

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

Bifocal Tree:
Uma Técnica para Visualização de
Estruturas Hierárquicas

por

RICARDO ANDRADE CAVA

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Profa. Dra. Carla Maria Dal Sasso Freitas
Orientadora

Porto Alegre, novembro de 2002

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Cava, Ricardo Andrade

Bifocal Tree: uma técnica de visualização de estruturas organizadas hierarquicamente / por Ricardo Andrade Cava. - Porto Alegre: PPGC da UFRGS, 2002.

97 f. : il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador: Freitas, Carla Maria Dal Sasso.

1. Visualização de informações 2. Estruturas hierárquicas. 3. Foco + Contexto. I. Freitas, Carla Maria Dal Sasso. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

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

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

Agradecimentos

A minha família, pela compreensão e apoio nos momentos mais difíceis.

A minha orientadora, profa. Carla Freitas, pela confiança em mim depositada e cujo apoio e sugestões foram fundamentais na realização desse trabalho.

Aos alunos e professores da Escola de Informática da UCPEL que participaram dos experimentos realizados.

Aos colegas Sérgio Santi e Paulo Luzzardi, companheiros de jornada e inúmeras viagens de Pelotas a Porto Alegre.

A UCPEL e ao CEFET-RS que possibilitaram a realização desse trabalho.

Sumário

| | |
|---|----|
| Lista de Figuras | 6 |
| Lista de Tabelas | 9 |
| Resumo | 10 |
| Abstract | 11 |
| 1 Introdução | 12 |
| 2 Visualização de estruturas organizadas hierarquicamente | 14 |
| 2.1 Técnicas que utilizam a abordagem do preenchimento do espaço | 16 |
| 2.1.1 TreeMaps..... | 16 |
| 2.1.2 Information Cube..... | 18 |
| 2.1.3 Information Slices..... | 19 |
| 2.1.4 Sunburst e extensões..... | 20 |
| 2.2 Técnicas que utilizam diagramas de nodos e arestas | 21 |
| 2.2.1 Cone Tree..... | 22 |
| 2.2.2 Reconfigurable Disc Tree..... | 24 |
| 2.2.3 Time Tube..... | 26 |
| 2.2.4 Hiperbolic Browser..... | 27 |
| 2.2.5 H3..... | 29 |
| 2.3 Comentários finais | 30 |
| 3 Visualização de informações de contexto e detalhe | 31 |
| 3.1 Abordagem Visão Geral+Detalhe | 31 |
| 3.1.1 Uma Interface Visão Geral+Detalhe para mapas..... | 32 |
| 3.1.2 Life Lines..... | 33 |
| 3.1.3 Microsoft Explorer Browser..... | 34 |
| 3.2 Abordagem Foco+Contexto | 35 |
| 3.2.1 Fisheye Views..... | 37 |
| 3.2.2 Graphical fisheye view..... | 39 |
| 3.2.3 Bifocal Display..... | 42 |
| 3.2.4 Perspective Wall..... | 43 |
| 3.2.5 Flip Zooming..... | 44 |
| 3.3 Informações hierárquicas e visualização de contexto e detalhes | 45 |
| 4 Bifocal Tree | 47 |
| 4.1 Representação visual | 47 |
| 4.2 Mecanismos de interação | 53 |
| 4.3 Implementação | 56 |
| 4.3.1 Layout da área de contexto..... | 57 |
| 4.3.2 Layout da área de detalhe..... | 61 |
| 4.3.3 Efeito fisheye..... | 65 |
| 4.4 O protótipo <i>Bifocal Browser</i> | 66 |
| 5 Avaliação da Bifocal Tree | 69 |
| 5.1 MagniFind | 70 |
| 5.2 Metodologia de avaliação | 71 |
| 5.2.1 Hierarquias..... | 71 |
| 5.2.2 Sessões de teste..... | 72 |

| | |
|---|-----------|
| 5.2.3 Execução das tarefas..... | 72 |
| 5.2.4 Definição da ordem das sessões | 74 |
| 5.3 Descrição e objetivos das tarefas..... | 74 |
| 5.3.1 Busca de um nodo | 74 |
| 5.3.2 Navegação através de um caminho determinado | 75 |
| 5.3.3 Identificação dos relacionamentos..... | 76 |
| 5.3.4 Comparação entre dois trechos da hierarquia..... | 76 |
| 5.3.5 Desorientação do usuário causada pela alteração do layout durante a navegação | 77 |
| 5.3.6 Representação de nodos que não são diretórios | 78 |
| 5.4 Descrição dos sujeitos e das medidas utilizadas..... | 78 |
| 5.5 Resultados obtidos | 79 |
| 5.6 Análise dos resultados | 80 |
| 5.6.1 Busca de um nodo | 80 |
| 5.6.2 Navegação através de um caminho determinado | 80 |
| 5.6.3 Identificação dos relacionamentos pai/filhos | 82 |
| 5.6.4 Comparação entre dois trechos da hierarquia..... | 83 |
| 5.6.5 Desorientação do usuário causada pela alteração do layout durante a navegação | 83 |
| 5.6.6 Representação de nodos que não são diretórios | 84 |
| 5.7 Avaliação subjetiva..... | 85 |
| 5.7.1. Representação visual | 85 |
| 5.7.2. Mecanismos de interação..... | 86 |
| 6 Conclusões | 87 |
| 6.1 Contribuições e avaliação geral..... | 87 |
| 6.2 Trabalhos futuros | 88 |
| Anexo 1 Principais algoritmos..... | 90 |
| Bibliografia | 92 |

| | |
|--|----|
| FIGURA 3.4 – Visualização normal de um programa em C..... | 38 |
| FIGURA 3.5 – Visualização de um programa em C mostrado na figura 3.4, através da fisheye view..... | 39 |
| FIGURA 3.6 – Visualização tradicional..... | 40 |
| FIGURA 3.7 – Visualização através da fisheye de Furnas | 40 |
| FIGURA 3.8 – Visualização através da GFV..... | 41 |
| FIGURA 3.9 – Grafo que representa o metrô da cidade de Paris – visualização tradicional | 41 |
| FIGURA 3.10 – Grafo que representa o metrô da cidade de Paris – Graphical Fisheye View | 42 |
| FIGURA 3.11 – Esquema da técnica Bifocal Display | 43 |
| FIGURA 3.12 – Esquema da técnica Perspective Wall | 43 |
| FIGURA 3.13 – Esquema da técnica Flip Zooming | 44 |
| FIGURA 3.14 – Zoom Browser | 45 |
| FIGURA 3.15 – Visualizador The Digital Variants | 45 |
| FIGURA 4.1 – Bifocal Tree – Uma hierarquia com aproximadamente 370 nodos | 48 |
| FIGURA 4.2 – Setores circulares utilizados para delimitar as áreas de contexto e detalhe..... | 48 |
| FIGURA 4.3 – Tipos de representação de nodos | 49 |
| FIGURA 4.4 – Destaque da última subárvore que ocupou a área de detalhe | 50 |
| FIGURA 4.5 – Porção angular ocupada pelos nodos do primeiro nível proporcional ao número de filhos da raiz da subárvore (nodo CTSND) | 51 |
| FIGURA 4.6 – Porção angular ocupada pelos nodos do primeiro nível proporcional ao número de nodos folhas existentes na subárvore (nodo CTSND)..... | 51 |
| FIGURA 4.7 – (a) Setor da área de detalhe com 315° (b) setor da área de detalhe com a medida angular proporcional ao número de nodos folhas..... | 52 |
| FIGURA 4.8 – Estrutura com o nodo raiz localizado na área de detalhe..... | 52 |
| FIGURA 4.9 – Distribuição dos nodos com distância fixa (a) e com distância variável em relação ao foco (b) | 53 |
| FIGURA 4.10 – Estado da representação antes da seleção do nodo navbar | 54 |
| FIGURA 4.11 – Estado da representação após a seleção do nodo navbar | 54 |
| FIGURA 4.12 – Área de detalhe antes (a) e depois da rotação (b)..... | 55 |
| FIGURA 4.13 – Janela para seleção de nodos irmãos sobrepostos..... | 56 |
| FIGURA 4.14 – Estado da representação antes (a) e depois da operação de poda (b) | 56 |
| FIGURA 4.15 – Hierarquia com o total de 18 nodos..... | 57 |

