

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RAQUEL MAINARDI PILLAT

**Coordenação Dinâmica de Visualizações de
Dados Multidimensionais**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof^a. Dr^a. Carla Maria Dal Sasso Freitas
Orientadora

Porto Alegre, julho de 2006.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Pillat, Raquel Mainardi

Coordenação Dinâmica de Visualizações de Dados Multidimensionais / Raquel Mainardi Pillat – Porto Alegre: Programa de Pós-Graduação em Computação, 2006.

106 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2006. Orientadora: Carla Maria Dal Sasso Freitas.

1. Visualização de informações. 2. Visualizações coordenadas. 3. Interface de usuário. I. Freitas, Carla Maria Dal Sasso. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Profa. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço à minha orientadora, Carla Dal Sasso Freitas, pelo auxílio e estímulo recebido durante a realização desse trabalho e a todos os integrantes do grupo de computação gráfica da UFRGS (um GRUPO de verdade, do qual eu tenho muito orgulho em fazer parte). Não posso deixar de agradecer também a paciência e boa vontade daqueles que participaram uma ou várias vezes dos meus experimentos de avaliação.

Por fim, agradecimentos muito especiais à minha família: minha mãe Lúcia, meu pai Luiz Clóvis, minha irmã Micheli e meu namorado Fábio.

SUMÁRIO

LISTA DE FIGURAS.....	7
LISTA DE TABELAS	10
RESUMO.....	11
ABSTRACT.....	12
1 INTRODUÇÃO	13
1.1 Motivação	14
1.2 Objetivos e Contribuições.....	15
1.3 Organização do Trabalho	15
2 INTERAÇÃO COM VISUALIZAÇÕES MULTIDIMENSIONAIS	16
2.1 Visualizações Multidimensionais	16
2.1.1 Técnicas de projeção geométrica.....	17
2.1.2 Técnicas Iconográficas	23
2.1.3 Técnicas Orientadas a Pixel.....	24
2.2 Técnicas de Interação.....	26
2.2.1 Mapeamento de Dados para Propriedades Visuais.....	26
2.2.2 Projeções.....	27
2.2.3 Filtragem.....	28
2.2.4 Zoom.....	29
2.2.5 Detalhes por Demanda.....	30
3 VISUALIZAÇÕES COORDENADAS: CONCEITOS E REVISÃO	
BIBLIOGRÁFICA.....	31
3.1 Tipos de Coordenações.....	31
3.1.1 <i>Brushing-and-Linking</i>	32
3.1.2 Reordenação de Dimensões Sincronizada.....	33
3.1.3 Filtragem Sincronizada.....	33
3.1.4 Mapeamento de Dados para Propriedades Visuais.....	34
3.1.5 <i>Scrolling</i> Sincronizado	34
3.1.6 Zoom Sincronizado	35
3.1.7 Rotação Sincronizada	36
3.1.8 <i>Overview</i> e Detalhe.....	36
3.1.9 <i>Drill Down</i>	37
3.2 Sistemas de Múltiplas Visualizações Coordenadas	38

3.2.1	XmdvTool.....	39
3.2.2	Snap-Together	40
3.2.3	GeoVista Studio.....	43
4	O TOOLKIT INFOVIS.....	45
4.1	Framework Geral.....	46
4.1.1	Entrada e Saída	46
4.1.2	Tabelas e Colunas.....	46
4.1.3	Visualizações	47
4.1.4	Componentes	48
4.2	Exibindo Visualizações.....	48
4.3	Extensão do Toolkit: Criando Novas Visualizações.....	49
4.4	Mecanismos de Interação.....	52
4.4.1	Mecanismos Adicionados.....	55
5	COORDENAÇÃO DINÂMICA DE VISUALIZAÇÕES MULTIDIMENSIONAIS	57
5.1	Coordenações para Sistemas de Visualização de Dados Multidimensionais .	57
5.1.1	Mapeamento de Dados para Propriedades Visuais.....	57
5.1.2	Reordenação de Dimensões.....	58
5.1.3	Filtragem.....	58
5.1.4	<i>Brushing-and-Linking</i>	59
5.2	Aplicação de Visualizações Coordenadas para Dados Multidimensionais	59
5.2.1	Visualizações	60
5.2.2	Coordenações	68
5.2.3	Arquitetura.....	71
5.2.4	Grafo de Coordenação	73
5.2.5	Especificando Coordenações Dinamicamente.....	74
5.2.6	Gerenciamento de Coordenações	76
5.2.7	Tornando uma Visualização Coordenável: A Interface <i>Visualization</i>	80
5.3	Comentários Finais.....	81
6	AVALIAÇÃO PRELIMINAR	82
6.1	Primeira Avaliação.....	82
6.1.1	Conjunto de Dados	82
6.1.2	Tarefas Experimentais	83
6.1.3	Procedimento	83
6.1.4	Resultados.....	84
6.2	Segunda Avaliação.....	87
7	ESTUDO DE CASO	88
7.1	Descrição e Objetivos do Domínio: Morfometria Craniana.....	88
7.2	Conjunto de Dados	89
7.3	Procedimento	89
7.4	Resultados	90
7.4.1	Identificação de Crânios com Medidas Fora do Padrão.....	90
7.4.2	Identificação de Variações nos Padrões de Medidas de Diferentes Localidades	91
7.4.3	Identificação de Variações nas Medidas de Animais com Número de Cromossomos Diferente	92
7.4.4	Verificação de Variações nas Medidas de Animais de Sexo Diferente	93

7.5	Comparação: Aplicação Desenvolvida x Aplicação do Domínio	95
7.6	Conclusões	96
8	CONCLUSÕES	98
8.1	Avaliação Final	98
8.2	Trabalhos Futuros	101
	REFERÊNCIAS.....	102

LISTA DE FIGURAS

Figura 2.1: Representação visual da técnica Coordenadas Paralelas	18
Figura 2.2: Técnica Coordenadas Paralelas com sobreposição de linhas (15.000 itens de dados representados)	18
Figura 2.3: Representação visual da técnica Matriz de <i>Scatter Plots</i>	19
Figura 2.4: Representação visual da técnica Radviz	20
Figura 2.5: Representação visual da técnica Coordenadas Paralelas Circulares.....	21
Figura 2.6: Representação visual da técnica Polyviz	22
Figura 2.7: Representação visual da técnica <i>Survey Plots</i>	22
Figura 2.8: Técnicas iconográficas: a) Faces de Chernoff e b) Glifos em Estrela	23
Figura 2.9: Mapeamento de <i>Stick Figures</i> (esquerda) e gráfico iconográfico de dados do censo da Inglaterra (direita).....	24
Figura 2.10: Visualizações orientadas a pixel de um conjunto de dados com nove dimensões, utilizando duas formas de distribuição espacial: espiral (esquerda) e por eixos (direita).....	25
Figura 2.11: Técnicas independentes de consulta: a) curvas de Peano-Hilbert, b) curvas de Morton e c) padrões recursivos.....	25
Figura 2.12: Técnica Segmentos de Círculo.....	26
Figura 2.13: Usando consultas dinâmicas para verificar a concentração espacial de itens de dados em uma região de interesse	28
Figura 2.14: <i>Magic lenses</i> : Lente de visão aramada 3D e lente de aumento 2D.....	29
Figura 2.15: Usando a técnica detalhes por demanda para obter informações adicionais sobre um item de dado.	30
Figura 3.1: Visualizações coordenadas pela técnica <i>brushing-and-linking</i>	33
Figura 3.2: Visualizações coordenadas com valores do atributo <i>Origin</i> mapeados para cor	35
Figura 3.3: Múltiplas visualizações 3D em grid.....	36
Figura 3.4: Coordenação de <i>overview</i> e detalhe no sistema PDQ Tree-browser	37
Figura 3.5: Usando a coordenação <i>drill down</i> para visualizar o conteúdo de agregações no Snap-Together.....	38

Figura 3.6: <i>Brushing-and-linking</i> <i>n</i> -dimensional no sistema XmdvTool. Os dados selecionados são exibidos nas imagens (a) e (b) em vermelho e a região de <i>brushing</i> , que indica os dados de interesse, é mostrada nas imagens (c) e (d) no espaço de um glifo em estrela e no espaço de coordenadas paralelas, respectivamente.	41
Figura 3.7: Cenário construído em Snap-Together para a exploração de duas tabelas relacionadas de dados multidimensionais.....	42
Figura 3.8: Cenário construído em GeoVista Studio com todas as suas técnicas de visualização multidimensionais	44
Figura 4.1: Estrutura interna do toolkit InfoVis. Retângulos representam estruturas de dados e elipses representam funções.	46
Figura 4.2: Programa para a visualização de Coordenadas Paralelas. O código do retângulo azul gera uma visualização não-interativa, mas quando substituído pelo código do retângulo vermelho a visualização recebe um painel de controle padrão.....	49
Figura 4.3: Hierarquia das classes de visualização do toolkit InfoVis.....	50
Figura 4.4: Código padrão executado para o desenho de itens de dados	51
Figura 4.5: Código padrão executado para o desenho de um item de dado	51
Figura 4.6: As quatro abas do painel de controle: (a) <i>Filters</i> ; (b) <i>Visual</i> ; (c) <i>Excentric</i> ; e (d) <i>Fisheyes</i>	52
Figura 4.7: Visualizações geradas pela aplicação dos mecanismos de interação <i>Color by</i> (à esquerda) e <i>Size by</i> (à direita) sobre as técnicas Radviz (acima) e Coordenadas Paralelas (abaixo).....	53
Figura 4.8: <i>Labels</i> dinâmicos na técnica Coordenadas Paralelas	54
Figura 4.9: Lente <i>Fisheye</i> aplicada na técnica Radviz	54
Figura 4.10: Visualização dos valores que o 142º item de dado do conjunto sobre carros possui para cada atributo	55
Figura 4.11: Componentes gráficos que compõem o mecanismo de interação <i>Attributes Manipulation</i>	56
Figura 5.1: Técnica Coordenadas Paralelas implementada no toolkit InfoVis	61
Figura 5.2: Criação e desenho das linhas que representam atributos na técnica Coordenadas Paralelas	61
Figura 5.3: Criação dos objetos que representam itens de dados na técnica Coordenadas Paralelas	62
Figura 5.4: Técnica Radviz implementada no toolkit InfoVis	63
Figura 5.5: Criação e desenho das linhas que representam atributos na técnica Radviz	64
Figura 5.6: Criação dos objetos que representam itens de dados na técnica Radviz.....	65
Figura 5.7: Técnica Matriz de <i>Scatter Plots</i> implementada no toolkit InfoVis.....	66
Figura 5.8: Procedimento para a criação da matriz de visualizações <i>Scatter Plots 2D</i> ..	67

Figura 5.9: Procedimento utilizado para garantir que ações realizadas em uma visualização 2D da matriz de <i>Scatter Plot</i> sejam executadas também nas suas demais visualizações.....	68
Figura 5.10: Cenário de exploração construído na aplicação protótipo	71
Figura 5.11: Arquitetura do protótipo desenvolvido	72
Figura 5.12: Estrutura do grafo de coordenação e seus relacionamentos com outros módulos da aplicação	73
Figura 5.13: Processo de criação e viabilização de uma nova coordenação	74
Figura 5.14: Representação gráfica de um grafo de coordenação.....	75
Figura 5.15: Conectando os nodos da nova coordenação.....	75
Figura 5.16: Seleção de interações a serem coordenadas.....	76
Figura 5.17: Nodos conectados pela ação de seleção.....	76
Figura 5.18: Removendo uma coordenação	76
Figura 5.19: Processo de gerenciamento de coordenações	77
Figura 5.20: Atividades realizadas pelo usuário e aplicação no processo de propagação de uma interação	78
Figura 5.21: Seqüência de operações realizadas pelos objetos envolvidos na propagação de interações	79
Figura 5.22: Métodos da interface <i>Visualization</i> relacionados ao processo de coordenação.....	81
Figura 7.1: Indicação das dezesseis medidas extraídas do crânio de um tuco-tuco para a análise morfométrica tradicional. (a) Vista dorsal, (b) vista ventral e (c) vista lateral.....	89
Figura 7.2: Identificação de um animal com medidas fora do padrão (item destacado)	91
Figura 7.3: Representação do conjunto inteiro de dados morfométricos com locais de coleta mapeados para cor	92
Figura 7.4: Representação em Coordenadas Paralelas das amostras morfométricas coletadas em Pelotas (cor verde), Bagé (cor azul) e Taim (cor vermelha)..	93
Figura 7.5: Representação em Matriz de <i>Scatter Plot</i> das amostras morfométricas coletadas em Pelotas (cor verde), Bagé (cor azul) e Taim (cor vermelha)..	94
Figura 7.6: Representação do conjunto inteiro de dados morfométricos com machos mapeados para a cor rosa e fêmeas mapeadas para a cor preta	94
Figura 7.7: Gráficos gerados com o sistema NCSS, <i>scatter plot</i> (esquerda) e <i>box plot</i> (direita), utilizados na análise do dimorfismo sexual.....	96
Figura 7.8: Gráficos gerados com o sistema NCSS, <i>scatter plot</i> (esquerda) e <i>box plot</i> (direita), utilizados na análise de variações por local de coleta.....	96

LISTA DE TABELAS

Tabela 2.1: Representação de uma tabela de dados.....	16
Tabela 2.2: Propriedades visuais que podem ser usadas para mapear atributos de dados e exemplos de mapeamentos típicos.....	27
Tabela 5.1: Interações coordenadas e as propriedades que notificam sua ocorrência.....	80
Tabela 8.1: Comparação das técnicas Coordenadas Paralelas, Radviz e Matriz de <i>Scatter Plots</i> segundo critérios encontrados na literatura.....	99

RESUMO

Técnicas de visualização de informações representam graficamente dados de um determinado domínio de aplicação e disponibilizam mecanismos para a interação com a representação gerada, a fim de que o usuário consiga interpretar e compreender as informações apresentadas. As técnicas de visualização multidimensionais, em particular, referem-se à visualização de informações onde cada elemento no conjunto de dados é descrito, ou caracterizado, por múltiplas variáveis (atributos), as quais devem ser codificadas em uma única estrutura visual.

Embora existam diversas técnicas de visualização para representar dados multidimensionais, nenhuma delas apresenta o melhor desempenho para todos os tipos de tarefas. Uma estratégia interessante é analisar várias representações simultaneamente mantendo uma ligação semântica entre elas, de forma que ações realizadas sobre uma técnica se reflitam automaticamente nas demais. Não existe, atualmente, um sistema de múltiplas visualizações coordenadas específico para a representação de dados multidimensionais, que disponha de um amplo e variado conjunto de coordenações. Os sistemas que suportam a representação coordenada deste tipo de dados são de propósito geral e, devido a isto, oferecem poucas possibilidades de coordenação.

O presente trabalho apresenta um estudo de técnicas de coordenação para sistemas de múltiplas visualizações, focando coordenação de representações visuais de dados multidimensionais. A partir do estudo dos diversos sistemas de visualização existentes, foi identificado um conjunto de coordenações que podem ser aplicadas entre visualizações multidimensionais. Esse trabalho apresenta estas coordenações e uma nova aplicação de múltiplas visualizações coordenadas (construída a partir do toolkit InfoVis), específica para a representação de dados multidimensionais, que implementa o conjunto de coordenações. A aplicação desenvolvida conta com um variado conjunto de coordenações e é altamente flexível, permitindo tanto a escolha das visualizações que representarão o conjunto de dados, bem como das coordenações ativas entre elas.

Foram realizados alguns estudos de avaliação como ensaios de interação com usuários. Primeiramente, as implementações das técnicas de visualização suportadas pela aplicação desenvolvida foram avaliadas, com o objetivo de encontrar possíveis problemas de usabilidade. Grande parte dos problemas identificados nesta avaliação, principalmente os mais graves, foram solucionados logo após sua realização. Na sequência, outro experimento, conduzido de maneira mais informal, avaliou algumas questões a respeito do uso das coordenações implementadas. Por fim, com um estudo de caso, verificou-se a aplicabilidade das visualizações multidimensionais suportadas pela aplicação para a exploração de dados de um domínio específico.

Palavras-Chave: Visualização de informações, múltiplas visualizações coordenadas, interface de usuário.

Dynamic Coordination of Multidimensional Data Visualizations

ABSTRACT

Information visualization techniques represent data of a specific domain graphically and provide mechanisms for interacting with this representation, allowing users to explore and understand their data. Multidimensional visualization techniques are employed to represent information where each data element is described by multiple variables (attributes) that must be mapped to a single visual structure.

Although there are several techniques to display multidimensional data, none of them performs best for all kinds of tasks. An interesting strategy is to analyze multiple visualization techniques simultaneously with a semantic connection between them, so that actions performed on a representation are broadcasted to the others.

This work investigates coordination techniques for multiple views systems, focusing the coordination of multidimensional data visualization. At the moment, there is not a specific coordinated-view system for multidimensional data representation that provides various forms of coordination. Systems allowing coordinated views are usually for general purpose. Due to this reason, they provide few coordination possibilities.

Starting from the study of several visualization systems, we identified coordination forms that can be used between multidimensional visualizations. This work presents these coordinations and a new specific coordinated-view application (built with the InfoVis toolkit) for multidimensional data representation that implements the identified coordinations. Our application provides various coordination forms and is very flexible: it allows the user to choose the views that will display the dataset as well as the active coordinations.

This work also presents some evaluation studies conducted as interaction tests with users. Firstly, the visualization techniques implemented in our application were evaluated in order to identify possible usability problems. Most of the problems identified in that study, mainly the more serious ones, were solved. Soon after, other experiment, conducted in a more informal way, evaluated some questions regarding the use of the implemented coordinations. Finally, we verified the suitability of the multidimensional visualizations supported by our application for a specific domain through a case study.

Keywords: Information visualization, coordinated multiple views, user interface.

1 INTRODUÇÃO

O desenvolvimento tecnológico, incorporando avanços em produtos e serviços, gerou condições para medições mais precisas, simulações mais velozes e registro de uma gama variada de tipos de dados, produzindo conjuntos de dados cada vez mais volumosos e complexos. A exploração e a análise desses volumes de dados tornou-se uma tarefa difícil que só pode ser realizada com a utilização de ferramentas computacionais apropriadas, muitas vezes dependentes do domínio dos dados. Isto é especialmente complexo quando os dados têm alta dimensionalidade, ou seja, correspondem a conjuntos de vários atributos.

A área de visualização de informações tem por objetivo facilitar a compreensão e a manipulação desta vasta quantidade de dados diversificados (CARD, 1999; SPENCE, 2001), através do emprego de técnicas que buscam tanto representar visualmente os dados quanto permitir ao usuário a interação com estas representações. As técnicas de visualização multidimensionais são especialmente projetadas para a representação de dados que têm alta dimensionalidade (descritos por múltiplos atributos). Conjuntos de dados multidimensionais geralmente são representados como tabelas, onde cada linha corresponde a um elemento de dado, com as colunas representando atributos.

As representações visuais de técnicas de visualização de informações, quando manipuladas dinamicamente, facilitam a interpretação dos dados pelo usuário, possibilitando a percepção de padrões, tendências, relacionamentos e exceções embutidos nos mesmos. Uma das grandes questões em relação a essas técnicas é o quão efetiva é uma determinada técnica, em um determinado domínio, ou para um determinado objetivo.

Embora existam diversas técnicas para mostrar tabelas e conjuntos de dados multidimensionais, dependendo da categoria dos dados e das tarefas específicas que deveriam ser suportadas, uma única técnica pode não ser suficiente: usuários podem precisar visualizar os dados mostrados com diferentes técnicas para encontrar relacionamentos e modelos.

Assim, dadas as necessidades mencionadas acima, alguns autores (BALDONADO et al., 2000; NORTH, 2000) passaram a trabalhar com uma estratégia denominada “Múltiplas Visualizações Coordenadas” (CMV - Coordinated Multiple Views). Nesta estratégia, diferentes visualizações podem ser integradas em um ambiente e usadas de forma coordenada, isto é, estabelecendo-se uma ligação semântica entre as visualizações, de maneira que ações sobre uma se reflitam nas demais. Desta forma, aproveita-se os pontos fortes de cada uma, e conjuntos de dados complexos e de alta dimensionalidade podem ser visualizados através de técnicas apropriadamente selecionadas (SHIMABUKURO, 2004).

North e Shneiderman (1997) observaram que a coordenação de múltiplas visualizações oferece as seguintes vantagens:

- *Melhora na performance do usuário*: O tempo de acesso à informação é reduzido, principalmente com a manipulação de informações hierárquicas, como mostrado por Chimera et al. (1994), bem como o número de ações requeridas do usuário para manter um estado sincronizado, como mostrado por Kandogan e Shneiderman (1997);
- *Descoberta de relacionamentos inesperados*: Seria difícil encontrar tais relacionamentos sem coordenação;
- *Unificação do desktop*: É mais sensato apresentar informações complexas não em uma única visualização complexa e integrada, mas em múltiplas visualizações mais simples, usando coordenação para navegar, manter relacionamentos, etc.

1.1 Motivação

Múltiplas visualizações de um mesmo conjunto de dados permitem observá-lo sob várias perspectivas, bem como explorar os pontos fortes e minimizar os efeitos dos pontos fracos das técnicas envolvidas (KEIM, 1996; EICK; KARR, 2000). Diversas ferramentas genéricas de visualização de informações oferecem várias técnicas e permitem o seu uso coordenado para visualizar um mesmo conjunto de dados. No entanto, em geral, o conjunto de técnicas e as formas de coordenação disponíveis são fixos, pré-estabelecidos durante o projeto da ferramenta, o que limita a potencialidade do uso das técnicas em contextos de aplicações diversos e a liberdade do usuário no processo de exploração das mesmas. Ambientes que permitem flexibilidade na configuração da coordenação entre múltiplas visualizações, possibilitando a criação de cenários personalizados para as necessidades do usuário, têm sido tema de estudos e já existem algumas iniciativas, como Snap-Together (NORTH, 2000) e GeoVISTA Studio (GAHEGAN et al., 2000), que serão abordadas com maior detalhamento no Capítulo 3. A primeira destas ferramentas é genérica, podendo ser aplicada para a análise de variados tipos de dados e domínios de aplicação; a última, entretanto, enfatiza o tratamento de dados espaciais (geo-referenciados).

Assim como a ferramenta GeoVISTA Studio, a maioria das aplicações existentes que suportam múltiplas visualizações coordenadas são projetadas para atender, com maior ênfase, dados relativos a um domínio específico. Assim, a tarefa de determinação das ações de interação coordenadas em uma ferramenta é facilitada e a mesma adquire condições para atender melhor as necessidades dos usuários no processo de compreensão e análise dos dados.

Um problema encontrado atualmente em ferramentas de múltiplas visualizações coordenadas genéricas é a baixa variedade de interações coordenadas que as mesmas suportam. Os usuários de Snap-Together, por exemplo, têm suas ações de interação limitadas à seleção e navegação, mas “...muito mais interações deveriam ser suportadas, tendo em vista que o objetivo da ferramenta é a visualização exploratória” (BOUKHELIFA et al., 2003).

A delimitação das coordenações suportadas por uma ferramenta de visualização genérica é uma tarefa difícil. Devido à natureza multiforma das diversas visualizações,

ações executadas sobre uma delas nem sempre podem ser diretamente aplicadas para outras. Além disso, algumas coordenações possíveis de serem alcançadas, podem não possuir utilidade; e ainda, outras podem ser impossíveis de realizar (BOUKHELIFA et al., 2003). Na área de visualização de informações multidimensionais, ainda não existe uma ferramenta genérica, capaz de atender os diversos domínios de aplicação em que dados multidimensionais podem ser gerados, flexível na escolha das visualizações e das ações coordenadas e com um conjunto rico e variado destas últimas. Esta observação foi a principal motivadora para a realização desse trabalho.

1.2 Objetivos e Contribuições

O objetivo principal desse trabalho é investigar técnicas de coordenação que podem ser aplicadas entre visualizações multidimensionais em geral. Como objetivo secundário está a implementação de tais técnicas em uma aplicação protótipo altamente flexível, com a finalidade de validar as coordenações definidas.

Assim, as contribuições desse trabalho são:

- A definição de um conjunto variado de coordenações que podem ser aplicadas entre visualizações multidimensionais em geral, reunindo coordenações utilizadas por diferentes ferramentas;
- A construção de uma aplicação protótipo de múltiplas visualizações coordenadas que:
 - É específica para visualização de informações multidimensionais, não sendo, no entanto, voltada para um domínio de aplicação específico;
 - É altamente flexível, pois permite que usuários possam definir quais, dentre as visualizações suportadas, irão compor um cenário de múltiplas visualizações e quais ações interativas serão coordenadas entre as mesmas.

1.3 Organização do Trabalho

Os demais capítulos dessa dissertação estão organizados como segue. O Capítulo 2 descreve as principais técnicas de visualização e de interação utilizadas para conjuntos de dados multidimensionais, organizadas em categorias. O Capítulo 3 aborda os principais tipos de coordenações utilizadas por sistemas de múltiplas visualizações coordenadas, além de descrever alguns destes sistemas relacionados com o presente trabalho. O Capítulo 4 apresenta características e a estrutura básica do toolkit de visualização InfoVis, sobre o qual foi construída a aplicação protótipo para validação de coordenações. O Capítulo 5 tem como objetivo apresentar e descrever apenas um conjunto específico de coordenações: aquelas que podem ser aplicadas em sistemas de múltiplas visualizações coordenadas, específicos para a visualização de dados multidimensionais. Além disso, este último capítulo aborda também a descrição de uma aplicação de visualizações multidimensionais coordenadas, que implementa o conjunto de coordenações apresentado no mesmo. O Capítulo 6 descreve dois estudos de avaliação preliminar da aplicação protótipo apresentada nesse trabalho e o Capítulo 7 descreve um estudo de caso em que a aplicação desenvolvida foi utilizada por especialistas de um domínio específico para a exploração de seus dados multidimensionais. Finalmente, o Capítulo 8 apresenta conclusões e trabalhos futuros.

2 INTERAÇÃO COM VISUALIZAÇÕES MULTIDIMENSIONAIS

O termo “dimensionalidade” é usado para referir tanto o número de dimensões do espaço-domínio onde determinado objeto está definido - podendo ser um espaço unidimensional (1D), bidimensional (2D) ou tridimensional (3D) -, quanto o número de atributos de um registro de dado - geralmente, referenciado como *n-dimensional* (nD), *multidimensional* ou *multivariado*. Assim, dados multidimensionais são aqueles que podem ser representados como uma tabela de dados multivariados, ou seja, que possui muitas variáveis (ou atributos) que devem ser codificadas em uma única estrutura visual 1D, 2D ou 3D (CARD, 1999). A representação de uma tabela de dados multivariados é mostrada na tabela 2.1.

Tabela 2.1: Representação de uma tabela de dados

	Variável _i	Variável _j	Variável _k	
Caso _x	Valor _{ix}	Valor _{jx}	Valor _{kx}	...
Caso _y	Valor _{iy}	Valor _{iy}	Valor _{ky}	...
...

As colunas da tabela em questão representam variáveis (ou atributos), conjuntos que representam a gama de valores nas colunas. As linhas representam casos (ou itens de dados, tuplas), que são conjuntos de valores para cada variável. Nesse trabalho, adotou-se o termo “item de dado” para denotar uma tupla descrevendo um relacionamento entre múltiplas variáveis (por exemplo, um caso ou registro de dado) e o termo “atributo” para denotar as variáveis (dimensões) do conjunto de dados.

2.1 Visualizações Multidimensionais

As visualizações que são especificamente projetadas para codificar dados que podem ser representados como tabelas multivariáveis são denominadas *visualizações multidimensionais*.

As técnicas de visualização multidimensionais podem ser categorizadas de acordo com diferentes critérios. Keim (1996) sugere uma categorização baseada em três critérios: a natureza do dado a ser visualizado, a abordagem de mapeamento adotado pela técnica e os métodos de interação e distorção disponibilizados para permitir a manipulação da representação visual.

Existem várias técnicas para a visualização de dados multidimensionais. Nessa seção serão apresentadas algumas técnicas clássicas, direcionadas à visualização de tabelas de dados, dando-se ênfase às técnicas mais representativas e citadas na literatura. As mesmas serão descritas seguindo o critério de classificação proposto por Keim (1996), que difere técnicas de acordo com a forma de mapeamento adotada para transformar dados em formas visuais. Este critério divide as técnicas de visualização multidimensionais em técnicas de projeção geométrica, iconográficas e orientadas a pixel.

2.1.1 Técnicas de projeção geométrica

Nessa classe estão incluídas todas as técnicas de visualização que empregam algum tipo de projeção geométrica para mapear dados para formas visuais. Entre as técnicas desta categoria estão: Coordenadas Paralelas, Matriz de *Scatter Plots*, gráficos de linhas, gráfico de barras e histogramas, *Survey Plots*, Curvas de Andrews, *Radviz* e Coordenadas Paralelas Circulares.

As principais técnicas representantes dessa categoria são descritas com maior detalhamento a seguir. Para demonstrar a representação visual gerada por cada uma destas técnicas, foi utilizado o conjunto de dados sobre carros (descrito na seção 6.1.1 desse trabalho).

2.1.1.1 Coordenadas Paralelas

A técnica Coordenadas Paralelas foi introduzida por Inselberg (1985) (1990) para representar múltiplas dimensões sem utilizar eixos cartesianos ortogonais. Nessa técnica, dados multidimensionais são representados utilizando linhas verticais e horizontais. Cada linha vertical indica um atributo (ou variável) e os valores dos atributos são mapeados para pontos nessas linhas verticais, de forma que cada item de dado seja representado como uma linha poligonal que intercepta cada linha vertical em seu ponto correspondente ao valor de cada atributo, formando um aglomerado de linhas horizontais. Os valores máximo e mínimo de um atributo são usualmente mapeados para as extremidades superior e inferior de uma linha vertical, respectivamente. A figura 2.1 mostra um exemplo da representação visual gerada pela técnica Coordenadas Paralelas (para representar os dados de carros). As diferentes intensidades de cor, das linhas horizontais da figura, foram usadas para diferenciar itens de dados com valores diferentes para o atributo *Origin*.

A representação visual gerada pela técnica Coordenadas Paralelas facilita principalmente a identificação de algumas características, como diferenças na distribuição dos dados e correlações entre atributos. Porém, visto que as linhas poligonais podem se sobrepor, o número de itens de dados que podem ser visualizados ao mesmo tempo é limitado a aproximadamente 1.000 itens (KEIM, 1996). Assim, a visualização de grandes volumes de dados requer que a técnica seja utilizada disponibilizando-se operações adequadas de interação. A figura 2.2 exhibe um exemplo da visualização Coordenadas Paralelas, onde há uma acentuada sobreposição de linhas.

2.1.1.2 Matriz de Scatter Plots

Scatter Plot é uma técnica classificada como geométrica, onde pontos ou marcadores visuais (representando itens de dados) são plotados em coordenadas (x, y)

de forma semelhante a outros gráficos bidimensionais. Matriz de *Scatter Plots* é uma extensão desta técnica, utilizada para a representação de dados multidimensionais. Na mesma, são exibidas simultaneamente múltiplas projeções bidimensionais (*scatter plots*) dos diferentes pares de atributos do conjunto de dados. Por exemplo, para um conjunto de dados com 6 dimensões, uma matriz 6 x 6 (6 linhas e 6 colunas) é usada para prover a visualização de cada dimensão versus todas as outras. A figura 2.3 mostra um exemplo de visualização gerada com a técnica Matriz de *Scatter Plots*.

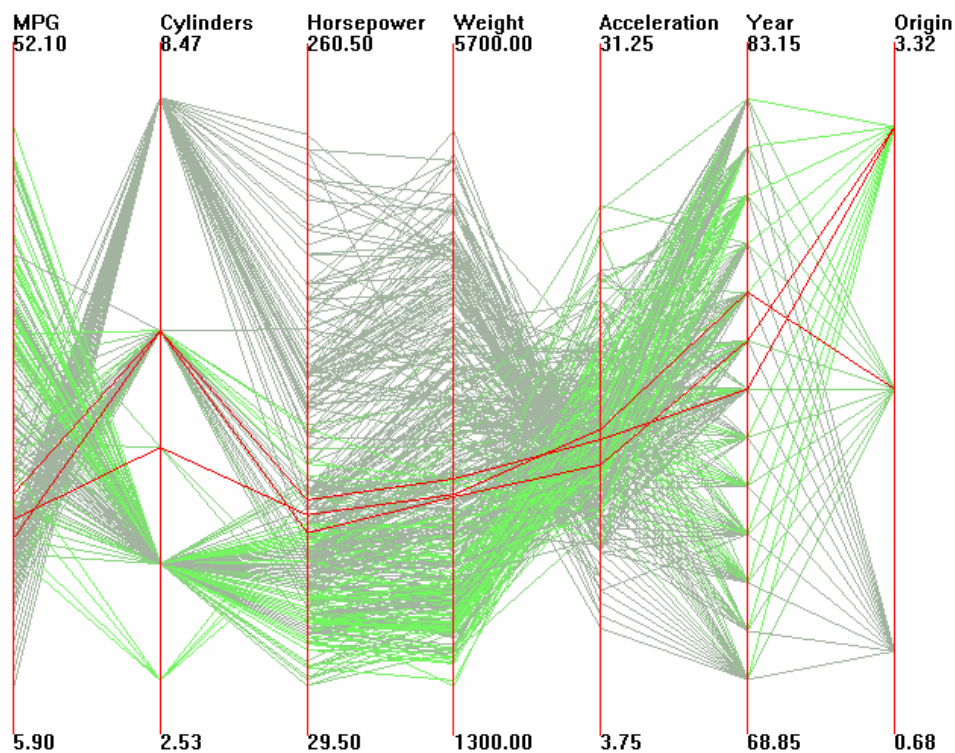


Figura 2.1: Representação visual da técnica Coordenadas Paralelas

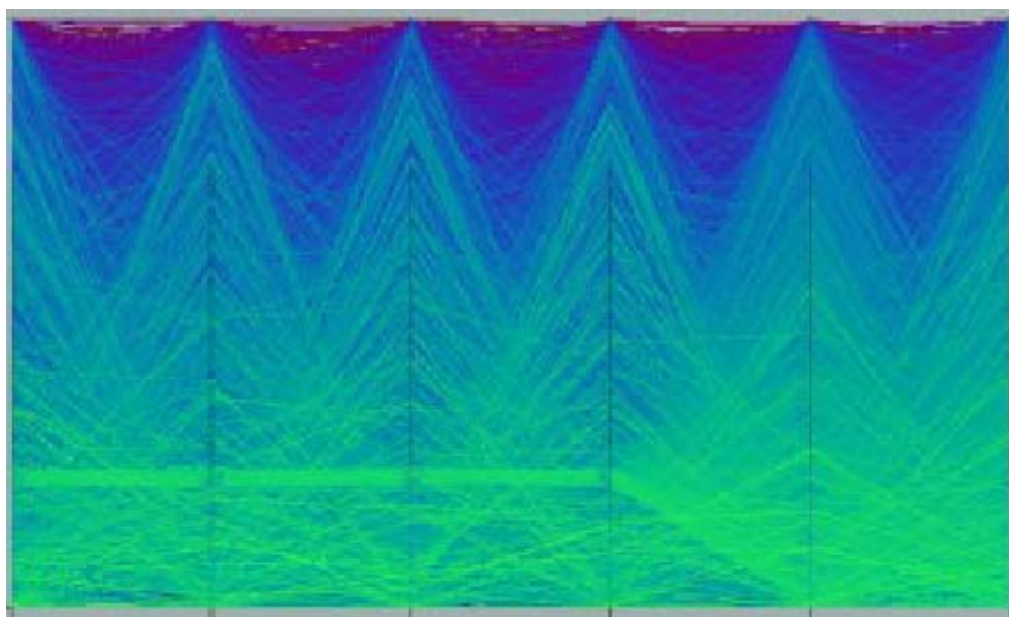


Figura 2.2: Técnica Coordenadas Paralelas com sobreposição de linhas (15.000 itens de dados representados) (KEIM, 1997)

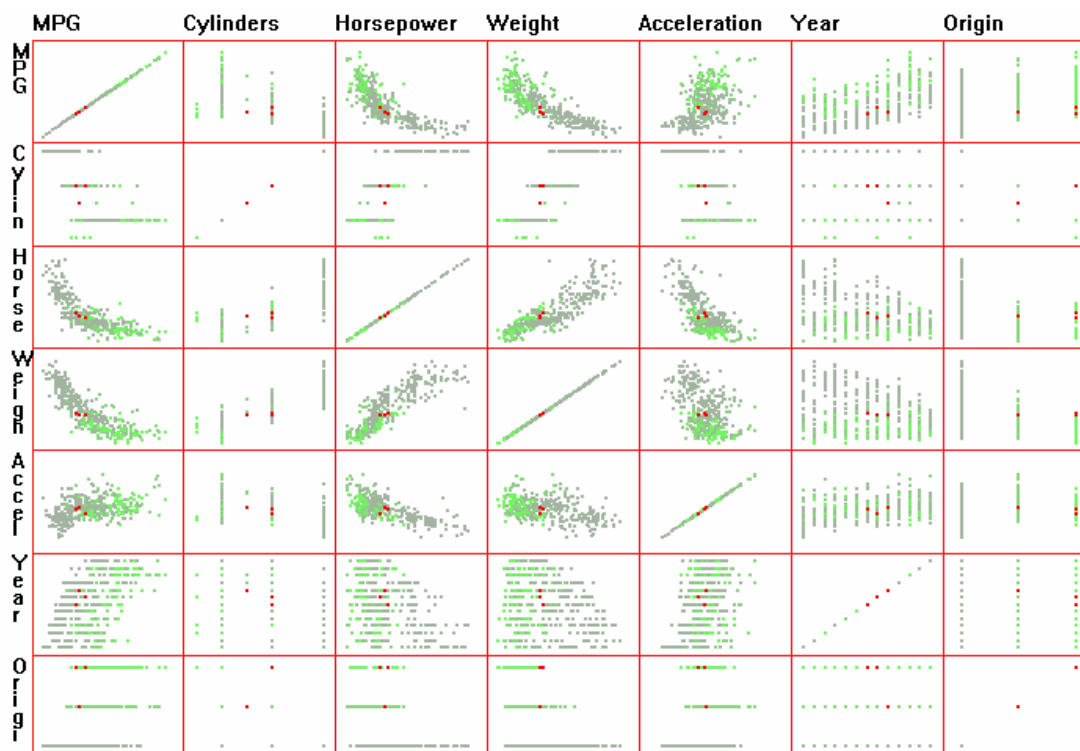


Figura 2.3: Representação visual da técnica Matriz de *Scatter Plots*

A Matriz de *Scatter Plots* é considerada de fácil interpretação, por permitir a visualização de todas as possíveis correlações entre os pares de dimensões. Contudo, a quantidade de dimensões que podem ser apresentadas simultaneamente é uma limitação, devido ao pouco espaço disponível para a projeção de cada gráfico. À medida que o número de dimensões aumenta, torna-se necessário o uso de operações de interação (como *zooming* ou *panning*) para uma efetiva interpretação dos resultados. Hyperslice (VAN, 1993) e HyperBox (ALPERN; CARTER, 1991) são exemplos de variações da Matriz de *Scatter Plots*, que fazem uso de uma matriz de painéis.

