de opacidade
GetValues(FILE)	Faz a leitura de um valor numérico armazenado em arquivo
GetSegmentOpacity()	Faz a leitura de valores de opacidade de cada segmento armazenados no vetor de informações sobre os segmentos
GetRGBValues()	Faz a leitura de valores de cor no modelo RGB de uma tabela armazenada em arquivo

GetRuleLookupTableHeader()	Faz a leitura do cabeçalho de uma tabela de cores armazenada em arquivo
GetParam(FILE)	Faz a leitura de metadados sobre o conjunto de dados que gerou a tabela de cores recuperada do arquivo
ExecuteVisualizationTask()	Faz a chamada para a função de definição da cor e da opacidade de cada entrada da tabela
MapDataValueToIndex(float)	Faz o mapeamento de um valor de dado em uma tabela de cores
IsValueIntoRange(float)	Verifica se o valor está dentro do intervalo de variação dos dados

Métodos Públicos

Método	Descrição
<code>vtkRuleLookupTable()</code>	Construtora da classe
<code>*GetClassName()</code>	Retorna o nome da classe
<code>SetTableSize(Número de entradas)</code>	Define o número de entradas da tabela de cores
<code>SetVisualizationTask(Tarefa de visualização)</code>	Define uma tarefa de visualização
<code>SetType(Tipo do dado)</code>	Define o tipo do dado
<code>SetTableColor(Código da cor)</code>	Define um código de cor/par de cores de uma tabela
<code>SetNumberOfSegments(Número de segmentos)</code>	Define o número de segmentos de uma tabela para a tarefa de segmentação
<code>SetSegmentationTable(Início do segmento, Código da cor, Valor da saturação, Valor da luminância)</code>	Define as informações sobre cada segmento para conjuntos de dados bidimensionais
<code>SetSegmentationTable(Início do segmento, Código da cor, Valor da saturação, Valor da luminância, Valor da opacidade)</code>	Define as informações sobre cada segmento para conjuntos de dados tridimensionais
<code>SetHighlightingInterval(Início do intervalo, Fim do intervalo, Código do par de cores)</code>	Define as informações sobre o intervalo de ênfase
<code>SetIntervalOpacity()</code>	Define o valor de opacidade do intervalo enfatizado e do restante da tabela
<code>SetTableOpacity(Valor mínimo, Valor máximo)</code>	Define o menor e maior valores de opacidade para as tabelas isomórfica, <i>default</i> e de escala de cinzas
<code>SetTableRange(Valor mínimo, Valor máximo)</code>	Define o menor e maior valores a serem mapeados em uma tabela de cores
<code>DataCharacteristics(Nome do arquivo)</code>	Define a frequência espacial e o intervalo de variação dos dados de um arquivo VTK
<code>ReadRuleLookupTableFromFile(Nome do arquivo)</code>	Recupera uma tabela de cores gerada pela classe <code>vtkRuleLookupTable</code> e armazenada em um arquivo de extensão (.txt)
<code>WriteRuleLookupTableToFile(Nome do arquivo)</code>	Armazena uma tabela de cores gerada pela classe <code>vtkRuleLookupTable</code> em um arquivo de extensão (.txt)

ApplyRules()	Aplica o conjunto de regras de acordo com a tarefa de visualização especificada
Build()	Constrói a tabela de cores no VTK

Classe SpatialFrequency

Atributos da Classe

Atributos	Descrição
dim_x, dim_y, dim_z	Dimensões da matriz
num_elem	Número de elementos da máscara
Volume	Indica conjunto de dados tridimensional
z_end	Dimensão do eixo z na matriz <i>copia</i>
*IS	<i>Handle</i> para a <i>stream</i> de entrada
somatoria	Valor da soma dos elementos da matriz <i>result</i>
***origin	Matriz com valores de dados originais
***copy	Matriz com valores de dados replicados e interpolados
***result	Matriz com valores de dados resultantes da diferença entre as matrizes <i>origin</i> e <i>copia</i>
min_val, max_val (atributo público)	Intervalo de variação do conjunto de dados

Métodos Protegidos

Método	Descrição
GetDimensions()	Faz a leitura das dimensões dos eixos x, y e z
Alocate()	Faz a alocação de memória para as matrizes (<i>origin</i> , <i>copia</i> e <i>result</i>)
Deallocate()	Libera a memória alocada para as matrizes
Coeficient()	Faz o cálculo do coeficiente amostral
SetZLimit()	Define o valor da dimensão z da matriz <i>copia</i>
Replicate()	Faz a replicação de elementos da matriz <i>copia</i>
ReadDataSet()	Faz a leitura do arquivo e verifica o intervalo de variação dos dados
CopyOrigin()	Copia os elementos da matriz <i>origin</i> para a matriz <i>copia</i>
NumElemMask()	Define o número de elementos da máscara utilizada na filtragem
DefineCorrespondent(int, int, int)	Define os índices (x, y, z) na matriz <i>copia</i> a partir dos índices da matriz <i>origin</i>
InitializeSubSet(float, float, float, int, int)	Copia um subconjunto para a filtragem
ChangeSubSet(int, int, int, float)	Copia os elementos de uma coluna/linha da matriz
ChangeSubSet(int, int, int, float, float, float)	Copia os elementos de uma camada da matriz
Addition(float, float, float)	Faz a soma de 3 valores
CalcMidle(float, float, float)	Calcula a média aritmética da vizinhança de um pixel
SubtractArrays()	Faz a diferença entre a matriz de dados original (<i>origin</i>) e a matriz de dados filtrados (<i>copia</i>)
SurfFilter()	Faz a filtragem dos elementos de uma superfície
VolumeFilter()	Faz a filtragem dos elementos de um volume

Métodos Públicos

Método	Descrição
SpatialFrequency()	Construtora da Classe
FrequencyAnalysis(istream) (método público)	Determina a frequência espacial dos dados