| | |
|--|----|
| FIGURA 4.16 – Distribuição das subárvores nos setores | 59 |
| FIGURA 4.17 – Cálculo das coordenadas polares do nodo N 4 | 60 |
| FIGURA 4.18 – Ângulo mínimo necessário para exibir os nodos sem sobreposição no primeiro nível | 61 |
| FIGURA 4.19 – Medida angular do setor da área de contexto sem (a) e com a existência de “netos” (b)..... | 62 |
| FIGURA 4.20 – Hierarquia com setor angular inferior a 9° | 62 |
| FIGURA 4.21 – Distribuição dos nodos no primeiro nível através do Subnode-Based Layout..... | 63 |
| FIGURA 4.22 – Distribuição dos nodos no primeiro nível através do Leaf-Based Layout..... | 64 |
| FIGURA 4.23 – Esquema do cálculo das distâncias no primeiro nível da área de detalhe . | 65 |
| FIGURA 4.24 – Largura do retângulo antes (a) e depois do mapeamento (b) | 66 |
| FIGURA 4.25 – Janela principal do Bifocal Browser..... | 67 |
| FIGURA 5.1 – Browser de arquivos MagniFind da Inxight | 70 |
| FIGURA 5.2 – Um exemplo de tarefa fornecida ao sujeito | 72 |
| FIGURA 5.3 – Um exemplo de tarefa utilizada com a hierarquia grande | 73 |
| FIGURA 5.4 – Um exemplo de uma questão utilizada no teste de satisfação do usuário ... | 74 |
| FIGURA 5.5 – Comparação das médias de tempo obtidas na execução das tarefas 1 e 2... 80 | |
| FIGURA 5.6 – Comparação das médias de tempo obtidas na execução das tarefas 3 e 4... 81 | |
| FIGURA 5.7 – Comparação das médias de tempo obtidas na execução das tarefas 5 e 6... 82 | |
| FIGURA 5.8 – Percentual de erros cometidos na execução da tarefa 7..... 82 | |
| FIGURA 5.9 – Comparação das médias de tempo obtidas na execução das tarefas 8 e 9... 83 | |
| FIGURA 5.10 – Comparação das médias de tempo obtidas na execução da tarefa 10..... 84 | |
| FIGURA 5.11 – Comparação das médias de tempo obtidas na execução da tarefa 11..... 84 | |
| FIGURA 5.12 – Percentual de erros cometidos na execução da tarefa 11..... 85 | |
| FIGURA 5.13 – Frequência de valores atribuídos às questões sobre a representação visual..... 85 | |
| FIGURA 5.14 – Frequência de valores atribuídos às questões sobre mecanismos de interação | 86 |

Lista de Tabelas

| | |
|--|----|
| TABELA 4.1 – Número de nodos folhas contidos em cada subárvore..... | 57 |
| TABELA 4.2 – Medida angular dos setores ocupados pelos nodos de 1 a 9 da hierarquia utilizada como exemplo..... | 59 |
| TABELA 4.3 – Medida angular dos setores ocupados pelos nodos de 10 a 18 da hierarquia utilizada como exemplo | 59 |
| TABELA 4.4 – Coordenadas polares dos nodos de 1 a 9 da hierarquia utilizada como exemplo | 60 |
| TABELA 4.5 – Coordenadas polares dos nodos de 10 a 18 da hierarquia utilizada como exemplo | 60 |
| TABELA 4.6 – Botões da barra de ferramentas de uso geral | 67 |
| TABELA 4.7 – Botões da barra de ferramentas relacionados com a alteração do layout ... | 68 |
| TABELA 5.1 – Relação das médias de tempo obtidas na execução das tarefas..... | 79 |
| TABELA 5.2 – Percentual de erros cometidos na execução das tarefas..... | 79 |

Resumo

Estruturas de informações organizadas hierarquicamente estão presentes em muitas áreas. Pode-se citar como exemplos diagramas organizacionais, árvores genealógicas, manuais, estruturas de diretórios, catálogos de bibliotecas, etc. Na última década, várias técnicas têm sido desenvolvidas a fim de permitir a navegação em espaços de informações organizados dessa forma. Essas técnicas buscam proporcionar uma melhor percepção de alguns atributos ou fornecer mecanismos de interação adicionais que vão além da tradicional navegação com barras de rolagem ou câmeras 3D em visualização bi e tridimensional, respectivamente.

Dentre as várias alternativas de representação utilizadas nas diversas técnicas para dados hierárquicos destacam-se dois grandes grupos: as que utilizam a abordagem de preenchimento do espaço e as baseadas em diagramas de nodos e arestas. Na primeira o espaço disponível para a representação da estrutura é subdividido recursivamente, de forma que cada subárea representa um nodo da hierarquia. Na segunda, os nodos são representados por figuras geométricas e os relacionamentos, por linhas. Outro critério utilizado para classificá-las é a estratégia que cada uma aplica para exibir os detalhes presentes na estrutura. Algumas técnicas utilizam o método *Foco+Contexto* de modo a fornecer uma representação visual inteira do espaço de informações, bem como uma visão detalhada de itens selecionados na mesma área de exibição. Outras utilizam a abordagem *Visão Geral+Detalhe* que possui a característica de exibir essas duas partes (conjunto total e subconjunto de interesse) em áreas separadas.

O objetivo do presente trabalho é investigar a integração dessas duas abordagens a partir da proposta da técnica *Bifocal Tree*. Esta estrutura utiliza um diagrama de nodos e arestas e incorpora os conceitos existentes na abordagem *Foco+Contexto* guardando, porém uma divisão mais perceptível da visão de contexto e de detalhe. Ela introduz o uso de um segundo foco proporcionando duas áreas de visualização onde são exibidos dois sub-diagramas conectados entre si. Um corresponde à subárvore que contém o trecho da estrutura de interesse do usuário, enquanto o outro representa o contexto da hierarquia visualizada. Possui ainda alguns mecanismos de interação a fim de facilitar a navegação e a obtenção das informações exibidas na estrutura. Experimentos baseados em tarefas realizadas por usuários com a *Bifocal Tree*, o *Microsoft Windows Explorer* e o *browser MagniFind* foram utilizados para a avaliação da técnica demonstrando suas vantagens em algumas situações.

Palavras-Chave: Visualização de informações, Estruturas hierárquicas, Abordagem foco+contexto.

TITLE: “BIFOCAL TREE: A TECHNIQUE FOR VISUALIZATION OF HIERARCHICAL STRUCTURES”

Abstract

Hierarchical information structures are present in many areas. Some examples are organizational charts, genealogical trees, file systems, etc. In the last decade many techniques have been developed in order to allow the navigation in these information spaces. These techniques either convey a better perception of some attributes or provide additional interaction functions that go beyond traditional navigation with 2D scrollbars or 3D cameras, in 2D or 3D visualizations respectively.

Among the many alternatives used to represent these structures graphically, two groups can be pointed out: those that use the space-filling approach and those based on node-edge diagrams. In the former, the available space for representing the structure is subdivided recursively, so that each subarea represents a node in the hierarchy. In the latter, node (information elements) are represented by geometric figures, while lines represent relationships between nodes. Another criterion used for classifying these techniques is the strategy each one applies for exhibiting the details present in the structure. Some techniques are classified as *Focus+Context* because they provide both a visual representation of the entire information space and a detailed view of selected items in the same screen area. Other techniques use the *Overview+Detail* approach that is characterized by exhibiting these two parts (whole set and interest subset) in separated areas.

This work aims at investigating the integration of these two approaches. A new visualization technique is proposed: the *Bifocal Tree*, which uses a node-edge diagram and incorporates the concepts from the *Focus+Context* approach but keeping a more perceptive division of the context and detail views. It introduces the use of a second focus providing two areas for visualization where are two interconnected subdiagrams are exhibited. One of the diagrams refers to a subtree which contains the structure the user is interested in, while the other represents the context (parent and siblings) of the selected subtree. The technique also provides some interaction mechanisms in order to facilitate the navigation and querying of information about the structure. Results from experiments with users performing tasks on hierarchies using the *Bifocal Tree*, the *Microsoft Windows Explorer* and *Magnifind* showed its advantages in some situations.

Keywords: Information visualization, Hierarchical structures, Focus+context approach.

1 Introdução

O rápido desenvolvimento de novas tecnologias e o decréscimo do custo do *hardware* vem proporcionando um aumento do poder de processamento e capacidade de armazenamento dos computadores. Esta proliferação da disponibilidade computacional associada com os avanços nas comunicações, particularmente no crescimento da *World Wide Web* (WWW), tem resultado em uma verdadeira explosão da quantidade de informações disponíveis. Usuários acessando essas grandes e diversificadas bases de dados ou realizando buscas na Internet obtêm facilmente um volume enorme de informações, dentre as quais muitas podem ser irrelevantes para os objetivos da tarefa que está sendo realizada. Dessa forma, minimizar a sobrecarga de informações tem sido uma das maiores preocupações na representação dos resultados obtidos através de mecanismos de recuperação de informações.