2.1.1.3 Radviz

Radviz (HOFFMAN et al., 1997) – visualização de coordenadas radiais – é uma técnica que segue um princípio semelhante a Coordenadas Paralelas. Nessa técnica, n linhas, correspondentes à n dimensões, emanam radialmente do centro de um círculo e terminam no seu perímetro, em pontos igualmente espaçados, denominados *dimensional anchors* (DAs). Cada item de dado projetado está ligado por meio de molas imaginárias às DAs e a posição onde são projetados é aquela onde existe equilíbrio das forças das molas associadas a cada dimensão (HOFFMAN, 1999). Cada ponto de dado é, então, exibido onde a soma das forças das molas é igual a zero. Todos os valores dos itens de dados são, geralmente, normalizados para valores entre 0 e 1 (FAYYAD; GRINSTEIN; WIERSE, 2002). A figura 2.4 mostra um exemplo da representação visual da técnica Radviz.

A técnica Radviz pode mapear vários itens de dados na mesma posição, dependendo da ordem em que os atributos estão distribuídos no círculo. Além disso, outras características desta técnica de visualização são (HOFFMAN, 1999) (FAYYAD; GRINSTEIN; WIERSE, 2002):

- Itens de dados com valores de atributos aproximadamente iguais (depois de normalizados) são posicionados próximos ao centro do círculo;
- Itens de dados com valores similares associados a dimensões em eixos opostos ficam, também, próximos ao centro;
- Valores maiores dos atributos (ou dimensões) atraem o ponto projetado para regiões próximas aos eixos associados a estes atributos.

A posição de um item de dado depende da representação das dimensões ao redor do círculo, sendo que, se a ordem das dimensões for alterada, a representação gráfica também o será.

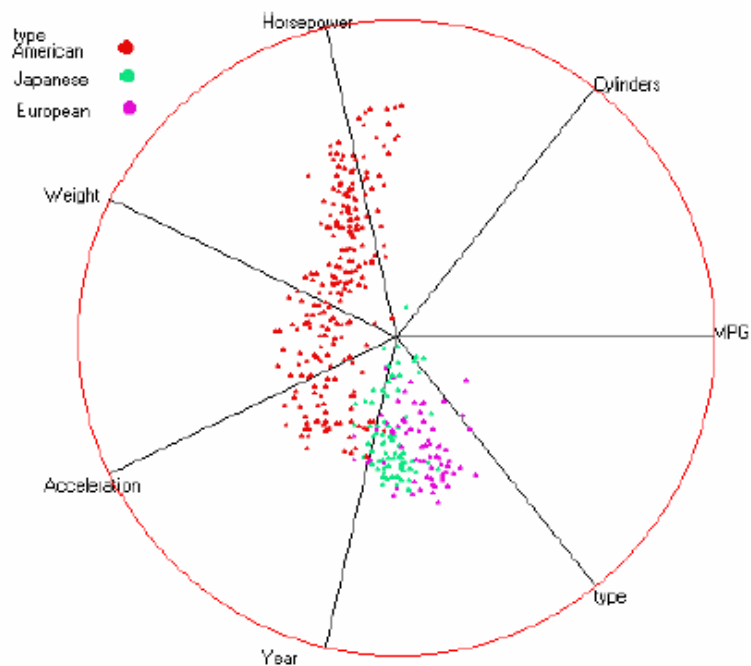


Figura 2.4: Representação visual da técnica Radviz (imagem extraída de VALIATI, 2004)

2.1.1.4 Coordenadas Paralelas Circulares

Coordenadas Paralelas Circulares (denominada, também, como *Gráfico de Estrelas Sobrepostas*) correspondem a uma outra versão circular da técnica Coordenadas Paralelas, consideravelmente, mais próxima à idéia original desta última (FAYYAD; GRINSTEIN; WIERSE, 2002). Basicamente, a forma de mapear e projetar os dados permanece a mesma, apenas as dimensões são dispostas de outra forma (em um formato circular), formando um gráfico em estrela (conforme mostra a figura 2.5).

Seguindo esta forma de projeção geométrica, os segmentos de linha são mais longos na parte externa do círculo, onde os valores maiores são tipicamente mapeados, enquanto os valores menores são projetados para o centro do círculo. Devido à assimetria formada na representação dos valores maiores e menores, certos padrões podem ser facilmente detectados com esta técnica (FAYYAD; GRINSTEIN; WIERSE, 2002).

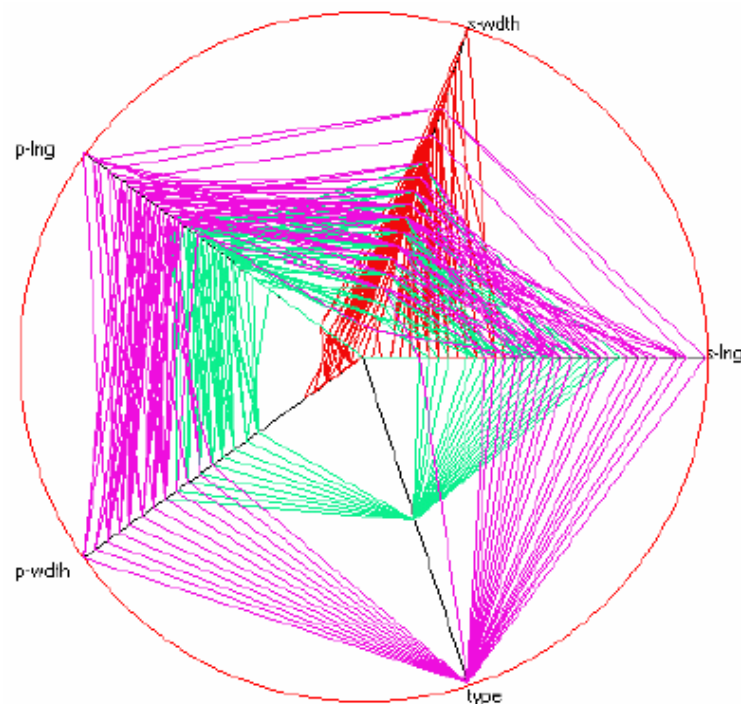


Figura 2.5: Representação visual da técnica Coordenadas Paralelas Circulares (imagem extraída de VALIATI, 2004)

2.1.1.5 Polyviz

Polyviz é uma técnica que busca combinar características de Coordenadas Paralelas e de Radviz. Um dos problemas apresentados pela técnica Radviz é que pontos n -dimensionais com valores bastante diferentes podem ser mapeados para a mesma posição na tela. Esta sobreposição pode ser contornada se os pontos fixados para as molas, em cada dimensão, forem expandidos ao longo dos eixos (FAYYAD; GRINSTEIN; WIERSE, 2002).

Na técnica Polyviz, além de ser empregada esta nova forma de mapeamento dos pontos, a distribuição para cada dimensão é mostrada por linhas emanando dos eixos e dispostas em uma configuração poligonal (ver figura 2.6). Deste modo, procura-se aproveitar a capacidade de exibição de agrupamentos, demonstrada pela técnica Radviz, e a habilidade de visualizar a distribuição de dados em cada dimensão, característica de Coordenadas Paralelas.

2.1.1.6 Survey Plots

Survey Plots (LOHNINGER; INSPECT, 1994) é outra técnica na qual os dados de cada dimensão são mapeados para linhas estendidas a partir de um ponto central. O comprimento de cada linha corresponde ao valor do item de dado sendo representado. Como em gráfico de barras, esta técnica utiliza a idéia de representar os dados estendendo cada ponto sendo mapeado para um determinado eixo. Um princípio semelhante a este é empregado também pelas técnicas *Table Lens* (RAO; CARD, 1994) e Matriz de Permutação (BERTIN, 1983).

A visualização de *Survey Plots* permite que correlações entre quaisquer duas dimensões possam ser percebidas, especialmente, quando os dados são classificados de acordo com uma dimensão particular (FAYYAD; GRINSTEIN; WIERSE, 2002). A figura 2.7 mostra um exemplo de visualização *Survey Plots* (representando os dados sobre carros), na qual os dados estão classificados por número de cilindros (valores do atributo *Cylinders*).

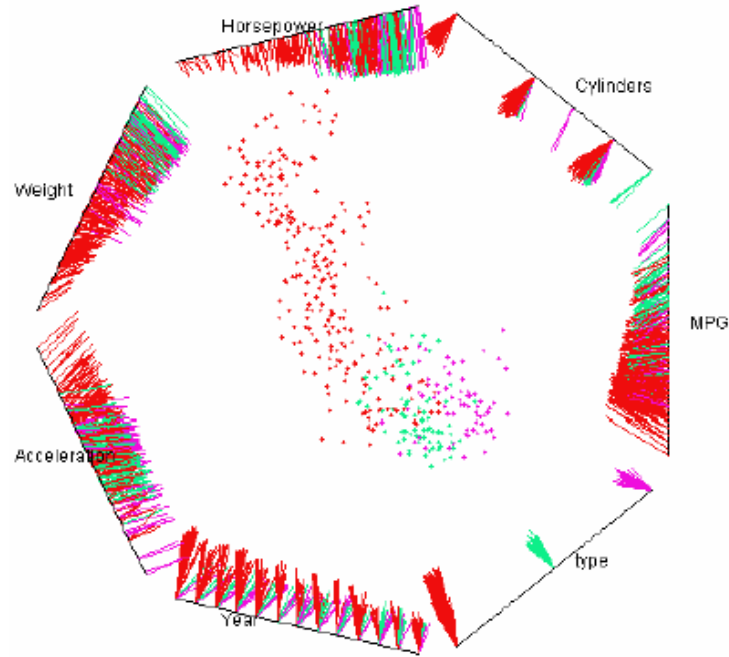


Figura 2.6: Representação visual da técnica Polyviz (HOFFMAN et al., 2000)

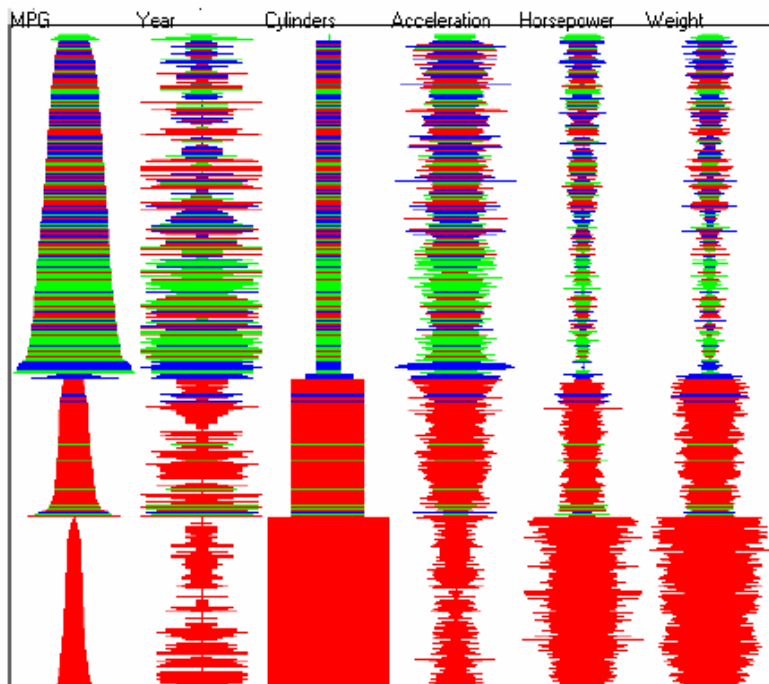


Figura 2.7: Representação visual da técnica *Survey Plots* (imagem extraída de VALIATI, 2004)

2.1.2 Técnicas Iconográficas

Há dois tipos de visualizações do tipo iconográficas: glifos e ícones. Em cada uma destas, as dimensões de um conjunto de dados são mapeadas para certas características dos glifos ou ícones. Assim, cada glifo ou ícone representa um item de dado com suas n -dimensões.

2.1.2.1 Faces de Chernoff

Uma das mais conhecidas visualizações iconográficas é Faces de Chernoff (CHERNOFF, 1973). Nesta última, as dimensões dos dados são mapeadas para características de uma face, como formato do nariz, boca, olhos e da face como um todo (ver figura 2.8(a)), resultando em diferentes formatos de faces (definidas segundo os valores sendo mapeados). Contudo, a efetividade desta técnica é amplamente discutida, devido à dificuldade de percepção de diferenças muito pequenas nas imagens e pela mesma ser inadequada para a identificação de agrupamentos.

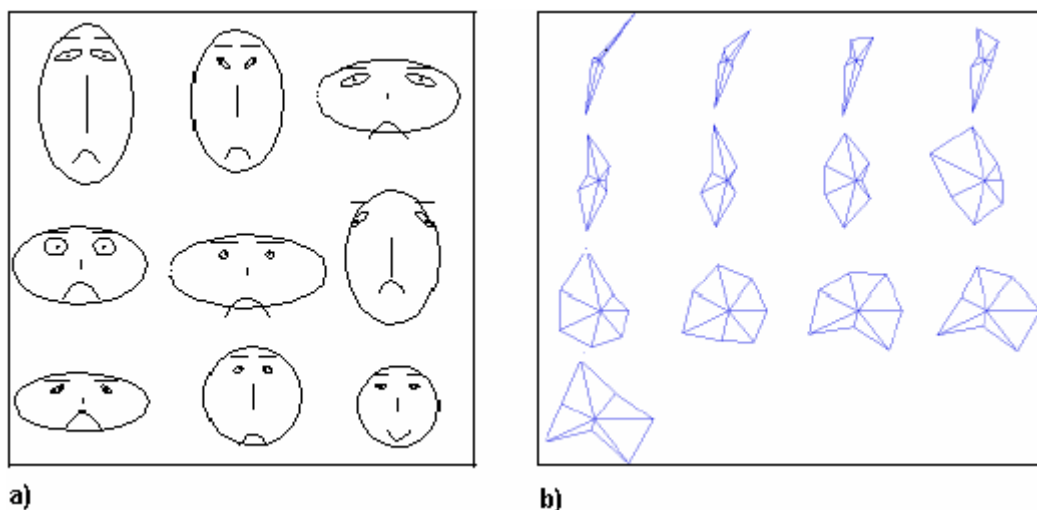


Figura 2.8: Técnicas iconográficas: a) Faces de Chernoff e b) Glifos em Estrela (KEIM, 1997)

2.1.2.2 Glifo em Estrela

Na técnica Glifo em Estrela (ver figura 2.8 (b)), cada item de dado é representado por um glifo em forma de estrela. Em cada um destes, as dimensões dos dados são representadas como raios de ângulos iguais, partindo do centro de um círculo. O comprimento de um raio indica o valor de um atributo. Raios mais extensos representam valores altos e o centro do círculo indica o valor mínimo da dimensão.

2.1.2.3 Stick Figure

A técnica *Stick Figure* caracteriza-se pelo uso de ícones básicos. As várias dimensões dos dados são mapeadas para parâmetros dos ícones, tais como o comprimento e o ângulo de rotação de uma aresta (ou galho). Variações de comprimento, espessura e cor das arestas fornecem outras possibilidades de representação, explorando a capacidade humana de percepção de texturas e interpretação de padrões (FAYYAD; GRINSTEIN; WIERSE, 2002). A imagem da esquerda da figura 2.9 apresenta uma tabela de ícones, que representam diversos

parâmetros do censo da Inglaterra, e a imagem da direita da mesma figura mostra estes ícones plotados em coordenadas x e y, indicando renda e idade.

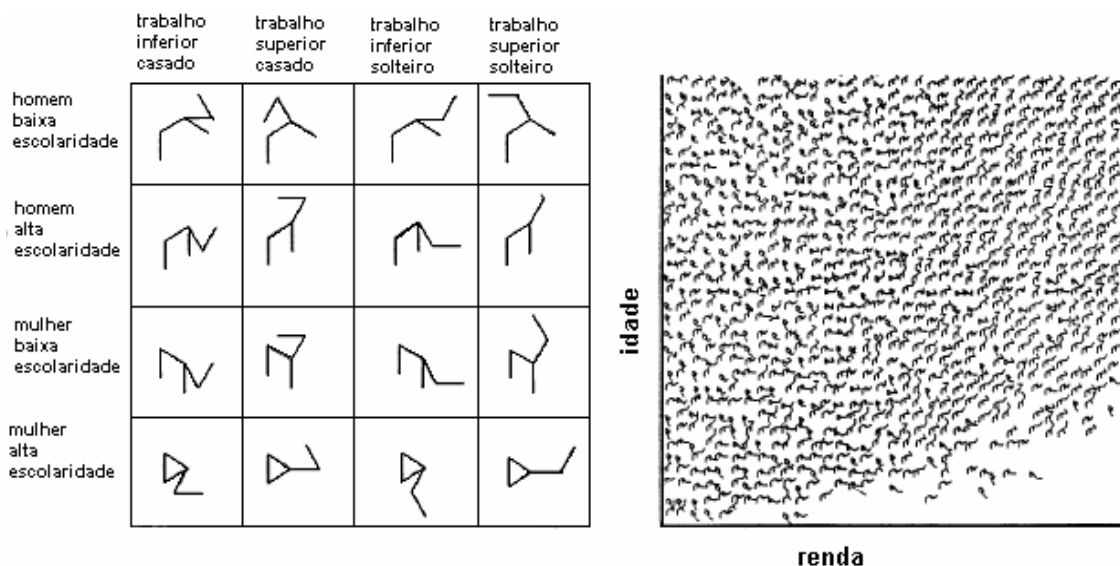


Figura 2.9: Mapeamento de *Stick Figures* (esquerda) e gráfico iconográfico de dados do censo da Inglaterra (direita) (FAYYAD et al., 2002)

2.1.3 Técnicas Orientadas a Pixel

Técnicas dessa categoria caracterizam-se por mapear o conjunto de valores de cada atributo (ou dimensão) dos dados em *pixels* na tela. O conjunto de valores de cada atributo é exibido em janelas individuais, ou seja, para um conjunto de dados que possui *n* atributos, a tela é dividida em *n* janelas. Em cada uma das janelas, cada valor do atributo é representado por um *pixel*, colorido conforme o valor sendo representado. A distribuição espacial dos *pixels* na janela pode ser determinada de diferentes maneiras (ver figura 2.10), de modo que relações ou significados nos dados possam ser percebidos pela análise das regiões correspondentes nas janelas (KEIM, 1996).

As formas utilizadas para a distribuição dos *pixels* nas janelas podem ser divididas em técnicas dependentes e independentes do resultado de uma consulta. Nas técnicas independentes de consulta, os valores dos atributos são mapeados diretamente para os *pixels* e exibidos nas janelas, utilizando-se curvas de preenchimento de espaço (curvas de Peano-Hilbert ou Morton) ou padrões recursivos (ver figura 2.11), que proporcionam um melhor agrupamento de itens de dados relacionados e uma distribuição semanticamente significativa.

Por outro lado, nas técnicas dependentes de consulta, ao invés dos valores dos atributos serem mapeados diretamente para *pixels*, as distâncias destes valores a uma determinada consulta é que são mapeadas. Desta forma, além dos itens de dados que satisfazem a consulta, também podem ser exibidos aqueles valores que se aproximam da resposta, através da exibição da distância de cada valor a um respectivo valor estabelecido na consulta. O cálculo da distância depende da aplicação e do tipo de dado. Os atributos referentes a um item de dado ocupam a mesma posição relativa nas suas respectivas janelas, sendo que a ordem de exibição dos itens na janela é determinada por uma distância global. Essa distância é calculada pela combinação das distâncias de cada atributo, ponderadas por um peso correspondente, que determina a relevância dada ao atributo.

Visto que o foco está na relevância dos dados com respeito a uma consulta, diferentes formas de distribuição dos *pixels* podem ser apropriadas: em espiral, onde as respostas corretas são apresentadas no centro da janela e os valores próximos ao redor desta região, ou por eixos, indicando distâncias positivas e negativas (ver figura 2.10).

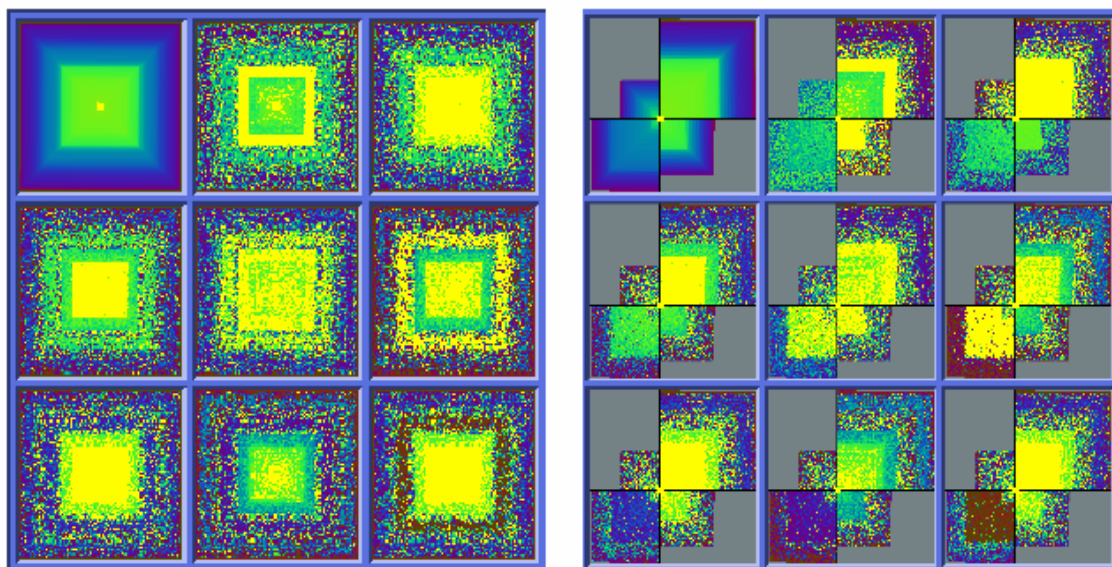


Figura 2.10: Visualizações orientadas a pixel de um conjunto de dados com nove dimensões, utilizando duas formas de distribuição espacial: espiral (esquerda) e por eixos (direita) (KEIM, 1997).

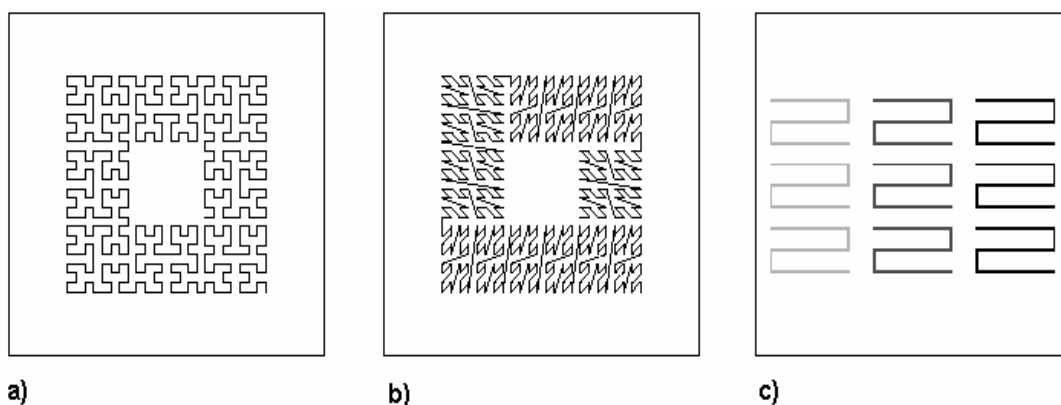


Figura 2.11: Técnicas independentes de consulta: a) curvas de Peano-Hilbert, b) curvas de Morton e c) padrões recursivos (KEIM, 1996) (KEIM, 1997)

Embora o formato retangular para as janelas permita um uso racional do espaço na tela, a percepção de relacionamentos entre os atributos pode ser dificultada quanto maior for o número de dimensões a serem visualizadas. Uma alternativa de solução a este problema é a utilização de um formato circular, adotado, por exemplo, na técnica Segmentos de Círculo (ver figura 2.12).

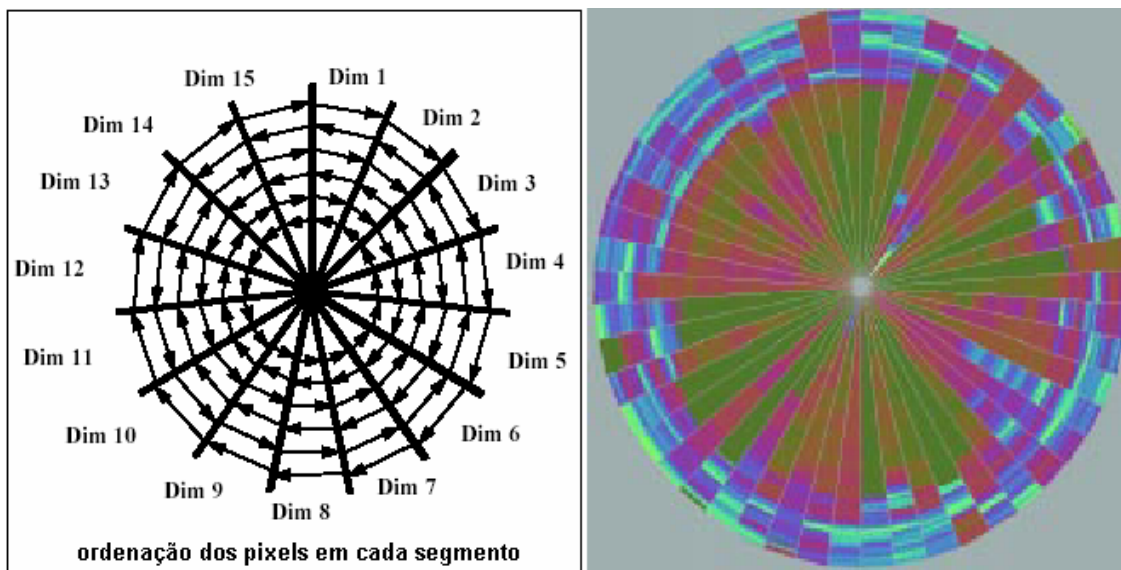


Figura 2.12: Técnica Segmentos de Círculo (KEIM, 1997).

2.2 Técnicas de Interação

Além das técnicas de visualização, para uma exploração de dados efetiva, é necessário também o uso de algumas técnicas de interação. Estas últimas permitem que um usuário possa interagir diretamente com visualizações e alterá-las dinamicamente de acordo com seus objetivos de exploração. A possibilidade de interagir com representações visuais pode reduzir consideravelmente as desvantagens e pontos fracos de algumas técnicas de visualização (principalmente, daquelas que apresentam desordem visual e sobreposição de objetos), fornecendo ao usuário mecanismos para manipular a complexidade de conjuntos de dados (OLIVEIRA; LEVKOWITZ, 2003).

Keim (1997) identificou as técnicas de interação que são utilizadas para a exploração de conjuntos de dados multidimensionais e organizou-as em seis classes: mapeamento de dados para propriedades visuais, projeções, filtragem (seleção e consulta), *brushing and linking*, zoom e detalhes por demanda. A seguir, as técnicas de interação aplicadas para a visualização de dados multidimensionais serão descritas, utilizando a categorização proposta por Keim.

2.2.1 Mapeamento de Dados para Propriedades Visuais

Esta categoria inclui as técnicas de interação que realizam mapeamentos de atributos (dimensões) dos dados para parâmetros de uma visualização. Estes parâmetros são atributos visuais de objetos representando itens de dados (como cor, tamanho, transparência, orientação, etc.). A tabela 2.2 mostra exemplos típicos de mapeamentos para algumas propriedades visuais.

A codificação de atributos de dados para propriedades visuais de uma visualização aumenta a densidade dos dados e a variedade de representações que podem ser geradas (STOLTE; HANRAHAN, 2001). Além disso, este tipo de mapeamento aumenta o número de dimensões que podem ser visualizadas simultaneamente. Por exemplo, um *scatter plot* 2D representa apenas dois atributos (dispostos nos eixos x e y), mas outros

atributos podem ser representados simultaneamente se estes forem mapeados para atributos visuais de objetos.







A fim de manter uma especificação breve, não deve ser requerida do usuário a construção de mapeamentos. O usuário deve apenas especificar qual atributo de dados deve ser mapeado para uma determinada propriedade visual. O sistema é que deve ser responsável por gerar um mapeamento efetivo dos valores do atributo para variações da propriedade visual.

Diferentes símbolos (ver tabela 2.2) podem ser utilizados para o mapeamento de atributos nominais, desde que o número de valores do atributo seja pequeno. O tamanho de objetos pode ser usado para mapear tanto atributos nominais como quantitativos. Quando mapeando atributos quantitativos, um mapa linear a partir dos valores de atributo para o tamanho dos objetos é criado. Por outro lado, se um atributo nominal é mapeado, o número de valores precisa ser pequeno (no máximo quatro ou cinco), de forma que o usuário consiga distinguir entre diferentes categorias (BERTIN, 1983).

Quando a cor é utilizada para mapear atributos nominais, os valores de atributo devem ser associados a cores bem distintas entre si (ver tabela 2.2). Quando os atributos mapeados são quantitativos, geralmente, apenas a nuance de cor é variada.

Um princípio chave na geração de mapeamentos de atributos nominais para a propriedade orientação é que a orientação precisa variar pelo menos 30 graus entre as categorias (KOSSLYN, 1994), o que restringe o número de categorias para no máximo seis. Para atributos quantitativos, a orientação varia linearmente com os valores de atributo.

Tabela 2.2: Propriedades visuais que podem ser usadas para mapear atributos de dados e exemplos de mapeamentos típicos

<i>propriedade visual</i>	<i>objetos</i>	<i>mapeamento nominal</i>	<i>mapeamento quantitativo</i>
figura	glifo	○ □ + △ S U	
tamanho	retângulo, círculo, linha, glifo, texto		
cor	retângulo, círculo, linha, glifo, texto		
orientação	retângulo, linha, texto		

Fonte: STOLDE; HANRAHAN, 2001

2.2.2 Projeções

A idéia das técnicas interativas de projeção é mudar, dinamicamente, os atributos de dados (dimensões) que são projetados em eixos de uma visualização. O objetivo é explorar um conjunto de dados multidimensional, permitindo correlações entre diferentes atributos de dados. Um exemplo clássico de sistema que emprega projeções é o Grand-Tour (ASIMOV, 1985). Ele tenta mostrar todas as projeções bi-dimensionais interessantes de um conjunto de dados multidimensional como uma série de *scatter plots*. O número de projeções possíveis é exponencial no número de dimensões e,

devido a isto, esta organização é inviável para conjuntos de dados de alta dimensionalidade. A seqüência de projeções mostrada pode ser randômica, manual ou pré-computada (KEIM, 2002).

2.2.3 Filtragem

Ao explorar grandes conjuntos de dados, é importante particionar, interativamente, estes dados em diferentes segmentos e, então, focar no subconjunto de interesse (KEIM, 2002). Isto pode ser feito através da seleção direta do subconjunto desejado (*browsing*) ou pela especificação de propriedades do subconjunto alvo (consulta). Ambas as possibilidades são formas de filtragem.

A seleção é realizada diretamente sobre uma representação visual, com o auxílio de um dispositivo de apontamento. Sua ênfase é na filtragem rápida, destacando um subconjunto dos dados. No entanto, o uso desta técnica é muito difícil para a filtragem de grandes conjuntos de dados, quando ocorre desordem visual e sobreposição de objetos.

As técnicas de consulta dinâmica permitem ao usuário manipular *widgets* gráficos (*sliders*, botões, etc.) para controlar a quantidade de dados visualizados (AHLBERG; WILLIAMSON; SHNEIDERMAN, 1992). Cada *widget* corresponde a um atributo do conjunto de dados. Esta técnica funciona melhor para dados quantitativos, mas pode também ser usada na manipulação de dados nominais (GOLDSTEIN; ROTH, 1994). Neste último caso, o usuário deve selecionar elementos a partir de uma lista de valores. A figura 2.13 mostra uma interface de consultas dinâmicas, usada para selecionar dinamicamente itens de dados, com base nos seus valores para os atributos *Latitude*, *Longitude* e *Interval* (intervalo de anos). O mapa maior mostra todos os itens de dados da região de interesse (delimitada pelos valores de *Latitude* e *Longitude*) e o mapa menor (acima à direita) mostra apenas os itens dentro do intervalo de 30 a 50 anos.

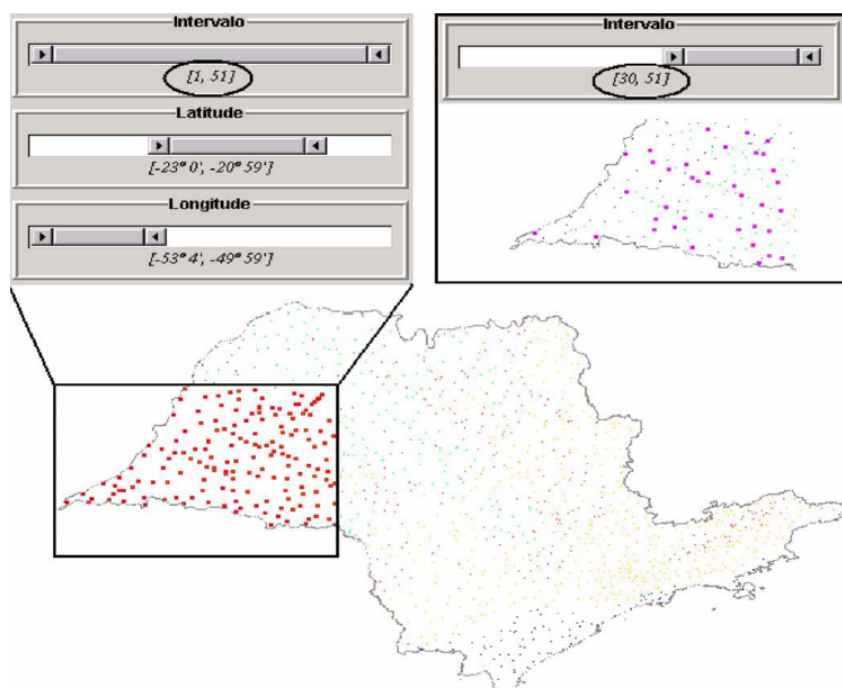


Figura 2.13: Usando consultas dinâmicas para verificar a concentração espacial de itens de dados em uma região de interesse (SHIMABUKURO, 2004)

As vantagens da consulta dinâmica são que *widgets* gráficos são fáceis de manipular e os efeitos da interação são vistos imediatamente na visualização. Ao utilizar *sliders*, por exemplo, o usuário pode acompanhar a mudança progressiva na quantidade de dados visualizados, à medida que a barra do *slider* vai sendo movida. Além disso, consultas dinâmicas permitem que o usuário possa descobrir rapidamente quais segmentos do espaço de pesquisa multidimensional estão muito ou pouco populadas, onde há agrupamentos, lacunas ou *outliers*, e quais tendências existem em dados nominais (SHNEIDERMAN, 1994). Por outro lado, a principal desvantagem dessa técnica interativa é a dificuldade de representar consultas mais elaboradas, como por exemplo, disjunções. Um exemplo de disjunção seria a combinação de dois intervalos distintos de valores em um único *slider*.

Como as técnicas de filtragem mencionadas acima (seleção e consultas dinâmicas) apresentam limitações claras, várias outras técnicas têm sido desenvolvidas para melhorar a filtragem interativa na exploração de dados. Um exemplo de ferramenta interativa que pode ser usada para filtragem é *magic lenses* (BIER et al., 1993) (FISHKIN; STONE, 1995). A idéia básica desta última é usar uma ferramenta como uma lente, para filtrar dados diretamente na visualização. Os dados debaixo da lente são processados por um filtro e exibidos diferentemente do restante dos dados (várias lentes com diferentes filtros podem ser usadas). A técnica mostra uma visualização modificada da região selecionada, enquanto o resto da visualização permanece igual. Como a “lente” pode ser movimentada por toda a área de uma visualização (utilizando o mouse), as *magic lenses* são também chamadas de filtros móveis. A figura 2.14 mostra o uso de dois filtros de *magic lenses*, um deles mostra uma visão aramada de blocos 3D e o outro uma visão aumentada 2D.

As principais vantagens do uso dessa técnica são a possibilidade de visualizar detalhes e contexto simultaneamente e modificar a apresentação de objetos para revelar informações escondidas, aumentar os dados de interesse ou eliminar informações desnecessárias do foco (BIER et al., 1993).

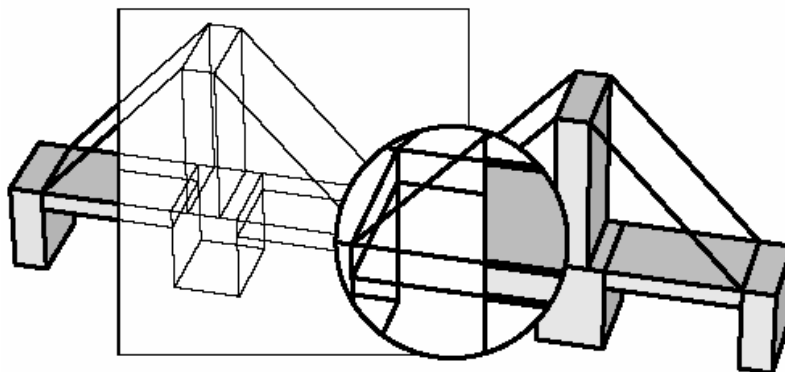


Figura 2.14: *Magic lenses*: Uma lente de visão aramada 3D e uma lente de aumento 2D (BIER et al., 1993).

2.2.4 Zoom

Zoom é uma técnica interativa muito conhecida e amplamente utilizada por diversas aplicações. Ao representar graficamente conjuntos de dados com um grande número de itens de dados, é importante que estes últimos sejam exibidos de forma comprimida, com o objetivo de fornecer uma visão de *overview* do conjunto inteiro. Mas, ao mesmo

tempo, é interessante fornecer também uma visualização diferenciada dos dados, em diferentes resoluções.

A técnica de zoom não significa apenas mostrar objetos de dados maiores, mas significa também que a representação dos dados muda automaticamente para apresentar mais detalhes, em níveis de zoom mais altos (KEIM, 2002). Os objetos podem, por exemplo, ser representados em um único pixel, em níveis de zoom baixos; como ícones, em um nível de zoom intermediário; e como objetos nomeados, em resoluções altas. Uma visão *overview* permite ao usuário detectar modelos, correlações e *outliers* no conjunto de dados, já uma visão com nível de zoom mais alto permite melhor exploração de uma área de interesse, pois os itens de dados são exibidos com maior detalhe.

2.2.5 Detalhes por Demanda

Essa técnica representa a possibilidade de obter, interativamente, mais detalhes sobre os dados visualizados. Estes detalhes podem ser, por exemplo, valores de atributo de um item de dado ou ícone (ou glifo) ou informações adicionais destes.

Detalhes por demanda é uma técnica oferecida como alternativa mais rápida a consultas dinâmicas. Em vez de compor consultas e interpretar seus resultados, o usuário pode, em qualquer momento que sentir necessidade, solicitar rapidamente a exibição de detalhes acerca de um item de dado. Esta solicitação, geralmente, é realizada por meio da seleção direta do item na visualização, e os dados de detalhe solicitados, usualmente, são exibidos em um frame *pop-up* (como na figura 2.15, que mostra informações adicionais sobre um filme selecionado) ou no painel de controle da visualização.