Uma abordagem para contornar as dificuldades de selecionar informações relevantes dentre os resultados de buscas em sistemas de recuperação de informações e de grandes bases de dados é utilizar técnicas de visualização de informações [GER 97] através das quais o usuário obtém uma representação visual que, se por um lado abstrai detalhes do conjunto de informações, por outro propicia uma organização desse conjunto segundo algum critério. Assim, representações gráficas tradicionalmente empregadas apenas para divulgação e apresentação de dados e resultados passaram a ser usadas como ferramentas de exploração, no que Tukey [TUK 77] pioneiramente denominou análise exploratória visual.

A visualização de informações está associada à apresentação de dados abstratos em uma forma gráfica a fim de que o usuário possa utilizar sua percepção visual para avaliá-los. Card et al.[CAR 99] a define como “uma representação visual de dados abstratos interativa baseada no computador para amplificar a cognição”. O uso do poder de processamento do computador, para rapidamente gerar uma representação, oferece ao usuário a habilidade de não somente obter uma apresentação clara e compreensível, mas também de modificá-la a fim de permitir o destaque dos pontos de interesse. A inclusão da palavra abstrato na definição acima advém da distinção desse novo campo de pesquisa da Visualização Científica, que tipicamente trata da visualização de dados obtidos de sensores, que medem fenômenos naturais, ou modelos matemáticos utilizados em simulações. Em muitos exemplos, este elo com a natureza ou mundo físico sugere uma forma para organizar e apresentar informações que é imediatamente reconhecida pelo usuário, como por exemplo, um mapa da densidade populacional.

No entanto, a visualização de informações é caracterizada pela necessidade do projetista criar uma forma para transformar os dados em uma representação gráfica. Esta representação deve expressar as suas propriedades mais relevantes e mostrar como os diferentes itens estão relacionados entre si. Algumas vezes, isto é relativamente simples, como por exemplo, a utilização de um gráfico de setores para representar uma pesquisa eleitoral. Outras vezes, isto não é tão imediato como no caso da exibição da interconexão das centenas de páginas *web* existentes em um site. Frequentemente, o problema refere-se a projetar uma representação que funcione para grandes conjuntos de dados, o que geralmente induz a necessidade da limitação das informações a serem exibidas através de filtros dirigidos pelo usuário.

Assim, no desenvolvimento de sistemas de visualização deve ser considerada a melhor forma de mapear os dados para uma representação que facilite a interpretação por parte do usuário. Além disso, devem ser previstos meios que limitem a quantidade de informações que estes recebem mantendo-os ao mesmo tempo informados sobre o conjunto global dos dados.

Estruturas de informações organizadas hierarquicamente estão presentes em muitas áreas. Pode-se citar como exemplos diagramas organizacionais, árvores genealógicas, manuais, estruturas de diretórios, catálogos de bibliotecas, etc. Na última década várias técnicas foram desenvolvidas [ROB91][SHN 92][LAM 95] a fim de permitir a navegação em espaços de informações organizados dessa forma. Essas técnicas buscam proporcionar uma melhor percepção de alguns atributos ou fornecer mecanismos de interação adicionais que vão além da tradicional navegação com barras de rolagem ou câmeras 3D em visualização bi e tridimensional, respectivamente.

Dentre as várias alternativas de representação utilizadas nas diversas técnicas para dados hierárquicos destacam-se dois grandes grupos: as que utilizam a abordagem de preenchimento do espaço e as baseadas em diagramas de nodos e arestas. Na primeira o espaço disponível para a representação da estrutura é subdividido recursivamente, de forma que cada subárea represente um nodo da hierarquia e, na segunda, os nodos são representados por figuras geométricas e os relacionamentos por linhas.

Outro critério utilizado para classificá-las é a estratégia que cada uma aplica para exibir os detalhes presentes na estrutura. Algumas técnicas utilizam o método *Foco+Contexto* em função de fornecer uma representação visual inteira do espaço de informações bem como uma visão detalhada dos itens selecionados na mesma área de exibição. Outras utilizam a abordagem *Visão Geral+Detalhe* que possui a característica de exibir essas duas partes (conjunto total e subconjunto de interesse) em áreas separadas.

O objetivo do presente trabalho é investigar a integração dessas duas abordagens a partir da proposta da técnica *Bifocal Tree*, que utiliza em sua representação diagramas de nodos e arestas. A técnica foi avaliada através de um experimento com um conjunto de usuários que executaram tarefas sobre hierarquias utilizando a *Bifocal Tree*, o *Microsoft Windows Explorer* e o browser hiperbólico *Magnifind*.

O trabalho está organizado como segue. No capítulo 2 são apresentadas as principais técnicas de visualização de estruturas hierárquicas que utilizam o preenchimento de espaço e diagramas de nodos e arestas, enquanto, no capítulo 3, são discutidas as abordagens *Visão Geral+Detalhe* e *Foco+Contexto*. No capítulo 4 é apresentada a técnica *Bifocal Tree*, objeto de estudo desse trabalho. A avaliação experimental da técnica é descrita no capítulo 5. Finalmente, o capítulo 6 contém as conclusões e propostas de trabalhos futuros.

2 Visualização de estruturas hierárquicas

Estruturas de informações organizadas hierarquicamente como árvores genealógicas, organogramas, e mapas de web sites geralmente são representados por diagramas compostos de nodos e arestas. Além dos tradicionais métodos de recuperação das informações associados às entidades que compõem essas estruturas, os usuários necessitam de mecanismos de exploração a fim de que possam perceber de forma mais intuitiva os relacionamentos existentes entre seus componentes. Neste capítulo será apresentada uma revisão das principais técnicas desenvolvidas na última década para a visualização de estruturas hierárquicas.

Vários *layouts* têm sido utilizados para a representação de árvores. Entre eles destacam-se os mais conhecidos como o diagrama tradicional *top-down*, a lista delimitada, a lista indentada e conjuntos aninhados semelhantes ao diagrama de Venn usado na representação de conjuntos. Para a geração do *layout top-down* (figura 2.1), o algoritmo clássico de Reingold e Tilford [REI 81] fornece uma solução efetiva, mas a relação entre a largura e profundidade da hierarquia pode trazer dificuldades na visualização. Conforme a árvore cresce linearmente em profundidade, o espaço exigido para a exibição do nível mais baixo cresce de forma exponencial.

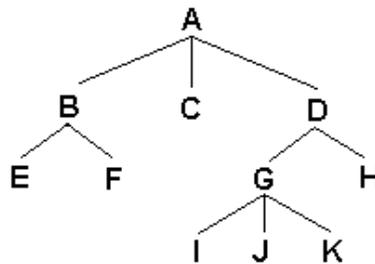


FIGURA 2.1 – Diagrama *top-down*

Nas listas delimitadas (figura 2.2) são listados todos os caminhos da raiz a cada nodo existente na hierarquia, separando cada elemento com uma barra. Essa representação possui a vantagem de exibir claramente os relacionamentos pai/filhos, mas faz com que o usuário tenha que percorrer o caminho de todos os nodos para construir o modelo mental da hierarquia. Para a percepção global em grandes hierarquias essa forma de representação torna-se inaceitável.

```

A
A/B
A/B/E
A/B/F
A/C
A/D
A/D/G
A/D/G
A/D/G/I
A/D/G/J
A/D/G/K
A/D/H
  
```

FIGURA 2.2 – Listas delimitadas

A lista indentada (figura 2.3), que é utilizada em alguns *browsers* de arquivos, como, por exemplo, no Windows Explorer da Microsoft, exibe para cada nodo, uma lista de filhos imediatamente após o seu respectivo pai, ligando-os através de linhas.

Essa representação resolve algumas dificuldades inerentes às listas delimitadas, como a exibição da estrutura da árvore, mas introduz outras dificuldades quando as árvores crescem em tamanho. Neste caso a quantidade de linhas paralelas ao longo da margem esquerda da tela podem conduzir a uma confusão sobre os relacionamentos pai/filhos existentes na estrutura. Além disso, a exibição de apenas um nodo por linha faz com que seja possível a visualização de um pequeno trecho da hierarquia, gerando problemas referente à perda de contexto durante a navegação. Entretanto essa forma permite que um maior número de detalhes sobre os nodos possam ser visualizados.

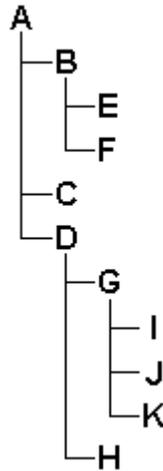


FIGURA 2.3 – Listas indentadas

Na representação através de conjuntos aninhados, a árvore é exibida como um grupo de conjuntos, onde cada nodo é desenhado como um subconjunto de seu pai conforme ilustrado na figura 2.4. A vantagem apresentada por esse *layout* é o aproveitamento do espaço da tela, embora a área para exibição de informações dentro dos nodos seja limitada. Essa abordagem é baseada no preenchimento do espaço, onde Johnson e Shneidermann [JOH 91] se apoiaram para criar uma técnica chamada *TreeMap* que será descrita adiante, neste mesmo capítulo.

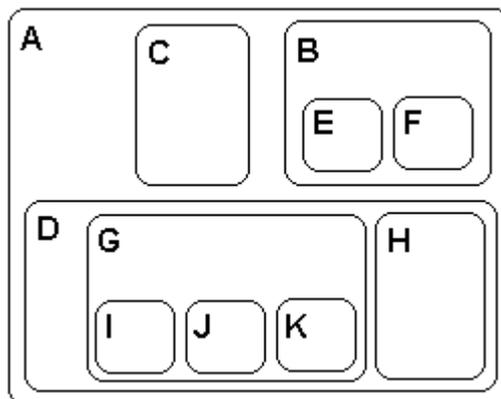


FIGURA 2.4 – Conjuntos aninhados

Além dessas representações, outras técnicas foram propostas no decorrer da última década a fim de proporcionar ao usuário a visualização de hierarquias de uma forma mais efetiva. A seguir serão apresentadas algumas técnicas encontradas na

literatura, divididas em dois grandes grupos: aquelas que utilizam a abordagem do preenchimento do espaço e as que utilizam os diagramas de nodos e arestas. Na primeira, o espaço disponível para a representação da estrutura é subdividido recursivamente, de forma que cada subárea represente um nodo da hierarquia. Na segunda abordagem os nodos que compõem a estrutura são representados por figuras geométricas e seus relacionamentos por linhas. Cabe salientar que algumas propostas não se enquadram nesta classificação, como por exemplo a Cheops [BEA 96].

2.1 Técnicas baseadas na abordagem de preenchimento do espaço

2.1.1 *Treemaps*

A técnica *Treemaps* foi proposta por Shneiderman [SHN 92] como forma de visualizar uma estrutura de diretórios e arquivos. Baseada no preenchimento do espaço bidimensional, divide uma área retangular em novos retângulos que representam cada nodo. A área de cada retângulo é proporcional a algum atributo, como por exemplo, o tamanho de arquivos. Na sua concepção foram levados em conta os seguintes objetivos [JOH 91]:

- utilizar o espaço de forma eficiente para permitir a visualização de grandes hierarquias;
- possuir um controle interativo da apresentação das informações com um *feedback* em tempo real;
- facilitar a interpretação da estrutura visualizada;
- produzir um desenho esteticamente agradável.

A figura 2.5 ilustra uma *Treemap* que representa uma estrutura de arquivos.

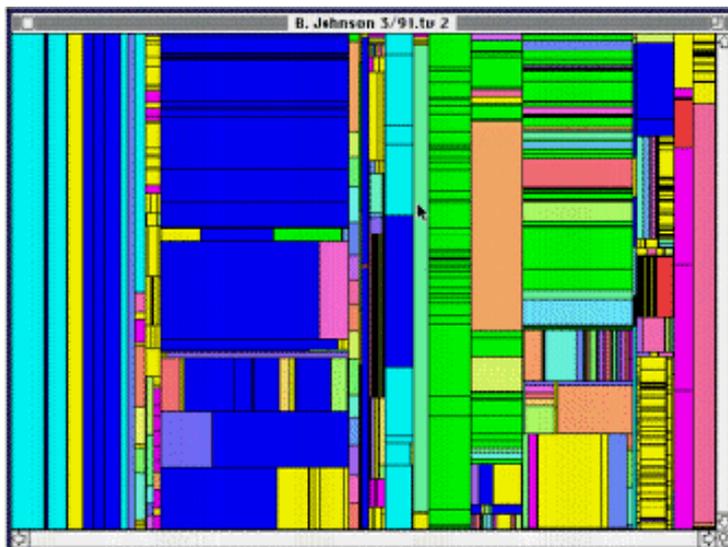


FIGURA 2.5 – *TreeMap* (extraída de [SHN 92])

A divisão dos retângulos começa segundo o eixo horizontal sendo rotacionado em 90 graus a cada nível. O algoritmo detalhado foi descrito por Johnson e Shneiderman [JOH 91]. No entanto, cabe salientar que, em determinados casos, alguns retângulos não serão exibidos. Isso ocorre quando o peso do nodo que ele representa é

muito pequeno e a divisão ao longo de um determinado eixo gera um retângulo com uma dimensão menor que um pixel. Regiões onde isso ocorre são indicadas por uma cor especial. A figura 2.6 apresenta a visualização de uma hierarquia de arquivos com 1500 nodos e 190 diretórios. As áreas em preto identificam zonas de agrupamento de nodos que não foram exibidos.

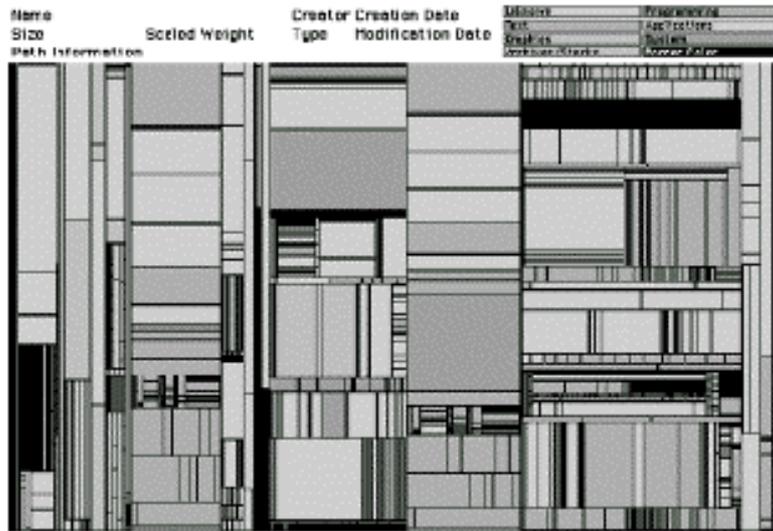
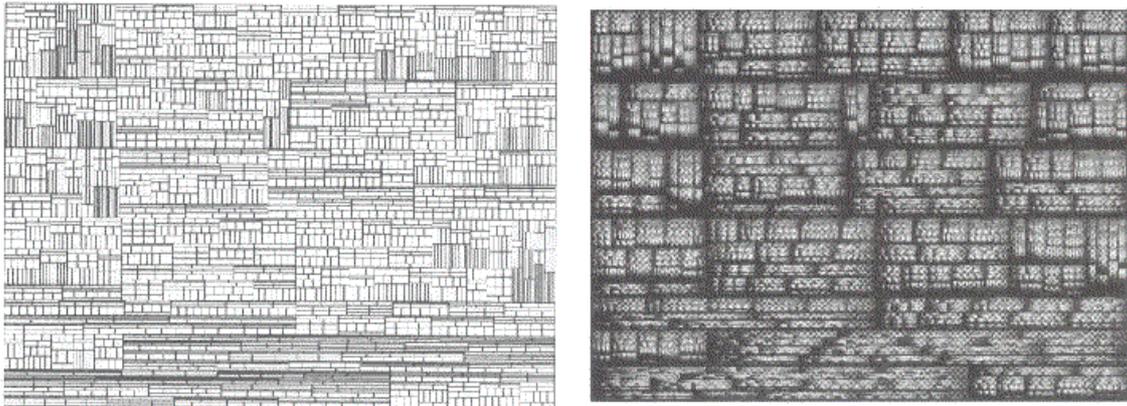


FIGURA 2.6 – *TreeMap* (extraída de [TUR 92])

Algumas características desejáveis em sistemas que utilizam a técnica *TreeMap* foram investigadas através de experimentação [TUR 92]. Uma delas é a existência de mecanismos que proporcionem aos usuários um maior controle sobre o mapeamento dos atributos para o tamanho e cor dos retângulos. A modificação deles permite ao usuário enfatizar os dados de seu interesse. A filtragem de nodos, através da ocultação daqueles indesejáveis, é outra característica que proporciona ao usuário a concentração em um determinado conjunto de dados. São propostas também duas funções que permitem a expansão de um nível de cada vez ou um nodo em particular. A utilização de animação também se faz presente, possibilitando que vários *Treemaps* de uma determinada hierarquia sejam visualizados, em tempos diferentes, na mesma área de vídeo. Dessa forma a percepção das alterações relativas ocorridas na estrutura é facilitada. Um mecanismo de *zoom* pode ser utilizado para ampliar um determinado nodo, fazendo com que ele se torne o nodo raiz da hierarquia exibida. Isso possibilita que os nodos eventualmente agrupados em regiões (por possuírem um tamanho muito reduzido) possam ser visualizados normalmente. Se houver espaço disponível nos retângulos, textos são adicionados fornecendo uma orientação ao usuário.