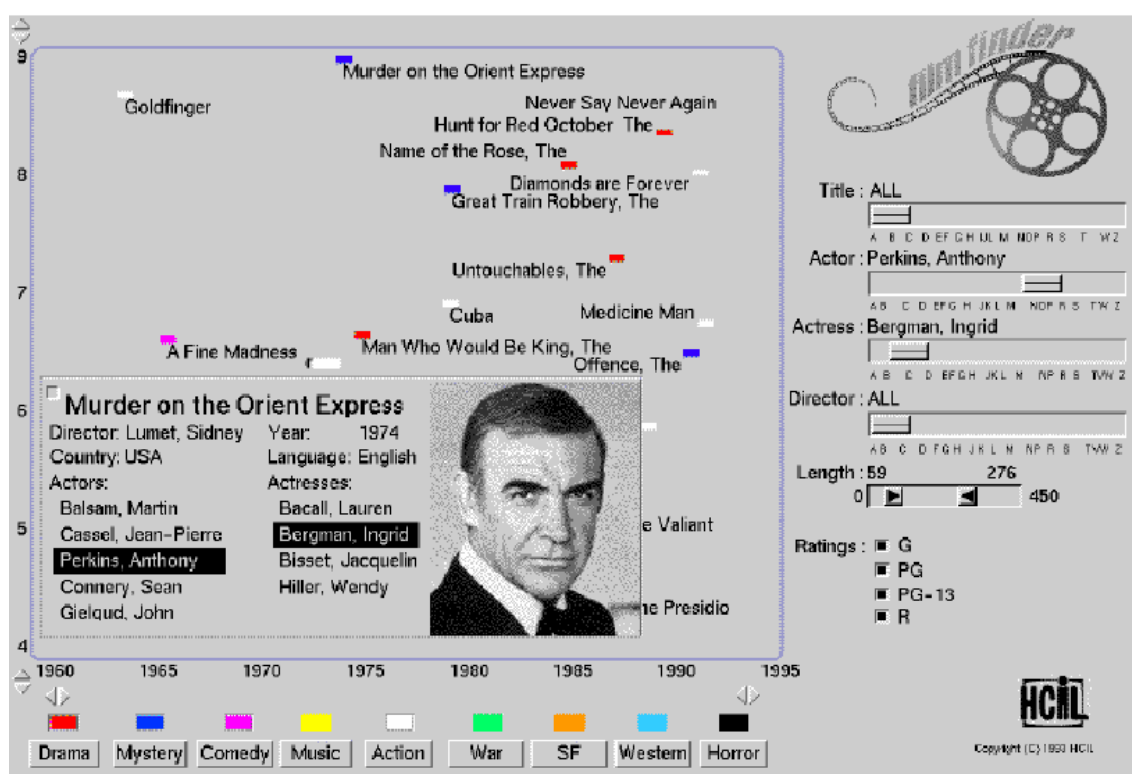


Figura 2.15: Usando a técnica detalhes por demanda para obter informações adicionais sobre um item de dado (extraído de AHLBERG; SHNEIDERMAN, 1994).

3 VISUALIZAÇÕES COORDENADAS: CONCEITOS E REVISÃO BIBLIOGRÁFICA

Conectar e relacionar informações em uma visualização com informações em outra auxilia o usuário no processo de exploração de um conjunto de dados e pode fornecer compreensões adicionais acerca do mesmo (ROBERTS, 2004). North e Shneiderman (1997), em seus experimentos com usuários, descobriram que a coordenação de múltiplas visualizações oferece como vantagens a melhora na performance do usuário, a descoberta de relacionamentos inesperados e a unificação do desktop.

Muitas formas diferentes de interações podem ser coordenadas, como operações de manipulação, consultas, seleções e mapeamentos visuais. Além disso, uma ação em determinada visualização pode ser conectada à mesma ação em outra visualização ou a uma ação diferente (a possibilidade de cada combinação deve ser verificada, pois nem todas são possíveis). Ainda, também há formas diferentes para especificar um relacionamento de coordenação, que pode ser unidirecional ou bidirecional. Considerando um par de visualizações interligadas, apenas uma delas pode afetar a outra, com uma ligação unidirecional (único sentido). Já se a ligação estabelecida for bidirecional (duplo sentido), ambas as visualizações são mutuamente afetadas.

Duas visualizações interligadas, geralmente, representam dados de um mesmo conjunto, cada uma utilizando uma técnica de representação visual diferente. Outra possibilidade, no entanto, é permitir que visualizações interligadas representem conjuntos de dados diferentes. Neste caso, os conjuntos de dados considerados devem possuir uma relação entre si, de forma que seja possível para uma aplicação estabelecer um mapeamento de informações em uma visualização para informações em outra.

A seção abaixo apresenta as principais formas de coordenações utilizadas por sistemas de múltiplas visualizações coordenadas e as vantagens que o uso das mesmas oferece, e a seção seguinte a esta descreve alguns dos principais sistemas de visualizações coordenadas relacionados a esse trabalho.

3.1 Tipos de Coordenações

North e Shneiderman (1997) determinaram que as ações realizadas por usuários no processo de exploração de dados são limitadas à seleção de itens e navegação em janelas (zoom, *scroll*, *pan*, rotação, etc.). Com base nisto, eles desenvolveram uma taxonomia de coordenações que abrange três possibilidades de combinação destas ações (uma ação em uma visualização ligada à outra ação numa segunda visualização): seleção $\leftarrow\rightarrow$ seleção, navegação $\leftarrow\rightarrow$ navegação e seleção $\leftarrow\rightarrow$ navegação. O símbolo $\leftarrow\rightarrow$ indica bidirecionalidade, o que significa que estas ligações são válidas nos dois

sentidos. Os mesmos autores também determinaram que as três possibilidades de coordenação podem ser aplicadas para visualizações que representam o mesmo conjunto de dados, bem como para visualizações que representam conjuntos relacionados.

No entanto, Pattison e Phillips (2001) discordaram das determinações acima, indicando outras formas de coordenação além da seleção e navegação, como por exemplo, a ordenação de itens e agrupamentos. Da mesma forma, Boukhelifa et al. (2003) acredita em um uso mais amplo de coordenações, exemplificado por um modelo em camadas, onde o usuário pode ligar qualquer aspecto do *dataflow* e processo de exploração.

Muitas são as coordenações citadas na literatura (muitas vezes a mesma técnica é tratada por diferentes nomenclaturas), mas não há uma única categorização amplamente aceita que organize as mesmas. Por esta razão, a seguir serão descritas algumas coordenações, que foram identificadas como as mais utilizadas em sistemas de múltiplas visualizações coordenadas para exploração de dados. As sete primeiras coordenações descritas (*brushing-and-linking*, reordenação de dimensões sincronizada, filtragem sincronizada, mapeamento de dados para propriedades visuais, *scrolling* sincronizado, zoom sincronizado e rotação sincronizada) conectam a mesma ação em todas as visualizações, ao contrário das duas últimas coordenações (*overview* e *detalhe* e *drill down*), que conectam ações diferentes em cada visualização. Entre todas as técnicas descritas abaixo, apenas *drill down* não costuma ser implementada de forma bidirecional.

3.1.1 *Brushing-and-Linking*

Brushing é um processo no qual um usuário pode destacar, selecionar ou apagar um subconjunto de elementos representados graficamente, apontando-os com o mouse (MARTIN; WARD, 1995). Quando múltiplas visualizações dos dados estão sendo exibidas simultaneamente, o *brushing* é associado com outro processo, conhecido como *linking*. Neste caso, elementos selecionados em uma visualização são automaticamente destacados em todas as demais, tornando possível a detecção de dependências e correlações. Por exemplo, na figura 3.1, a seleção dos estados com mais alto nível de educação nos EUA, na visualização *scatter plot*, revela que estes estados estão localizados na região nordeste do país. Através da coordenação *brushing-and-linking*, os itens selecionados na *scatter plot* (em amarelo) são automaticamente selecionados também na representação de mapa.

Brushing-and-linking é o tipo de coordenação mais utilizada pelos sistemas de visualizações coordenadas e é empregada para a análise de dados há muitos anos. Recentemente, esta técnica tem sido usada também para ajudar usuários a obterem informações adicionais sobre pontos de dados selecionados. Alguns sistemas que implementam esse tipo de coordenação são Spotfire (AHLBERG; WISTRAND, 1995), Visage (ROTH et al., 1996) e Snap-Together (NORTH, 2000), sistema no qual foi gerado o cenário da figura 3.1.

A coordenação *brushing-and-linking* é utilizada com maior frequência para visualizar equivalências entre diferentes representações do mesmo conjunto de dados, mas a mesma é também utilizada, por alguns sistemas, para visualizar relações arbitrárias entre diferentes conjuntos de dados. Por exemplo, registros de dados relacionados, mas em diferentes tabelas, podem ser destacados consultando uma base de dados relacional durante o processo de *brushing-and-linking*.

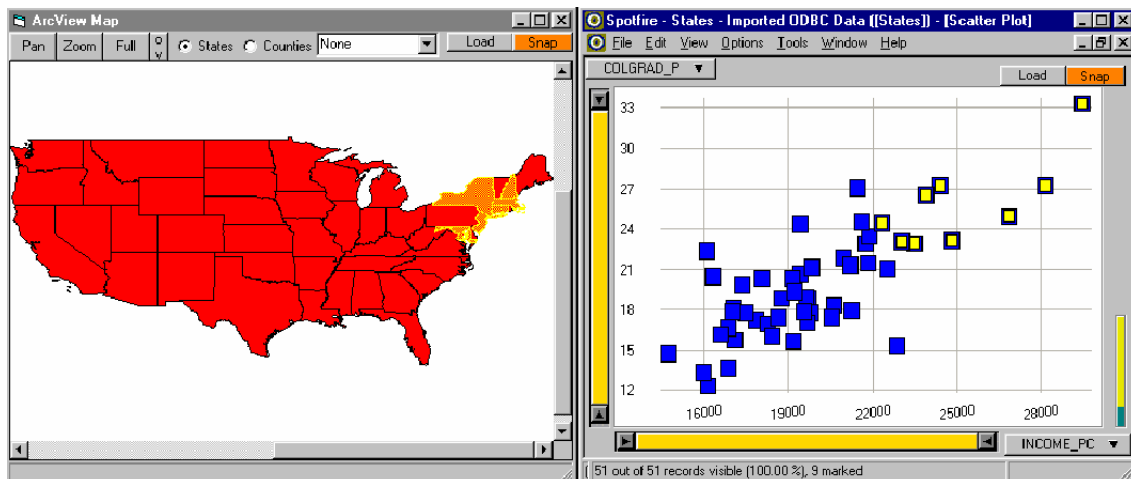


Figura 3.1: Visualizações coordenadas pela técnica *brushing-and-linking* (NORTH, 2000)

3.1.2 Reordenação de Dimensões Sincronizada

Esse tipo de coordenação permite que um usuário possa mudar a disposição em que atributos de dados (dimensões) aparecem em diferentes visualizações multidimensionais, simultaneamente. Na maioria das técnicas de visualização multidimensionais, os atributos do conjunto de dados são posicionados na tela em um arranjo uni ou bi-dimensional (YANG et al., 2003). Devido a isto, deve ser escolhida alguma ordem para as dimensões. Esta ordem geralmente interfere na expressividade da visualização, pois diferentes arranjos de dimensões podem revelar aspectos diferentes dos dados e afetar a clareza e estrutura da representação visual (PENG; WARD; RUNDENSTEINER, 2004). Assim, características bem diferentes podem ser percebidas do conjunto de dados, dependendo do arranjo das dimensões na visualização.

Apenas alguns sistemas de visualizações multidimensionais permitem a reordenação de dimensões e esta é realizada manualmente pelo usuário. XmdvTool (WARD, 1994) é um exemplo de sistema que permite coordenar a reordenação de dimensões. Quando o usuário solicita manualmente uma alteração na ordem das mesmas, essa alteração é realizada em todas as visualizações multidimensionais disponíveis, simultaneamente.

3.1.3 Filtragem Sincronizada

O processo de filtragem consiste em particionar um conjunto de dados, dinamicamente, com o objetivo de focar em um determinado subconjunto de interesse (KEIM, 2002). No contexto de visualizações coordenadas, a filtragem é, geralmente, utilizada como consultas dinâmicas (AHLBERG; SHNEIDERMAN, 1994). O usuário habilita a eliminação de pontos de dados das visualizações, especificando intervalos de valores para atributos em consultas. Cada *widget* gráfico (geralmente, um *slider*) de consulta dinâmica representa um atributo de dados. Conforme o usuário muda o intervalo de um filtro de consulta dinâmica, todas as visualizações coordenadas são progressivamente alteradas, para exibir o novo montante de pontos de dados.

City'O'Scope (BRODBECK; GIRARDIN, 2003) é um exemplo de sistema que coordena o processo de filtragem da forma descrita acima. Outra forma de coordenar filtragem é implementada por sistemas de *filter-flow*, como Linkwinds (JACOBSON; BERKIN; ORTON, 1994). Neste tipo de sistema, o usuário conecta controles de filtro

de consulta dinâmica e visualizações em uma rede *pipeline*. Selecionando intervalos para dimensões em um controle ou filtros da visualização, os dados mostrados seguem no *pipeline*. Em todos os sistemas pesquisados na literatura, a filtragem sincronizada é utilizada apenas em visualizações que representam o mesmo conjunto de dados.

A filtragem é uma técnica muito útil para analisar a distribuição dos dados (agrupamentos e lacunas) e tendências, assim como reduzir a desordem visual das representações. Quando empregada de forma coordenada, esta técnica evita que o usuário tenha que, repetidas vezes, focar uma visualização em um determinado subconjunto de dados, para visualizá-lo em todas as representações visuais.

3.1.4 Mapeamento de Dados para Propriedades Visuais

Esse tipo de coordenação mapeia valores de atributos (dimensões) para parâmetros visuais de objetos de dados de visualizações, como cor, tamanho, transparência, orientação, entre outros (ver tabela 2.2 para exemplos de mapeamentos típicos). O usuário deve indicar qual atributo de dados deve ser mapeado para uma determinada propriedade visual, selecionando o nome do atributo no componente de interface gráfica que representa a propriedade em questão. Logo após a seleção, todas as visualizações coordenadas exibirão seus objetos de dados com os valores do atributo selecionado pelo usuário, mapeados para a propriedade visual por ele também escolhida. O mesmo tipo de mapeamento visual é aplicado em todas as visualizações.

A coordenação de mapeamentos de dados para propriedades visuais auxilia o usuário a identificar o posicionamento de determinados objetos de dados nas diferentes visualizações e perceber tendências gerais do conjunto de dados. O sistema XmdvTool (WARD, 1994) permite a coordenação de mapeamentos de dados para diferentes mapas de cores. A figura 3.2 mostra as técnicas Coordenadas Paralelas e Glifos em Estrela utilizando o mesmo mapa de cores, que representa os valores do atributo *Origin*. Preto representa o valor mais baixo do atributo e verde claro o valor mais alto.

3.1.5 Scrolling Sincronizado

Esse tipo de coordenação conecta as *scroll bars* de duas ou mais janelas, mantendo-as sincronizadas. Uma *scroll bar* é um mecanismo típico de navegação em janela (*panning*), sendo assim, as técnicas *scrolling* e *panning* sincronizado são análogas. Quando o usuário movimenta a *scroll bar* de uma janela, as demais são deslocadas automaticamente para a posição correspondente à primeira. Esta navegação sincronizada evita a perda do relacionamento entre as visualizações e a repetição tediosa da ação de navegação em cada uma das janelas, pelo usuário (NORTH; SHNEIDERMAN, 1997). Com isto, as tarefas do usuário, como fazer comparações e examinar diferentes pontos de vista, são agilizadas.

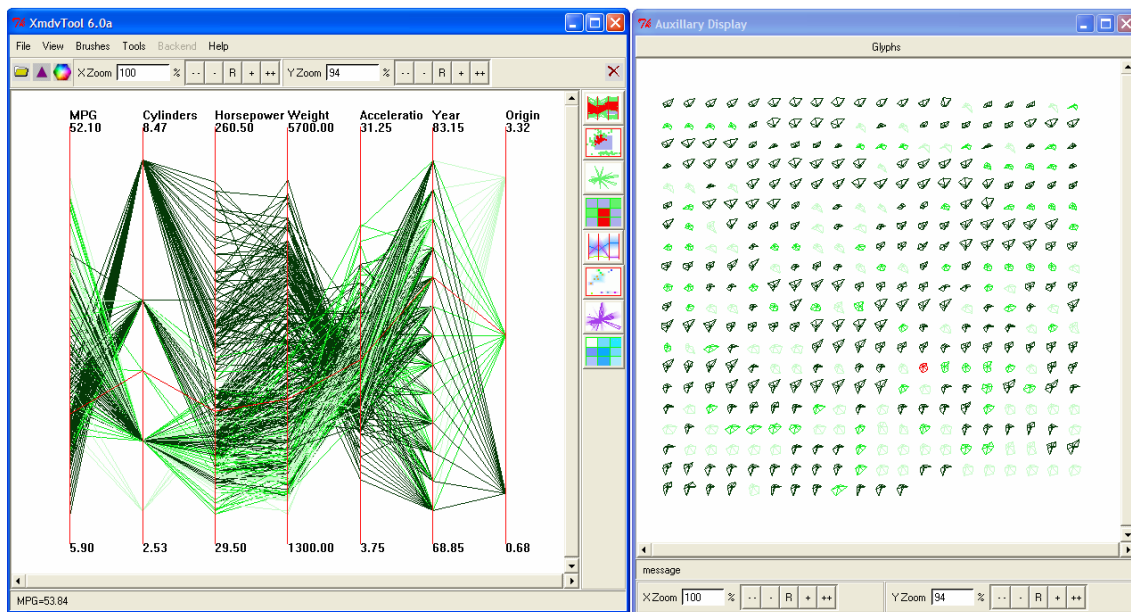


Figura 3.2: Visualizações coordenadas com valores do atributo *Origin* mapeados para cor

O *scrolling* sincronizado pode ser usado para navegar através do mesmo conjunto de dados em todas as janelas ou através de múltiplos conjuntos correspondentes (NORTH; SHNEIDERMAN, 1997). No primeiro caso, por exemplo, quando editando um documento HTML (representação de texto) em uma janela, é útil simultaneamente visualizar a saída desse documento (representação do layout) como uma página web, em outra janela. No caso de conjuntos de dados diferentes, o Microsoft Word fornece *scrolling* sincronizado entre as janelas de documento, notas de rodapé e anotações. O deslocamento através do documento provoca o deslocamento das listas de notas de rodapé e anotações, para mostrar os itens que são referenciados dentro da atual parte visível do documento.

Os exemplos de *scrolling* sincronizado citados acima são uni-dimensionais. No entanto, este tipo de coordenação pode também ser aplicado para representações visuais de dimensionalidade maior. O sistema DEVise (LIVNY et al., 1997), por exemplo, implementa *panning* sincronizado em 2D, permitindo que o usuário conecte o eixo x ou y de múltiplos gráficos, para que estes mostrem sempre os mesmos domínios e intervalos.

3.1.6 Zoom Sincronizado

O zoom sincronizado permite que duas ou mais visualizações apresentem o mesmo nível de detalhamento. Se o usuário executa uma operação de zoom em uma visualização, as visualizações coordenadas a esta serão automaticamente atualizadas para refletir a mesma operação.

O sistema DEVise (LIVNY et al., 1997), por exemplo, permite ao usuário sincronizadamente realizar zoom em múltiplos gráficos 2D, que possuam eixos x ou y em comum. Da mesma forma, Chi et al. (1997) arranhou várias visualizações 3D pequenas em um grid e sincronizou suas operações de zoom (ver figura 3.3). O zoom sincronizado pode ser aplicado com visualizações que representam o mesmo conjunto de dados ou conjuntos diferentes (como DEVise).

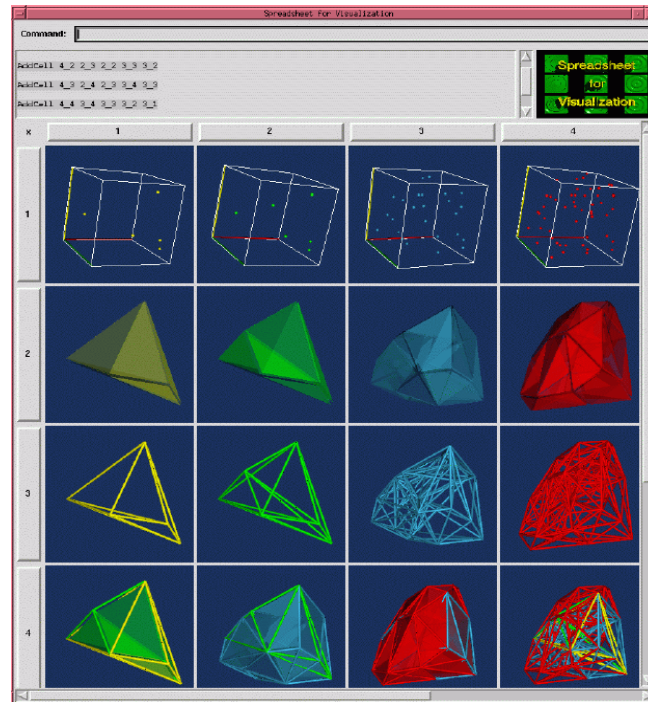


Figura 3.3: Múltiplas visualizações 3D em grid (CHI et al., 1997)

3.1.7 Rotação Sincronizada

Esse tipo de coordenação consiste apenas em sincronizar rotações executadas no espaço visual de diferentes visualizações, de forma que a orientação rotacional seja a mesma em todas elas. Esta técnica é utilizada apenas em representações 3D.

O sistema de visualizações em grid proposto por Chi et al. (1997) utiliza rotações sincronizadas para permitir a visualização de pontos oclusos dos objetos em cada célula do grid. Na figura 3.3, é possível observar que todas as visualizações 3D do grid representado possuem a mesma orientação rotacional.

3.1.8 Overview e Detalhe

Ao contrário de todas as coordenações descritas acima, o *overview* e detalhe não corresponde ao mesmo tipo de interação nas visualizações interligadas. Este tipo de coordenação envolve, geralmente, apenas duas visualizações e estas são conectadas por interações diferentes.

A idéia é fornecer uma visualização de *overview* para orientação e uma visualização de detalhe para tarefas adicionais. Selecionando um item na visão *overview* faz com que a visão de detalhe seja automaticamente atualizada para exibir este item (e vice-versa). Note que a ação de seleção na vista *overview* corresponde a uma ação de navegação (*panning* ou *zoom*) na vista de detalhe, pois o conjunto inteiro de dados não pode ser exibido simultaneamente nesta última vista. O *overview* fornece uma visão do conjunto de dados inteiro e, por isto, os itens são exibidos de forma bastante reduzida. O detalhe, ao contrário, fornece informações detalhadas de uma pequena parte do conjunto, exibindo-a em proporções bem maiores que o *overview*.

A figura 3.4 mostra a coordenação *overview* e detalhe na representação de árvores. A visualização da esquerda mostra a representação reduzida da árvore (*overview*) e a

visualização da direita mostra em detalhe apenas a sub-árvore selecionada no *overview*. Nessa figura, a visualização de detalhe é apenas uma versão ampliada do *overview*. No entanto, a aparência, ou até mesmo o conteúdo, pode mudar entre as visualizações de *overview* e de detalhe (CARD, 1999). Por exemplo, no sistema SeeSoft (BALL; EICK, 1996), o texto do código de um programa é mostrado com a largura de um único pixel na visão de *overview*, já na visão de detalhe, ele é mostrado com letras. A grande maioria dos sistemas que implementam a coordenação *overview* e detalhe representam o mesmo conjunto de dados em ambas as visualizações.

Ter uma visualização de *overview* dos dados apresenta muitas vantagens. O *overview* reduz o tempo de busca, permite a detecção de modelos gerais e ajuda o usuário a escolher o próximo movimento (NORTH; SHNEIDERMAN, 1997). Sua coordenação com a visualização de detalhe permite ainda um acesso muito mais rápido a informações detalhadas.

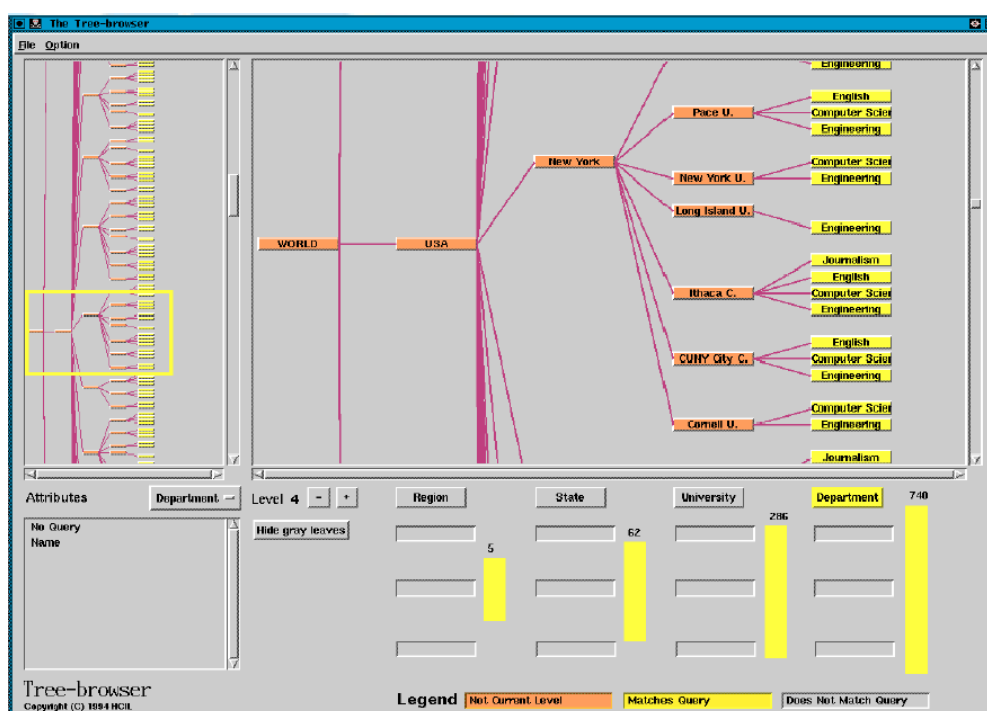


Figura 3.4: Coordenação de *overview* e detalhe no sistema PDQ Tree-browser (KUMAR; PLAISANT; SHNEIDERMAN, 1997)

3.1.9 Drill Down

A coordenação *drill down* permite ao usuário navegar através de sucessivas camadas de uma base de dados hierárquica, com visualizações separadas para cada camada. Selecionando um item pai em uma visualização, os itens filhos deste são carregados em uma outra visualização, como ocorre no Windows Explorer. Este tipo de coordenação não costuma ser implementado de forma bidirecional, ou seja, a coordenação ocorre somente em um sentido (da visualização que exibe itens pais para aquela que exibe itens filhos).

Esta coordenação torna possível a exploração de dados hierárquicos com um grande número de camadas e ainda de conjuntos de dados com um grande número de itens, que suportam agregações. Agregações são mostradas em uma visualização e o conteúdo de

uma agregação selecionada é exibido em outra visualização (FREDRIKSON et al., 1999). Por exemplo, a figura 3.5 mostra agregações de itens pelo número de automóveis envolvidos em incidentes, no gráfico superior da figura, e os incidentes com 2 carros envolvidos (agregação selecionada) são exibidos no gráfico inferior da figura. Uma forma diferente de implementação da técnica *drill down* é utilizada no sistema Visage (ROTH et al., 1996). Neste último, o usuário precisa arrastar uma agregação de itens de uma visualização e soltá-la em outra para ver seu conteúdo detalhado.

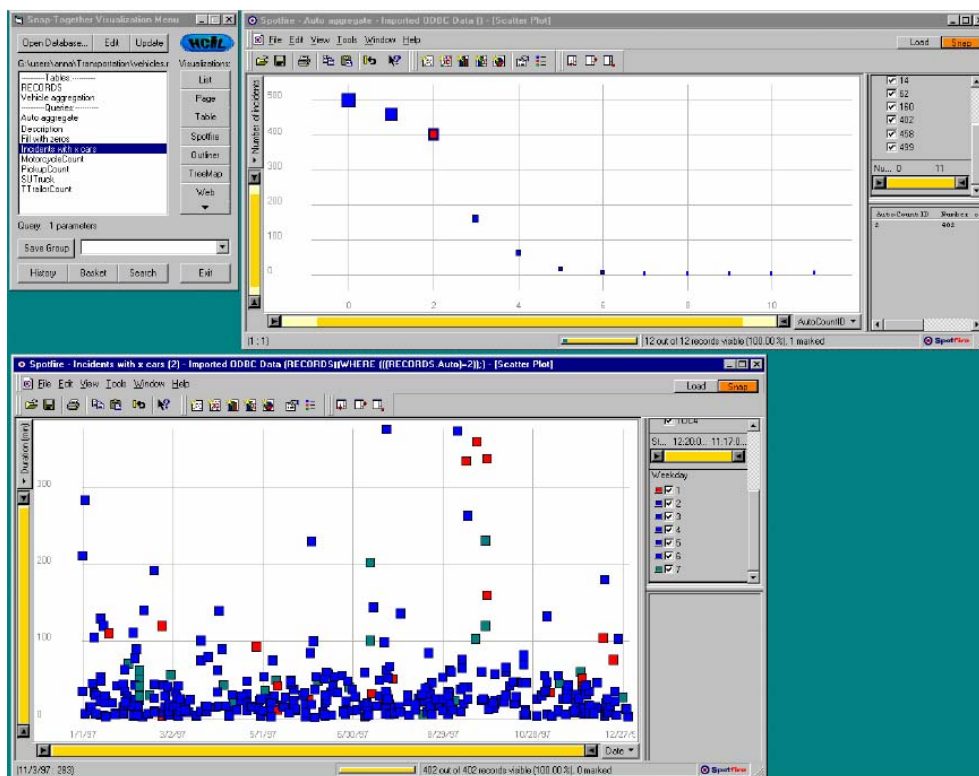


Figura 3.5: Usando a coordenação *drill down* para visualizar o conteúdo de agregações no Snap-Together (FREDRIKSON et al., 1999)

3.2 Sistemas de Múltiplas Visualizações Coordenadas

Sistemas de visualizações coordenadas podem ser classificados por seus níveis de flexibilidade em manipulações de dados, visualizações e coordenações (NORTH; SHNEIDERMAN, 2000). Sistemas flexíveis em dados (nível 1) permitem que os usuários carreguem conjuntos de dados diferentes nas visualizações. Sistemas flexíveis em visualizações (nível 2) permitem a escolha de diferentes conjuntos de visualizações, conforme mais apropriado para os dados. Já os sistemas flexíveis em coordenações permitem que usuários escolham diferentes tipos de coordenações entre pares de visualizações.

A maioria dos sistemas de visualizações coordenadas é de nível 1, ou seja, flexíveis para dados, mas não para visualizações ou coordenações. Usuários podem carregar seus próprios dados, mas estes são sempre apresentados com a mesma interface.

Sistemas de nível 2 incluem flexibilidade na escolha de visualizações. Ferramentas como EDV (EICK; WILLS, 1995), Spotfire (AHLBERG; WISTRAND, 1995) e Visage (ROTH et al., 1996) podem representar conjuntos de dados através de muitas técnicas

de visualização diferentes, escolhidas pelo usuário. Todas as visualizações são coordenadas através da técnica *brushing-and-linking*, o que permite o estabelecimento de relações entre itens de dados em diferentes visualizações.

Finalmente, sistemas de nível 3 incluem flexibilidade nas coordenações entre visualizações. Alguns deles fornecem apenas um tipo de coordenação, mas deixam o usuário escolher quais visualizações coordenar. Como em sistemas desse nível as coordenações são dinâmicas, é importante que os mesmos disponham de algum método para representar visualmente as coordenações ativas entre pares de visualizações. O objetivo disto é evitar a desorientação do usuário, que pode esquecer facilmente quais conexões estabeleceu. Ferramentas como Snap-Together (NORTH, 2000), GeoVISTA Studio (GAHEGAN et al., 2000), DEVise (LIVNY et al., 1997) e Improvise (WEAVER, 2004) apresentam flexibilidade nas coordenações.

Os sistemas DEVise e Improvise podem ser utilizados para a representação de dados multidimensionais, porém ambos não disponibilizam visualizações que mostram todas as dimensões destes dados em uma única representação visual. DEVise tem como única visualização um *plot 2D* e Improvise oferece também *plots 3D* e algumas tabelas. Assim, a forma utilizada para representar simultaneamente n -dimensões de dados nestas ferramentas é a criação de múltiplos *plots 2D*. Embora este tipo de representação não faça parte do foco desse trabalho, é importante mencionar que o mesmo permite a utilização de coordenações interessantes (como *panning* e *zoom* sincronizado de *plots* com eixos comuns), que são disponibilizadas por DEVise e Improvise. Um ponto negativo destas ferramentas é que elas não oferecem um recurso visual de rápido acesso para o usuário verificar as coordenações ativas.

A aplicação apresentada nesse trabalho é classificada como um sistema flexível em dados, visualizações e coordenações, pois suporta a escolha dos dados a serem representados, das visualizações que formarão um cenário, bem como também das coordenações que conectarão estas últimas.

A seguir, serão abordadas com maior detalhamento as ferramentas de visualizações coordenadas que possuem uma relação mais forte com o presente trabalho. São elas: XmdvTool, Snap-Together e GeoVista Studio.

Snap-Together e GeoVista Studio são flexíveis em dados, visualizações e coordenações, enquanto XmdvTool é flexível somente em dados e visualizações. Esta última será abordada aqui por ser específica para a visualização de dados multidimensionais e apresentar técnicas de coordenação relevantes para o trabalho.

3.2.1 XmdvTool

A aplicação XmdvTool (WARD, 1994) (RUNDENSTEINER et al., 2002) tem como foco a visualização exploratória de conjuntos de dados multidimensionais. Para representar estes dados, ela disponibiliza quatro técnicas tradicionais: Coordenadas Paralelas, Matriz de *Scatter Plots*, Glifos em Estrela e Pilha Dimensional. No entanto, como estas técnicas geram grande confusão visual e sobreposição de objetos quando representando conjuntos com grande número de itens de dados, XmdvTool fornece ainda uma versão hierárquica para cada uma das técnicas mencionadas. O usuário pode, então, escolher qual versão de uma técnica utilizar. A única diferença entre uma técnica hierárquica e sua versão tradicional está no fato de que a primeira possibilita ao usuário filtrar hierarquicamente objetos de dados separados em *clusters*, por meio da interação

com uma árvore de mínimos e máximos, onde intervalos de consultas são especificados (RUNDENSTEINER et al., 2002).

As visualizações geradas por essa ferramenta são todas interligadas por um conjunto fixo e pré-determinado de coordenações e apenas duas visualizações podem ser exibidas simultaneamente (uma visualização principal e outra “auxiliar”). Além disso, todas as visualizações representam o mesmo conjunto de dados. As características mais interessantes dessa ferramenta, no entanto, são suas possibilidades de coordenação, que abrangem *brushing-and-linking* n-dimensional, reordenação e filtragem sincronizada de dimensões, filtragem hierárquica de objetos de dados e mapeamento de dados para cor (ver figura 3.2 para um exemplo).

Brushing-and-linking n-dimensional permite ao usuário selecionar, através de manipulação direta, regiões diferentes para cada dimensão do conjunto de dados, as quais serão consideradas também em outras visualizações devido à coordenação. Logo, apenas os objetos de dados contidos inteiramente dentro da região de *brushing* n-dimensional serão destacados em todas as visualizações. Na figura 3.6, as imagens (c) e (d) mostram em roxo as regiões de *brushing* n-dimensional definidas por um usuário interagindo diretamente nestas representações e as imagens (a) e (b) mostram destacados (em vermelho) os dados contidos inteiramente dentro da região de *brushing* definida. As técnicas de visualização da figura estão interligadas, devido a isto as regiões de *brushing* de ambas são análogas.

A reordenação e filtragem sincronizada de dimensões são realizadas pelo usuário modificando componentes gráficos em uma janela independente das visualizações. A filtragem faz com que um eixo vertical seja retirado da visualização Coordenadas Paralelas, uma aresta (ou raio) da visualização Glifos em Estrela e uma linha e uma coluna da Matriz de *Scatter Plots*. A reordenação sincronizada, diferentemente, apenas muda a posição destes elementos nas visualizações citadas, conforme determinado pelo usuário.

Embora o sistema XmdvTool apresente técnicas de coordenação interessantes e variadas para visualizações multidimensionais, ele não é flexível em coordenações e não permite a criação de múltiplas visualizações (apenas duas), restringindo a liberdade de usuários no processo de exploração de dados. Além disso, ele emprega apenas uma possibilidade de mapeamento de dados para propriedades visuais, quando muitas outras poderiam ser utilizadas.

3.2.2 Snap-Together

Snap-Together (NORTH, 2000) é uma ferramenta de propósito geral, que suporta a visualização exploratória de conjuntos de dados de diversos tipos. Usuários podem criar cenários com múltiplas visualizações e coordená-las da forma que desejarem. Uma coordenação pode corresponder a ações diferentes em cada visualização (por exemplo, uma seleção de itens correspondendo a uma operação de zoom), mas o conjunto de ações suportadas por cada representação visual é limitado à seleção de itens e navegação em janela (*scroll*, *pan*, *zoom*, *load*, etc.). Além disso, muitas combinações de ações não são possíveis. As principais coordenações suportadas por Snap-Together são: *brushing-and-linking*, *overview* e detalhe, *drill down* e *scrolling* sincronizado (NORTH, SHNEIDERMAN, 2000).

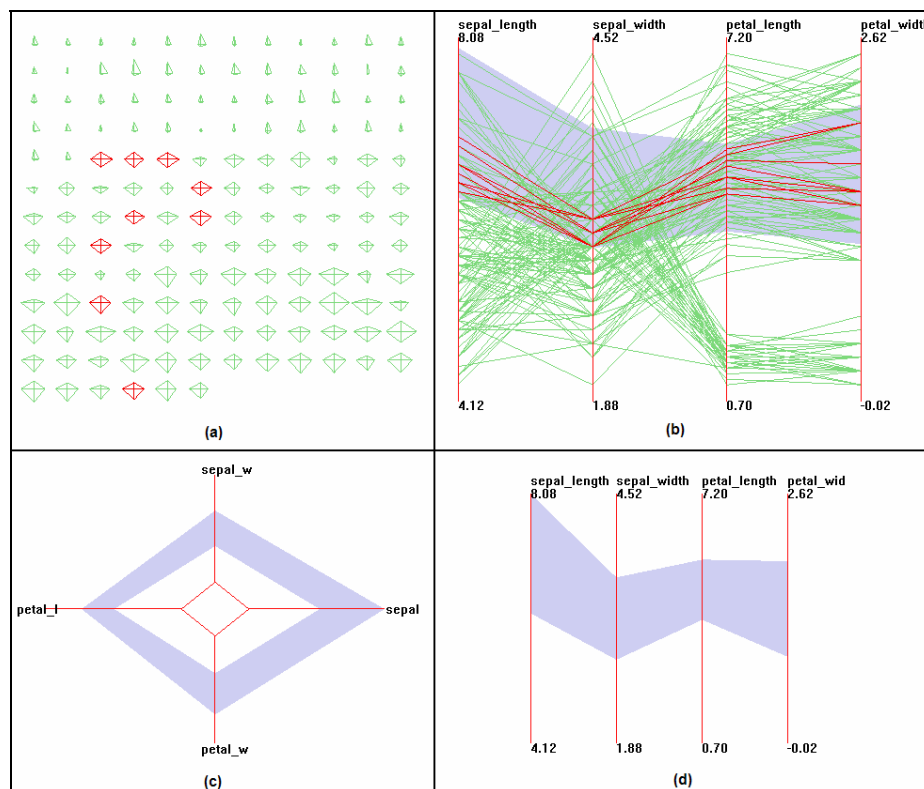


Figura 3.6: *Brushing-and-linking* n-dimensional no sistema XmdvTool. Os dados selecionados são exibidos nas imagens (a) e (b) em vermelho, e a região de *brushing* que indica os dados de interesse é mostrada nas imagens (c) e (d), no espaço de um glifo em estrela e no espaço de coordenadas paralelas, respectivamente.