Uma extensão para a *Treemap* denominada de *Cushion Treemap* [WIJ 99], foi proposta com objetivo de melhorar a identificação da estrutura visualizada através da aplicação de um sombreamento 3D para cada retângulo, já que em determinadas situações como em estruturas balanceadas, a percepção da hierarquia torna-se difícil. A figura 2.7(a) mostra uma estrutura de 3060 nodos visualizada através da *Treemap* original e a figura 2.7(b) exhibe a mesma hierarquia com a *Cushion Treemap*.



(a) (b)
 FIGURA 2.7 – (a) *Treemaps* (b) *Cushion Treemaps* (extraída de [WIJ 99])

Além disso, são propostas algumas alternativas de navegação. Através da seleção de um nodo, indicada por uma moldura vermelha sobre o respectivo retângulo, é possível consultar os seus atributos que são mostrados em uma janela separada. O teclado é utilizado para alternar a seleção entre nodos irmãos (seta para a esquerda ou para a direita), ou entre pai (seta para cima) e o primeiro filho (seta para baixo).

2.1.2 *Information Cube*

Uma técnica baseada na subdivisão do espaço tridimensional chamada *Information Cube* [REK 93] foi proposta por Rekimoto e Green, em 1993. Nela, as informações são exibidas como cubos translúcidos encaixados um dentro do outro, onde o mais externo representa o nodo raiz da hierarquia (figura 2.8). Os cubos que denotam o segundo nível estão localizados no interior do mais externo, os do terceiro dentro daqueles que ocupam o segundo nível e assim por diante. Títulos são exibidos na superfície dos cubos representando as informações contidas nos nodos.

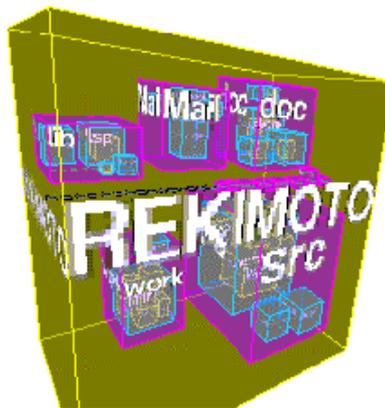


FIGURA 2.8 – *Information Cube* (extraída de [REK 93])

Para que o usuário possa ver o seu interior, cada cubo é exibido aplicando-se à cor um certo grau de transparência. O uso de dessas superfícies translúcidas permite que, até mesmo em hierarquias muito profundas, a complexidade da imagem mantenha-se em um nível razoável em função de que os cubos que representam os nodos mais internos são enevoados gradualmente. Além disso, elas podem representar informações adicionais pela variação de seu nível de transparência.

São fornecidas facilidades de *zoom*, movimentação e rotação da estrutura que são executadas através de animações para que o usuário não se sinta confuso com a troca brusca do foco. A rotação é permitida somente em torno do eixo vertical para não causar desorientação. Embora o usuário possa utilizar um monitor comum para a visualização, Rekimoto e Green [REK 93] propõem a utilização de equipamento de Realidade virtual como HMD (*Head Mounted Display*) e a *DataGlove*, como mecanismo de interação. Para permitir a inspeção de uma informação localizada em um nível mais baixo da hierarquia foi implementada uma operação chamada foco. Nela o usuário dispara um feixe virtual (raio de apontamento) na direção do ponto de interesse (figura 2.9). Quando o raio intercepta um objeto ele automaticamente torna-se um candidato a seleção, que pode ser confirmada pelo usuário.

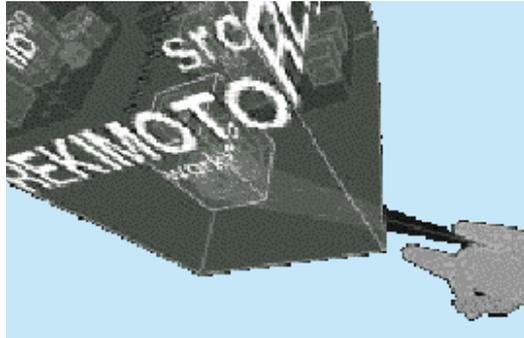


FIGURA 2.9 – Operação foco (extraída de [REK 93])

2.1.3 Information Slices

Information Slices [AND 98] é mais uma técnica baseada na abordagem de preenchimento do espaço e, assim como a *Treemap*, foi proposta para a exibição de estrutura de diretórios. Ela utiliza um ou mais discos semicirculares para representar a hierarquia. Seus diversos níveis são exibidos em regiões concêntricas, onde cada fatia representa um nodo. O espaço ocupado pelas fatias é proporcional ao tamanho em bytes ocupados pelo total de arquivos que ela representa. Tipicamente são visualizados entre 5 e 10 níveis da estrutura em cada disco. Hierarquias compostas por muitos níveis são representadas através de uma série de discos em cascata, onde apenas 2 são exibidos simultaneamente como mostra a figura 2.10.

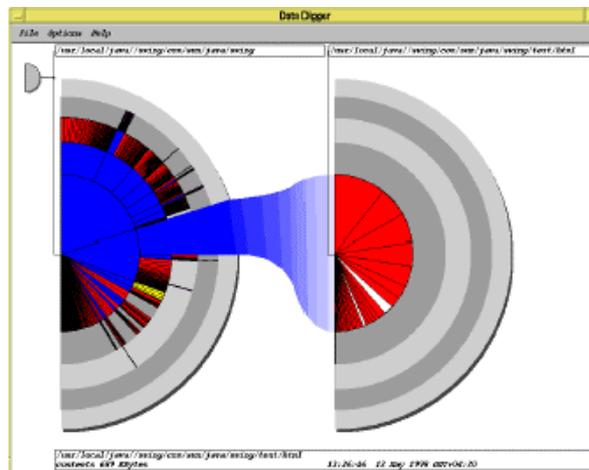


FIGURA 2.10 – *Information Slices* (extraída de [AND 98])

A navegação é proporcionada por um mecanismo de seleção que permite que um determinado nodo pertencente ao disco da esquerda, seja exibido com mais detalhes no disco que se situa à direita., com seus filhos distribuídos em uma fatia de 180 graus. Caso a seleção seja feita em um nodo situado no disco da direita, o primeiro é transformado em um ícone, representando os níveis iniciais, e um novo disco é aberto para exibir a sua respectiva subárvore. Cores são utilizadas para identificar os tipos dos nodos e as informações do nodo selecionado são exibidas em uma barra de status externa à representação gráfica.

2.1.4 *Sunburst* e extensões

Sunburst é uma ferramenta para visualização de estrutura de diretórios que também utiliza uma abordagem semelhante à técnica *Treemap*, diferenciando-se pela utilização de um layout circular ao invés do retangular. Os níveis são representados por círculos concêntricos com a medida de seus raios variando de forma proporcional à distância do respectivo nível em relação ao nodo raiz da estrutura (figura 2.11).

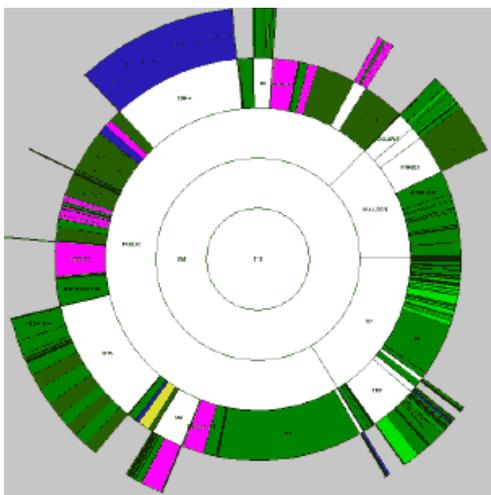


FIGURA 2.11 – *Sunburst* (extraída de [STA 2000])

Assim como a técnica *Information Slices* [AND 98], o ângulo utilizado para exibir cada fatia é proporcional ao tamanho do arquivo ou diretório que ele representa, e a cor identifica o tipo de arquivo ou a data da mais recente alteração. Ao fazer um experimento de avaliação com um grupo de usuários, foi identificado que os nodos referentes a pequenos arquivos tinham uma medida angular muito pequena, dificultando a leitura de seus atributos, o que é agravado em grandes hierarquias. Dessa forma, Stasko e Zhang [STA 00] propuseram três formas alternativas para o layout, a fim de permitir o exame de trechos da estrutura com um maior nível de detalhes.

Na primeira (*Angular Detail*), quando o usuário seleciona um nodo, o desenho que representa a estrutura inteira é reduzido de tamanho e movido para um local próximo a borda da janela no lado diametralmente oposto ao item selecionado. Logo após o setor que representa a subárvore do nodo selecionado é exibido fora do desenho original da estrutura, com suas medidas radial e angular ampliadas. A transição entre os dois momentos é executada através de animação a fim de minimizar a desorientação do usuário. Os dois parâmetros (raio e ângulo) podem ser alterados conforme a necessidade. A figura 2.12 mostra 3 momentos dessa seleção.