Essa ferramenta permite também que cada visualização de um cenário represente graficamente dados de tabelas diferentes, mas relacionadas. Para isto, Snap-Together utiliza um modelo conceitual baseado no modelo relacional. Usuários podem, interativamente, estabelecer relacionamentos entre tabelas de dados relacionadas, conectando um atributo de uma tabela com um atributo de outra. Este relacionamento é utilizado pela aplicação para mapear dados de uma visualização para a outra.

Apesar de Snap-Together prover um conjunto amplo de técnicas de visualização, apenas Coordenadas Paralelas e a visualização Tabela representam técnicas multidimensionais. Para estas técnicas, as ações possíveis são ainda mais limitadas: *selection*, *highlight* e *load*, para Coordenadas Paralelas, e *selection*, *load* e *scroll*, para a visualização Tabela. A ação *highlight* (destaque) provoca o mesmo efeito visual que a ação *selection*, mas difere desta última porque é executada quando o usuário simplesmente passa o mouse sobre um objeto de dado, destacando-o. A ação *load*, que permite apenas o recebimento de dados, é utilizada para isolar itens de interesse, exibindo apenas estes últimos na visualização. Finalmente, a ação *scroll* faz com que a barra de *scroll* da visualização Tabela seja deslocada para exibir um determinado item de dado. Esta última ação também apenas recebe dados. A figura 3.7 mostra um cenário construído em Snap-Together, para a exploração de duas tabelas relacionadas de dados multidimensionais. A primeira tabela (com dados sobre o censo em estados dos EUA) foi relacionada à segunda (com dados sobre o censo em cidades dos EUA) através do atributo “nome do estado”, contido em ambas. Este relacionamento é que determina o mapeamento de informações da visualização Coordenadas Paralelas, que representa a primeira tabela, para informações nas visualizações *Scatter Plot* e Tabela, que

representam a segunda tabela de dados. Isto pode ser comprovado observando a figura 3.7: a ação *selection* da visualização Coordenadas Paralelas está coordenada com a ação *load* da visualização Tabela, então a seleção de um item na primeira representação, no caso o estado *California*, causa a exibição de informações acerca de todas as cidades deste estado, apenas, na segunda visualização.

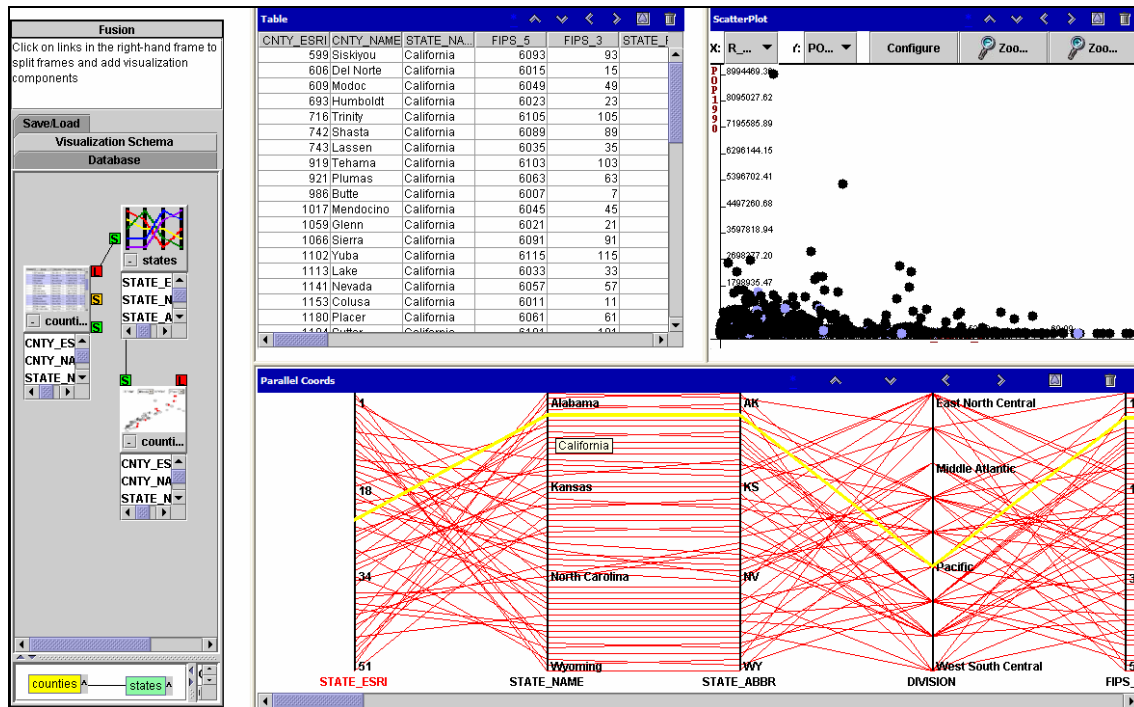


Figura 3.7: Cenário construído em Snap-Together para a exploração de duas tabelas relacionadas de dados multidimensionais

Como pode ser observado também na figura 3.7 (*frame* da esquerda), o sistema Snap-Together mantém visível em sua interface a representação visual de um diagrama de coordenações. Neste, cada visualização de um cenário e suas possíveis ações (pequenos retângulos coloridos) são representadas. As coordenações estabelecidas pelo usuário em cada momento também podem ser identificadas, pois são representadas pelas linhas que conectam os símbolos de ações. Além de orientar o usuário quanto às coordenações atuais, o diagrama de coordenações em questão também é manipulado interativamente pelo usuário para criar e remover coordenações.

Enfim, as características mais interessantes de Snap-Together são a possibilidade de representar mais de uma tabela de dados em um mesmo cenário de exploração e sua aplicação em diversos domínios de dados. No entanto, estas duas características acarretam limitações quanto às possibilidades de coordenação. A ferramenta oferece um conjunto pouco variado de ações interativas (apenas seleção e navegação em janelas), sendo suas combinações ainda mais limitadas. No caso específico da exploração de dados multidimensionais, além da disponibilidade de poucas visualizações adequadas, o usuário fica limitado, basicamente, à seleção coordenada de objetos de dados. Coordenações como filtragem sincronizada e mapeamento de dados para propriedades visuais não são suportadas.

3.2.3 GeoVista Studio

GeoVista Studio (GAHEGAN et al., 2000) consiste em uma ferramenta de programação visual orientada a componentes, que tem como foco principal a visualização e análise de dados geo-científicos. Ela é baseada em um modelo de fluxo de dados, em que cenários de exploração personalizados podem ser dinamicamente construídos conectando entradas e saídas de componentes independentes – JavaBeans. Cada componente possui uma lista pré-definida de métodos que permitem a entrada ou saída de determinadas informações, e cabe ao usuário definir quais destes métodos serão ativados para habilitar a troca de informações entre *beans*.

O conjunto de componentes que acompanham a ferramenta é bastante variado e permite a execução das mais diversas funcionalidades, desde a leitura de arquivos e entradas fornecidas pelo usuário até a representação visual dos dados e análises. Em especial, para a visualização de dados multidimensionais, GeoVista Studio oferece quatro técnicas tradicionais: Coordenadas Paralelas, Matriz de *Scatter Plots*, *Space Fill* (orientada a pixel e independente de consulta) e Tabela de Dados. A figura 3.8 mostra um cenário de exploração de dados multidimensionais formado por estas quatro técnicas de visualização. A janela acima à direita, na figura, mostra a representação dos componentes utilizados neste cenário e do fluxo de dados entre eles. É interagindo diretamente com esta representação que o usuário cria um cenário.

As múltiplas visualizações criadas para um cenário de exploração podem ser coordenadas dinamicamente. Para isto, os componentes que as representam devem ser conectados a um componente especial, denominado *Coordinator*, que é responsável por gerenciar coordenações entre componentes de visualização (ver figura 3.8 acima à direita). O componente *Coordinator* cria uma janela (como a mostrada na figura 3.8, abaixo, à esquerda) que mostra todas as possibilidades de coordenação para cada visualização. Por *default*, todas as coordenações possíveis entre pares de visualizações iniciam ativadas, mas o usuário pode desabilitar/habilitar as mesmas através dos botões *check box* da janela do coordenador.

Cada coordenação, em GeoVista Studio, corresponde a ações iguais em diferentes visualizações. No entanto, essa ferramenta disponibiliza poucas ações para coordenação. No caso das visualizações multidimensionais mencionadas, é possível coordenar apenas a seleção ou destaque (*highlight*) de itens e o mapeamento de dados para cor.

Enfim, embora disponha de diferentes técnicas para a visualização de dados multidimensionais, GeoVista Studio, assim como os demais sistemas descritos nessa seção, também deixa a desejar quanto às possibilidades de coordenação, que são bastante limitadas.

A limitação constatada nas técnicas de coordenação para representações de dados multidimensionais nos sistemas atuais inspirou o desenvolvimento do presente trabalho, onde foram identificados os tipos de coordenações para visualizações multidimensionais, e seu uso é demonstrado em uma aplicação protótipo.

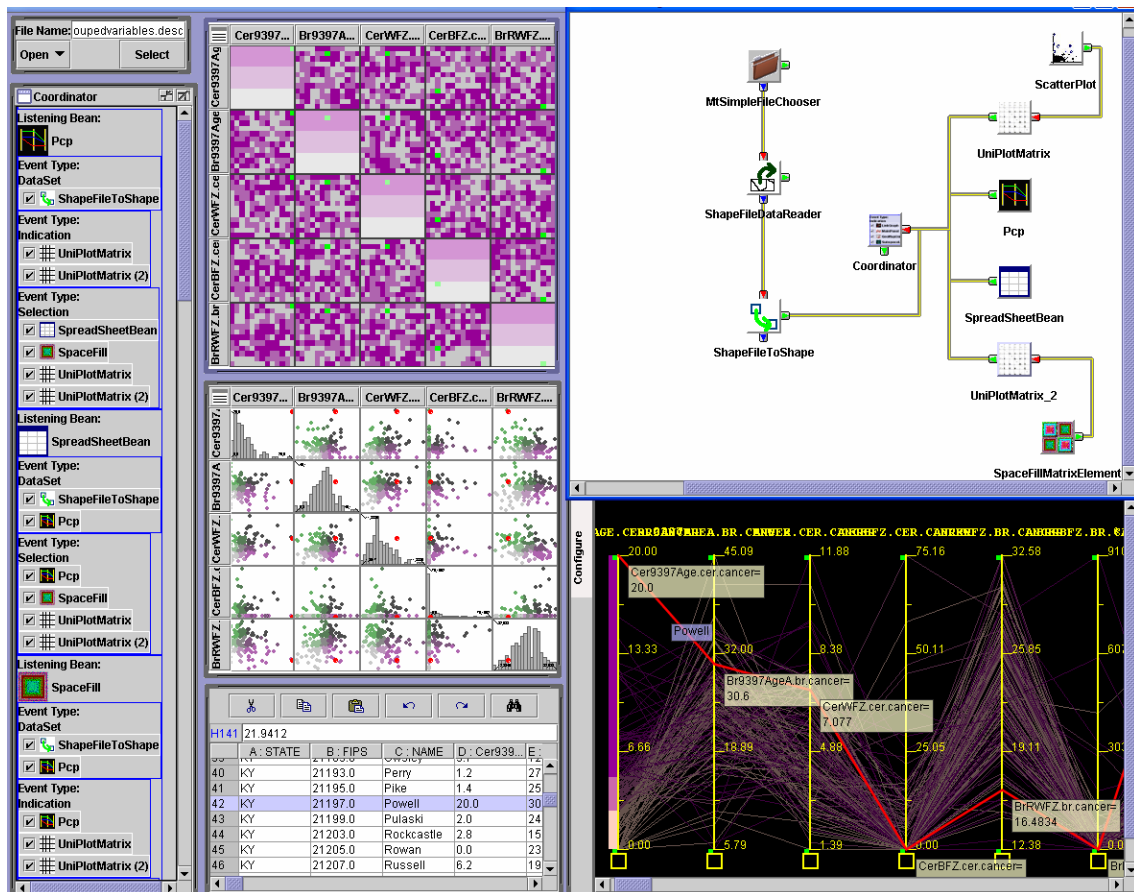


Figura 3.8: Cenário construído em GeoVista Studio com todas as suas técnicas de visualização multidimensionais

4 O TOOLKIT INFOVIS

Esse capítulo descreve as principais características e a estrutura básica do toolkit de visualização de informações InfoVis (FEKETE, 2003; FEKETE, 2004), sobre o qual foi construída a aplicação de visualizações coordenadas descrita nesse trabalho. InfoVis foi escolhido por ser específico para visualização de informações, por apresentar uma arquitetura simples de usar e extensível e, principalmente, porque implementa várias técnicas de visualização e fornece vários componentes para suportar manipulações interativas nestas últimas. O mesmo pode tanto ser usado na sua forma original, como modificado e estendido livremente, já que seu código fonte encontra-se disponível.

O toolkit InfoVis foi projetado para suportar a criação, extensão e integração de componentes avançados de visualização de informações 2D dentro de aplicações Java interativas. Segundo Fekete, suas principais características são:

- Estruturas de dados genéricas;
- Algoritmos específicos para visualizar estas estruturas de dados;
- Mecanismos e componentes para executar manipulação direta sobre visualizações;
- Mecanismos e componentes para selecionar, filtrar e executar tarefas de visualização de informações genéricas;
- Componentes para nomeação dinâmica (*dynamic labeling*) e deformação espacial.

Algumas das possibilidades de interação oferecidas pelo InfoVis são padrões na ferramenta, ou seja, são suportadas pela maioria das técnicas de visualização implementadas. Estas interações são:

- Filtragem;
- Mapeamento de valores de atributos de dados para atributos visuais (cor, tamanho e transparência);
- Mapeamento de valores de atributos de dados para a ordenação em profundidade das figuras;
- Nomeação dinâmica (*dynamic labeling*);
- Deformação espacial.

Atualmente, o toolkit InfoVis suporta três diferentes estruturas de dados – tabelas, árvores e grafos – e oito tipos de visualizações: (1) séries de tempo, (2) coordenadas paralelas e (3) *scatter plots* 2D, para tabelas; (4) diagramas de nodos e arestas, (5)

árvores Icicle e (6) treemaps, para árvores; (7) diagramas de nodos e arestas e (8) matrizes de adjacência, para grafos.

4.1 Framework Geral

O toolkit InfoVis é uma biblioteca Java e uma arquitetura de software organizada em cinco módulos principais (FEKETE, 2004): tabelas, colunas, visualizações, componentes e entrada/saída. A figura 4.1 detalha estes módulos, descritos abaixo.

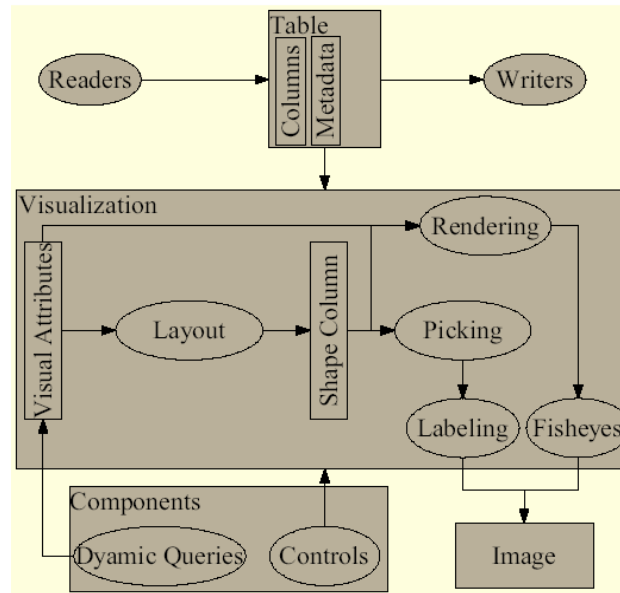


Figura 4.1: Estrutura interna do toolkit InfoVis. Retângulos representam estruturas de dados e elipses representam funções (FEKETE, 2003).

4.1.1 Entrada e Saída

O toolkit InfoVis suporta um conjunto limitado, mas extensível de formatos de arquivos. Atualmente, existem *readers* e *writers* para os seguintes formatos: CSV, XML, TQD, Newick, TM3, DOT, XML TreeML DTD e XML GraphML DTD.

Os *readers* podem ser criados instanciando diretamente, a partir do programa, a classe que trata de um determinado formato de arquivo ou podem ser criados através de objetos *factory*. Estes últimos são usados para criar objetos indiretamente, de acordo com alguns parâmetros especificados. Eles analisam o nome do arquivo a ser carregado e seu conteúdo para criar o *reader* mais satisfatório para ele (FEKETE, 2003).

4.1.2 Tabelas e Colunas

InfoVis fornece uma estrutura de dados unificada que é baseada em tabelas. Segundo Fekete (2003), “as razões disto recaem sobre os ganhos com eficiência que a estrutura permite e a facilidade com que qualquer estrutura de dados pode ser implementada sobre tabelas”.

Uma tabela é uma lista de colunas nomeadas, metadados e dados do usuário. Uma coluna gerencia linhas com elementos de tipos homogêneos (inteiro, ponto flutuante, string, etc.). Os elementos são indexados, de forma que as colunas são implementadas

com arrays. Algumas linhas podem ter seus valores não definidos. Este mecanismo é importante, porque conjuntos de dados do mundo real frequentemente possuem valores faltando, e, além disso, facilita a representação de estruturas de dados gerais.

Um conjunto de dados é armazenado como uma tabela, onde cada linha representa um registro (item de dado) e cada coluna representa um atributo. Isto é natural para conjuntos de dados tabulares, mas árvores e grafos também são representados com essa estrutura de dados. Árvores e grafos são implementados como *wrappers* no topo de tabelas com informação topológica representada por colunas internas (FEKETE, 2003). Estas últimas contêm informações internas e não são salvas para arquivos e nem disponíveis para consultas dinâmicas.

4.1.3 Visualizações

Visualizações transformam um conjunto de atributos armazenados em colunas de tabela para representações visuais. Além disso, elas executam filtragem, zoom, navegação e *picking*. Cada visualização dispõe de uma lista de atributos visuais que podem ser associados com colunas de dados. Esses atributos são utilizados para mudar a cor ou propriedades geométricas dos itens exibidos (FEKETE, 2002). Para seu controle, a visualização mantém uma coluna interna de objetos gráficos (FEKETE, 2003). Alguns atributos visuais são genéricos e padrões para todas as visualizações, como cor (*color*), tamanho (*size*), transparência (*alpha*) e label. Outros, são específicos de uma visualização, como eixo x e eixo y no *scatter plot* 2D, ou de algumas, como os atributos visuais de aresta e vértice nas visualizações de grafos.

As visualizações são redesenhadas quando pelo menos uma coluna a que elas se referem é modificada. Além disso, quando um atributo visual usado para computar as figuras é modificado, estas últimas são invalidadas e recomputadas para o próximo *rendering* ou *picking*. A seleção, filtragem e ordenação, que não são atributos visuais, comportam-se exatamente do mesmo modo. Por default, elas são associadas com colunas internas. As permutações são usadas para ordenação e filtragem. Elas especificam uma ordem para as linhas da tabela e são capazes de esconder (filtrar) uma linha da visualização apenas não a especificando na ordenação. Permutações também mantêm o mapeamento de um número de linha para seu índice e a contagem das linhas visíveis.

Visualizações usam vários subcomponentes para gerenciar cores, permutações, redesenho, *labeling* e deformações espaciais. O mapeamento de atributos de dados para cores é feito através de uma interface, chamada *color visualization*, que retorna uma cor a partir de uma linha da tabela.

O redesenho está dividido entre o *layout* e o *rendering*. Na maioria dos casos, o layout pode ser reusado. Seleções apenas causam redesenho sem refazer *layout*, *picking* também reusa o *layout* já computado e a filtragem, em geral, apenas muda o conjunto de itens redesenhados, não seu *layout*. Este último poderia ser recomputado cada vez que um item é filtrado, no entanto, o tempo necessário para o redesenho dos itens mudaria drasticamente, sendo talvez impossível atualizar as mudanças de um frame para o próximo (FEKETE, 2003).

O *rendering* de itens de dados está relacionado com figuras, computação de cores e, opcionalmente, com lentes *fisheye*. Por default, o *rendering* itera sobre cada linha não-filtrada da tabela, na ordem de permutação, computando a cor com o componente *color*

visualization. A figura representando um item de dado é pintada, bem como sua borda – geralmente preta para itens não-selecionados e vermelha para os selecionados.

As visualizações opcionalmente suportam labels dinâmicos e deformações espaciais. O primeiro componente usa o mecanismo *picking* para computar os labels sob o ponteiro do mouse. Dois métodos são fornecidos pelo toolkit para *picking*: o primeiro retorna o item desenhado mais à frente em uma posição espacial e o segundo retorna uma lista de itens que interseccionam um retângulo. O segundo componente opcional, deformação espacial, é aplicado pelo *rendering* depois que as figuras dos itens de dados foram computadas pelo *layout*. O toolkit suporta, atualmente, um subconjunto das deformações de Carpendale (CARPENDALE; MONTAGNESE, 2001). Um objeto *Fisheye* é usado para transformar uma figura Java na sua deformação através de lentes. A implementação verifica se uma figura específica intersecciona a lente e, se isto não ocorre, ela retorna sem alterações. Mas, se a interseção acontece, itera-se sobre a borda da figura, aplicando-se a deformação de lentes para cada vértice de controle (FEKETE, 2003).

4.1.4 Componentes

O toolkit InfoVis fornece vários componentes para suportar manipulações interativas. Por default, a cada visualização está associado um painel de controle padrão organizado em um grupo de abas. Programadores podem usar seus próprios painéis de controle, substituindo ou acrescentando componentes e mecanismos de interação.

O painel de controle padrão apresenta os seguintes componentes: consultas dinâmicas e filtros, seleção, ordenação, *labels* dinâmicos, lente *fisheye* e manipulação de atributos visuais. Todos esses componentes são descritos e ilustrados na seção 4.4.

4.2 Exibindo Visualizações

O procedimento seguido para exibir a representação visual de uma técnica já suportada pelo toolkit InfoVis é bastante simples. A figura 4.2 demonstra, de forma simplificada, como uma visualização da técnica Coordenadas Paralelas pode ser gerada para um arquivo de dados, cujo nome é especificado no primeiro argumento do programa principal.

No programa da figura 4.2, primeiramente, uma tabela é criada e identificada com o nome do arquivo de dados especificado (linhas 4-5). O carregamento do arquivo é feito em dois passos: encontrando um *reader* e carregando o arquivo a partir deste último. O *reader* é criado através de um objeto *factory* (linhas 6-7). Este último analisa o nome do arquivo e seu conteúdo para criar o *reader* mais satisfatório para ele. Se nenhum *reader* é retornado ou ele não pôde carregar o arquivo, o programa é encerrado com uma mensagem de erro (linhas 8-11). Caso contrário, um objeto visualização é criado (linhas 12-13) e inserido em um *JFrame Java/Swing* padrão, dentro de um painel de visualização (linhas 14-15).

Como descrito acima, a visualização gerada não é interativa, porque nenhum painel de controle foi associado a ela. A visualização terá um painel de controle se o código para a criação do painel de visualização (retângulo azul na figura) for substituído pelo código contido no retângulo vermelho.


```

1. public class Example {
2.     public static void main(String args[]) {
3.         String fileName = args[0];
4.
5.         DefaultTable t = new DefaultTable();
6.         t.setName(fileName);
7.
8.         AbstractReader reader =
9.             TableReaderFactory.createReader(fileName, t);
10.
11.         if (reader == null || !reader.load()) {
12.             System.err.println("nao deu pra carregar " + fileName);
13.             return;
14.         }
15.         ParallelCoordinatesVisualization visualization =
16.             new ParallelCoordinatesVisualization(t);
17.
18.         VisualizationPanel panel =
19.             new VisualizationPanel(visualization);
20.
21.         JFrame frame = new JFrame(fileName);
22.         frame.getContentPane().add(panel);
23.         frame.setVisible(true);
24.         frame.pack();
25.     }
26. }

```

Figura 4.2: Programa para a visualização de Coordenadas Paralelas. O código do retângulo azul gera uma visualização não-interativa, mas quando substituído pelo código do retângulo vermelho a visualização recebe um painel de controle padrão.

Como pode ser observado na mesma figura, a criação do painel de controle associado à visualização também é realizada através de um objeto *factory*. A classe painel de controle a ser instanciada dependerá do tipo de visualização recebida como parâmetro pelo objeto. Como foi mencionado na seção anterior, visualizações diferentes podem ter painéis de controle com componentes ou mecanismos de interação distintos.

4.3 Extensão do Toolkit: Criando Novas Visualizações

Antes de iniciar a implementação de novas técnicas com o toolkit InfoVis, é necessário conhecer a organização e hierarquia das classes de visualização e dos seus métodos mais relevantes para essa tarefa.

Todas as visualizações do toolkit implementam a interface *Visualization*. Uma visualização está sempre relacionada a uma única tabela e deveria ser instalada em um *JComponent* Java para ser exibida em uma janela (FEKETE, 2002). O toolkit InfoVis fornece a classe *VisualizationPanel* para este propósito.

A principal implementação da interface *Visualization* é a classe *DefaultVisualization*. Todas as visualizações específicas derivam dela. A figura 4.3 exhibe a hierarquia das classes de visualização do toolkit.

A implementação de uma nova técnica no toolkit InfoVis envolve, antes de tudo, a criação de uma nova classe de visualização específica, com o nome da técnica, e esta deve também ser derivada de *DefaultVisualization*. A classe específica contém apenas métodos que são utilizados exclusivamente pela nova técnica.

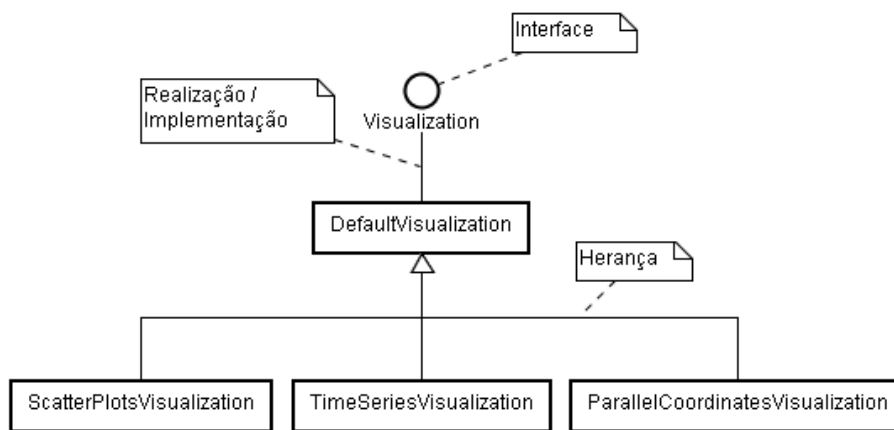


Figura 4.3: Hierarquia das classes de visualização do toolkit InfoVis

Quando uma nova visualização é criada, ela deve também ser configurada para mapear colunas da tabela para colunas visuais. Visualizações usam vários tipos de propriedades visuais, algumas são padrões para todas elas e outras são específicas para uma técnica. A configuração de um atributo visual é realizada com a seguinte linha de código:

```
visualization.setVisualColumn(string_constante,
                             table.getColumn("nome_coluna"));
```

string_constante é uma string que contém o nome da coluna visual para a propriedade a ser mapeada e *nome_coluna* é o nome de uma coluna numérica da tabela.

Os atributos visuais padrões são os seguintes:

- *Color* – A coluna usada para especificar a cor dos itens. Sua constante string é *Visualization.VISUAL_COLOR*.
- *Size* – A coluna usada para especificar o tamanho dos itens. Sua constante string é *Visualization.VISUAL_SIZE*.
- *Alpha* – A coluna usada para especificar a transparência ou canal alpha de itens. Sua constante string é *Visualization.VISUAL_ALPHA*.
- *Label* – A coluna usada para especificar o nome ou *label* de itens. Sua constante string é *Visualization.VISUAL_LABEL*.

Outros atributos podem ser especificados utilizando o mesmo mecanismo, embora não sejam atributos “visuais”:

- *Selection* – A coluna usada para especificar quais itens estão selecionados. Ela deve ser uma coluna booleana e sua constante string é *Visualization.VISUAL_SELECTION*.
- *Filter* – A coluna usada para especificar quais itens estão filtrados. Ela deve ser uma *FilterColumn* e sua constante string é *Visualization.VISUAL_FILTER*.
- *Sort* – A coluna usada para especificar a ordem (de sobreposição) dos itens na visualização. Sua constante string é *Visualization.VISUAL_SORT*.

A computação e desenho dos itens de dados de uma visualização são implementados através da derivação de classes. Há um elemento gráfico para cada item de dado

visualizado: *Shape* é uma interface Java que descreve uma geometria geral. As visualizações específicas podem decidir que tipo de figura elas criam e em qual posição. Geralmente, é um *Rectangle2D* Java que implementa a interface *Shape*.

As figuras computadas são armazenadas em uma coluna visual chamada *Visualization.VISUAL_SHAPE*.

O método usado para computar o layout das figuras, geralmente implementado nas classes específicas de visualização, é denominado *computeShapes*. Este método é chamado apenas antes do desenho, quando um dos seguintes eventos ocorre (FEKETE, 2002):

- A visualização deve ser mostrada na tela e nenhum *layout* foi computado antes;
- Uma coluna visual envolvida na computação do *layout* foi associada com uma nova coluna da tabela;
- Uma coluna visual envolvida na computação do *layout* foi modificada.

Depois que todas as figuras foram computadas, os itens podem ser mostrados de uma única vez, na ordem definida pelo atributo *sorting*. A implementação padrão usa o método *paintItems* para essa tarefa, geralmente definido na classe *DefaultVisualization*, e cujo código é exibido na figura 4.4.

```

1. public void paintItems(Graphics2D graphics, Rectangle2D bounds) {
2.     for (RowIterator iter = iterator(); iter.hasNext();) {
3.         int row = iter.nextRow();
4.         if (isFiltered(row))
5.             continue;
6.         paintItem(graphics, row);
7.     }
8. }

```

Figura 4.4: Código padrão executado para o desenho de itens de dados

Esse método usa um *RowIterator* para iterar sobre todos os itens na ordem determinada pelo *sorting*, pula os itens filtrados através do método *isFiltered* e desenha os demais usando o método *paintItem*. O mesmo pode ser sobrescrito em uma classe derivada se, por exemplo, a visualização deve mostrar em cinza os itens filtrados.

O desenho de um item precisa de vários passos: configurar todos os atributos gráficos, transformar a figura com uma lente *fisheye*, desenhar a figura, desenhar sua linha de contorno e seus *labels*. Cada um desses passos é realizado dentro do método *paintItem*, cujo código é mostrado na figura 4.5.

```

1. public void paintItem(Graphics2D graphics, int row) {
2.     Shape s = getShapeAt(row);
3.     if (s == null)
4.         return;
5.     s = transformShape(s);
6.
7.     installAlpha(graphics, row);
8.     installColor(graphics, row, s);
9.     displayedItems++;
10.    paintShape(graphics, row, s);
11.    paintOutline(graphics, row, s);
12.    paintLabel(graphics, row, s);
13. }

```

Figura 4.5: Código padrão executado para o desenho de um item de dado

Todos os métodos mencionados podem ser sobrescritos em subclasses, se requerido. Na maioria dos casos, as novas técnicas acrescentadas ao toolkit InfoVis precisam implementar apenas o método *computeShapes* para a exibição dos itens de dados. Geralmente, a implementação padrão dos métodos de desenho (*paintItems* e *paintItem*) pode ser reusada.

4.4 Mecanismos de Interação

O painel de controle padrão do toolkit InfoVis oferece vários componentes, por meio dos quais um usuário pode interagir com visualizações. As opções de interação disponibilizadas encontram-se distribuídas dentro de quatro abas, denominadas *Filters*, *Visual*, *Excentric* e *Fisheyes*, que compõem o painel de controle.

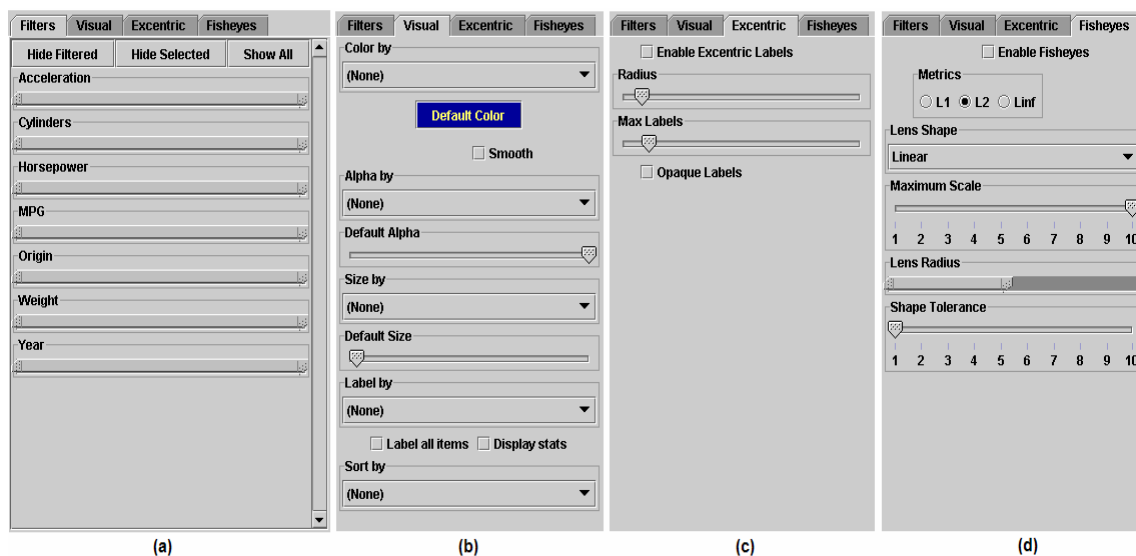


Figura 4.6: As quatro abas do painel de controle: (a) *Filters*; (b) *Visual*; (c) *Excentric*; e (d) *Fisheyes*.

A aba *Filters* contém, basicamente, barras de filtragem correspondentes aos atributos do conjunto de dados, conforme pode ser observado na figura 4.6(a). As barras de filtragem permitem que o usuário possa ocultar da representação visual itens de dados dentro de uma faixa de valores (para atributos numéricos). Quando uma barra é movida da esquerda para a direita, os itens são ocultados, progressivamente, daqueles que possuem o menor valor para o atributo relativo à barra até aqueles que apresentam o maior valor para o mesmo. Já quando o sentido de movimentação da barra é invertido, o desaparecimento de itens se dá em ordem reversa. Para atributos de dados categóricos, o processo de filtragem é realizado através da seleção de *strings* em uma lista. Apenas os itens de dados que possuem as *strings* selecionadas como valores para um atributo categórico são desenhados na representação visual.

A aba *Visual*, cujos componentes podem ser observados na figura 4.6(b), disponibiliza mecanismos de interação que alteram atributos visuais dos objetos representando itens de dados. Os mecanismos denominados *Color by*, *Alpha by* e *Size by* utilizam variações progressivas na intensidade da cor, nível de transparência e tamanho dos objetos itens de dados, respectivamente, para representar quão alto é o valor que cada objeto apresenta para um determinado atributo (escolhido pelo usuário). No recurso *Color by*, a menor intensidade de cor é atribuída ao item de dado que possui

o menor valor para o atributo sendo mapeado e a maior intensidade é aplicada ao item que apresenta o maior valor para o mesmo. Os recursos *Alpha by* e *Size by* aplicam suas variações segundo este mesmo critério. As visualizações geradas pela aplicação dos mecanismos de interação *Color by* e *Size by*, com a técnica Coordenadas Paralelas e Radviz, podem ser observadas na figura 4.7.

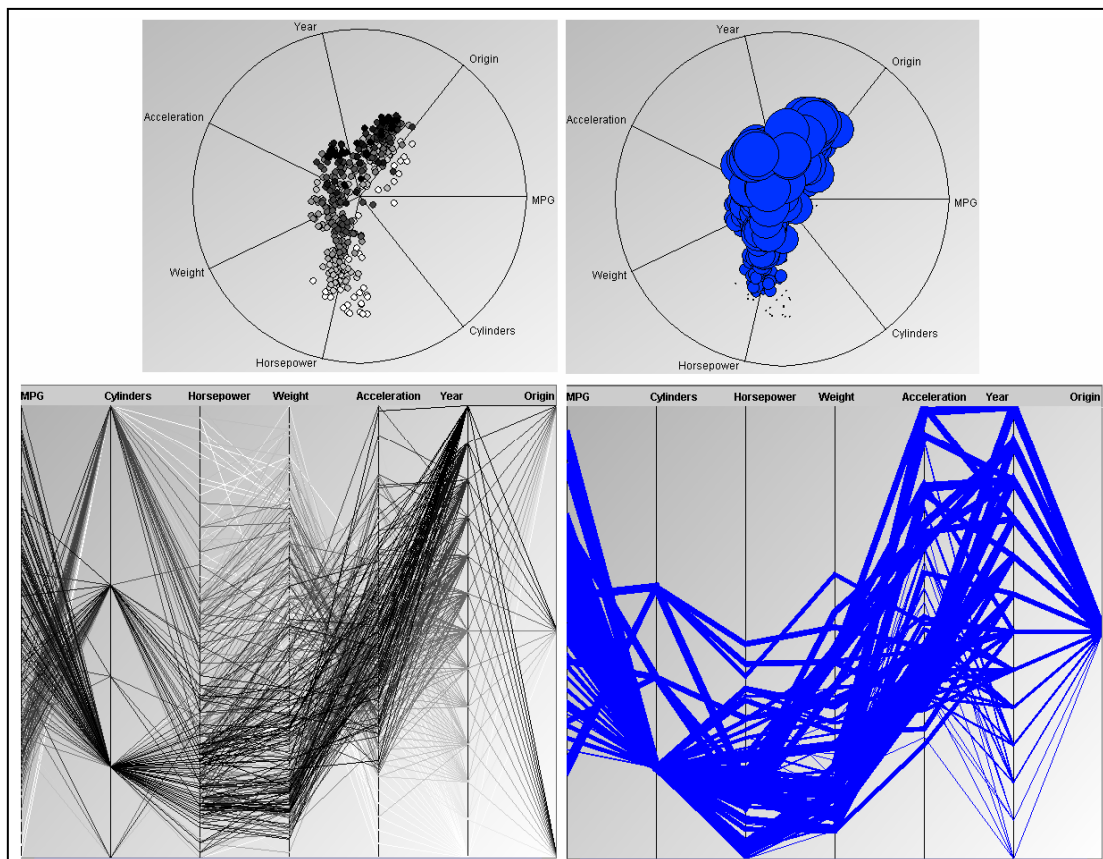


Figura 4.7: Visualizações geradas pela aplicação dos mecanismos de interação *Color by* (à esquerda) e *Size by* (à direita) sobre as técnicas Radviz (acima) e Coordenadas Paralelas (abaixo)

O mecanismo denominado *Sort by* trabalha de forma semelhante aos mencionados acima. No entanto, ele utiliza o valor que cada objeto item de dado possui para um determinado atributo (especificado pelo usuário) como critério para a ordenação dos objetos em profundidade. Ou seja, a ordem em que objetos se sobrepõem na representação visual pode ser determinada pela escala de valores de um atributo do conjunto de dados. O primeiro item a ser desenhado é aquele que possui o menor valor para o atributo determinante e o último item desenhado é aquele que apresenta o maior valor para o mesmo.

O usuário pode também especificar um atributo de dados, que determina os *labels* exibidos para cada item de dado, através do mecanismo de interação *Label by*. Quando o componente *Label all items* da aba *Visual* encontra-se habilitado, cada item aparece na visualização com um *label* associado e o conteúdo deste último é o valor que o objeto possui para o atributo determinante.

Além de todos os mecanismos de interação já citados, a aba *Visual* também permite ao usuário modificar algumas configurações padrão dos itens de dados, como o tamanho (componente *Default size*), o nível de transparência (componente *Default alpha*) e a cor

(componente *Default color*). O número de itens desenhados na visualização a cada momento pode também ser acompanhado habilitando o componente *Display stats*.

Em muitos casos, os *labels* fixos gerados pelo componente *Label all items* não podem ser visualizados adequadamente, principalmente quando o número de itens de dados é muito elevado. Diante disso, um meio alternativo para visualizar esses *labels* é oferecido pela aba *Excentric* (figura 4.6(c)), habilitando o componente *Enable excentric labels*. Este último exhibe *labels* em posições dinâmicas (sempre indicando o objeto ao qual cada um se refere) e apenas para os itens que ficam sob uma região quadrada (limitada por linhas vermelhas) que acompanha o cursor do mouse, conforme ilustrado na figura 4.8. O tamanho do quadrado, que representa a área de abrangência desse mecanismo de interação e o número máximo de *labels* exibidos podem ser configurados através dos componentes *Radius* e *Max Labels*, respectivamente.

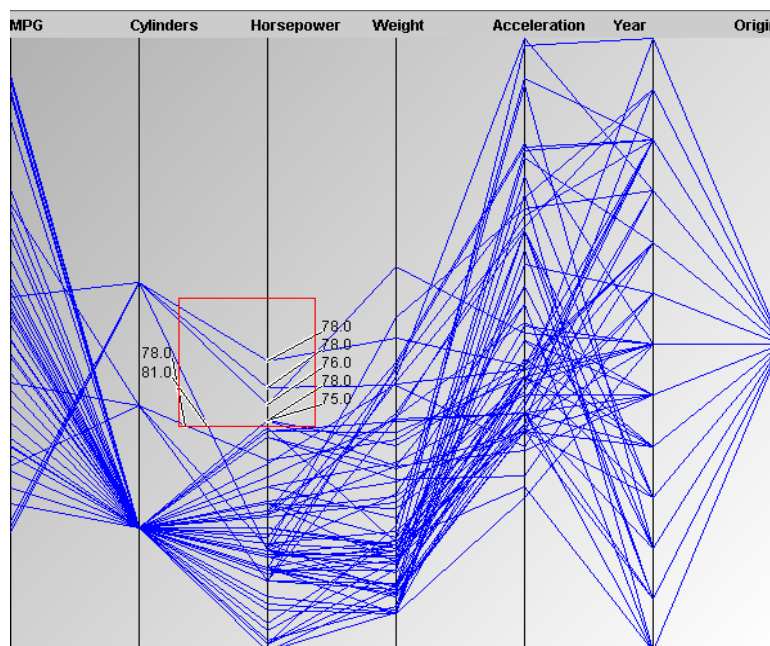


Figura 4.8: *Labels* dinâmicos na técnica Coordenadas Paralelas

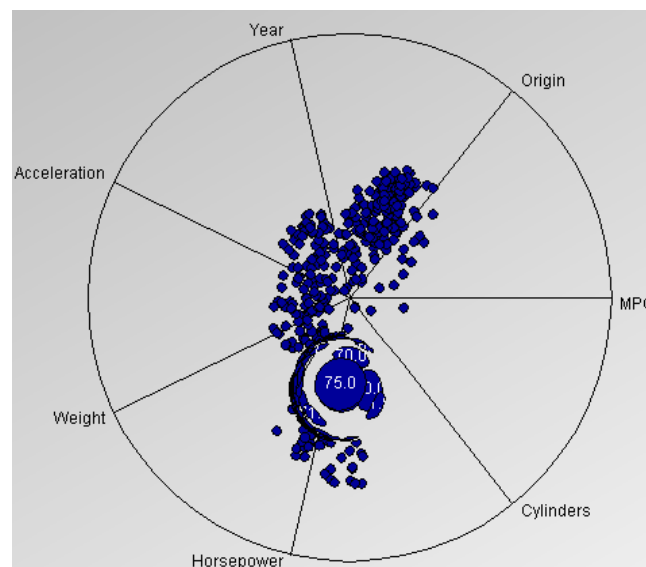


Figura 4.9: Lente *Fisheye* aplicada na técnica Radviz

Finalmente, a última aba do painel de controle, denominada *Fisheyes* (figura 4.6(d)), refere-se a um mecanismo de filtro móvel, que amplia a representação dos objetos de dados localizados sob uma “lente” que acompanha o cursor do mouse. O efeito dessa ampliação pode ser comparado com o de uma lupa, conforme pode ser observado na figura 4.9, e facilita a identificação de objetos pequenos, próximos ou levemente sobrepostos.

Acima das quatro abas do painel de controle, pertencendo também a este último, existe ainda um espaço reservado apenas para a visualização dos valores que um ou mais itens de dados, selecionado(s) na representação visual, possui (possuem) para cada um dos atributos do conjunto de dados. A exibição desses valores é feita em forma de tabela, como pode ser observado na figura 4.10, com a primeira coluna mostrando os nomes dos atributos e a segunda mostrando os valores que o item selecionado apresenta para estas dimensões. Com a finalidade de facilitar a identificação do objeto de dado selecionado na visualização, este último tem sua cor de contorno modificada para vermelho.

Attribute	Row 142
Acceleration	16.5
Cylinders	4.0
Horsepower	52.0
MPG	31.0
Origin	3.0
Weight	1649.0
Year	74.0

Figura 4.10: Visualização dos valores que o 142º item de dado do conjunto sobre carros possui para cada atributo

4.4.1 Mecanismos Adicionados

Com o objetivo de ampliar os recursos de interação oferecidos pelo toolkit InfoVis, foram adicionados novos mecanismos no painel de controle das técnicas de visualização multidimensionais. Os mesmos foram acrescentados após um estudo de avaliação preliminar (relatado no capítulo 6), que revelou a necessidade destes mecanismos para a realização de algumas tarefas de usuário, envolvidas no processo de exploração de conjuntos de dados.

Assim, no final da aba *Visual* do painel de controle, foram adicionados os componentes gráficos exibidos na figura 4.11, que compõem o novo mecanismo de interação, denominado *Attributes Manipulation*. Por meio deste mecanismo, o usuário pode remover (filtrar) atributos de dados da visualização associada ao painel de controle, bem como também alterar a ordem em que estes atributos aparecem na mesma. Para remover um atributo, o usuário deve selecionar o nome deste na lista de atributos que aparece à direita na figura 4.11 e pressionar o botão *Remove*. Ao contrário, para adicionar este atributo novamente à visualização, o procedimento é selecionar o nome deste na lista de atributos à esquerda na figura 4.11 e pressionar o botão *Add*. No entanto, se o usuário deseja apenas trocar a localização de um atributo por outro, na representação visual, ele deve selecionar seu nome na lista de atributos da direita e pressionar o botão *Up*, para trocá-lo de posição com o atributo acima na lista, ou o botão *Down*, para trocá-lo de posição com o atributo abaixo na lista.

Outro componente de interação adicionado ao painel de controle do toolkit, agora na aba *Filters*, é um botão para filtragem (um para cada atributo de dados). Ao pressionar o

mesmo, é apresentada uma janela onde um intervalo numérico (valor inicial e final) de filtragem pode ser rapidamente fornecido. A finalidade deste é a mesma das barras de filtragem (*scroll bars*), no entanto ele oferece maior agilidade para filtrar um intervalo previamente conhecido de valores.

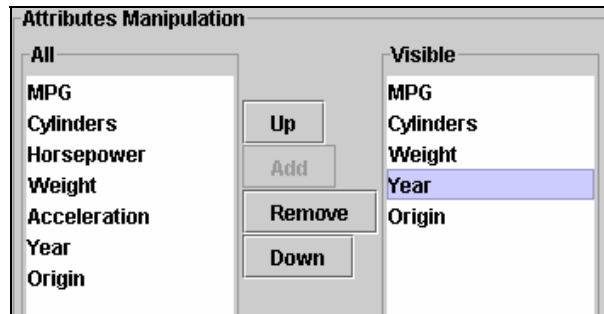


Figura 4.11: Componentes gráficos que compõem o mecanismo de interação *Attributes Manipulation*

5 COORDENAÇÃO DINÂMICA DE VISUALIZAÇÕES MULTIDIMENSIONAIS

No capítulo 2 desse trabalho, foram descritas as técnicas de interação que costumam ser associadas a representações visuais isoladas de dados multidimensionais e no capítulo seguinte, foram apresentados os principais tipos de coordenações implementadas pelos sistemas com múltiplas visualizações coordenadas em geral. Nesse capítulo, o objetivo é apresentar e descrever um conjunto específico de coordenações (interações): aquelas que podem ser aplicadas em sistemas de múltiplas visualizações coordenadas, específicos para a visualização de dados multidimensionais. Além disso, a fim de validar o conjunto de coordenações apresentado aqui, será abordada também a descrição de uma aplicação de visualizações multidimensionais coordenadas, que implementa o conjunto de coordenações apresentado.

5.1 Coordenações para Sistemas de Visualização de Dados Multidimensionais

Analisando-se os tipos de coordenações disponibilizadas por sistemas de visualizações coordenadas em geral, procurou-se identificar quais coordenações poderiam ser usadas para conectar diferentes visualizações multidimensionais e auxiliariam no processo de exploração de um conjunto de dados com múltiplas dimensões. Como resultado deste estudo, definiu-se um conjunto com quatro categorias gerais de coordenações: *mapeamento de dados para propriedades visuais*, *reordenação de dimensões*, *filtragem* e *brushing-and-linking*. Todas estas coordenações podem ser implementadas de forma bidirecional, o que garante que, para um par de visualizações interligadas, uma interação do usuário com a primeira visualização será propagada para a segunda, e vice-versa.

A seguir, são discutidas quais categorias de visualizações multidimensionais suportam as coordenações acima e como estas coordenações podem ser implementadas em cada categoria.

5.1.1 Mapeamento de Dados para Propriedades Visuais

Todas as técnicas multidimensionais de projeção geométrica suportam várias possibilidades de mapeamentos de dados para propriedades visuais, pois os itens de dados, nestas técnicas, são representados por figuras geométricas (círculos, retângulos, linhas, etc.). Assim, em qualquer técnica de projeção geométrica, valores de atributos podem ser mapeados para cor, tamanho ou transparência de objetos de dados. Algumas técnicas dessa categoria, como Radviz, Matriz de *Scatter Plots* e Polyviz, podem

suportar ainda mapeamentos de dados para o tipo de figura utilizada como objeto de dado e sua orientação. Além disso, outro tipo de mapeamento visual pode ainda ser utilizado em técnicas de projeção geométrica: a ordenação em profundidade. Este tipo de mapeamento utiliza valores de atributo para determinar a ordem em que objetos de dados são desenhados na representação visual. Como o problema da sobreposição de objetos costuma ser muito acentuado em técnicas de projeção geométrica, a possibilidade de alterar a ordem de desenho destes objetos é muito útil para uma exploração mais efetiva do conjunto de dados.

Em técnicas multidimensionais iconográficas, é possível o emprego de mapeamentos de dados para cor, espessura e transparência. Um glifo ou ícone que representa um item de dado nestas técnicas pode refletir seu valor para um determinado atributo através da cor ou nível de transparência que é desenhado ou ainda através da espessura de suas linhas (ver a figura 3.2 para um exemplo). Geralmente, os glifos ou ícones dessa categoria de técnicas não possuem uma localização determinada onde devem ser exibidos na tela e, por esta razão, não costumam haver sobreposições dos mesmos. Se não há sobreposição de objetos, não faz sentido o emprego do mapeamento de dados para ordenação em profundidade.

As técnicas orientadas a pixel, finalmente, não permitem nenhum tipo de mapeamento de dados para propriedades visuais definido pelo usuário. Estas técnicas já possuem uma forma de mapeamento fixa, que não pode ser alterada pelo usuário. Ou seja, todas as técnicas dessa categoria utilizam um mapa de cores fixo, que é utilizado para mapear os valores de itens de dados para todas as dimensões. Não é possível utilizar um mapa de cores para representar apenas valores de uma dimensão, como é feito em técnicas de projeção geométrica e iconográficas. Outros tipos de mapeamentos também não são possíveis: diferentes níveis de transparência tornariam a visualização confusa; diferentes tamanhos não podem ser aplicados para pixels; e ordenação em profundidade não faz sentido, pois não há sobreposições.

5.1.2 Reordenação de Dimensões

Todas as técnicas de visualização multidimensionais suportam alguma forma de reordenação de dimensões. Em técnicas de projeção geométrica que representam dimensões como eixos, esse tipo de interação é bastante intuitivo: consiste apenas em mudar a disposição dos eixos na visualização. Na técnica Matriz de *Scatter Plots*, a reordenação de dimensões pode ser executada alterando a numeração da linha e coluna da matriz, representativas de uma dimensão.

Na técnica iconográfica Faces de *Chernoff*, a reordenação de dimensões pode ser realizada alterando a característica da face (olhos, nariz, boca, etc.) que representa cada dimensão. Na técnica Glifos em Estrela, a reordenação é realizada alterando a posição, em graus, dos raios que representam as dimensões.

Finalmente, nas técnicas orientadas a pixel, a reordenação de dimensões pode ser aplicada simplesmente trocando a ordem em que as janelas de dimensões aparecem na tela.

5.1.3 Filtragem

Em visualizações multidimensionais, dois tipos de filtragem podem ser realizadas: de objetos de dados e de dimensões.

A filtragem de objetos de dados causa o mesmo efeito em todas as técnicas de visualização multidimensionais. Os itens de dados com valores dentro de um intervalo filtrado deixam de ser representados nas visualizações. Em técnicas de projeção geométrica, são figuras geométricas que deixarão de ser desenhadas; em técnicas iconográficas, são glifos ou ícones que não serão representados; e em técnicas orientadas a pixel, para cada item de dado filtrado, um pixel em cada janela de dimensão será cedido para a representação de um outro item de dado.

A filtragem de dimensões faz com que uma ou mais dimensões do conjunto de dados deixem de ser representadas nas visualizações multidimensionais. Em técnicas de projeção geométrica, são eixos que deixarão de ser desenhados (ou linhas e colunas, para a Matriz de *Scatter Plots*); em técnicas iconográficas, são características de faces ou raios que não serão representados; e em técnicas orientadas a pixel, janelas de dimensões serão excluídas da representação visual.

5.1.4 *Brushing-and-Linking*

Quaisquer visualizações multidimensionais podem ser coordenadas pela técnica *brushing-and-linking*. Em técnicas de projeção geométrica, objetos de dados selecionados são geralmente exibidos com preenchimento vermelho ou com esta cor de borda; em técnicas iconográficas, as arestas de ícones ou glifos são exibidas com a cor de destaque escolhida; e em técnicas orientadas a pixel, os pixels selecionados são exibidos com uma borda de destaque. A coordenação *brushing-and-linking* possui várias formas de implementação e não necessariamente requer que o usuário selecione objetos de dados clicando com o mouse sobre eles. Algumas implementações destacam os objetos de dados correspondentes em outras visualizações quando o usuário simplesmente passa o mouse sobre eles.

Todas as coordenações mencionadas nessa seção foram implementadas em uma aplicação de múltiplas visualizações coordenadas, flexível em coordenações, com o objetivo de validá-las e de criar uma ferramenta para visualização de dados multidimensionais que oferece variadas possibilidades de conexões e liberdade na construção de cenários.

5.2 Aplicação de Visualizações Coordenadas para Dados Multidimensionais

A aplicação de múltiplas visualizações coordenadas desenvolvida para validar o conjunto de coordenações definido nesse trabalho foi projetada para trabalhar apenas com a representação gráfica de dados multidimensionais. Assim, tanto as visualizações, como as coordenações por ela suportadas, são direcionadas para a representação desse tipo de dados.

Desenvolvida a partir das classes que compõem o toolkit de visualização InfoVis (FEKETE, 2003), a aplicação integra as técnicas de visualização e interações do toolkit que podem ser utilizadas para dados multidimensionais. Foi acrescentada a possibilidade de coordenar dinamicamente visualizações através destas interações.

A aplicação permite que usuários construam dinamicamente cenários com múltiplas visualizações coordenadas, sem a necessidade de programação. Com isto, pretende-se oferecer flexibilidade na representação de informações. Tal flexibilidade é suportada em

relação aos dados representados, técnicas de visualização e coordenações. Ou seja, a aplicação permite que usuários possam escolher o conjunto de dados a ser carregado, definir quais dentre as visualizações suportadas irão compor um cenário de múltiplas visualizações e quais ações interativas serão coordenadas entre as mesmas. Usuários têm liberdade para criar e remover coordenações a qualquer momento, sendo capazes de sincronizar as ações que são importantes para suas tarefas.

5.2.1 Visualizações

No capítulo 2 desse trabalho, foi apresentada uma categorização, frequentemente utilizada na literatura, para técnicas de visualização multidimensionais, bem como também se descreveu as características gerais das principais técnicas representantes de cada categoria.

Essa seção apresenta uma breve descrição das técnicas de visualização suportadas pela aplicação desenvolvida, bem como também mostra como estas técnicas foram incorporadas ao toolkit InfoVis. Atualmente, três técnicas de visualização de informações multidimensionais estão disponíveis: Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots*. A primeira técnica já se encontrava implementada no toolkit InfoVis e as outras duas foram incorporadas especialmente para comporem a aplicação relatada nesse trabalho.

5.2.1.1 Coordenadas Paralelas

A técnica Coordenadas Paralelas faz parte do conjunto de visualizações disponibilizadas pelo InfoVis. Sua implementação no toolkit foi levemente modificada com a finalidade de acrescentar características importantes.

A implementação dessa técnica, utilizada na aplicação desenvolvida, representa os atributos (dimensões) do conjunto de dados por meio de linhas verticais dispostas de forma regular (espaçamentos iguais). A ordenação inicial dos atributos (da esquerda para a direita) segue a mesma ordem em que estes aparecem no arquivo de dados. Cada linha vertical possui acima dela a indicação do nome do atributo que a mesma representa.

Os valores mínimo e máximo que cada atributo assume em determinado conjunto de dados são mapeados para as extremidades inferior e superior das linhas verticais, respectivamente, e valores intermediários a estes têm suas posições, ao longo das mesmas, calculadas com base no intervalo mínimo-máximo.

Os itens de dados, representados por linhas horizontais, aparecem, inicialmente, todos com a mesma cor. A figura 5.1 mostra a técnica Coordenadas Paralelas no toolkit InfoVis, exibindo o conjunto de dados sobre carros (descrito na seção 6.1.1).

Os procedimentos seguidos para a criação das linhas verticais, que representam os atributos, e das linhas horizontais, que representam os itens de dados, serão detalhadamente descritos a seguir, pois os mesmos representam as operações fundamentais para a criação da representação visual dessa técnica.

Na implementação de Coordenadas Paralelas no toolkit InfoVis, as linhas verticais foram incorporadas ao plano de fundo (*background*) do gráfico 2D. Por este motivo, elas são computadas e desenhadas dentro do método *paintBackground*. A figura 5.2 mostra o código Java executado por este último.

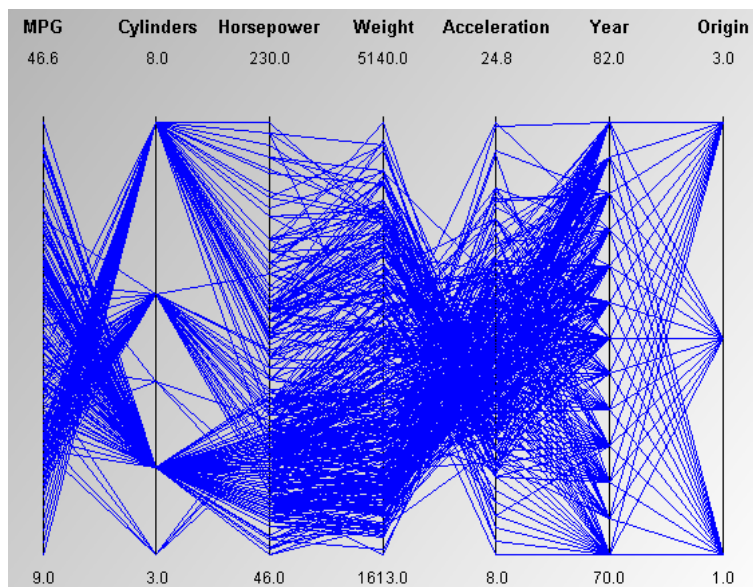


Figura 5.1: Técnica Coordenadas Paralelas implementada no toolkit InfoVis

Na figura 5.2, para o desenho da linha vertical de cada atributo, é necessário fornecer ao método *drawLine* (linha 9) as coordenadas (x,y) de dois pontos, que representam as extremidades da linha. Como se trata de uma linha vertical, a coordenada x de ambos os pontos é a mesma e seu valor é obtido pela fórmula da linha 6. A coordenada y do primeiro ponto é igual ao valor do limite superior do gráfico 2D (linha 7) e a coordenada y do segundo ponto é igual ao valor do limite inferior do mesmo gráfico (linha 13).

```

1. public void paintBackground(Graphics2D graphics, Rectangle2D bounds) {
2.     // espaçamento entre linhas verticais
3.     double sx = bounds.getWidth() / (columns.size() - 1);
4.     // para cada atributo...
5.     for (int i = 0; i < columns.size(); i++) {
6.         float x = (float) (sx * i + bounds.getX());
7.         float y = (float) bounds.getY();
8.         // desenha linha vertical
9.         graphics.drawLine(
10.            (int) x,
11.            (int) y,
12.            (int) x,
13.            (int) bounds.getHeight());
14.     }
15. }

```

Figura 5.2: Criação e desenho das linhas que representam atributos na técnica Coordenadas Paralelas (implementado por Fekete)

A figura 5.3 apresenta o código Java executado pelo método *computeShapes*, responsável pela criação dos objetos que representam os itens de dados na técnica Coordenadas Paralelas. O código das linhas 6 à 35 é executado para cada item de dado, conforme determina o laço *for* da linha 5. As linhas 8 à 12 verificam se já existe um objeto *GeneralPath* – caminho geométrico formado por segmentos de linha reta – para o item de dado atual. Se tal objeto existe, ele tem seus atributos inicializados, caso contrário, um novo objeto *GeneralPath* é criado.

O laço *for* da linha 15 indica que cada item de dado será considerado em relação a cada atributo do conjunto de dados a partir desse ponto. Com isto, as linhas de código contidas no laço têm como objetivo calcular a posição em que as linhas horizontais

(itens de dados) atravessam as linhas verticais (atributos). Se o item de dado possui um valor numérico válido para o atributo em questão, a posição onde a linha horizontal corta a linha vertical desse atributo, no eixo x, é dada pela fórmula da linha 26 e a posição no eixo y é dada pela fórmula da linha 28. Finalmente, o método `setShapeAt` armazena o caminho de segmentos de linha para o item de dado.

```

1. public void computeShapes(Rectangle2D bounds) {
2.     // espaçamento entre linhas verticais
3.     double sx = bounds.getWidth() / (columns.size() - 1);
4.     // para cada item de dado...
5.     for (RowIterator iter = iterator(); iter.hasNext(); ) {
6.         int i = iter.nextRow();
7.         // obtém caminho de segmentos de reta para o item de dado
8.         GeneralPath p = (GeneralPath) getShapeAt(i);
9.         if (p == null)
10.            p = new GeneralPath();
11.         else
12.            p.reset(); // inicializa
13.         boolean first = true;
14.         // para cada atributo...
15.         for (int col = 0; col < columns.size(); col++) {
16.             NumberColumn n = getNumberColumnAt(col);
17.             // se valor do item para o atributo não é válido...
18.             if (n.isValueUndefined(i)) {
19.                 first = true;
20.                 continue;
21.             }
22.             double min = n.getDoubleMin(); // valor min. do atributo
23.             double max = n.getDoubleMax(); // valor máx. do atributo
24.             double diff = (max - min);
25.             double sy = diff == 0 ? 1 : bounds.getHeight() / diff;
26.             float x = (float) (sx * col + bounds.getX());
27.             float h = (float) (sy * (n.getDoubleAt(i) - min));
28.             float y = (float) (bounds.getY() + bounds.getHeight() - h);
29.             if (first) {
30.                 p.moveTo(x, y); // primeiro ponto do caminho
31.                 first = false;
32.             } else
33.                 p.lineTo(x, y); // define segmentos de reta
34.             }
35.         setShapeAt(i, p); // armazena caminho (linhas horizontais)
36.     }
37. }

```

Figura 5.3: Criação dos objetos que representam itens de dados na técnica Coordenadas Paralelas (implementado por Fekete)

5.2.1.2 Radviz

A técnica Radviz não faz parte do conjunto de técnicas de visualização que acompanham o toolkit InfoVis. Ela foi incorporada ao mesmo para a construção da aplicação relatada nesse trabalho.

A implementação dessa técnica no toolkit representa os atributos (dimensões) do conjunto de dados por meio de linhas que emanam radialmente do centro de um círculo e terminam em seu perímetro. A ordenação inicial dos atributos no círculo (em sentido horário) segue a mesma ordem em que estes aparecem no arquivo de dados. No final de cada linha, no perímetro do círculo, o nome do atributo que a mesma representa permanece visível. Além disso, para facilitar a compreensão do mapeamento dos pontos

de dados realizado pela técnica, todos os seus valores são normalizados para o intervalo de 0 à 1.

Os itens de dados, representados nessa implementação por pequenas elipses, aparecem, inicialmente, todos com a mesma cor. A figura 5.4 mostra a representação visual da técnica Radviz implementada no toolkit InfoVis, exibindo o conjunto de dados sobre carros.

Os procedimentos seguidos para a criação das linhas e das pequenas elipses, que representam os atributos e os itens de dados, respectivamente, serão detalhadamente descritos a seguir, por representarem as operações fundamentais da técnica Radviz.

Na implementação dessa técnica no toolkit InfoVis, as linhas representando atributos também foram computadas e desenhadas dentro do método *paintBackground*, como ocorreu com as linhas de atributos na técnica Coordenadas Paralelas. No entanto, nesse caso, as mesmas não foram incorporadas ao plano de fundo (*background*) do gráfico 2D, mas sim desenhadas como objetos separados do mesmo. A razão para isto é a precisão necessária (*double*) para o desenho das linhas de atributos do Radviz, precisão esta não alcançada com o método *drawLine* (figura 5.2 – linha 9) empregado em Coordenadas Paralelas, que trabalha apenas com valores inteiros. A figura 5.5 mostra o código Java executado pelo método *paintBackground* da técnica Radviz.

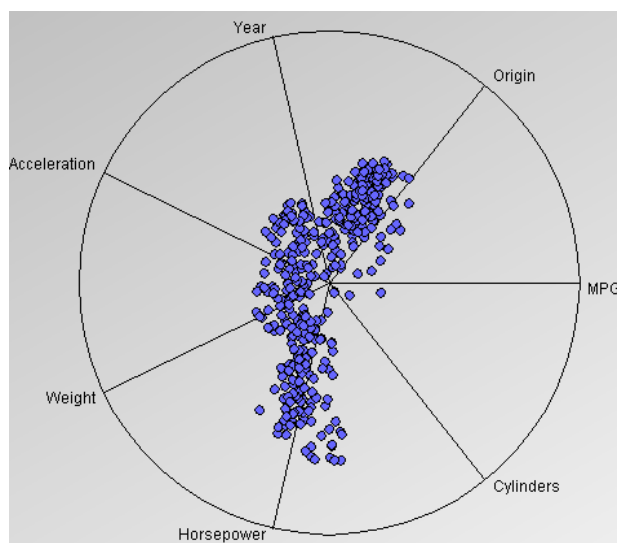


Figura 5.4: T cnica Radviz implementada no toolkit InfoVis

Todas as linhas de atributos s o computadas por esse m todo, no interior do laço *while* (linha 15), que executa seu bloco de opera es somente para colunas num ricas do conjunto de dados. Cada nova linha criada tem seu ponto inicial no centro do c rculo Radviz e seu ponto final sobre a borda do mesmo (linha 20-21). O primeiro ponto   igual para todas as linhas e previamente conhecido, mas o segundo ponto precisa ser calculado. As f rmulas das linhas 17 e 18 calculam a posi o do ponto final de cada linha, considerando um c rculo que abrange o intervalo $[-1, 1]$ tanto em x como em y . As coordenadas resultantes s o ent o multiplicadas pelo raio do c rculo Radviz e somadas aos valores do centro do mesmo (linha 21) para a obten o da posi o correta de t rmino da linha de atributo. Nos comandos seguintes da figura 5.5, a linha computada e o nome do atributo que ela representa s o desenhados no gr fico 2D. As opera es necess rias para o c lculo e desenho da borda do c rculo Radviz tamb m s o

executadas pelo método *paintBackground*, mas foram ocultadas da figura 5.5 porque não possuem importância significativa para esse trabalho.

```

1. public void paintBackground(Graphics2D graphics, Rectangle2D bounds) {
2.     double raio, map_x, map_y;
3.     double width = bounds.getWidth(); // largura do gráfico 2D
4.     double height = bounds.getHeight(); // altura do gráfico 2D
5.     double center_x = width / 2; // centro do círculo em x
6.     double center_y = height / 2; // centro do círculo em y
7.     // escolhe menor dimensão como base para o raio do círculo
8.     if (width < height)
9.         raio = center_x - margin;
10.    else
11.        raio = center_y - margin;
12.    int n = getTotalNumberColumn(); // número total de atributos
13.    int j = 0;
14.    // para cada atributo...
15.    while (getNumberColumn(table, j) != null) {
16.        // calcula posição da fatia do círculo relativa ao atributo
17.        map_x = Math.cos((2 * PI * j) / n);
18.        map_y = Math.sin((2 * PI * j) / n);
19.        // cria nova linha representando atributo
20.        Line2D.Double lin = new Line2D.Double(center_x, center_y,
21.                                               (center_x + raio * map_x), (center_y + raio * map_y) );
22.        lin = transformShape(lin); // aplica lente fisheye, se habilitada
23.        graphics.setColor(Color.BLACK); // seta cor preta
24.        graphics.draw(lin); // desenha a linha
25.        paintAxisLabel(graphics, j, lin); // desenha nome do atributo que a linha representa
26.        j++;
27.    }
28. }

```

Figura 5.5: Criação e desenho das linhas que representam atributos na técnica Radviz

A figura 5.6 apresenta o código Java executado pelo método *computeShapes*, responsável pela criação dos objetos que representam itens de dados na técnica Radviz. As operações que determinam a posição onde estes objetos devem aparecer são executadas para cada item de dado, conforme determina o laço *for* da linha 12, e considerando-os em relação a cada atributo, como determina o laço *while* da linha 18.

Todos os valores de um item de dado são, primeiramente, normalizados para o intervalo [0, 1], com base nos valores mínimo e máximo do atributo ao qual ele se refere (linhas 20-21). Nas linhas 22 à 26, a posição do item de dado para cada atributo isolado e seu valor normalizado são acumulados e registrados. A localização final de um item (onde a soma das forças das molas dos atributos é zero) é encontrada dividindo a soma das posições do item para cada atributo isolado pela soma dos valores normalizados, multiplicando o resultado pelo raio do círculo Radviz e, finalmente, somando as coordenadas do seu centro (linhas 31 e 32). Um novo objeto elipse é criado nessa posição (linha 36) e armazenado (linha 37).


```

1. public void computeShapes(Rectangle2D bounds) {
2.     double raio, sum_x, sum_y, normaliz, map_x, map_y, ksum;
3.     double width = bounds.getWidth();           // largura do gráfico 2D
4.     double height = bounds.getHeight();        // altura do gráfico 2D
5.     double center_x = width / 2;              // centro do círculo em x
6.     double center_y = height / 2;            // centro do círculo em y
7.     // escolhe menor dimensão como base para o raio do círculo
8.     if (width < height) raio = center_x - margin;
9.     else raio = center_y - margin;
10.    int n = getTotalNumberColumn();           // número total de atributos
11.    // para cada item de dado...
12.    for (RowIterator iter = iterator(); iter.hasNext();) {
13.        int row = iter.nextRow();
14.        sum_x = 0; sum_y = 0; ksum = 0;
15.        int j = 0;
16.        NumberColumn numCol = getNumberColumn(table, j); // obtém coluna numérica
17.        // para cada atributo...
18.        while (numCol != null) {
19.            // normaliza valor do item de dado para o intervalo [0, 1]
20.            normaliz = (numCol.getDoubleAt(row) - numCol.getDoubleMin()) /
21.                (numCol.getDoubleMax() - numCol.getDoubleMin());
22.            map_x = Math.cos((2 * PI * j) / n); // posição máxima para o atributo (em x)
23.            sum_x += normaliz * map_x;         // acumula posição do item p/ cada atributo
24.            map_y = Math.sin((2 * PI * j) / n); // posição máxima para o atributo (em y)
25.            sum_y += normaliz * map_y;         // acumula posição do item p/ cada atributo
26.            ksum += normaliz;                 // acumula valores normalizados do item
27.            j++;
28.            numCol = getNumberColumn(table, j);
29.        }
30.        if (ksum == 0) ksum = 0.00001;        // evita divisão por zero
31.        double xi = center_x + raio * (sum_x / ksum); // posição final do item (em x)
32.        double yi = center_y + raio * (sum_y / ksum); // posição final do item (em y)
33.        double s = getSizeAt(row);           // tamanho do objeto elipse
34.        double s2 = s / 2;
35.        // cria nova elipse representando item de dado
36.        Ellipse2D.Double elip = new Ellipse2D.Double(xi - s2, yi - s2, s, s);
37.        setShapeAt(row, elip);              // armazena item de dado
38.    }
39. }

```

Figura 5.6: Criação dos objetos que representam itens de dados na técnica Radviz

5.2.1.3 Matriz de Scatter Plots

A técnica multidimensional Matriz de *Scatter Plots* também não faz parte do conjunto de técnicas de visualização do toolkit InfoVis. No entanto, a técnica *Scatter Plot 2D*, que é exibida em cada célula dessa matriz, já se encontrava implementada no mesmo. Assim, construiu-se a implementação da Matriz de *Scatter Plots* no toolkit como um agrupamento de n *Scatter Plots 2D*. O valor de n corresponde ao quadrado do número de dimensões dos dados (dimensoes²).

Cada *Scatter Plot 2D* criado para compor a matriz é uma visualização independente, que implementa os métodos *paintBackground* e *computeShapes* da interface de visualização, assim como as técnicas descritas acima. Logo, não é necessário fornecer, para a classe de visualização que representa a Matriz, uma implementação para estes métodos, pois a representação visual dessa matriz é formada pelo conjunto das representações de seus *Scatter Plots 2D*.

A representação visual de um *Scatter Plot 2D* é bastante simples: as figuras de itens de dados são mapeadas como em gráficos cartesianos, considerando seus valores para um eixo x e um eixo y . Nesse caso, os eixos x e y são representados por dois atributos do conjunto de dados. Assim, dada a simplicidade do mapeamento utilizado por essa técnica, não será apresentada aqui a implementação para os métodos *computeShapes* e *paintBackground* da mesma.

A figura 5.7 mostra a representação visual da Matriz de *Scatter Plots* implementada no toolkit InfoVis, exibindo o conjunto de dados sobre carros. Os itens de dados são representados em cada *Scatter Plot 2D* por pequenas elipses e o nome dos atributos que correspondem aos eixos x e y destas são exibidos nas laterais da matriz. Por outro lado, a figura 5.8 mostra o código Java executado pelo método *createVisualizationMatrix* da classe Matriz, que gera o agrupamento de *Scatter Plots 2D* (matriz).

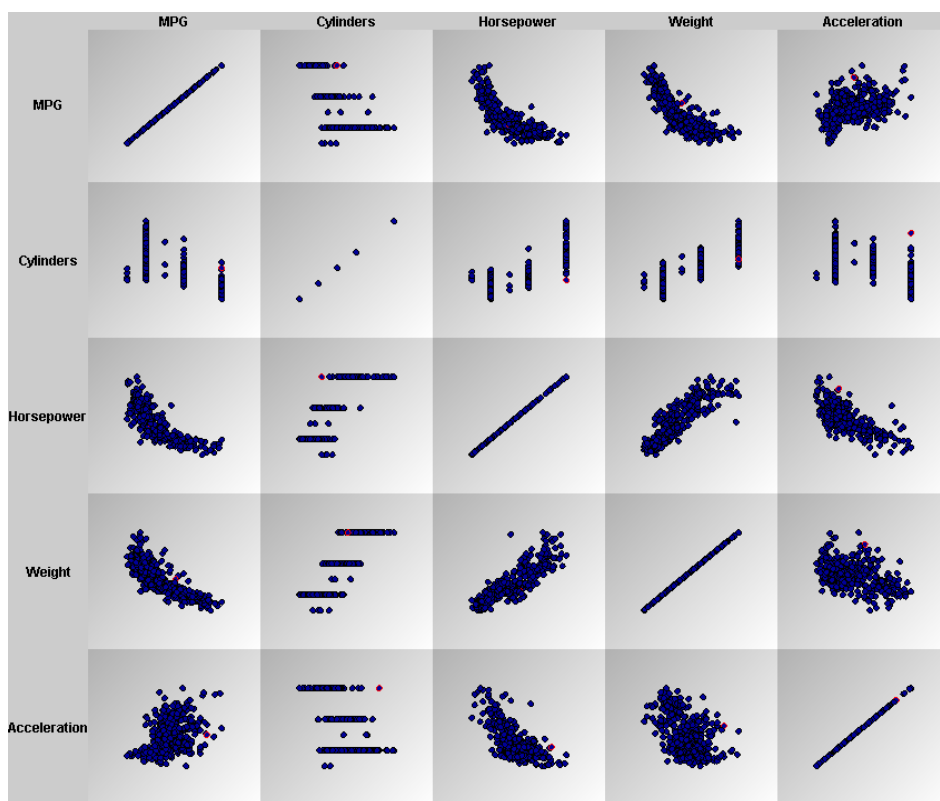


Figura 5.7: Técnica Matriz de *Scatter Plots* implementada no toolkit InfoVis

É importante salientar que devido à classe de visualização Matriz de *Scatter Plots* não fornecer implementação para o método *computeShapes*, ela não gera um conjunto de objetos gráficos para representar itens de dados. Apenas as visualizações *Scatter Plots 2D* geram estes objetos. Assim, para que um usuário possa interagir com a visualização Matriz através de um painel de controles para interação, que modificam a representação de objetos de dados, uma de suas visualizações *Scatter Plot 2D* deve ser associada diretamente a este painel de controle. Quando esta visualização sofrer alterações na representação de seus objetos de dados (causada por uma interação com o painel de controle), as demais visualizações da matriz deverão ser igualmente alteradas. Para isto, um *Scatter Plot 2D* é identificado como a principal (associada ao painel de controle) e as demais são registradas como “ouvintes” (*listeners*) de mudanças em determinadas propriedades desta visualização.

```

1. public void createVisualizationMatrix() {
2.     //IntVector numericColumnIndices = índices das colunas numéricas da tabela;
3.     int n = numericColumnIndices.size();
4.     JPanel panel = new JPanel( new GridLayout(n, n) );
5.
6.     mainVisualization = null; // visualização associada com um painel de controle
7.
8.     for (int i = 0; i < n; i++) {
9.         // obtém uma coluna da tabela
10.        Column c = table.getColumnAt( numericColumnIndices.get(i) );
11.
12.        for (int j = 0; j < n; j++) {
13.            // cria uma visualização scatter plot 2D
14.            ScatterPlotVisualization visualization = new ScatterPlotVisualization(
15.                table, selectionColumn, filterColumn );
16.            VisualizationPanel visualizationPanel =
17.                new VisualizationPanel(visualization);
18.
19.            // adiciona a scatter plot na matriz (grid)
20.            panel.add( visualizationPanel );
21.
22.            // seta duas colunas da tabela para os eixos x e y da scatter plot 2D
23.            visualization.setVisualColumn( ScatterPlotVisualization.VISUAL_X_AXIS,
24.                table.getColumnAt( numericColumnIndices.get(i) ) );
25.            visualization.setVisualColumn( ScatterPlotVisualization.VISUAL_Y_AXIS,
26.                table.getColumnAt( numericColumnIndices.get(j) ) );
27.
28.            if ( i== 0 && j == 0 ) {
29.                // a primeira scatter plot criada será associada com o painel de controle
30.                mainVisualization = visualization;
31.            }
32.            else {
33.                addListenerForMainVisualization( mainVisualization, visualization );
34.            }
35.        }
36.    }
37. }

```

Figura 5.8: Procedimento para a criação da matriz de visualizações *Scatter Plots 2D*

No código da figura 5.8, é inicialmente criado um *grid* de n linhas e n colunas (linha 4), onde serão adicionadas as representações dos *Scatter Plots 2D*. Cada visualização criada desta técnica (linhas 14-15) tem seus eixos x e y associados com um par diferente de atributos do conjunto de dados (linhas 23-26). Estes eixos são representados na classe de visualização *ScatterPlotVisualization* como propriedades visuais. A primeira visualização *Scatter Plot 2D* criada é identificada como a principal da matriz (linha 30), ou seja, aquela que será associada com um painel de controle. Todas as visualizações criadas após esta são registradas como suas “ouvintes” pelo método *addListenerForMainVisualization*. Este último, cujo código é mostrado na figura 5.9, é executado para cada *Scatter Plot 2D* que forma a matriz (exceto a principal).