FIGURA 2.12 – *Angular Detail* (extraída de [STA 2000])

Na segunda alternativa (*Detail Outside* - figura 2.13), o desenho da estrutura é reduzido de tamanho, mas mantém-se no centro da janela. A subárvore do nodo selecionado também é exibida externamente ao desenho da estrutura, mas ocupa uma medida angular de 360 graus formando um anel em torno do desenho original.

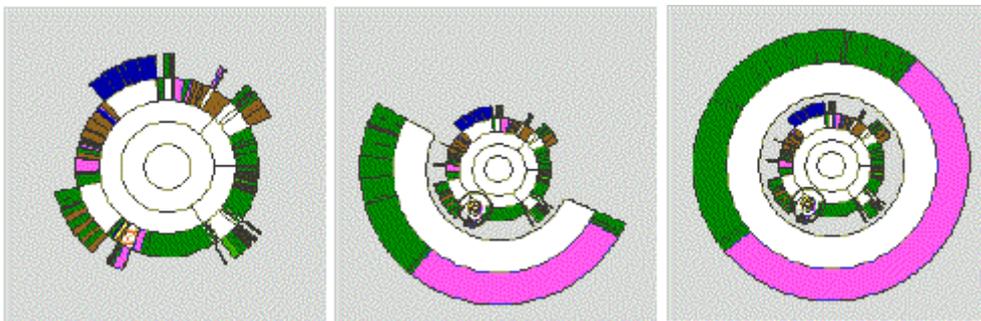


FIGURA 2.13 – *Detail Outside* (extraída de [STA 2000])

Na terceira (*Detail Inside*), o diâmetro do círculo onde é exibida a estrutura é ampliado de forma que a subárvore do nodo selecionado seja exibida em seu interior conforme mostra a figura 2.14. Nesta situação é a estrutura geral que forma um anel circundando a área detalhada.

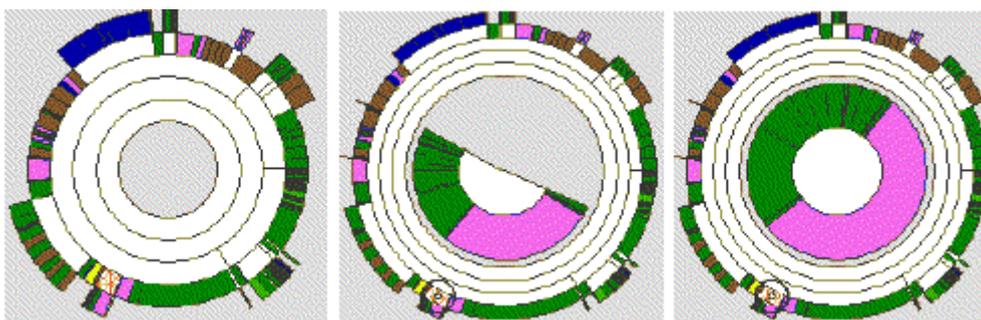


FIGURA 2.14 – *Detail Inside* (extraída de [STA 2000])

2.2 Técnicas baseadas em diagramas de nodos e arestas

Algumas técnicas que utilizam diagramas de nodos e arestas têm se utilizado do espaço tridimensional para representar estruturas hierárquicas. Elas fundamentam-se na idéia de que a terceira dimensão fornece um espaço adicional para a visualização, facilitando a exibição de grandes estruturas. No entanto alguns novos problemas são introduzidos, com por exemplo, o aumento da possibilidade de que os elementos que as

compõem se sobreponham tornando difícil a escolha do melhor ângulo de visualização. Conseqüentemente alguns mecanismos devem ser fornecidos ao usuário para facilitar a exploração das informações, permitindo que o usuário possa alterar interativamente a posição de visão, de forma que ele possa mover-se ao redor do espaço. A utilização de transparência da geração das imagens também é um aspecto bastante abordado. A seguir serão descritas algumas técnicas que utilizam diagramas de nodos e arestas para a representação das hierarquias.

2.2.1 Cone Tree

A técnica *Cone Tree* foi originalmente proposta por Robertson et al. [ROB 91], sendo implementada como parte integrante de um sistema de visualização experimental chamado *Information Visualizer* [CAR 91] desenvolvido no Xerox Palo Alto Research Center (PARC). Nela, a hierarquia é exibida dentro de um espaço tridimensional no formato de um paralelepípedo (*3D/Room*), onde o nodo raiz se localiza próximo ao topo e representa o vértice de um cone com os nodos filhos posicionados, igualmente distanciados, em torno de sua base. Cada nodo filho (não folha) dá origem a um novo cone. As dimensões dos cones são calculadas de forma que toda a estrutura se encaixe perfeitamente dentro das medidas do paralelepípedo. A altura de cada cone é a mesma em todos os níveis sendo obtida dividindo-se a dimensão vertical pelo número de níveis da árvore. Já o diâmetro da base dos cones é reduzido a cada nível à medida que eles se aproximam da parte inferior, de forma que aqueles localizados no nível mais baixo da hierarquia se encaixem na largura disponível.

A superfície de cada cone é exibida como se fosse constituída de um material transparente com o objetivo de ser percebida sem ocultar aqueles que estão localizados na parte de trás. As sombras dos cones são projetadas no plano inferior para reforçar a percepção do aspecto tridimensional da estrutura. Os nodos se apresentam no formato de um pequeno cartão, com as medidas na proporção de 3 x 5, onde as informações são impressas. A figura 2.15 apresenta uma *cone tree*.

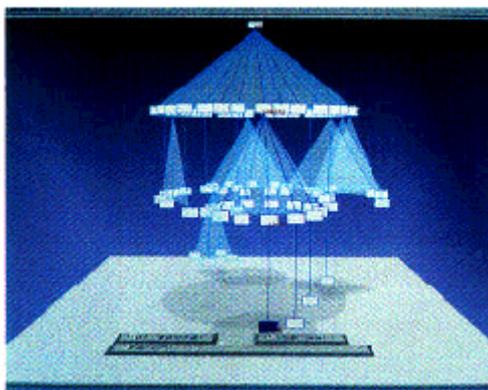


FIGURA 2.15 – *Cone Tree* (extraída de [ROB 91])

Robertson et al. [ROB 91] justificam a utilização da visualização no espaço 3D comparando com o espaço de tela necessário para a exibição de um layout 2D tradicional. Ignorando-se as dimensões dos nodos, uma árvore com n níveis e grau máximo igual a g terá como largura em sua base g^{n-1} e uma relação de aspecto igual a g^{n-1}/n . Nota-se que esta relação aumenta em função do número de níveis existentes de forma exponencial. Em contraste, a relação de aspecto da *Cone Tree* se mantém

constante, pois as dimensões dos cones dependem apenas do tamanho do espaço onde serão inseridos, e não mais do número de nodos ou grau máximo da árvore. Neste caso a profundidade é utilizada para possibilitar o preenchimento da tela com mais informações. Entretanto, o fato de que a estrutura tenha que se encaixar dentro deste espaço, traz uma limitação (em torno de 10) do número de níveis que poderão.

Além das características espaciais já citadas, a *Cone Tree* oferece um conjunto de mecanismos que permitem a poda ou expansão de trechos da hierarquia e a localização de nodos que contenham a informação desejada.

Os mesmos autores propõem também um layout alternativo chamado *Cam Tree* (figura 2.16), onde a hierarquia é visualizada com uma orientação horizontal, posicionando o nodo raiz à esquerda da tela. Esse formato facilita a exibição do texto em cada nodo.

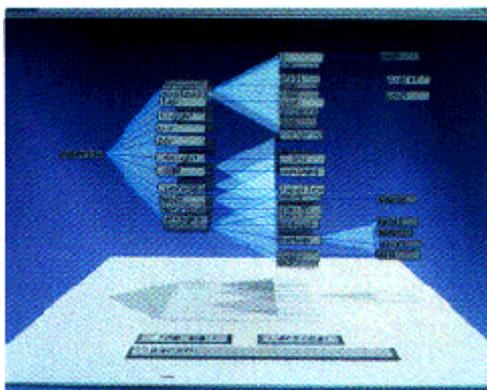


FIGURA 2.16 – *Cam Tree* (extraída de [ROB 91])

Várias melhorias foram incorporadas à idéia original da técnica. Tversky [TVE 93] incluiu a capacidade de redimensionamento dos cones e a utilização de canais visuais (tamanho, cor e forma) para permitir a representação de informações adicionais, baseando-se no fato de que a mente humana pode interpretá-los muito mais rapidamente do que fazer a leitura através de texto. Este sistema foi utilizado na visualização de estrutura de diretórios onde os programas executáveis são representados através de pirâmides, os diretórios na forma de elipsóides e o restante como paralelepípedos (figura 2.17).

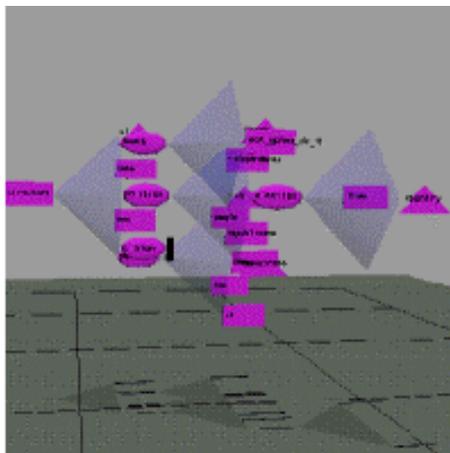


FIGURA 2.17 – Uso da forma e cor na representação dos nodos (extraída de [TVE 93])

Carrière e Kazman [CAR 95] em seu sistema *fsviz* propõem ainda a utilização da cor no desenho dos arcos que conectam os nodos a fim de indicar a profundidade na árvore, exibindo-os em cor diferente a cada nível. Começam com azul no topo da hierarquia e conforme descem alteram o tom suavemente até chegar ao vermelho na base da árvore. Essa utilização da cor é especialmente útil quando a árvore está sendo visualizada do topo, onde a terceira dimensão praticamente desaparece.

A fim de reduzir a desordem visual e aumentar a eficiência do processo de visualização foi proposto um mecanismo para a redução dos nodos efetivamente visualizados. Os que estiverem localizados além de uma distância arbitrária do ponto de vista do observador serão incluídos em uma única representação gráfica (neste caso um tetraedro), isto é, todos os nodos que estão localizados hierarquicamente abaixo do nodo em questão são substituídos por um único tetraedro (figura 2.18).

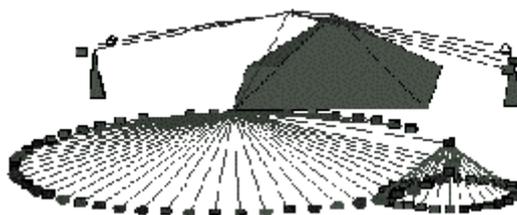


FIGURA 2.18 – Agrupamento de nodos (extraída de [CAR 95])

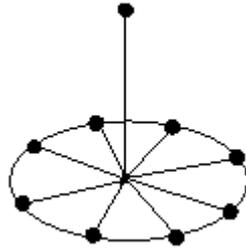
2.2.2 Reconfigurable Disc Tree

A técnica *Reconfigurable Disc Tree* (RDT) foi proposta por Jeong e Pang [JEO 98] e implementada no sistema VISIT (*Visual Information System for reconfigurable dIsc Tree*). Ela é baseada na *Cone Tree* [ROB 91] descrita anteriormente e tem como objetivo aperfeiçoar esta última nos seguintes aspectos:

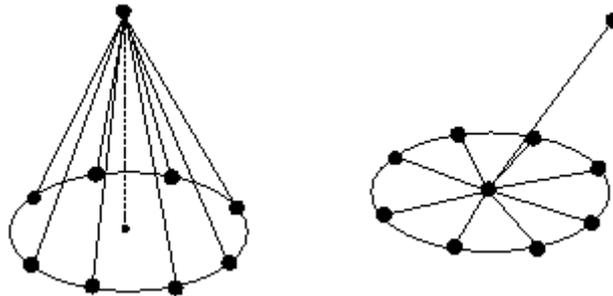
- diminuir a desordem visual;
- aumentar o número de nodos visualizados;
- permitir o mapeamento da estrutura no plano bidimensional sem sobreposição.

A RDT pode ser considerada uma generalização da *Cone Tree*, onde os cones foram substituídos por discos, sendo incluída ainda a capacidade de alteração da sua forma através de alguns parâmetros que podem ser configurados conforme a necessidade. Combinações dos valores desses parâmetros dão origem a três tipos de representações denominadas: *Disc Tree*, *Compact Disc Tree* e *Plane Disc Tree*, essa última gerada no plano.

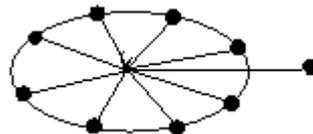
Uma *Disc Tree* é uma RDT onde o cone é transformado em um disco. Embora o nodo pai continue sendo representado no vértice, as ligações entre ele e seus filhos ocorrem em duas etapas. Inicialmente, as linhas têm origem nos nodos filhos (distribuídos em torno do disco) com suas extremidades finais localizadas no centro. A partir desse ponto, uma linha que se situa na mesma vertical de onde está posicionado o nodo pai, completa a ligação. A figura 2.19 exemplifica o layout de um nível de uma *disc tree*.

FIGURA 2.19 – Esquema de uma *Disc Tree* [JEO 98]

A *Compact Disc Tree* utiliza duas representações diferentes que são utilizadas de forma alternada para exibir cada nível da hierarquia. Nos níveis de ordem par os nodos são visualizados conforme a *cone tree* original, e nos de ordem ímpar é utilizado uma *disc tree* onde a linha que liga o centro do disco e o nodo pai não se situa na mesma vertical. A figura 2.20 exibe cada uma das duas representações.

FIGURA 2.20 – Esquema de uma *Compact Disc Tree* [JEO 98]

A *Plane Disc Tree* diferencia-se das anteriores pelo fato de que todos os seus nodos são exibidos no mesmo plano. O seu layout é gerado levando-se em conta que não haja sobreposição alguma entre os discos que compõem a estrutura. A figura 2.21 mostra um nível da hierarquia nessa representação.

FIGURA 2.21 – Esquema para um *Plane Disc Tree* [JEO 98]

No sistema VISIT, onde a técnica foi implementada, são oferecidos mecanismos para alterar a aparência da RDT. É possível a alteração dinâmica dos atributos que definem o layout utilizado, bem como a técnica de exibição utilizada na geração final da imagem, que vai desde uma representação *wireframe* até um sombreamento suave. Ela também pode ser visualizada com transparência, para facilitar a compreensão global da hierarquia, ou de forma opaca permitindo que o usuário concentre-se em alguma subárvore localizada na parte frontal da estrutura.

2.2.3 Time Tube

A técnica *Time Tube* foi criada por Chi et al. [CHI 98] como parte integrante de um sistema denominado WEEV (*Web Ecology and Evolution Visualization*) que tem como objetivo facilitar a interpretação dos relacionamentos existentes nos conteúdos da *Web*, bem como a sua topologia e alterações que possam ocorrer através do tempo.

Um *Time Tube* é formado por uma ou mais *Disk Trees* que representam a estrutura de hiperlinks de um site em um determinado momento no tempo. A figura 2.22 apresenta a evolução do site da Xerox (www.xerox.com) durante o mês de abril de 1997. A estrutura é formada por 4 discos, onde cada um representa um estado do site em cada uma das 4 semanas de abril.

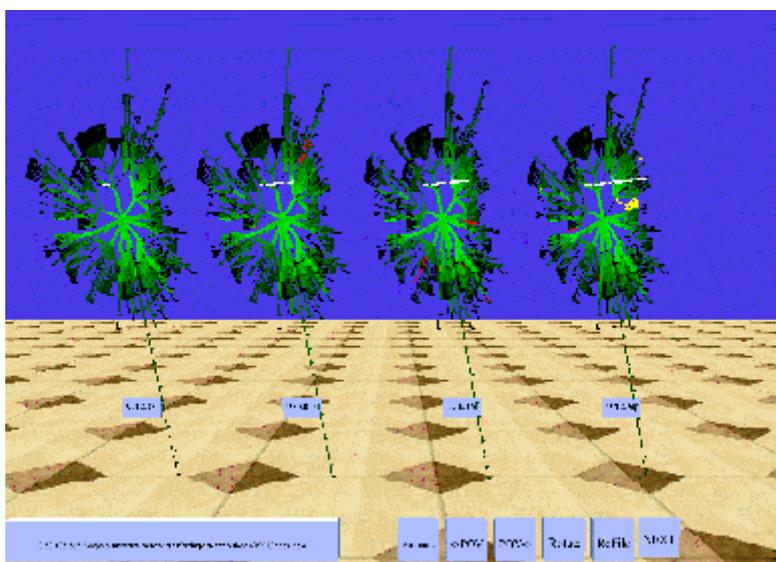


FIGURA 2.22 – *Time Tube* (extraída de [CHI 98])

Cada disco é representado através de uma técnica denominada *Disk Tree*. Ela exhibe a estrutura em um layout circular localizando o nodo raiz no centro. Cada nível é representado através de círculos concêntricos, onde o mais externo denota o nível mais profundo. Os nodos folha são exibidos como fatias do círculo que representa o nível onde eles estão localizados. O ângulo que determina o espaço ocupado por cada fatia é obtido dividindo-se 360° pelo número de nodos folha existentes na árvore.