No código da figura 5.9, um objeto “ouvinte” de mudanças de propriedades é criado (linha 5) e registrado para a visualização principal da matriz (linha 4). Uma propriedade é identificada por uma constante string e representa a ocorrência de uma interação em uma visualização. Assim, quando uma determinada propriedade da visualização principal é alterada, o método *propertyChange* (linha 6-49) é invocado. Este identifica qual foi exatamente a propriedade modificada e a altera da mesma forma nas demais visualizações *Scatter Plot 2D*. Assim, o método *addListenerForMainVisualization* garante que todas as visualizações que compõem a matriz estejam sempre iguais, utilizando, para isto, coordenadas fixas e pré-determinadas entre elas. A figura 5.7, por exemplo, mostra um mesmo item selecionado em todas as células da Matriz.

```

1. void addListenerForMainVisualization( ScatterPlotVisualization mainVisualization,
2.                                     final ScatterPlotVisualization visualization )
3. {
4.     mainVisualization.addPropertyChangeListener(
5.         new PropertyChangeListener() {
6.             public void propertyChange( final PropertyChangeEvent evt ) {
7.                 new Thread( new Runnable() {
8.                     public void run() {
9.
10.                        if ( evt.getPropertyName().equals(VISUAL_SIZE) ||
11.                            evt.getPropertyName().equals(VISUAL_SORT) ||
12.                            evt.getPropertyName().equals(VISUAL_ALPHA) ||
13.                            evt.getPropertyName().equals(VISUAL_LABEL) ||
14.                            evt.getPropertyName().equals(VISUAL_COLOR) ) {
15.
16.                            visualization.setVisualColumn( evt.getPropertyName(),
17.                                                            (Column) evt.getNewValue() );
18.                        }
19.                    else {
20.
21.                        if ( evt.getPropertyName().equals(PROPERTY_SMOOTH) ) {
22.                            visualization.setSmooth( !visualization.isSmooth() );
23.                            return; }
24.
25.                        if ( evt.getPropertyName().equals(PROPERTY_DEFAULT_SIZE) ) {
26.                            visualization.setDefaultSize( (Double) evt.getNewValue() );
27.                            return; }
28.
29.                        if ( evt.getPropertyName().equals(PROPERTY_DEFAULT_ALPHA) ) {
30.                            visualization.setDefaultAlpha( (Double) evt.getNewValue() );
31.                            return; }
32.
33.                        if ( evt.getPropertyName().equals(PROPERTY_SHOWING_LABEL) ) {
34.                            visualization.setShowingLabel(
35.                                !visualization.isShowingLabel() );
36.                            return; }
37.
38.                        if ( evt.getPropertyName().equals(PROPERTY_FISHEYES) ) {
39.                            visualization.setFisheyes( (Fisheyes) evt.getNewValue() );
40.                            return; }
41.
42.                        if ( evt.getPropertyName().equals(PROPERTY_EXCENTRIC_LABEL) ) {
43.                            visualization.setExcentric( (evt.getNewValue() != null) ?
44.                                                            new DefaultExcentricLabels() : null );
45.                            return; }
46.                    }
47.                }
48.            } ).run();
49.        }
50.    }
51.    );
52. }

```

Figura 5.9: Procedimento utilizado para garantir que ações realizadas em uma visualização 2D da matriz de *Scatter Plots* sejam executadas também nas suas demais visualizações

5.2.2 Coordenações

O toolkit InfoVis, utilizado para a construção da aplicação em questão, implementa e disponibiliza para todas as suas visualizações (sem coordenação) grande parte das técnicas interativas que podem ser coordenadas na visualização de dados multidimensionais (apenas as manipulações de atributos de dados não são suportadas). Assim, na aplicação desenvolvida foi necessário apenas habilitar a coordenação dinâmica destas técnicas e acrescentar aquelas ainda não suportadas pelo toolkit.

Na aplicação desenvolvida, o tipo de coordenação *mapeamento de dados para propriedades visuais* possui cinco formas específicas: mapeamento para cor (denominado *Color by*), mapeamento para tamanho (denominado *Size by*), mapeamento para transparência (denominado *Alpha by*), mapeamento para ordenação em profundidade (denominado *Sort by*) e, finalmente, mapeamento para texto de *labels* (denominado *Label by*). A *filtragem de dimensões e reordenação* destas foram agrupadas em uma única coordenação, denominada *Attributes manipulation*. A denominação *Filtering* foi limitada para designar apenas a filtragem coordenada de objetos de dados e a denominação *Selection* foi adotada para representar a coordenação *brushing-and-linking*.

Analisando os efeitos de cada ação interativa disponibilizada pela aplicação, decidiu-se coordenar apenas ações iguais. Muitas combinações não são possíveis e aquelas que o são, não auxiliam significativamente na compreensão do conjunto de dados. Além disso, desta forma, todas as coordenações realizadas na aplicação serão bidirecionais.

Como interagir com uma visualização e os efeitos gerados por cada interação já foram detalhadamente descritos na seção 4.4 do Capítulo 4, que abordou os mecanismos de interação oferecidos pelo toolkit InfoVis. Cada visualização suportada pela aplicação desenvolvida possui um painel de controle próprio, que o usuário deve utilizar para interagir com sua representação. Este painel de controle é igual àquele apresentado na seção 4.4 (incluindo os mecanismos adicionados). Na aplicação de visualizações coordenadas, o usuário deverá apenas estabelecer e remover dinamicamente coordenações, o que será tratado mais adiante nesse capítulo.

Atualmente, oito ações interativas podem ser coordenadas dinamicamente entre as três visualizações suportadas pela aplicação. Como a execução e efeitos destas ações já foram discutidos no capítulo anterior, descreve-se, agora, resumidamente, o conjunto completo de ações coordenadas suportadas pela aplicação desenvolvida:

- *Selection*: Objetos de dados selecionados com o mouse em uma visualização são destacados (exibidos com uma borda vermelha) em todas as visualizações coordenadas. Além disso, os valores que estes itens possuem para cada atributo de dados são exibidos nos painéis de controle das visualizações (figura 5.10, em baixo à esquerda).
- *Filtering*: Intervalos numéricos de filtragem podem ser especificados para cada atributo numérico do conjunto de dados, separadamente. Itens de dados com valores contidos em um intervalo filtrado deixam de ser desenhados nas visualizações coordenadas. Para atributos categóricos, a filtragem de valores é realizada selecionando o nome destes em uma lista. Objetos de dados com valores não selecionados nesta lista também deixam de ser desenhados na visualização.
- *Size by*: Aplica variações progressivas no tamanho dos objetos de dados para representar quão alto é o valor que cada objeto apresenta para um determinado atributo (escolhido pelo usuário). O menor tamanho é atribuído ao objeto de dado que possui o menor valor para o atributo sendo mapeado e o maior tamanho é aplicado ao objeto de dado que apresenta o maior valor para o mesmo (figura 5.10, centro).

- *Color by*: Aplica variações progressivas na intensidade da cor dos objetos de dados para representar quão alto é o valor que cada objeto apresenta para um determinado atributo (escolhido pelo usuário). A menor intensidade de cor é atribuída ao objeto de dado que possui o menor valor para o atributo sendo mapeado e a maior intensidade de cor é aplicada ao objeto de dado que apresenta o maior valor para o mesmo (figura 5.10, centro).
- *Sort by*: Utiliza o valor que cada item de dado possui para um determinado atributo (especificado pelo usuário) como critério para a ordenação dos objetos de dados em profundidade. Ou seja, a ordem em que figuras de itens se sobrepõem na representação visual pode ser determinada pela escala de valores de um atributo do conjunto de dados. O primeiro item a ser desenhado é aquele que possui o menor valor para o atributo determinante e o último item desenhado é aquele que apresenta o maior valor para o mesmo.
- *Label by*: Determina que o conteúdo de *labels* exibidos para cada objeto de dado será o valor que este possui para um determinado atributo (escolhido pelo usuário).
- *Alpha by*: Aplica variações progressivas no nível de transparência dos objetos de dados para representar quão alto é o valor que cada objeto apresenta para um determinado atributo (escolhido pelo usuário). O menor nível de transparência é atribuído ao objeto de dado que possui o maior valor para o atributo sendo mapeado e o maior nível de transparência é aplicado ao objeto de dado que apresenta o menor valor para o mesmo.
- *Attributes manipulation*: Permite que atributos (dimensões) do conjunto de dados sejam retirados/acrescentados de representações visuais pelo usuário. Além disso, o usuário pode também alterar a ordem em que tais atributos são exibidos.

Dentre as ações mencionadas acima, apenas a seleção (*selection*) é realizada interagindo diretamente sobre as representações visuais. As demais ações são realizadas através da interação do usuário com componentes gráficos dos painéis de controle.

A figura 5.10 mostra um cenário de exploração construído na aplicação relatada aqui, para análise do conhecido conjunto de dados sobre carros. Nas três visualizações do cenário, o mecanismo de interação *Color by* foi utilizado para mapear os valores do atributo de dados *Cylinders* para intensidades de cor. Nos gráficos das técnicas Radviz e Coordenadas Paralelas foi utilizado ainda o recurso *Size by*, para mapear valores do atributo de dados *Year* para o tamanho (ou espessura) das figuras de itens de dados. As figuras que aparecem com o tom mais escuro de verde e com o maior tamanho são aquelas que possuem os valores mais altos para os atributos *Cylinders* e *Year*, respectivamente.

Ainda em relação à figura 5.10, há um item de dado selecionado em todas as visualizações. Este item pode ser identificado porque aparece como uma linha vermelha na técnica Coordenadas Paralelas e como um círculo de borda vermelha nas outras técnicas. Os valores que o item selecionado possui para cada atributo do conjunto de dados são mostrados em uma tabela na parte superior do painel de controle das visualizações. Na figura, esta tabela aparece no painel de controle da técnica Radviz.

Em nenhuma das visualizações da mesma figura, o conjunto de dados inteiro está representado. Uma pequena parcela dos itens de dados foi filtrada: aqueles que

possuíam os mais baixos valores para o atributo de dados *Acceleration*. A aplicação deste filtro pode ser notada observando o painel de controle da técnica Radviz.

As coordenações entre visualizações são estabelecidas e removidas dinamicamente pelo usuário, interagindo com o grafo de coordenações. Este último é representado, na figura 5.10, no interior da janela *Main Window*. A descrição do grafo de coordenação e de como interagir com ele serão abordadas nas próximas seções desse capítulo.

É importante mencionar que além das interações que podem ser coordenadas entre visualizações, as quais foram abordadas nessa seção, a aplicação desenvolvida também disponibiliza para as três visualizações suportadas outras possibilidades de interação (não-coordenadas), como as lentes *Fisheyes*, *labels* dinâmicos e configurações padrão de objetos de dados. Estas técnicas interativas foram descritas na seção 4.4 do capítulo 4.

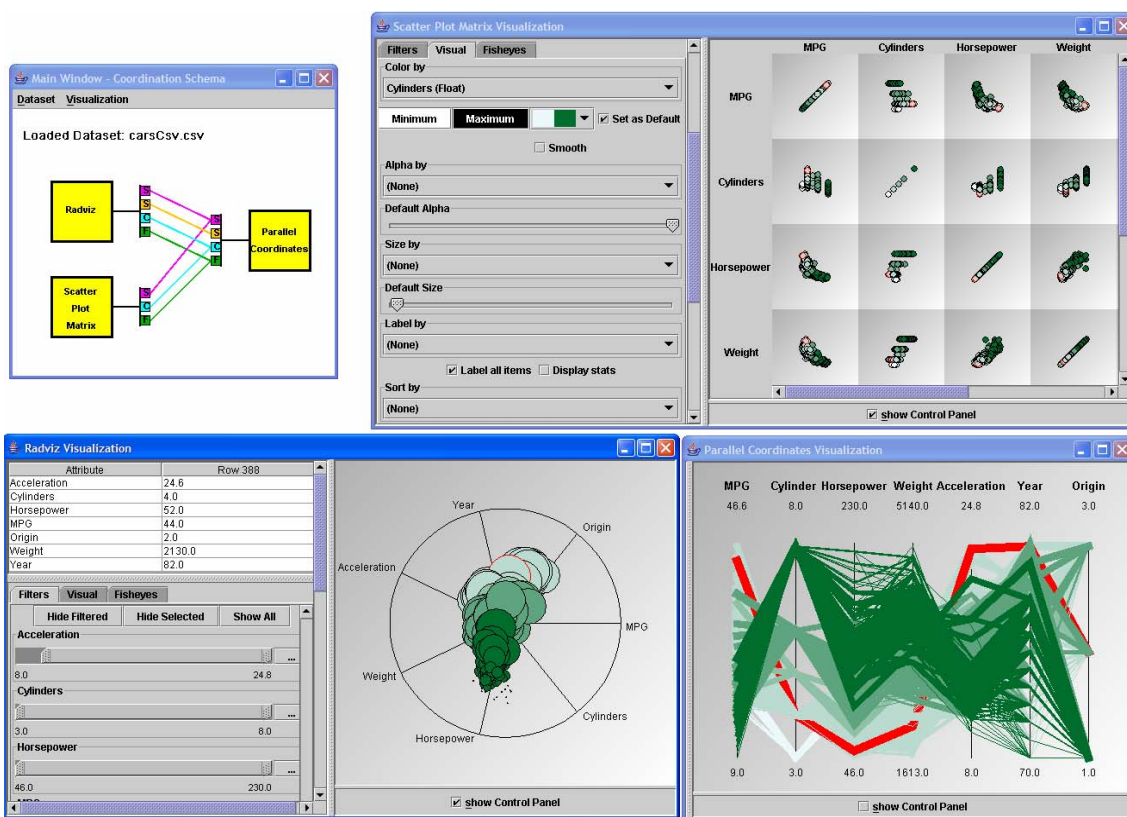


Figura 5.10: Cenário de exploração construído na aplicação protótipo

5.2.3 Arquitetura

A arquitetura da aplicação de visualizações coordenadas desenvolvida é uma combinação de dois estilos arquiteturais. A comunicação entre componentes internos da aplicação segue o mesmo padrão utilizado pelo toolkit InfoVis, ou seja, uma organização orientada a objetos. Estes últimos encapsulam dados e operações a eles associadas (SHAW; GARLAN, 1996). Cada objeto no sistema precisa conhecer a identidade de outros objetos para que possam ocorrer trocas de mensagens. Por outro lado, a propagação de interações (coordenação) foi construída com arquitetura baseada em eventos. A ocorrência destes últimos causa chamadas implícitas de métodos.

A aplicação pode ser dividida em cinco estruturas principais: tabela, visualizações, painéis de controle, controlador de coordenação e grafo de coordenação. As classes que compõem as três primeiras estruturas pertencem ao toolkit InfoVis. No entanto, algumas extensões foram feitas nas classes de visualização e de controles de interação, com o objetivo de viabilizar coordenações e ampliar suas possibilidades. A figura 5.11 demonstra as relações existentes entre os módulos da aplicação.

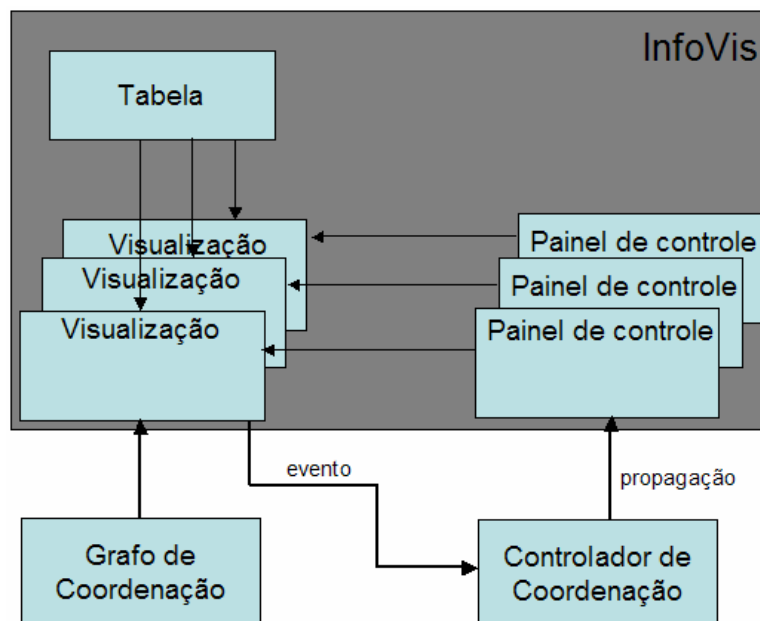


Figura 5.11: Arquitetura do protótipo desenvolvido

Os dados multidimensionais obtidos a partir de um arquivo são armazenados na estrutura Tabela. As visualizações transformam estes dados armazenados em representações visuais. Todas as visualizações suportadas pela aplicação representam graficamente a mesma tabela de dados, porém utilizando técnicas diferentes para o mapeamento de itens. Mais de uma visualização da mesma técnica pode ser gerada, se o usuário desejar.

Cada visualização possui sempre um painel de controle associado a ela. Utilizando os componentes de interface contidos neste último, o usuário pode interagir com a visualização. Apenas a seleção de itens é realizada com manipulação direta.

O grafo de coordenação é a estrutura por meio da qual o usuário pode definir e remover dinamicamente coordenações entre visualizações. Além de armazenar as propriedades relevantes de cada coordenação, essa estrutura também mantém atualizada uma representação gráfica interativa das conexões atuais. Logo após o estabelecimento de uma nova coordenação, o grafo atualiza a lista de conexões das duas visualizações envolvidas.

Finalmente, o controlador de coordenação é a estrutura encarregada de receber eventos de interação ocorridos em qualquer das visualizações e propagar a interação para aquelas conectadas à primeira. Como todas as visualizações possuem um painel de controle associado e todas as ligações são bidirecionais, o controlador de coordenação propaga interações modificando os componentes do painel de controle. Este, por sua vez, atualiza a visualização correspondente.

5.2.4 Grafo de Coordenação

Como a aplicação desenvolvida permite que conexões entre visualizações sejam definidas e removidas dinamicamente, é necessário manter uma estrutura atualizada, que armazene as coordenações atuais. Além disso, deve ser oferecida ao usuário uma forma visual de construir cenários com múltiplas visualizações coordenadas assim como visualizar, a qualquer momento, esta estrutura, já que ele pode esquecer com facilidade quais ligações estabeleceu. O grafo de coordenação é a estrutura que oferece todos estes serviços mencionados, sendo representado graficamente por um diagrama de nodos e linhas de conexão (figura 5.12).

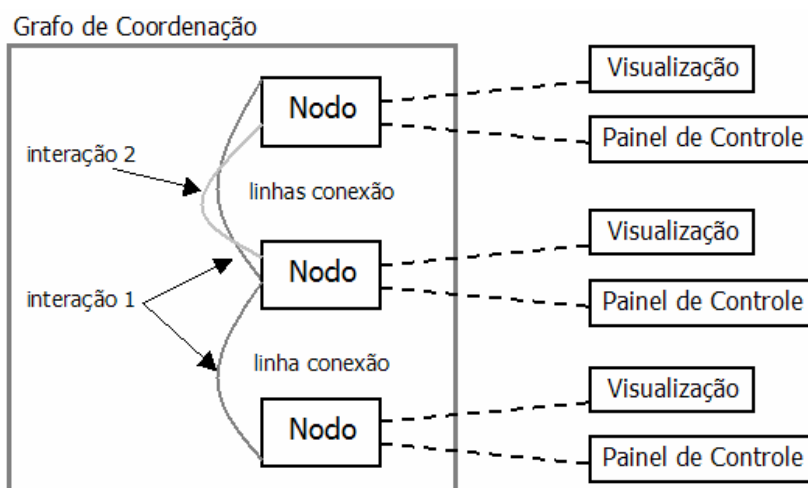


Figura 5.12: Estrutura do grafo de coordenação e seus relacionamentos com outros módulos da aplicação

Cada nodo do grafo de coordenação encapsula uma visualização e seu painel de controle. Além disso, ele também mantém armazenadas as conexões associadas com o mesmo.

As informações contidas nos nodos são utilizadas para manter o grafo de coordenação em estado coerente e para atualizar a lista de conexões das visualizações envolvidas em novas ligações.

Cada linha de conexão mantém referências para os dois nodos envolvidos na ligação que ela representa. Além disso, toda linha de conexão possui também uma informação que identifica o tipo de interação associado a ela. Como pode ser observado na figura 5.12, dois nodos podem ter mais de uma linha de conexão ligando-os. No entanto, cada linha destas deve representar um tipo de interação diferente. A flexibilidade em coordenações, oferecida pela aplicação, permite que o usuário possa coordenar apenas as ações interativas que lhe parecerem convenientes.

No momento em que uma nova linha de conexão é criada pelo usuário, três ações importantes são executadas pela aplicação para viabilizar esta coordenação. Primeiramente, todas as visualizações envolvidas na coordenação, direta ou indiretamente, são inicializadas para a ação coordenada. Ou seja, se a nova linha de conexão representa a interação *filtering*, todos os itens que estavam filtrados anteriormente em visualizações envolvidas na coordenação, deixarão de estar. O componente gráfico do painel de controle que invoca esta ação será forçado a voltar a seu estado de inicialização. Com isto, conseqüentemente, a visualização será também

atualizada. O que se pretende com estas inicializações é evitar que visualizações coordenadas apresentem estados diferentes logo após a conexão.

Em seguida, a aplicação adiciona o controlador de coordenação como “ouvinte” da ação coordenada em ambas as visualizações envolvidas na ligação. Sempre que a ação em questão for realizada pelo usuário em uma destas visualizações, o controlador de coordenação tomará conhecimento disto e propagará a ação para as visualizações coordenadas.

Finalmente, a última providência necessária para viabilizar a coordenação é atualizar a lista de conexões das duas visualizações envolvidas na ligação. A primeira visualização é adicionada à lista de conexões da segunda, e a segunda é adicionada à lista de conexões da primeira.

A seqüência completa de operações realizadas desde que o usuário solicita a criação de uma nova conexão até sua inteira viabilização pela aplicação é representada no diagrama de seqüência UML da figura 5.13.

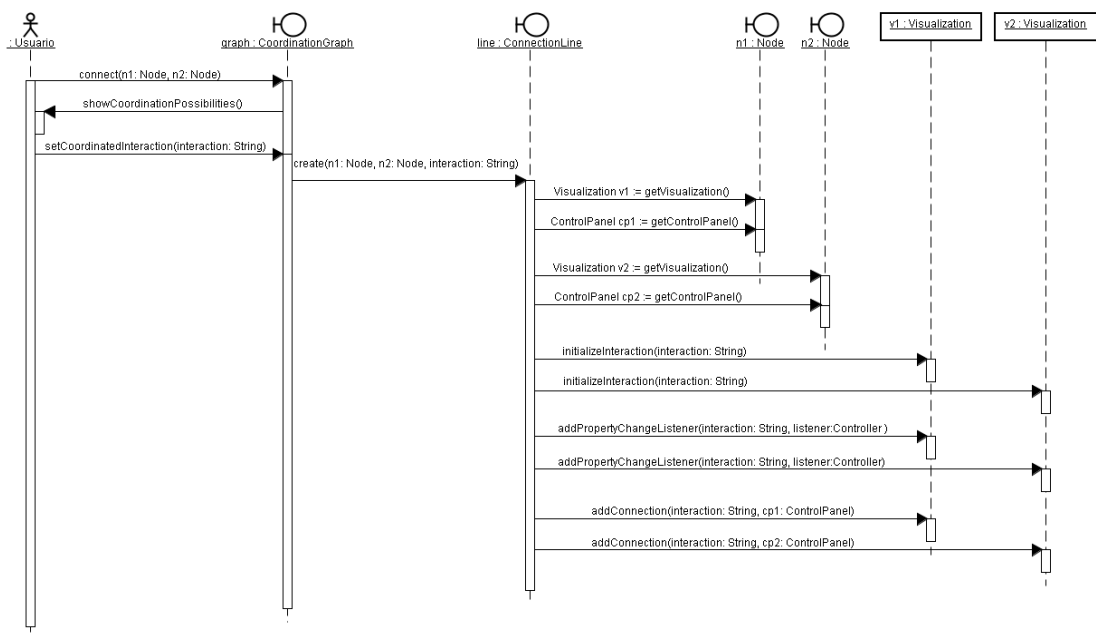


Figura 5.13: Processo de criação e viabilização de uma nova coordenação

5.2.5 Especificando Coordenações Dinamicamente

Com o intuito de permitir que o usuário construa dinamicamente relacionamentos entre visualizações, a aplicação mantém visível na janela principal uma representação gráfica atualizada do grafo de coordenação. O usuário pode interagir a qualquer momento com esta representação para criar ou remover conexões e também para alterar a disposição dos elementos que compõem o grafo.

Os nodos do grafo são representados por retângulos amarelos. Os mesmos são identificados pelo nome da visualização que representam. As linhas de conexão são representadas graficamente por linhas coloridas. Cada interação possui uma cor que a identifica.

Para permitir a identificação da ação representada por cada linha de conexão, além da cor, foram adicionados pequenos conectores na representação do grafo de

coordenação. Cada um destes representa uma interação diferente e, devido a isto, são exibidos como retângulos com a cor de sua interação e a primeira letra do nome da mesma. Cada conector está associado a um nodo. Posicionando o ponteiro do mouse sobre ele, o nome completo da interação é exibido.

As linhas de conexão não aparecem conectando diretamente nodos, mas sim, conectores. Dessa forma, é possível identificar com facilidade a interação representada por cada uma delas. A figura 5.14 mostra a representação gráfica de um grafo de coordenação com três nodos e todas as suas coordenações possíveis ligadas.

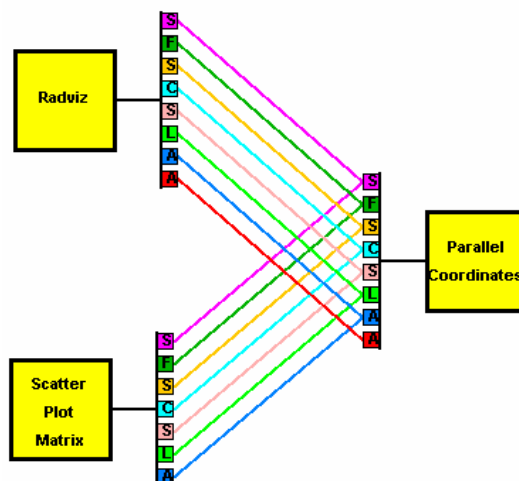


Figura 5.14: Representação gráfica de um grafo de coordenação

No cenário representado pelo grafo da figura 5.14, todas as visualizações estão coordenadas. Embora não existam linhas de conexão ligando diretamente as visualizações Radviz e Matriz de *Scatter Plots*, elas estão conectadas indiretamente por meio das ações coordenadas em comum com a visualização Coordenadas Paralelas. Dessa forma, não é necessário ligar explicitamente essas duas visualizações.

5.2.5.1 Criando Coordenações

Uma ação interativa pode ser coordenada entre duas visualizações interagindo no grafo de coordenação:

1. Os nodos envolvidos na coordenação devem ser conectados por uma linha pontilhada vermelha, conforme mostra a figura 5.15. O usuário cria esta linha clicando uma vez sobre o primeiro nodo e depois sobre o segundo.



Figura 5.15: Conectando os nodos da nova coordenação

2. Com a criação da linha, aparece uma janela de diálogo (como a da figura 5.16). Nesta, o usuário deve selecionar quais das ações exibidas serão coordenadas.

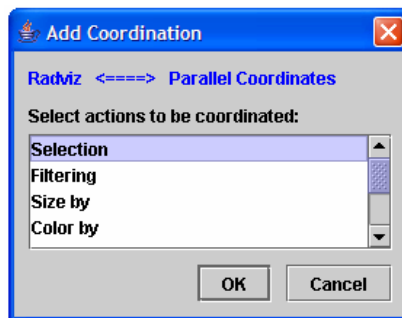


Figura 5.16: Seleção de ações a serem coordenadas

3. Após a escolha, as linhas de conexão são criadas e desenhadas no grafo (figura 5.17).

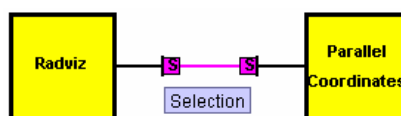


Figura 5.17: Nodos conectados pela ação de seleção

5.2.5.2 Removendo Coordenações

Uma coordenação entre duas visualizações pode ser removida interagindo no grafo de coordenação:

1. O usuário deve pressionar o botão direito do mouse sobre a linha que representa a coordenação a ser removida.
2. No menu que aparece, selecionar a opção *Remove coordination* (mostrada na figura 5.18).

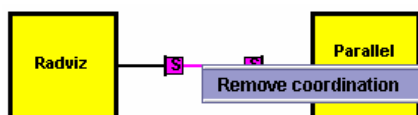


Figura 5.18: Removendo uma coordenação

5.2.6 Gerenciamento de Coordenações

Quando o usuário interage com uma visualização, a aplicação propagará os efeitos desta ação para as visualizações coordenadas através das conexões estabelecidas. Serão afetadas pela interação as visualizações conectadas direta ou indiretamente à primeira.

O controlador de coordenação recebe eventos que indicam que uma ação foi executada em uma visualização e seu estado mudou. A partir deste evento, ele identifica a ação e provoca sua execução em cada uma das visualizações coordenadas à primeira. O controlador recebe eventos a partir de uma visualização, mas propaga os efeitos da interação representada por este evento através dos painéis de controle. Assim, ele não intermedia a comunicação entre objetos de visualização, mas sim, entre estes últimos e seus painéis de controle.

A figura 5.19 mostra as classes da aplicação envolvidas no processo de gerenciamento de coordenação e as mensagens trocadas entre seus objetos. É importante mencionar que a seqüência de operações do diagrama não termina com a execução de

repaint() no objeto de visualização. Após esta operação, um novo evento de notificação de mudança será gerado pela visualização atualizada e enviado ao controlador de coordenação. Deve-se ressaltar que os componentes do diagrama não possuem indicação de nomes de objetos, o que significa que eles são utilizados para representar n objetos da classe.

A comunicação entre objetos de visualização e o controlador de coordenação se dá conforme especifica o padrão de projeto *Observer* (GAMMA et al., 1995). O controlador de coordenação é um observador dos objetos de visualização. Quando um destes últimos tem seu estado alterado, ele envia uma notificação de mudança para o controlador, e este, por sua vez, obterá o novo estado do objeto de visualização, a fim de propagar este estado para outros objetos.

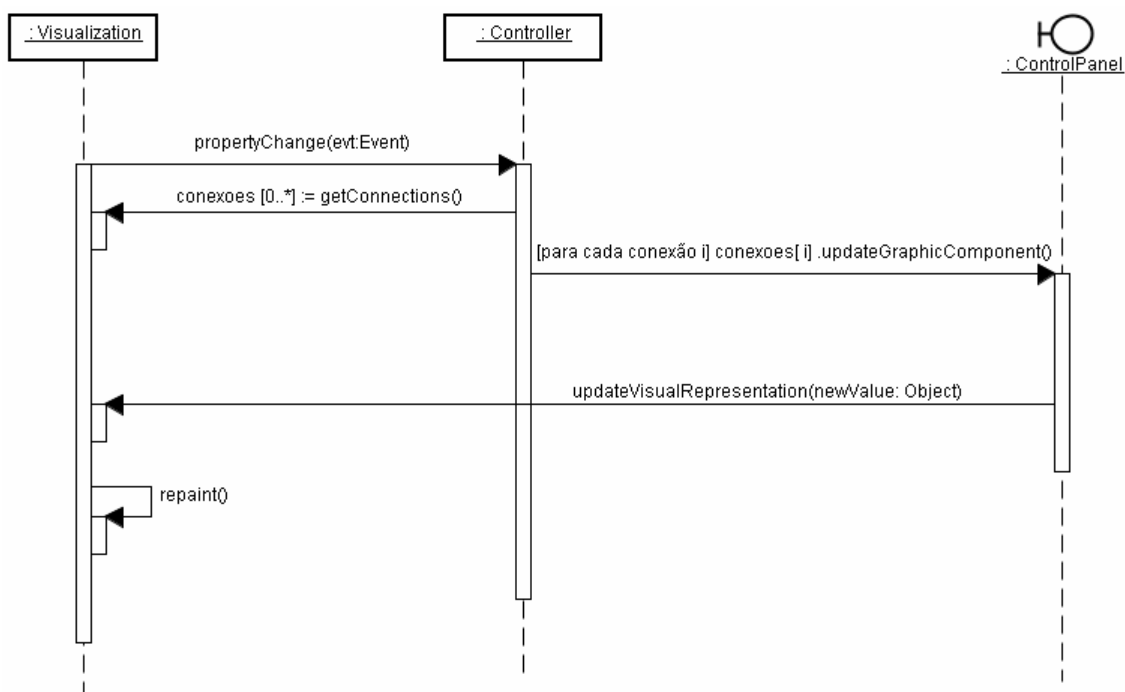


Figura 5.19: Processo de gerenciamento de coordenações

Para cada interação que pode ser coordenada, a aplicação possui uma constante string que armazena um nome atribuído a ela. Esta mesma constante é utilizada como uma propriedade para a notificação de mudanças em uma visualização, provocadas pela interação em questão. Dessa forma, quando o usuário interage com uma determinada visualização, o método da mesma que processa a ação é invocado. Este último atualiza a visualização (fornecendo *feedback* ao usuário) e provoca a geração de um evento de mudança de propriedade, cujo nome é a string mencionada acima, que representa a interação executada.

O evento gerado pela visualização é enviado ao controlador de coordenação (registrado como “ouvinte” de todas as visualizações), provocando uma chamada implícita ao método deste que propaga interações. O controlador de coordenação obtém a lista de visualizações conectadas diretamente (pela interação em questão) àquela que gerou o evento e força a atualização do painel de controle associado a elas. Assim como o usuário faz uso de componentes gráficos do painel de controle para interagir com as visualizações, o controlador de coordenação também modifica o estado destes componentes para provocar interações nas visualizações coordenadas.

O diagrama da figura 5.20 representa graficamente as atividades de alto nível descritas acima, que são executadas pelo usuário e pelo sistema. Por outro lado, o diagrama da figura 5.21 fornece uma visão mais detalhada do processo de propagação de interações. Ele mostra a seqüência de operações de objeto executadas desde a interação do usuário com uma visualização até o término da propagação dessa interação para as visualizações coordenadas à primeira.

Logo que as visualizações coordenadas diretamente àquela com a qual o usuário interagiu forem atualizadas, estas também provocarão a geração de um novo evento de mudança de propriedade. Este último, por sua vez, também será tratado pelo controlador de coordenação e a interação por ele representada será propagada para outras visualizações coordenadas a elas.

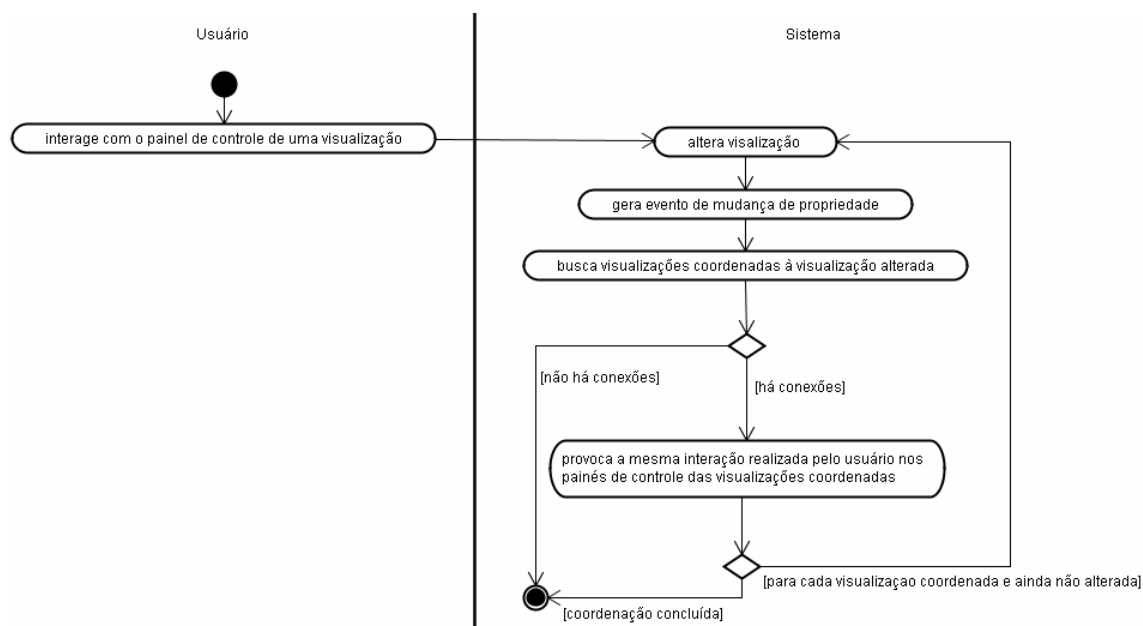


Figura 5.20: Atividades realizadas pelo usuário e aplicação no processo de propagação de uma interação

Conforme já mencionado em seção anterior, no momento em que uma nova coordenação é estabelecida entre duas visualizações, a primeira delas é adicionada à lista de conexões da segunda e a segunda é adicionada à lista de conexões da primeira. Embora o controlador de coordenação atualize o componente gráfico que causa uma interação sendo propagada nos painéis de controle de todas as visualizações coordenadas com a que o usuário interagiu, não ocorrem ciclos de propagação infinitos. Isto porque um painel de controle somente irá atualizar sua visualização correspondente se foi gerado um evento de mudança em um de seus componentes gráficos. Assim, se o componente gráfico atualizado pelo controlador já se encontrava exatamente no estado desejado antes da ação do controlador, nenhum evento de mudança será gerado para este componente. Conseqüentemente, a visualização ligada ao painel de controle não será novamente atualizada e a propagação termina.

A propagação de interações é realizada utilizando múltiplas *threads*. A atualização de cada painel de controle de uma visualização coordenada é executada por uma *thread* diferente. Isto permite ao controlador de coordenação continuar propagando interações, enquanto cada painel de controle e sua visualização são atualizados.

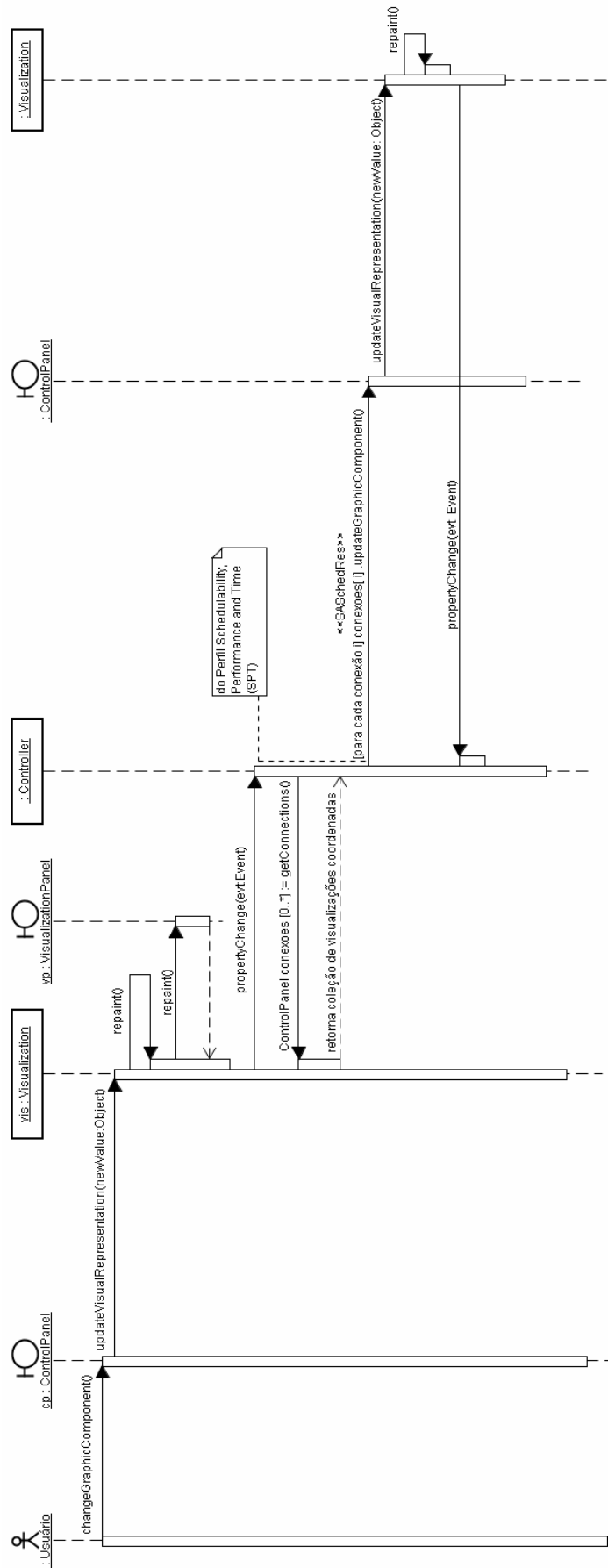


Figura 5.21: Seqüência de operações realizadas pelos objetos envolvidos na propagação de interações

5.2.7 Tornando uma Visualização Coordenável: A Interface *Visualization*

No topo da estrutura de classes de visualização do toolkit InfoVis está uma interface denominada *Visualization*. Nesta última, estão contidas as assinaturas de todos os métodos que uma visualização precisa implementar para ser incorporada ao toolkit e disponibilizar ao usuário todos os mecanismos de interação padrão oferecidos pelo mesmo. Como a implementação de muitos destes métodos é igual em diferentes visualizações, o InfoVis conta com uma classe abstrata denominada *DefaultVisualization*, que contém justamente implementações padrão para os métodos da interface *Visualization*. Assim, a maioria das classes de visualizações específicas (Radviz, Coordenadas Paralelas, Matriz de *Scatter Plots*, etc.) derivam da classe *DefaultVisualization* e implementam apenas os métodos que possuem um comportamento específico.

Para adicionar uma nova visualização à aplicação protótipo tema desse trabalho, ela deve apenas ser incorporada ao toolkit InfoVis. Para isto, a classe que representa a nova visualização deve implementar a interface *Visualization* ou estender a classe abstrata *DefaultVisualization*.

A interface *Visualization* contém a assinatura de muitos métodos e a declaração de algumas constantes. Dentre estas últimas, estão as strings que armazenam um nome para cada uma das interações que podem ser coordenadas em nossa aplicação. Estas strings também são utilizadas como propriedades para a notificação de mudanças em uma visualização, provocadas pela interação que elas representam. A tabela 5.1 mostra as interações suportadas pela aplicação e suas respectivas constantes.

Tabela 5.1: Interações coordenadas e as propriedades que notificam sua ocorrência

<i>Interação</i>	<i>Nome da Propriedade que Notifica a Ocorrência da Interação</i>
<i>Selection</i>	<code>public static final String VISUAL_SELECTION = "Selection";</code>
<i>Filtering</i>	<code>public static final String VISUAL_FILTER = "Filtering";</code>
<i>Sort by</i>	<code>public static final String VISUAL_SORT = "Sort by";</code>
<i>Label by</i>	<code>public static final String VISUAL_LABEL = "Label by";</code>
<i>Color by</i>	<code>public static final String VISUAL_COLOR = "Color by";</code>
<i>Size by</i>	<code>public static final String VISUAL_SIZE = "Size by";</code>
<i>Alpha by</i>	<code>public static final String VISUAL_ALPHA = "Alpha by";</code>
<i>Attributes Manipulation</i>	<code>public static final String PROPERTY_DATA_ATTRIBUTES = "Attributes manipulation";</code>

Em relação aos métodos definidos pela interface *Visualization*, a maioria deles são voltados para a manutenção da representação visual. No entanto, é nesta interface que estão definidos também os métodos que tornam possível uma visualização ser coordenada. Os métodos de *Visualization* diretamente relacionados ao processo de coordenação são mostrados na figura 5.22.

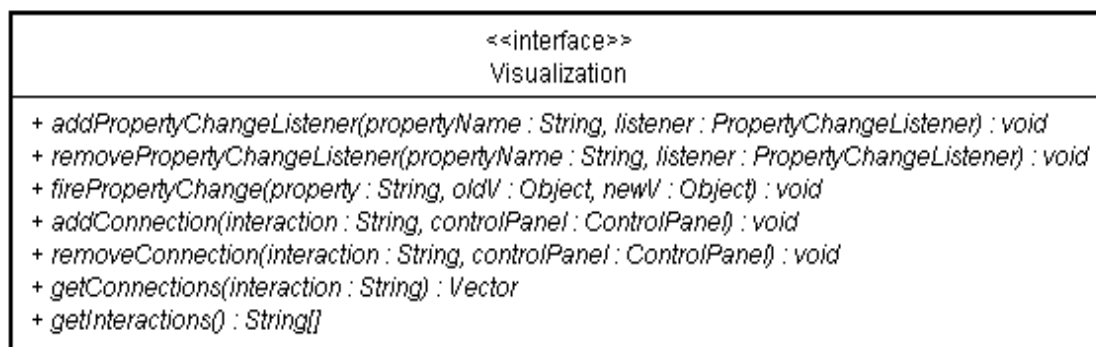


Figura 5.22: Métodos da interface *Visualization* relacionados ao processo de coordenação

Os métodos *addPropertyChangeListener* e *removePropertyChangeListener* acrescentam o controlador de coordenação como “ouvinte” de eventos de mudança de propriedade ocorridos em uma visualização. Estes métodos possuem uma implementação padrão, presente na classe *DefaultVisualization*. O método *firePropertyChange* é invocado por uma visualização logo que uma interação acabou de ser processada na mesma. Ele envia um evento de mudança de propriedade, que será recebido pelo controlador de coordenação. Este método também possui uma implementação padrão em *DefaultVisualization*.

Os métodos *addConnection* e *removeConnection* são invocados pelo grafo de coordenação para atualizar a lista de conexões de uma visualização. Esta última armazena referências não para outras visualizações, mas para painéis de controle. Uma visualização pode facilmente ser obtida através de seu painel de controle, já o contrário não é possível. O método *getConnections* apenas retorna as conexões atuais que uma visualização possui. Finalmente, o método *getInteractions* fornece a lista de interações que são implementadas por uma visualização.

5.3 Comentários Finais

A aplicação desenvolvida e relatada nesse capítulo é flexível em relação aos dados, visualizações e coordenações. Isto significa que, dado um determinado conjunto de dados multidimensionais, os usuários podem escolher as visualizações mais apropriadas para representá-lo e também determinar, dinamicamente, quais interações serão coordenadas entre essas visualizações.

Apenas três técnicas de visualização (Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots*) são suportadas, mas esse conjunto pode ser facilmente ampliado. Para isto, basta apenas incorporar uma nova técnica ao toolkit. As técnicas escolhidas para comporem o grupo de visualizações da aplicação (todas elas geométricas) são bastante conhecidas e apresentam muitas características úteis para a compreensão efetiva dos dados. As possibilidades de interação disponibilizadas pelo toolkit InfoVis foram estendidas para uso coordenado e outras foram adicionadas a esse conjunto inicial.

A descrição da aplicação protótipo foi publicada recentemente, na última International Working Conference on Advanced Visual Interfaces (PILLAT; FREITAS, 2006).

6 AVALIAÇÃO PRELIMINAR

Nesse capítulo serão relatados dois estudos de avaliação preliminares, realizados com a aplicação protótipo apresentada nesse trabalho. O primeiro estudo de avaliação foi realizado logo após o término da implementação das técnicas de visualização multidimensionais sobre o toolkit InfoVis. Estas já possuíam os mecanismos de interação disponíveis, no entanto, coordenações ainda não eram possíveis.

O segundo estudo de avaliação foi realizado logo após o término da implementação da estrutura de coordenações. Já era possível, nesse momento, estabelecer conexões dinamicamente entre visualizações (através do diagrama de coordenações).

Ambos os estudos foram realizados como ensaios de interação com usuários.

6.1 Primeira Avaliação

Este primeiro estudo experimental teve como objetivos avaliar e comparar duas técnicas de visualização de informações, Coordenadas Paralelas e Radviz, bem como seus recursos de interação. A avaliação foi realizada para identificar possíveis problemas de usabilidade nas implementações das duas técnicas e na interface de usuário de ambas, com o intuito de preparar a implementação para as facilidades de coordenação. Um segundo objetivo foi enquadrar os problemas dentro dos critérios ergonômicos de Bastien e Scapin (1993), como colaboração com outro projeto em andamento. É importante mencionar também que as implementações avaliadas nesse estudo eram versões preliminares das visualizações e algumas características e recursos de interação (como a manipulação de atributos) foram adicionados após essa avaliação, com base em seus resultados. A técnica de visualização Matriz de *Scatter Plots* não foi incluída nesse experimento a fim de não torná-lo muito extenso e cansativo para os usuários.

O estudo de avaliação relatado nessa seção foi descrito em artigo publicado na Conferência Latino Americana de Interação Humano Computador (PILLAT; VALIATI; FREITAS, 2005).

6.1.1 Conjunto de Dados

Para esse experimento de avaliação, apenas um conjunto de dados foi utilizado, o conjunto de dados de automóveis, que frequentemente acompanha pacotes para visualização e mineração de dados (GRINSTEIN et al., 2000). Este conjunto é bastante conhecido entre pesquisadores de visualização de informações e é muito utilizado em experimentos desse tipo, como por exemplo, Grinstein et al. (2000) e Kobsa (2001).

O conjunto demonstra as diferentes características dos carros americanos, europeus e japoneses comercializados de 1970 a 1982. Consta de 392 registros (cinco registros foram excluídos da versão original da *CMU Statlib Library* porque havia valores faltando) com 7 atributos (seis numéricos e um categórico). As dimensões são denominadas de *MPG* (milhas por galão), *Cylinders*, *Horsepower*, *Weight*, *Acceleration*, *Year* e *Origin*.

As características que esse conjunto de dados pode ilustrar são (GRINSTEIN et al., 2000):

- Itens situados fora da concentração principal;
- Clusters;
- Clusters de classe (origem do carro, por exemplo).

6.1.2 Tarefas Experimentais

Todos os usuários que realizaram esse teste de avaliação de usabilidade foram instruídos a explorar as visualizações das técnicas avaliadas enquanto buscavam respostas para quatro tarefas/questões experimentais.

Essas tarefas/questões, definidas por Valiati (2004), são as seguintes:

1. A maior produção de carros japoneses foi fabricada entre que anos?
2. Analise os dados e descreva as principais características dos carros americanos.
3. Os carros japoneses com 4 cilindros geralmente são mais pesados do que os americanos com 6 cilindros?
4. Que tendência os carros europeus apresentaram com o passar dos anos em relação às suas características?

6.1.3 Procedimento

O experimento de avaliação de usabilidade em questão foi realizado na forma de ensaios de interação com usuários. Devido a limitações de voluntários com experiência na área, cinco estudantes de visualização de informações participaram do experimento.

Algumas semanas antes do mesmo, os estudantes receberam, em grupo, treinamento de uma hora com relação ao uso das duas técnicas avaliadas. Nesse período, eles receberam explicações gerais sobre a forma como cada técnica organiza o layout dos itens de dados em sua representação visual e exploraram livremente todos os componentes e mecanismos de interação disponibilizados para cada técnica. Durante essa seção de treinamento, foi utilizado, para a exploração das técnicas, um conjunto de dados chamado *Salivary*, que acompanha o toolkit *InfoVis*. O mesmo contém informações temporais a respeito de glândulas salivares de larvas.

No momento do experimento propriamente dito, cada estudante recebeu um re-treinamento de 15 minutos. Após este tempo, cada sujeito recebeu uma lista contendo as quatro tarefas/questões a serem resolvidas utilizando as visualizações e foi instruído a verbalizar todas as ações que iam sendo realizadas e problemas encontrados, assim como, a sua gravidade.

Os experimentos foram conduzidos individualmente, em laboratório, na presença de dois observadores, sendo que um deles anotou os problemas de usabilidade observados e/ou relatados pelo usuário (em cada técnica de visualização avaliada) e o outro observou o procedimento de interação do usuário.

Os estudantes foram randomicamente selecionados para utilizar as duas técnicas em ordem inversa de forma alternada, ou seja, dois usuários utilizaram Coordenadas Paralelas e depois Radviz e os outros três utilizaram Radviz e depois Coordenadas Paralelas, totalizando dez observações. Não foi estipulado limite de tempo para a completude das tarefas/questões.

6.1.4 Resultados

Os problemas relatados pelos usuários e/ou identificados pelo observador do experimento foram classificados dentro de nove critérios de usabilidade, definidos por Bastien e Scapin (1993). Ao todo, eles são dezoito critérios, mas apenas em nove deles foram identificados problemas com as implementações avaliadas, segundo o resultado do experimento.

Os dezoito critérios elementares definidos por Bastien e Scapin são: Incitação ou presteza, agrupamento/distinção por localização, agrupamento/distinção por formato, feedback imediato, legibilidade, concisão, ações mínimas, densidade informacional, ações explícitas do usuário, controle do usuário, flexibilidade, consideração da experiência do usuário, proteção contra erros, qualidade das mensagens de erro, correção dos erros, consistência, significância de códigos e compatibilidade.

A fim de classificar o grau de severidade dos problemas de usabilidade identificados, foram usados os seguintes códigos:

- Severidade 0 (sem importância): Não afeta o funcionamento da ferramenta;
- Severidade 1 (cosmético): Não há necessidade de reparar o problema imediatamente;
- Severidade 2 (simples): Problema de baixa prioridade (não precisa ser reparado);
- Severidade 3 (grave): Problema de alta prioridade (precisa ser reparado);
- Severidade 4 (catastrófico): Muito grave (precisa ser reparado).

A seguir, será descrito cada um dos problemas de usabilidade identificados, organizados de acordo com os critérios em que foram enquadrados.

6.1.4.1 Incitação/Presteza

Três usuários que participaram do experimento sentiram falta da indicação dos valores mínimo e máximo de cada atributo próximo às barras de filtragem. Além desses valores, eles também mencionaram que o intervalo correspondente aos itens não filtrados, em cada momento, deveria permanecer visível ao lado das barras de filtragem. Os componentes e mecanismos de interação de ambas as técnicas avaliadas possuem o mesmo funcionamento. O recurso utilizado pela interface para exibir os valores mencionados acima é um label, que aparece apenas quando o usuário posiciona o cursor do mouse sobre a barra de filtragem, o que não agradou os usuários devido ao tempo e esforço requerido para a obtenção da informação.

Na técnica Coordenadas Paralelas, os mesmos usuários mencionaram ainda que os valores mínimo e máximo de cada atributo deveriam permanecer exibidos também nas extremidades inferior e superior, respectivamente, da linha vertical da visualização correspondente a ele.

Classificou-se os problemas identificados para esse critério como graves (severidade 3), que devem ser reparados.

6.1.4.2 *Agrupamento/Distinção por Localização*

Todos os usuários (5 sujeitos) manifestaram alguma dificuldade quanto à interpretação do layout da técnica Radviz. Isto foi sentido, principalmente, nos momentos em que era necessário conhecer a ordenação dos itens de dados com respeito a um determinado atributo. Para essa tarefa, todos os usuários recorreram ao uso de atributos visuais, já que o layout da técnica, em muitos casos, não fornecia essa informação com clareza. Um dos usuários sugeriu que a interface poderia fornecer um mecanismo que mapeasse todos os itens sobre um determinado eixo do Radviz, definido pelo usuário, nas posições correspondentes a seus valores para o atributo.

O problema identificado para esse critério foi considerado de tipo cosmético (severidade 1), pois se trata de uma deficiência inerente à técnica Radviz.

6.1.4.3 *Feedback Imediato*

Nesse critério, o observador do experimento identificou problemas relativos ao tempo de resposta do sistema para os mecanismos de alteração do tamanho default dos itens, alteração do nível de transparência default dos mesmos e de exibição dos labels dinâmicos (excentric labels).

Classificou-se estes problemas como simples (severidade 2), que podem ser reparados.

6.1.4.4 *Legibilidade*

O primeiro problema de legibilidade apontado pelos usuários foi a respeito do label que exhibe os valores mínimo e máximo de cada atributo, bem como o intervalo dos itens mostrados pelas barras de filtragem. Esse label aparece quando o cursor é posicionado sobre uma destas últimas. O formato em que estes valores são exibidos foi considerado de difícil compreensão por todos os usuários (5 sujeitos).

Na técnica Radviz, a oclusão de objetos (itens de dado) foi apontada por três usuários como um fator que dificulta a interpretação dos dados, em alguns casos. Esse problema é mais evidente, quando o atributo visual de tamanho dos itens (*Size by*) é usado.

A disposição dos atributos no layout da visualização também afeta a interpretação dos dados em ambas as técnicas avaliadas. No entanto, em Radviz o problema é mais acentuado. Três usuários sugeriram que a interface permitisse a alteração da ordenação dos atributos (posições) nas visualizações e a retirada de um ou mais atributos das mesmas, como mecanismos para amenizar o problema da oclusão de itens.

Classificou-se todos os problemas de legibilidade mencionados como graves (severidade 3).

6.1.4.5 *Concisão*

O único problema relatado, por dois usuários, quanto a esse critério foi que a interface de ambas as técnicas avaliadas não permite a eliminação (filtragem) de itens desnecessários para uma análise, quando seus valores para um determinado atributo estão entre os valores de itens relevantes para a mesma. Ou seja, não é permitido especificar dois intervalos diferentes para a filtragem. Esse problema foi considerado simples (severidade 2).

6.1.4.6 *Ações Mínimas*

O problema identificado por um dos usuários para esse critério está relacionado com a volta da aplicação para o estado inicial. Para que a visualização de uma técnica volte a ser exibida exatamente como apareceu inicialmente, o usuário precisa desfazer, uma por uma, todas as alterações realizadas na representação visual por meio de mecanismos de interação, o que pode envolver um grande número de ações. O mesmo usuário sugeriu a criação de um botão reset (de inicialização) para a resolução desse problema. Este último foi considerado do tipo cosmético (severidade 1), pois não há necessidade imediata de solução.

6.1.4.7 *Densidade Informacional*

O observador do experimento notou que um número elevado de itens na visualização influencia a interpretação da técnica Coordenadas Paralelas. Quando havia muitas linhas horizontais, os usuários sentiam dificuldade para acompanhar o caminho que elas seguiam após atravessar uma linha vertical de um atributo. Esse problema foi considerado do tipo cosmético (severidade 1), pois é inerente à técnica, embora existam formas de amenizá-lo (trocando a ordenação dos atributos ou retirando alguns, por exemplo).

6.1.4.8 *Flexibilidade*

A interface de ambas as técnicas avaliadas possui poucos recursos alternativos que dão flexibilidade à execução de tarefas. Os recursos não oferecidos e que os usuários sentiram falta foram:

- Componente para entrada de texto do intervalo a ser filtrado (severidade 1);
- Permitir a seleção de múltiplos itens e exibir no painel de controle informações relevantes sobre o conjunto selecionado, como número de itens, porcentagem dessa quantia em relação ao todo e informações estatísticas (severidade 0);
- Permitir que os itens filtrados apareçam em cinza, ao invés de serem eliminados da visualização (severidade 1);
- Permitir mudança na ordenação dos atributos (severidade 3);
- Permitir eliminação de um ou mais atributos da visualização (severidade 3).

6.1.4.9 *Compatibilidade*

Um dos usuários notou que a seleção de um item com o mouse considera sempre o item mais ao fundo como o selecionado. Quando há oclusão de objetos, o normal seria a seleção considerar o objeto mais à frente, como acontece em outras aplicações.

Classificou-se esse problema como grave (severidade 3), pois o resultado da ação não corresponde ao esperado pelo usuário.

6.2 Segunda Avaliação

Esse segundo estudo de avaliação foi conduzido com o objetivo de avaliar o uso e contribuições das coordenações dinâmicas implementadas na aplicação desenvolvida.

O estudo foi realizado na forma de avaliação subjetiva e observação, com dez estudantes de mestrado, que possuem conhecimento básico sobre visualização de informações. Os estudantes já conheciam as técnicas de visualização suportadas pela aplicação, mas receberam explicações gerais sobre os mecanismos de interação possíveis, assim como sobre a especificação de coordenações.

O procedimento de avaliação correspondeu ao uso livre da aplicação para responder quatro questões experimentais sobre o mesmo conjunto de dados de carros, utilizado na avaliação anterior. Além do conjunto de dados, também as questões experimentais foram as mesmas utilizadas no estudo de avaliação anterior.

Observando o procedimento de análise adotado por cada estudante para responder as questões, verificou-se que:

- O uso de coordenações é mais útil para a análise de conjuntos de dados complexos, já que nesse estudo de avaliação algumas das questões do experimento foram respondidas pelos sujeitos analisando a representação visual de apenas uma técnica de visualização.
- Para análises mais complexas do conjunto de dados, em que era necessário observar mais de uma visualização, todos os estudantes usaram algum mecanismo de interação coordenado.
- Coordenações foram úteis, principalmente, para evitar interações repetidas em múltiplas visualizações (economizando tempo), localizar um mesmo item de dado nas diferentes representações e identificar padrões não evidentes.
- Cada sujeito utilizou apenas três interações coordenadas: filtragem, mapeamento de valores para cor e seleção.

7 ESTUDO DE CASO

Esse capítulo descreve um estudo de caso em que a aplicação de visualização desenvolvida foi utilizada por especialistas de um domínio específico para a exploração de seus dados multidimensionais. O objetivo principal desse estudo foi verificar se as técnicas de visualização implementadas na aplicação, em conjunto com os mecanismos de interação associados a elas, permitem a extração de informações relevantes e significativas quando utilizadas para a representação de dados multidimensionais de um domínio específico. Como se sabe, as técnicas Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots* são voltadas para a visualização de dados multidimensionais em geral e, por esta razão, torna-se também importante verificar sua utilização por usuários “reais”, em um domínio determinado.

A fim de reduzir a complexidade no processo de exploração de dados e simplificar o aprendizado do usuário especialista na manipulação da aplicação desenvolvida, não foi habilitado, nesse estudo, nenhum tipo de coordenação entre as três técnicas de visualização.

7.1 Descrição e Objetivos do Domínio: Morfometria Craniana

A morfometria consiste na obtenção de informações a respeito da forma de determinadas estruturas através de suas medidas. Os biólogos frequentemente utilizam medidas morfométricas para distinguir espécies animais ou vegetais. Para esse estudo de caso, a aplicação desenvolvida foi utilizada para a visualização de dados morfométricos cranianos do roedor conhecido popularmente como Tuco-tuco, a partir de exemplares coletados por biólogos do departamento de Genética da Universidade Federal do Rio Grande do Sul. O projeto de pesquisa destes biólogos tem como objetivos:

- Estabelecer a distribuição geográfica de roedores desse gênero (*Ctenomys*) no Estado do Rio Grande do Sul;
- Avaliar dimorfismo sexual e variações cromossômicas e geográficas e sua relação com a morfometria craniana do animal;
- Propor uma filogenia (história genealógica das espécies) baseada também na morfometria craniana;
- Analisar a influência de barreiras geográficas na evolução do gênero no Rio Grande do Sul.

Nesse contexto, os biólogos que participaram do presente estudo de caso, utilizando as visualizações da aplicação desenvolvida para a exploração dos dados morfométricos,

separados por local da coleta e sexo, esperavam obter as seguintes informações a partir da representação destes dados:

- A identificação de crânios com medidas fora do padrão (possíveis erros);
- A identificação de variações entre os padrões de medidas cranianas de diferentes localidades (variações geográficas);
- A distinção entre medidas de animais de sexo diferente (dimorfismo sexual);
- A identificação de variações nas medidas de animais com número de cromossomos diferente.

A ferramenta de software utilizada atualmente pelos especialistas desse domínio para a verificação das informações acima é denominada NCSS (*Number Cruncher Statistical System*), e o método aplicado é a Análise de Componentes Principais (PCA).

7.2 Conjunto de Dados

Para esse estudo de caso, apenas um conjunto de dados do domínio morfometria animal foi utilizado. Este conjunto contém dados de 157 animais de uma espécie desse gênero e apresenta como atributos 17 variáveis numéricas, 16 destas correspondentes a medidas do crânio destes animais (realizadas como mostra a figura 7.1) e uma correspondente ao número de cromossomos dos mesmos, e três variáveis categóricas: *código* (valor diferente para cada animal), *local* (local da coleta da amostra) e *sexo*. Isto totaliza 20 atributos de dados.

As amostras presentes nesse conjunto de dados foram obtidas a partir de 10 locais de coleta diferentes e todas elas deveriam representar animais adultos. No entanto, há grandes possibilidades de que dados de animais “jovens” estejam incluídos na amostragem acidentalmente pelos biólogos e, caso isto ocorra, estas amostras devem ser identificadas e retiradas do conjunto de dados, pois podem levar a conclusões errôneas na caracterização do exemplar típico da espécie.

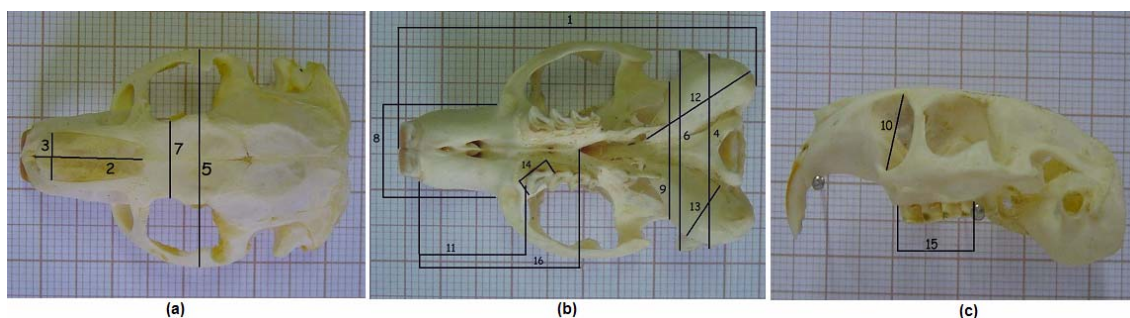


Figura 7.1: Indicação das dezesseis medidas extraídas do crânio de um tuco-tuco para a análise morfométrica tradicional. (a) Vista dorsal, (b) vista ventral e (c) vista lateral.

7.3 Procedimento

Esse estudo de caso foi realizado de maneira informal e contou com a participação de dois especialistas em morfometria craniana. Ambos os sujeitos são alunos de doutorado da área de Genética, nesta universidade.

Os dois sujeitos não conheciam as técnicas de visualização a serem avaliadas (Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots*), mas possuíam algum conhecimento básico de visualização de informações, adquirido através do uso da ferramenta de software utilizada para a análise de seus dados. Assim, eles receberam um treinamento de aproximadamente uma hora com relação ao uso das três técnicas avaliadas. Nesse período, eles receberam explicações gerais sobre a forma como cada técnica organiza o layout dos itens de dados em sua representação visual e sobre o efeito das possibilidades de interação disponibilizadas. Não foi comentada a possibilidade de coordenar visualizações.

Após o treinamento, os sujeitos puderam utilizar a aplicação desenvolvida livremente por três dias. Não foram estabelecidas questões experimentais para essa avaliação. Ao invés disto, os sujeitos foram orientados a explorarem o máximo possível as representações visuais do conjunto de dados descrito na seção anterior, a fim de encontrarem informações relevantes e de interesse para a análise morfométrica (como as informações mencionadas na seção 7.1). Durante esse período, os sujeitos anotaram todas as informações relevantes obtidas com o uso da aplicação desenvolvida, as visualizações que foram úteis para a obtenção dessas informações e o procedimento de interação seguido. Além disso, eles anotaram também sugestões e observações sobre a interface gráfica da aplicação.

Ao final do terceiro dia, conversou-se novamente com os sujeitos para a verificação das informações anotadas, inclusão de informações que estivessem faltando e para ouvir pessoalmente as conclusões subjetivas sobre a aplicação desenvolvida e sua aplicabilidade para a exploração de dados morfométricos.

7.4 Resultados

No final da seção 7.1 foram citadas as informações que os especialistas desse estudo esperavam obter com a exploração de seus dados morfométricos na aplicação desenvolvida. Nessa seção, serão apresentadas as informações que eles conseguiram realmente obter ao final do estudo e as técnicas de visualização e interação que possibilitaram isto.

7.4.1 Identificação de Crânios com Medidas Fora do Padrão

Os usuários especialistas destacaram em suas avaliações que a visualização Coordenadas Paralelas foi muito útil para a rápida identificação de crânios com medidas fora do padrão e/ou medidas que apresentam diferenças significativas entre os indivíduos. Estas ocorrências, segundo eles, podem indicar erros na coleta das medidas e, por esta razão, a identificação das mesmas é muito importante.

A visualização do conjunto total de indivíduos contidos nos dados analisados nesse estudo já permite a identificação de crânios fora do padrão, conforme mostra a figura 7.2, e medidas que apresentam diferenças significativas entre indivíduos. A possibilidade de destacar na representação visual um determinado item e, paralelamente, observar seus valores de atributo na tabela do painel de controle facilita ainda mais a observação de um determinado item em relação ao todo, segundo os usuários especialistas. Por exemplo, o item destacado na visualização da figura 7.2 foi identificado pelos mesmos como sendo um indivíduo “jovem”, devido às medidas

pequenas apresentadas. Este, então, deveria ser retirado do conjunto de dados, já que apenas indivíduos adultos devem compor a amostragem.

As visualizações Radviz e Matriz de *Scatter Plots* não chegaram a ser exploradas pelos usuários nessa tarefa.

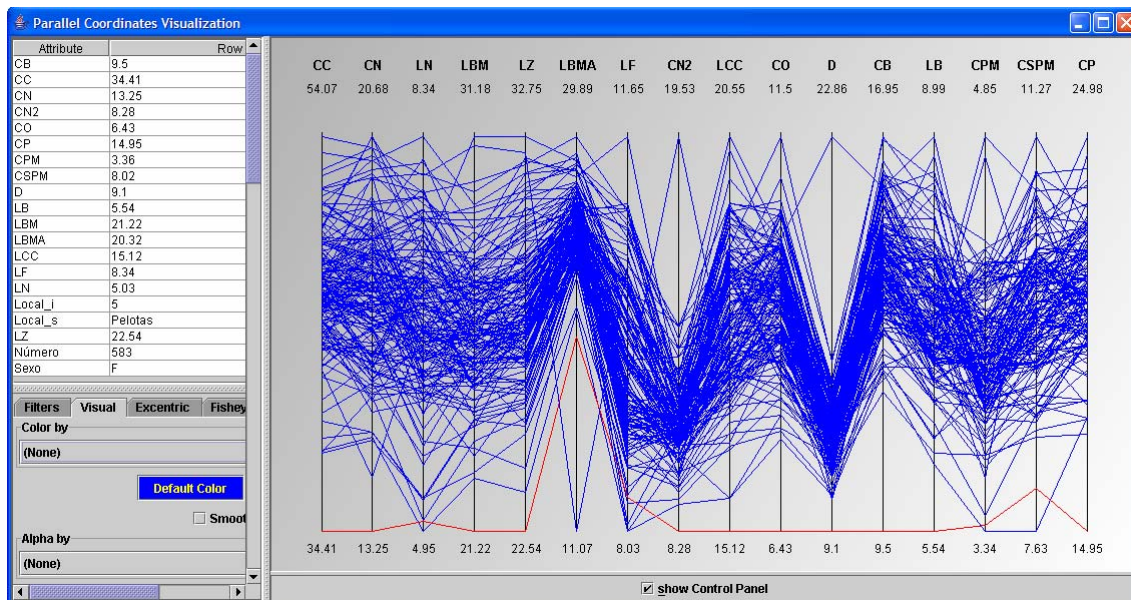


Figura 7.2: Identificação de um animal com medidas fora do padrão (item destacado)

7.4.2 Identificação de Variações nos Padrões de Medidas de Diferentes Localidades

Um dos objetivos da análise morfométrica de crânios é verificar se existem variações nas medidas de crânios em função da localização geográfica das amostras. Para tentar identificar estas variações entre diferentes localidades (o conjunto de dados analisado contém 10 localidades), os especialistas do estudo de caso inicialmente retiraram o atributo *cromossomos* da representação visual e mapearam o conteúdo do atributo *local* (categórico) para cor, na visualização Coordenadas Paralelas. Com isto, os itens de dados representados foram divididos em 10 cores diferentes, cada cor representando um local de coleta. No entanto, isto não foi suficiente para obter as informações de variações desejadas, devido à grande sobreposição e cruzamento de linhas, que pode ser observada na figura 7.3.

Para melhorar a visualização, os especialistas aplicaram diversas combinações de filtros por *local*, ou seja, variaram o número de locais de coleta visualizados simultaneamente, bem como também os locais propriamente ditos. Com isto, eles conseguiram obter informações de variações interessantes: as amostras coletadas em Pelotas (cor verde claro), por exemplo, apresentam um padrão de medidas um pouco mais alto, em geral, que as amostras coletadas em outros locais; as amostras coletadas na Estação Ecológica do Taim (em cor vermelha) apresentam o padrão de medidas mais baixo entre todos os locais analisados; e as amostras coletadas nos demais locais apresentam um padrão semelhante de medidas, intermediário entre Taim e Pelotas. A visualização Matriz de *Scatter Plots*, com o atributo *local* mapeado para cor também, foi observada em paralelo com Coordenadas Paralelas e ajudou a validar as informações de variação mencionadas acima.

As figuras 7.4 e 7.5 mostram apenas os itens de dados que representam amostras coletadas em Pelotas (cor verde), Bagé (cor azul) e Taim (com vermelha), utilizando as técnicas Coordenadas Paralelas e Matriz de *Scatter Plots*, respectivamente. Claramente, é possível perceber três padrões distintos de medidas: o padrão mais alto representa Pelotas, o intermediário representa Bagé e o mais baixo representa Taim. Os demais locais de coleta, não representados nessas figuras, seguem aproximadamente o mesmo padrão de medidas de Bagé.

7.4.3 Identificação de Variações nas Medidas de Animais com Número de Cromossomos Diferente

A análise morfométrica tem como objetivo também verificar se ocorrem variações nas medidas de animais que possuem um número diferente de cromossomos. No caso do conjunto de dados analisado nesse estudo, há apenas 3 numerações diferentes de cromossomos e estas possuem relação direta com os locais de coleta, ou seja, os indivíduos de Taim e Rio Grande têm 46 cromossomos, os de Alegrete têm 40 e os indivíduos dos demais locais de coleta têm 44 cromossomos. Assim, para a análise de variações nas medidas de crânios em função do número de cromossomos, o procedimento seguido é o mesmo utilizado para a análise de variações em decorrência do local de coleta (descrito acima) e os resultados também são semelhantes. Portanto, os biólogos puderam confirmar variações nas medidas de crânios relacionadas com o número de cromossomos das amostras.

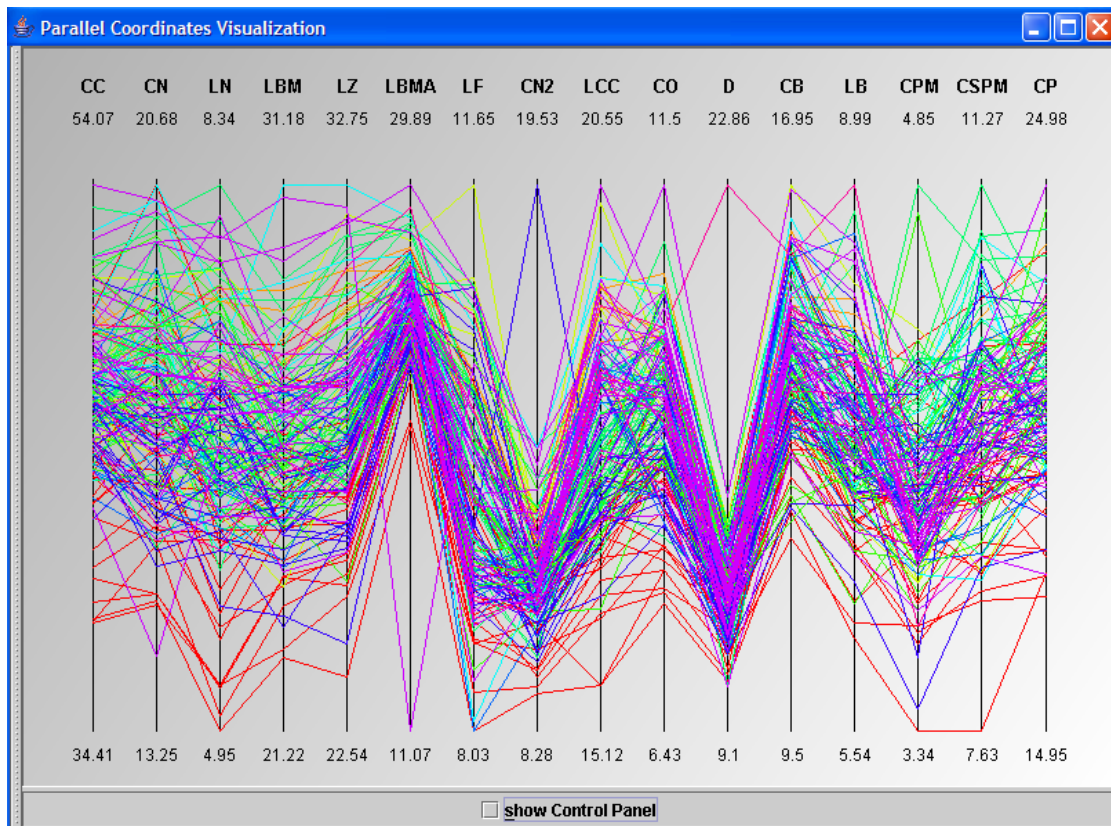


Figura 7.3: Representação do conjunto inteiro de dados morfométricos com locais de coleta mapeados para cor

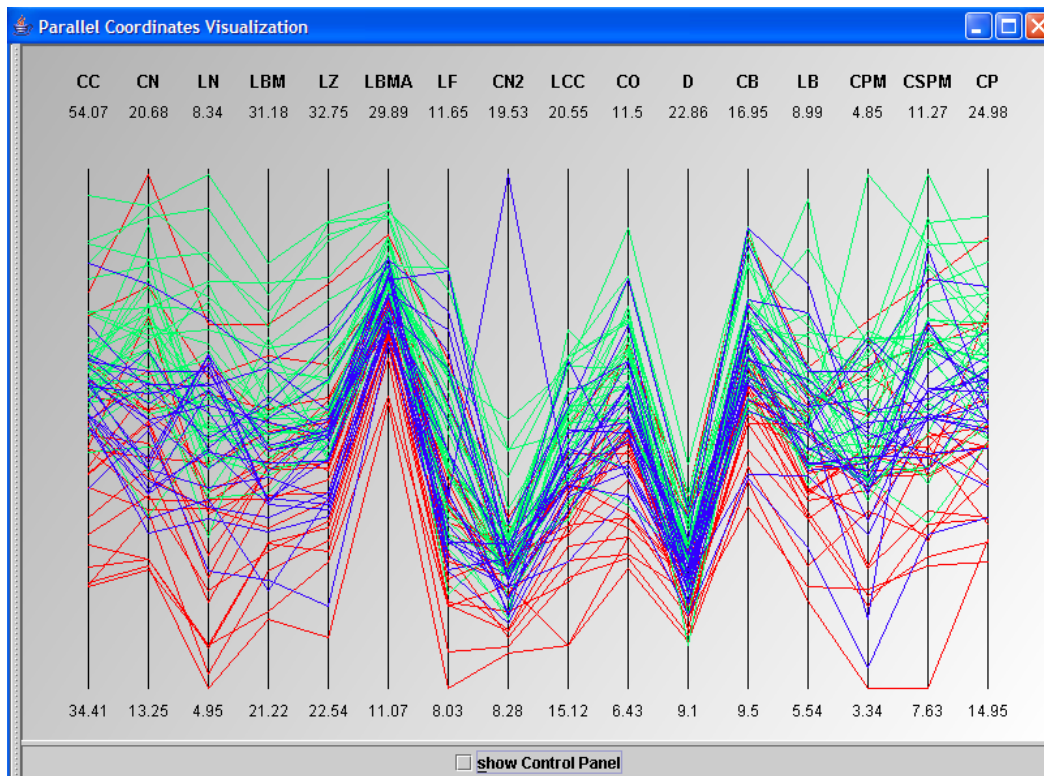


Figura 7.4: Representação em Coordenadas Paralelas das amostras morfométricas coletadas em Pelotas (cor verde), Bagé (cor azul) e Taim (cor vermelha)

7.4.4 Verificação de Variações nas Medidas de Animais de Sexo Diferente

Um outro objetivo ainda da análise morfométrica é avaliar o dimorfismo sexual de uma espécie, ou seja, comparar diferenças nas formas de machos e fêmeas. Os especialistas desse estudo de caso consideraram a visualização Coordenadas Paralelas muito útil e rápida para a avaliação do dimorfismo sexual, permitindo, inclusive, acompanhar quais medidas apresentam maiores variações.

Para esta avaliação, os usuários especialistas basearam-se, principalmente, na representação da técnica Coordenadas Paralelas com o atributo *sexo* mapeado para cor, conforme mostra a figura 7.6. O mecanismo de interação *Sort by* foi utilizado também para mudar a ordem de sobreposição das linhas na visualização mostrada na figura. Nesta última, as linhas que representam machos (cor rosa) sobrepõem as linhas que representam fêmeas (cor preta), mas é útil inverter essa ordem a fim de visualizar melhor a relação entre os dois grupos.

Além da representação da figura 7.6, os especialistas observaram também, separadamente, as representações de machos e fêmeas, utilizando para isto o filtro para o atributo *sexo*. Eles observaram também as medidas de machos e fêmeas de cada local de coleta, isoladamente. Apenas a técnica Coordenadas Paralelas foi observada para essa tarefa.

Como resultado da análise de dimorfismo sexual, os especialistas concluíram que os machos apresentam, em geral, medidas um pouco maiores que as fêmeas da espécie. No entanto, para algumas medidas (como *D* e *LB*), os dois grupos apresentam padrões praticamente iguais e, para outras medidas (como *CC*, *CN*, *LN*, *LBM* e *LZ*), há variações bem evidentes.



Figura 7.5: Representação em Matriz de *Scatter Plots* das amostras morfométricas coletadas em Pelotas (cor verde), Bagé (cor azul) e Taim (cor vermelha)

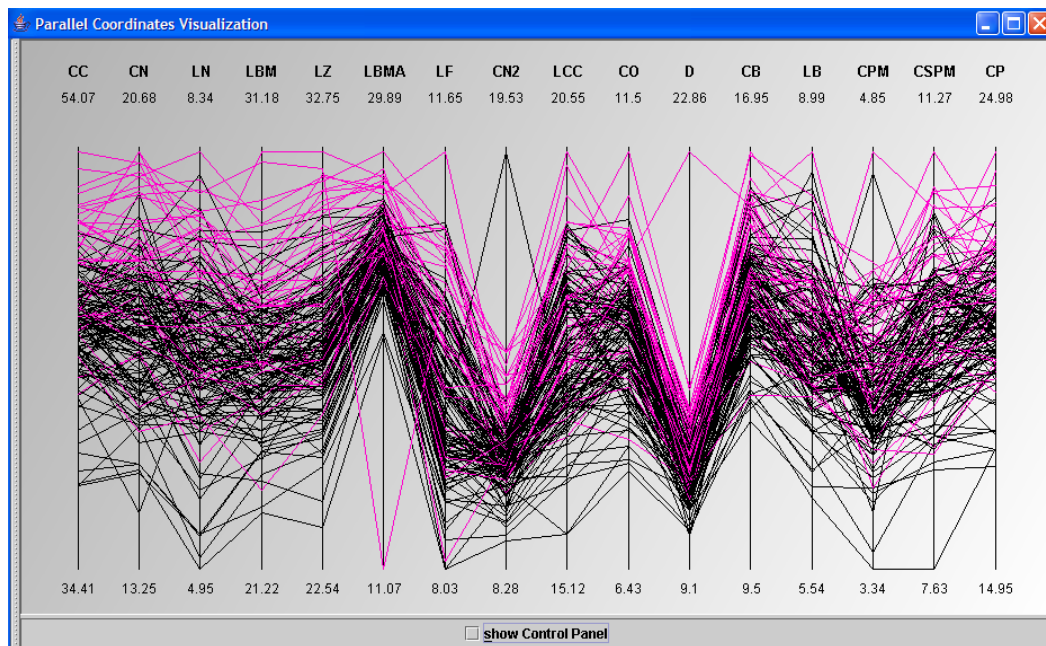


Figura 7.6: Representação do conjunto inteiro de dados morfométricos com machos mapeados para a cor rosa e fêmeas mapeadas para a cor preta

7.5 Comparação: Aplicação Desenvolvida x Aplicação do Domínio

A ferramenta de software utilizada atualmente pelos especialistas desse estudo de caso para a análise morfométrica é a NCSS (*Number Cruncher Statistical System*), um sistema comercial. Com esta ferramenta, eles realizam análises estatísticas dos dados gerados, utilizando Função de Análise Canônica Discriminante (representação de *escores* calculados pelo sistema) e Análise dos Componentes Principais (representação apenas dos componentes mais significativos do conjunto de dados, também determinados pelo sistema).

As únicas representações visuais geradas pelos especialistas com NCSS são *scatter plots* e *box plots*, ambas representações 2D. Como não é possível, com estas visualizações, observar simultaneamente todas as medidas das amostras coletadas, eles utilizam os dois eixos disponíveis do gráfico 2D para representar apenas os componentes principais dos dados ou atributos que resumem o conjunto inteiro de medidas de uma amostra. No caso de visualizações *scatter plots*, a cor e a forma das figuras gráficas são utilizadas ainda para representar outros atributos dos dados.

Para uma análise mais segura dos dados com NCSS, é necessário criar vários gráficos 2D, variando os atributos que são mapeados para seus eixos. A análise do dimorfismo sexual, por exemplo, é feita através de *scatter plots* e *box plots* que representam em seus eixos os componentes principais dos dados. As figuras 7.7 e 7.8 mostram alguns dos gráficos gerados com NCSS, para a análise do dimorfismo sexual e de variações entre localidades, respectivamente. Para a primeira análise, são representados e analisados apenas os componentes principais dos dados. Já para a segunda análise, além dos componentes principais, é feita também a representação e observação de discriminantes (*escores*).

O que é importante notar a respeito destes gráficos é que eles fornecem apenas uma visão geral dos dados, ou seja, representam informações estatísticas que resumem mais de uma medida em um único valor. Com isto, não é possível obter informações detalhadas acerca dos dados, como por exemplo, os valores precisos das medidas de um determinado crânio ou a identificação das amostras com medidas fora do padrão. Além da desvantagem de ser necessária a criação e observação de mais de um gráfico, estes não oferecem ainda possibilidades de interação ao usuário.

A aplicação proposta nesse trabalho, por outro lado, fornece principalmente visões detalhadas dos dados representados, permitindo a obtenção das informações que não podem ser obtidas com a ferramenta NCSS. Tanto amostras como medidas individuais podem ser detalhadamente acompanhadas. Informações gerais acerca dos dados também podem ser obtidas, de maneira diferenciada, porém com menor precisão quando comparado ao sistema NCSS. Além disso, a aplicação desenvolvida apresenta também como vantagem a disponibilidade de gráficos capazes de representar todos os atributos (medidas) e amostras do conjunto de dados simultaneamente, permitindo ainda que o usuário interaja de diversas formas com estes gráficos.

Em suma, os especialistas desse estudo de caso afirmaram que a aplicação avaliada é complementar ao sistema NCSS e muito útil para análises prévias dos dados morfométricos. As informações detalhadas que eles não conseguiam obter com o sistema NCSS foram fornecidas pela aplicação descrita nesse trabalho, graças às técnicas de visualização multidimensionais e seus recursos de interação.

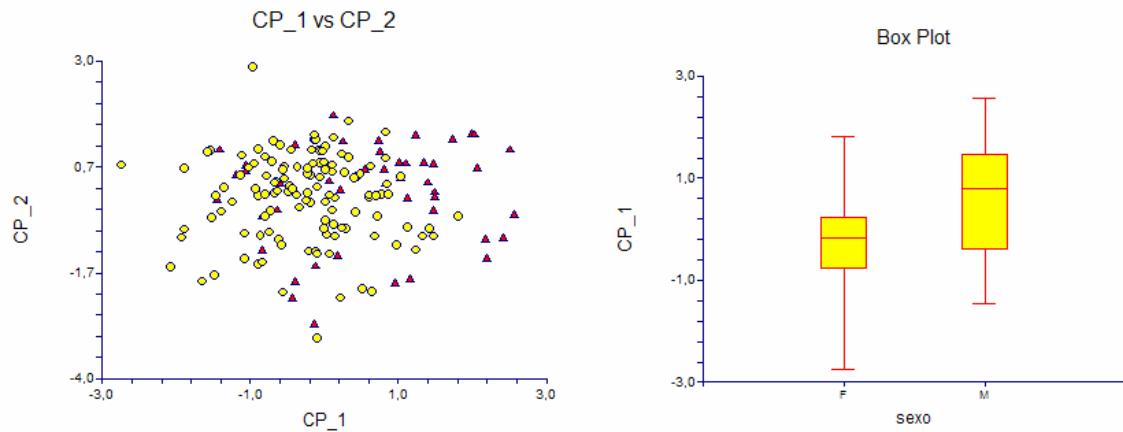


Figura 7.7: Gráficos gerados com o sistema NCSS, *scatter plot* (esquerda) e *box plot* (direita), utilizados na análise do dimorfismo sexual.

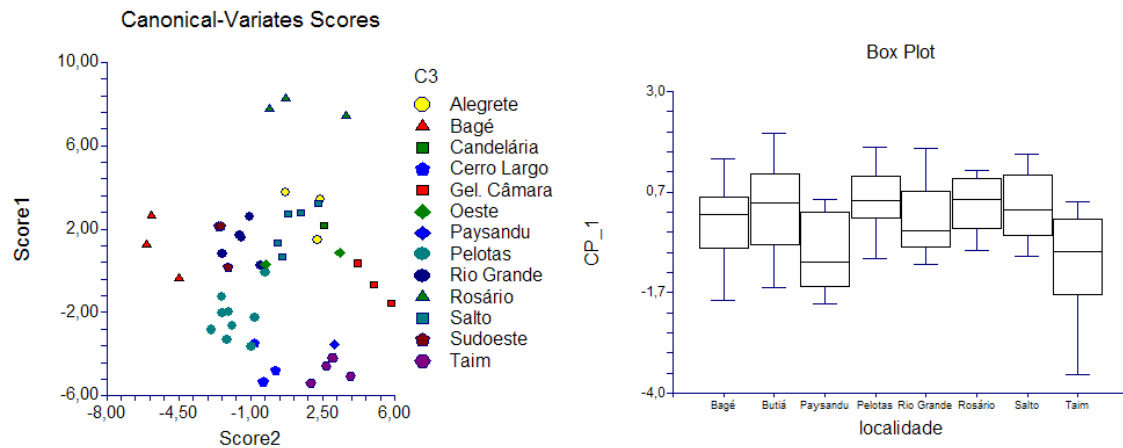


Figura 7.8: Gráficos gerados com o sistema NCSS, *scatter plot* (esquerda) e *box plot* (direita), utilizados na análise de variações por local de coleta.

7.6 Conclusões

A técnica de visualização multidimensional que melhor auxiliou os especialistas desse estudo na obtenção das informações desejadas foi Coordenadas Paralelas. Os especialistas consideraram o mapeamento desta técnica bastante claro e o único capaz de permitir deduções precisas e detalhadas a respeito dos dados, que era o seu principal objetivo.

Em relação à técnica Radviz, os especialistas consideraram seu mapeamento complicado e de difícil compreensão. Além disso, os dados analisados nesse estudo de caso não apresentam características adequadas para a representação com essa técnica, pois todos os itens de dados possuem valores com pesos semelhantes, o que faz com que estes sejam desenhados todos muito próximos, no centro do círculo. Por estas razões, a técnica Radviz não contribuiu para a análise realizada.

A técnica Matriz de *Scatter Plots*, por outro lado, foi observada paralelamente à técnica Coordenadas Paralelas e ajudou a validar informações extraídas a partir desta última, como as variações nas medidas coletadas em diferentes localidades. Além disso, a correlação entre atributos, representada na Matriz, demonstra claramente que todas as

medidas de um crânio são alométricas, ou seja, quando uma delas aumenta, todas as demais também aumentam (ver figura 7.5).

Além das visualizações, os recursos de interação também foram imprescindíveis para a análise realizada. A obtenção de muitas das informações mencionadas nos resultados desse estudo só foi possível devido à disponibilidade de variados recursos interativos. Considerando todas as análises realizadas durante esse estudo de caso, os mecanismos de interação utilizados pelos especialistas foram: *Selection* (seleção individual de itens), *Filtering* (filtragem principalmente por local e sexo), *Color by* (mapeamento de local e sexo para cor), *Sort by* (alteração na ordem de desenho dos objetos de dados) e *Attributes Manipulation* (remoção do atributo cromossomos das visualizações).

Assim, por meio das visualizações multidimensionais e seus recursos de interação, todas as informações que os especialistas esperavam obter com o uso da aplicação avaliada foram realmente encontradas.

Finalmente, em relação à aplicação de visualização como um todo, os especialistas do estudo de caso consideraram-na muito útil e ágil para a análise morfométrica de crânios, principalmente para análises prévias, e descreveram sua interface como simples e de fácil manipulação.

8 CONCLUSÕES

8.1 Avaliação Final

Esse trabalho apresentou um conjunto de coordenações que podem ser aplicadas entre visualizações multidimensionais, e demonstrou a aplicabilidade destas em uma nova aplicação de múltiplas visualizações coordenadas, específica para a representação de dados multidimensionais. Esta aplicação foi construída a partir do toolkit de visualização InfoVis. Incorporou-se ao conjunto de visualizações multidimensionais da mesma aquelas já implementadas pelo toolkit (Coordenadas Paralelas) e acrescentou-se outras (Radviz e Matriz de *Scatter Plots*). Os recursos de interação oferecidos pelo toolkit foram estendidos para uso coordenado e alguns outros recursos adicionados.

A possibilidade de coordenar interações através de múltiplas visualizações é uma característica muito útil para a exploração de conjuntos de dados, pois como se sabe, as técnicas de visualização existentes possuem pontos fortes e fracos diferentes entre si. Coordenações são especialmente úteis quando se dispõe de técnicas de visualização complementares, ou seja, que auxiliam na percepção de informações diferentes.

Alguns pesquisadores, como Keim (1996, 1997), Hoffman (1999) e Grinstein et al. (2000), já compararam as representações visuais das técnicas Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots* em seus estudos de avaliação. Os mesmos consideraram diferentes critérios (relacionados aos dados, tarefas e visualização) e, os resultados obtidos mostram o grau de adequação das técnicas avaliadas para cada critério.

A tabela 8.1 mostra uma comparação das técnicas Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots*, segundo os principais critérios considerados nos estudos de avaliação mencionados acima. A classificação mostrada na tabela foi obtida a partir da análise dos resultados destes estudos, encontrados na literatura, e dos experimentos com usuários realizados nesse trabalho. Observando essa classificação, pode-se verificar que as visualizações avaliadas possuem algumas características complementares, que tornam a coordenação entre elas um recurso muito interessante.

A técnica Coordenadas Paralelas é muito útil para a percepção de diversas informações acerca dos dados representados e há poucos critérios para os quais outras técnicas são mais adequadas. Coordenadas Paralelas destaca-se das demais por permitir a melhor percepção dos valores aproximados de itens de dados e da distribuição destes entre múltiplas dimensões. Entre as poucas fraquezas dessa técnica estão o número de

itens de dados que podem ser mostrados efetivamente e o alto índice de sobreposição visual destes. Quando exibindo conjuntos de dados com muitos itens, a ocupação do espaço de tela torna-se muito densa e o grande número de cruzamentos e sobreposições de linhas horizontais causam confusão visual, que impede a compreensão dos dados representados. Devido a isto, o número máximo de itens de dados que podem ser exibidos por Coordenadas Paralelas fica limitado em torno de mil itens (Keim, 1996). No entanto, mesmo para conjuntos de dados pequenos, pode haver sobreposição de itens e confusão visual, o que prejudica a percepção de *clusters* de dados.

A técnica Radviz pode representar um grande número de dimensões e itens de dados simultaneamente, sem provocar confusão visual. A mesma é muito boa para a identificação de *outliers* e *clusters* de dados, mas, por outro lado, sua representação dos dados também pode ser prejudicada pela sobreposição de itens de dados. Este problema pode ser contornado, contudo, através de mecanismos de interação que permitam alterar a disposição dos atributos no layout da técnica. A maior deficiência dessa técnica é, sem dúvida, o fato de não ser possível estimar os valores dos itens de dados. Além disso, os estudos de avaliação relatados nesse trabalho mostram claramente que a técnica Radviz tem um aprendizado mais difícil.

A Matriz de *Scatter Plots* é, entre as três técnicas avaliadas, a de mais fácil aprendizagem. No entanto, ela possui limitações quanto ao número de dimensões e itens de dados que podem ser representados. Para visualizar adequadamente características de um conjunto de dados com mais de 10 dimensões, é necessária a utilização de recursos interativos. Mas, mesmo assim, não será possível, nestes casos, observar todas as células da matriz simultaneamente. A visualização dessa técnica é muito útil para a percepção de *outliers*, *clusters* e correlações entre dimensões dos dados.

Tabela 8.1: Comparação das técnicas Coordenadas Paralelas, Radviz e Matriz de *Scatter Plots* segundo critérios encontrados na literatura. Os símbolos ++, +, 0, -, -- correspondem à excelente, bom, médio, pobre e muito pobre (simbologia proposta por KEIM, 1996).

	<i>Coordenadas Paralelas</i>	<i>Radviz</i>	<i>Matriz de Scatter Plots</i>
Número de dimensões	++	++	0
Número de itens de dados	0	++	0
Curva de aprendizagem	+	-	++
Percepção de valores de dados	++	--	+
Percepção de regras, modelos	+	+	0
Representação de <i>Outliers</i>	++	++	++
Representação de <i>Clusters</i>	0	++	++
Sobreposição visual	--	0	0

As técnicas Radviz e Coordenadas Paralelas mostram visões bem diferentes para um mesmo conjunto de dados. Enquanto a primeira enfatiza a representação de características gerais, como a distribuição dos dados em relação ao todo e a formação de *clusters*, a segunda técnica enfatiza a representação de informações mais detalhadas,

como os valores de dimensões e seus intervalos. Essas técnicas possuem pontos fortes diferentes, sendo que uma complementa as fraquezas da outra. De um lado, Radviz oferece pouca confusão visual e representação clara de *clusters* de dados, mas ela não permite a visualização de informações quantitativas; por outro lado, Coordenadas Paralelas mostra claramente valores de itens de dados e intervalos de atributos, mas apresenta grande confusão visual (até mesmo para conjuntos de dados pequenos).

Um outro enfoque de representação dos dados é fornecido ainda pela técnica Matriz de *Scatter Plots*, que possibilita que determinadas análises dos dados sejam mais efetivas utilizando essa técnica do que as anteriores. Por exemplo, a *Matriz de Scatter Plots* oferece a representação mais clara e fácil para a identificação de correlações entre quaisquer dimensões dos dados.

Com o uso de coordenações, as possibilidades de colaboração entre essas técnicas aumentam consideravelmente. Por exemplo, um *cluster* de dados pode ser facilmente identificado na técnica Radviz. Com a coordenação *brushing-and-linking* habilitada, selecionando, nessa técnica, os itens que compõem o *cluster* irá destacar os mesmos nas outras duas visualizações. Com isto, pode-se então observar a representação de Coordenadas Paralelas e verificar os valores destes itens para cada atributo. Bertini et al. (2005) realizaram estudos de avaliação que demonstram outros ganhos com o uso coordenado das técnicas Radviz e Coordenadas Paralelas.

Enfim, além de ser específica para dados multidimensionais, a aplicação relatada nesse trabalho apresenta ainda várias vantagens para a exploração deste tipo de dados em relação a outros sistemas:

- *Conta com um conjunto variado de coordenações e recursos de interação:* reúne coordenações e interações utilizadas por diversos sistemas de visualização;
- *Oferece visualizações complementares para a construção de cenários:* conforme foi discutido nessa seção, as visualizações multidimensionais suportadas pela aplicação são complementares, pois cada uma delas auxilia na percepção de informações diferentes;
- *É altamente flexível:* permite ao usuário escolher os dados a serem representados, as visualizações que formarão um cenário de exploração e, ainda, quais coordenações estarão ativas e entre quais visualizações;
- *É construída sobre um framework de visualização consistente e extensível:* estas características são “herdadas” pela aplicação, que pode facilmente ser estendida para incorporar novas visualizações ou recursos.

A utilidade e eficácia da aplicação desenvolvida foram demonstradas experimentalmente na exploração de dados multidimensionais de um domínio específico.

8.2 Trabalhos Futuros

Há vários aspectos da aplicação relatada nesse trabalho que precisam ou poderiam ser melhorados. Entre estes, pode-se destacar:

- *Interface gráfica:*

O resultado do experimento de avaliação de usabilidade realizado com a aplicação em questão (relatado no capítulo 6) apontou diversas falhas e problemas com a interface gráfica inicial que se tinha naquele momento. Por questões de tempo, apenas os problemas mais graves foram resolvidos após o experimento, restando ainda muitos melhoramentos para serem realizados. Por exemplo, devem ser disponibilizadas opções para abrir/salvar cenários de exploração e recursos alternativos para a interação do usuário.

- *Visualizações:*

A implementação da técnica Matriz de *Scatter Plots* é, dentre as técnicas de visualização já suportadas, aquela que mais necessita de melhoramentos. Para aumentar a efetividade dessa técnica na representação de conjuntos de dados com grande número de dimensões ou itens de dados, interações como o zoom devem ser habilitadas para uso nas células da matriz.

- *Coordenações:*

No contexto das técnicas de coordenação, poderiam ser disponibilizadas diferentes formas de implementação para uma mesma técnica. Por exemplo, a filtragem de itens de dados poderia ser realizada através de um outro *widget* gráfico que não uma *slider bar*. Da mesma forma, o *brushing-and-linking* poderia ser ativado também apenas pela passagem do mouse sobre um item de dado, sem necessitar da seleção.

Além disso, para aumentar a efetividade da aplicação na exploração dos mais diversos conjuntos de dados multidimensionais, pretende-se, futuramente, ampliar o conjunto de visualizações suportadas (acrescentando técnicas iconográficas e orientadas a pixel) e de interações.

Finalmente, vale ressaltar que a aplicação aqui desenvolvida vem sendo (e continuará a ser) usada em trabalho de definição de uma taxonomia de tarefas de visualização (VALIATI; PIMENTA; FREITAS, 2006) para permitir a definição de cenários de avaliação de usabilidade de técnicas de visualização. Espera-se que a aplicação venha a ser um dos componentes de um sistema de avaliação de usabilidade de técnicas de visualização.

REFERÊNCIAS

AHLBERG, C.; WILLIAMSON, C.; SHNEIDERMAN, B. Dynamic Queries for Information Exploration: An Implementation and Evaluation. In: HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1992. **Proceedings...** Monterey, CA: ACM/SIGCHI, 1992.

AHLBERG, C.; SHNEIDERMAN, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In: HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1994. **Proceedings...** [S.l.:s.n.], 1994.

AHLBERG, C.; WISTRAND, E. IVEE: An Information Visualization and Exploration Environment. In: IEEE INFORMATION VISUALIZATION, 1995. **Proceedings...** [S.l.:s.n.], 1995.

ALPERN, B.; CARTER, L. Hyperbox. In: IEEE VISUALIZATION, 1991. **Proceedings...** San Diego, California: [s.n.], 1991.

ASIMOV, D. The Grand Tour: A Tool for Viewing Multidimensional Data. **SIAM Journal of Science & Stat. Comp.**, [S.l.], v. 6, p. 128-143, 1985.

BALDONADO, M. Q.; WOODRUFF, W. A.; KUCHINSKY, A. Guidelines for Using Multiple Views in Information Visualization. In: ADVANCED VISUAL INTERFACES, AVI, 2000. **Proceedings...** [S.l.:s.n.], 2000.

BALL, T.; EICK, S. Software Visualization in the Large. **IEEE Computer**, [S.l.], v. 29, n. 4, p. 33-43, Apr. 1996.

BASTIEN, C.; SCAPIN, D. **Ergonomic Criteria for the Evaluation of Human-Computer Interfaces**. [S.l.]: INRIA, 1993. (RT n. 156).

BERTIN, J. **Semiology of Graphics**. Madison, WI: The University of Wisconsin Press, 1983.

BERTINI, E.; AQUILA, L. D.; SANTUCCI, G. SpringView: Cooperation of Radviz and Parallel Coordinates for View Optimization and Clutter Reduction. In: INTERNATIONAL CONFERENCE ON COORDINATED & MULTIPLE VIEWS IN EXPLORATORY VISUALIZATION, CMV, 2005. **Proceedings...** [S.l.:s.n.], 2005.

BIER, E. A. et al. Toolglass and Magic Lenses: The See-Through Interface. In: SIGGRAPH, 1993. **Proceedings...** Anaheim, CA: [s.n.], 1993.

- BOUKHELIFA, N.; ROBERTS, J. C.; RODGERS, P. J. A Coordination Model for Exploratory Multi-View Visualization. In: INTERNATIONAL CONFERENCE ON COORDINATED & MULTIPLE VIEWS IN EXPLORATORY VISUALIZATION, CMV, 2003. **Proceedings...** London, England: [s.n.], 2003.
- BRODBECK, D.; GIRARDIN, L. Design Study: Using Multiple Coordinated Views to Analyse Geo-referenced High-dimensional Datasets. In: COORDINATED & MULTIPLE VIEWS IN EXPLORATORY VISUALIZATION, CMV, 2003. **Proceedings...** [S.l.:s.n.], 2003.
- CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN B. **Readings in Information Visualization: Using Vision to Think**. San Francisco, California: Morgan Kaufmann Publishers, 1999.
- CARPENDALE, M. S. T.; MONTAGNESE, C. A. Framework for Unifying Presentation Space. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 2001. **Proceedings...** Orlando, Florida: [s.n.], 2001.
- CHERNOFF, H. The Use of Faces to Represent Points in k-Dimensional Space Graphically. **Journal Amer. Statistical Association**, [S.l.], v. 68, p. 361-368, 1973.
- CHI, E. H. et al. A Spreadsheet Approach to Information Visualization. In: IEEE INFORMATION VISUALIZATION, 1997. **Proceedings...** [S.l.:s.n.], 1997.
- CHIMERA, R. et al. An Exploratory Evaluation of Three Interfaces for Browsing Large Hierarchical Tables of Contents. **ACM Transactions on Information Systems**, New York, v. 12, n. 4, p. 383-406, 1994.
- EICK, S; WILLS, G. High Interaction Graphics. **European Journal of Operations Research**, [S.l.], p. 445-459, 1995.
- EICK, S. G.; KARR, A. F. **Visual Scalability**. USA: National Institute of Statistical Sciences (NISS), 2000. (TR - 106).
- FAYYAD, U.; GRINSTEIN, G. G.; WIERSE, A. **Information Visualization in Data Mining and Knowledge Discovery**. San Francisco: Academic Press, 2002.
- FEKETE, J. D. **The InfoVis Toolkit Manual**. 2002. Disponível em: <<http://www.lri.fr/~fekete/InfovisToolkit/manual.html>>. Acesso em: abr. 2006.
- FEKETE, J. D. **The InfoVis Toolkit**. [S.l.]: INRIA Futurs, 2003. (RR - 4818).
- FEKETE, J. D. The InfoVis Toolkit. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, InfoVis, 2004. **Proceedings...** [S.l.]: IEEE Press, 2004.
- FISHKIN, K.; STONE, M. C. Enhanced Dynamic Queries Via Movable Filters. In: HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1995. **Proceedings...** Denver, CO: [s.n.], 1995.
- FREDRIKSON, A. et al. Temporal, Geographical and Categorical Aggregations Viewed through Coordinated Displays: A Case Study with Highway Incident Data. In: WORKSHOP ON NEW PARADIGMS IN INFORMATION VISUALIZATION AND MANIPULATION, ACM CIKM, 1999. **Proceedings...** [S.l.:s.n.], 1999.

- GAHEGAN, M. et al. GeoVISTA Studio: A Geocomputational Workbench. In: INTERNATIONAL CONFERENCE ON GEOCOMPUTATION, 2000. **Proceedings...** [S.l.:s.n.], 2000.
- GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software.** [S.l.]: Addison-Wesley, 1995.
- GOLDSTEIN, J.; ROTH, S. F. Using Aggregation and Dynamic Queries for Exploring Large Data Sets. In: HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1994. **Proceedings...** [S.l.:s.n.], 1994.
- GRINSTEIN, G. et al. Benchmark Development for the Evaluation of Visualization for Data Mining. In: FAYYAD, U. M.; GRINSTEIN, G. G.; WIERSE, A. (Ed.). **Information Visualization in Data Mining and Knowledge Discovery.** San Francisco: Morgan-Kaufmann Publishers, 2000.
- HOFFMAN, P. E. et al. DNA Visual and Analytic Data Mining. In: IEEE VISUALIZATION, 1997. **Proceedings...** Phoenix, AZ: [s.n.], 1997.
- HOFFMAN, P. E. **Table Visualization: A Formal Model and Its Applications.** 1999. Tese (Doutorado) - Computer Science Dept., University of Massachusetts at Lowell.
- HOFFMAN, P.; GRINSTEIN, G.; PINKNEY, D. **Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations.** New York: ACM Press, 2000.
- INSELBERG, A. The Plane with Parallel Coordinates. **The Visual Computer**, New York, v. 1, p. 69-91, 1985. Special Issue on Computational Geometry.
- INSELBERG, A.; DIMSDALE, B. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. In: IEEE VISUALIZATION, 1990. **Proceedings...** [S.l.:s.n.], 1990.
- JACOBSON, A.; BERKIN, A.; ORTON, M. LinkWinds: Interactive Scientific Data Analysis and Visualization. **Communications of the ACM**, New York, v. 37, n. 4, p. 43-52, Apr. 1994.
- KANDOGAN, E.; SHNEIDERMAN, B. Elastic Windows: Evaluation of Multi-Window Operations. In: ACM CHI, 1997. **Proceedings...** [S.l.:s.n.], 1997.
- KEIM, D. A. Visualization Techniques for Mining Large Database: A Comparison. **Transactions on Knowledge and Data Engineering**, New York, v. 8, n. 6, 1996.
- KEIM, D. A. Visual Database Exploration Techniques. In: INTERNATIONAL CONFERENCE IN KNOWLEDGE DISCOVERY AND DATA MINING, KDD, 1997. Tutorial. [S.l.:s.n.], 1997.
- KEIM, D. A. Information Visualization and Visual Data Mining. **IEEE Transactions on Visualization and Computer Graphics**, [S.l.], v. 7, n. 1, Jan./Mar. 2002.
- KOBSA, A. An Empirical Comparison of Three Commercial Information Visualization Systems. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, InfoVis, 2001. **Proceedings...** San Diego, CA: [s.n.], 2001.

- KOSSLYN, S. M. **Elements of Graph Design**. New York: Freeman and Company, 1994.
- KUMAR, H.; PLAISANT, C.; SHNEIDERMAN, B. Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning. **International Journal of Human Computer Studies**, [S.l.], v. 46, n. 1, p. 103-124, 1997.
- LIVNY, M. et al. DEVise: Integrated Querying and Visual Exploration of Large Datasets. In: ACM SIGMOD, 1997. **Proceedings...** [S.l.:s.n.], 1997.
- LOHNINGER, H.; INSPECT. **A Program System to Visualize and Interpret Chemical Data**. 1994. Disponível em: <<http://qspr03.tuwien.ac.at/lo/>>. Acesso em: mar. 2006.
- MARTIN, A.; WARD, M. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In: VISUALIZATION, 1995. **Proceedings...** [S.l.:s.n.], 1995.
- NORTH, C.; SHNEIDERMAN, B. **A Taxonomy of Multiple-Window Coordination**. [S.l.]: University of Maryland Computer Science Dept., 1997. (Technical Report #CS-TR-3854).
- NORTH, C. **A User Interface for Coordinating Visualizations Based on Relational Schemata: Snap-Together Visualization**. 2000. Tese (Doutorado) - Computer Science Department, University of Maryland.
- NORTH, C.; SHNEIDERMAN, B. **Snap-Together Visualization: Can Users Construct and Operate Coordinated Views?**. 2000. Computer Science Dept, University of Maryland.
- OLIVEIRA, M. C. F.; LEVKOWITZ, H. From Visual Data Exploration to Visual Data Mining: A survey. **IEEE Transactions on Visualization and Computer Graphics**, [S.l.], v. 9, n. 3, July-Sept. 2003.
- PATTISON, T.; PHILLIPS, M. View Coordination Architecture for Information Visualisation. In: AUSTRALIAN SYMPOSIUM ON INFORMATION VISUALISATION, 2001. **Proceedings...** Sydney: [s.n.], 2001.
- PENG, W.; WARD, M. O.; RUNDENSTEINER, E. A. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, InfoVis, 2004. **Proceedings...** [S.l.]: IEEE Press, 2004.
- PILLAT, R. M.; VALIATI, E.; FREITAS, C. M. D. S. Experimental Study on Evaluation of Multidimensional Information Visualization Techniques. In: LATIN AMERICAN CONFERENCE ON HUMAN-COMPUTER INTERACTION, CLIHC, 2005. **Proceedings...** Cuernavaca, México: [s.n.], 2005.
- PILLAT, R. M.; FREITAS, C. M. D. S. Coordinating Views in the InfoVis Toolkit. In: ADVANCED VISUAL INTERFACES, AVI, 2006. **Proceedings...** Veneza, Itália: [s.n.], 2006.
- RAO, R.; CARD, S. K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular

Information. In: HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1994. **Proceedings...** Boston: ACM Press, 1994.

ROBERTS, J. C. Exploratory Visualization with Multiple Linked Views. In: EXPLORING GEOVISUALIZATION, 2004. **Proceedings...** Oxford: Elsevier, 2004.

ROTH, S. et al. Visage: A User Interface Environment for Exploring Information. In: IEEE INFORMATION VISUALIZATION, 1996. **Proceedings...** [S.l.:s.n.], 1996.

RUNDENSTEINER, E. A. et al. XmdvTool: Visual Interactive Data Exploration and Trend Discovery of High-dimensional Data Sets. In: ACM SIGMOD CONFERENCE, 2002. **Proceedings...** Wisconsin: [s.n.], 2002.

SHAW, M.; GARLAN, D. **Software Architecture: Perspectives on an Emerging Discipline.** Upper Sadle River: Prentice Hall, 1996.

SHIMABUKURO, M. H. **Visualizações Temporais em uma Plataforma de Software Extensível e Adaptável.** 2004. Tese (doutorado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Paulo.

SHNEIDERMAN, B. Dynamic Queries for Visual Information Seeking. **IEEE Software**, Los Alamitos, v. 11, n. 6, Nov. 1994.

SPENCE, R. **Information Visualization.** Harlow: Addison-Wesley, 2001.

STOLTE, C.; HANRAHAN, P. Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2001. **Proceedings...** [S.l.:s.n.], 2001.

VALIATI, E. **Avaliação de Técnicas de Visualização de Informações Multidimensionais.** 2004. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

VALIATI, E.; PIMENTA, M.; FREITAS, C. D. S. A Taxonomy of Tasks for Guiding the Evaluation of Multidimensional Visualizations. In: BEYOND TIME AND ERRORS: NOVEL EVALUATION METHODS FOR INFORMATION VISUALIZATION, BELIV; AVI, 2006. **Proceedings...** Veneza: [s.n.], 2006.

VAN WIJK, J.J.; VAN LIERE, R. HyperSlice. In: IEEE VISUALIZATION, 1993. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1993.

WARD, M. Xmdvtool: Integrating Multiple Methods for Visualizing Multivariate Data. In: VISUALIZATION, 1994. **Proceedings...** [S.l.:s.n.], 1994.

WEAVER, C. Building Highly-coordinated Visualizations in Improvise. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2004. **Proceedings...** Austin, TX: [s.n.], 2004.

YANG, J. et al. Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration of High Dimensional Datasets. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2003. **Proceedings...** [S.l.:s.n.], 2003.