Os *links* da árvore são representados através de linhas, que podem variar em sua espessura em função da frequência de acesso de cada link. Aqueles que ligam páginas de acesso frequentes são representados com linhas mais espessas. A adição de brilho ao atributo cor reforça a identificação das páginas mais visitadas, onde as cores claras indicam as mais acessadas. A cor propriamente dita representa o estágio do ciclo de vida que a página se encontra. Vermelho indica novas páginas, amarelo páginas excluídas, e verde páginas existentes. A figura 23 apresenta um exemplo de *Disk Tree*.

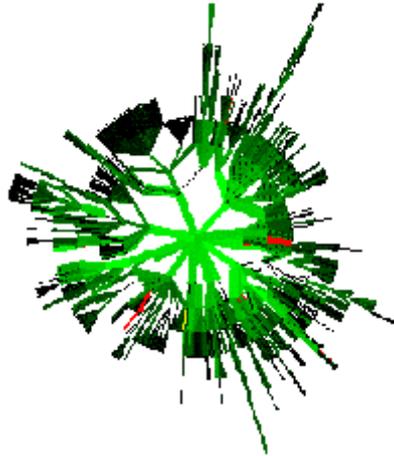


FIGURA 2.23 – *Disk Tree* (extraída de [CHI 98])

Em função da *Disk Tree* ser representada no plano, a terceira dimensão foi utilizada para mapear o tempo. Através da distribuição de um conjunto de *Disk Trees* ao longo de um eixo no espaço, foi possível apresentar a estrutura de um site em vários momentos no tempo na mesma imagem. O conjunto de discos é exibido de forma que eles não se situem paralelos entre si. Uma rotação é aplicada a cada um deles, de forma que ao serem exibidos ocupem a mesma porção de tela, a fim de evitar o efeito causado pela projeção perspectiva que produz uma redução nas dimensões de cada disco em função da sua localização na janela. Essa compensação angular evita que aqueles que estão localizados próximos ao centro possuam uma pequena porção de sua face exibida. Além disso, também evita que os discos localizados à esquerda do centro tenham a sua face frontal exibida, enquanto que os da direita sejam visualizados por sua face posterior, o que causaria uma maior dificuldade na compreensão global da estrutura. Essa característica também permite que um determinado disco possa sofrer uma rotação diferenciada a fim de ser enfatizado em relação ao demais.

O sistema mantém também diversas formas de interação com o usuário para permitir uma exploração das informações. Entre elas destacamos os seguintes mecanismos:

- rotação simultânea de todos os discos posicionando as suas faces em uma vista frontal a tela;
- *zoom* para permitir a visualização de um nodo com um maior nível de detalhes;
- destaque de um nodo selecionado em todos os discos simultaneamente;
- destaque das linhas que representam os *links* de um nodo selecionado.

2.2.4 *Hiperbolic Browser*

O uso da geometria hiperbólica na visualização de estruturas hierárquicas tem sido objeto de investigação por parte de alguns pesquisadores. A principal justificativa de sua utilização vem do fato de que nela, o perímetro de um círculo cresce exponencialmente em função de seu raio. Deste modo, estruturas hierárquicas, que tendem a expandir-se exponencialmente conforme o incremento da profundidade, podem ser posicionadas no espaço hiperbólico de forma uniforme, mantendo as distâncias entre pais e filhos aproximadamente iguais.

O *Hyperbolic Browser*, desenvolvido por Lamping et al. [LAM 95] exibe hierarquias no espaço 2D através dessa geometria. O processo de visualização consiste em obter o layout da estrutura no plano hiperbólico e então mapeá-la para o plano Euclidiano, através de modelos como o Poincaré ou Klein, durante a etapa de exibição [HER 2000]. Dessa forma, a posição de cada nodo no plano hiperbólico não precisa ser alterada durante a manipulação do foco. Neste mapeamento, um segmento de reta que possui um determinado comprimento próximo ao centro da superfície, tem seu tamanho diminuído conforme a aproximação de sua borda. Isso produz uma visualização com características *fisheye* [FUR 86] que permite exibir detalhes da hierarquia sem perder o contexto global. A técnica utiliza um layout radial posicionando o nodo raiz no centro de um círculo, exibindo cada subárvore em uma fatia com medida angular proporcional ao número de nodos folhas pertencentes a ela (figura 2.24).

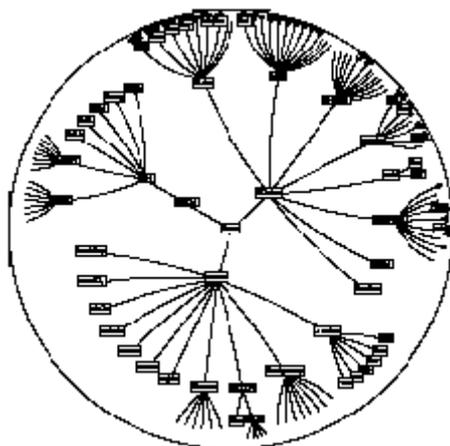


FIGURA 2.24 – *Hyperbolic Browser* (extraída de [LAM 95])

O posicionamento dos nodos em cada subárvore é obtido da seguinte forma: A figura 2.25 mostra um esquema da determinação da posição de cada nodo. O ponto P representa a localização de um nodo, e o setor QPR, com ângulo α , a área onde a subárvore de P será exibida. Inicialmente, o algoritmo busca determinar a medida de cada setor onde serão exibidas as subárvores de P que possuem como raiz os seus filhos P1, P2 e P3. O ângulo α é subdividido conforme o número de filhos, posicionando-os no centro desse setor com uma distância arbitrária de P. A seguir, a partir de linhas paralelas a PQ, PQ' e PQ'' com ponto inicial em P1, P2 e P3, são determinadas as medidas dos novos setores que irão conter as respectivas subárvores de P. Esse algoritmo é aplicado recursivamente para cada subárvore existente na hierarquia.

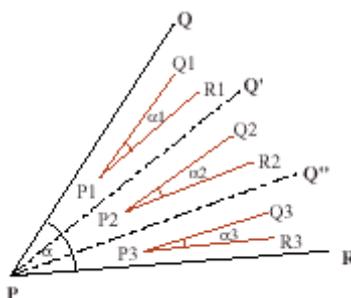


FIGURA 2.25 – Localização dos nodos no plano Euclidiano

A utilização de linhas paralelas garante que não haja sobreposição entre os nodos. Como se pode observar, essa subdivisão aplicada ao espaço Euclidiano, faz com que os ângulos tornem-se cada vez menores conforme o incremento da profundidade da árvore. No entanto, a sua aplicação no espaço hiperbólico, faz com que os ângulos obtidos para a exibição de cada subárvore não sejam reduzidos. Essa característica é obtida em função da forma como as linhas paralelas (PQ, PQ' e PQ'') são calculadas, através de funções trigonométricas hiperbólicas. A figura 2.26 mostra a estrutura anterior utilizando um mapeamento através do modelo Klein. No plano hiperbólico as linhas paralelas se interseccionam sobre a circunferência que representa o espaço (círculo unitário) utilizado para o mapeamento.

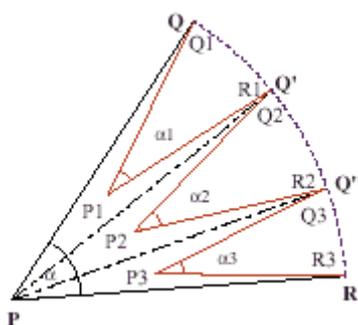


FIGURA 2.26 – Localização dos nodos no plano hiperbólico, visualizado através do modelo Klein

Como mencionado anteriormente, essa técnica possui a característica de exibir os nodos que estão próximos ao centro (foco) com seu tamanho ampliado em relação aqueles que estão mais distantes. Dessa forma, é fornecido um mecanismo para permitir que qualquer ponto visível (pertencente ou não a hierarquia) possa tornar-se o novo foco, a fim de possibilitar uma visualização com mais detalhes do trecho de interesse do usuário. Isso pode ser executado através da seleção de um ponto que será movido de forma automática para o foco, ou através de seu deslocamento (arrasto) para outra posição.

As translações executadas no espaço hiperbólico podem causar rotações nos elementos da estrutura gerando uma dificuldade de preservação do sentido de orientação do usuário. Para minimizar esse efeito, todas as transições são executadas com o uso de animação, através da exibição de visões intermediárias.

2.2.5 H3

Outra técnica que utiliza a geometria hiperbólica é a H3[MUN 95][MUN 97], desenvolvida para a visualização de estruturas tridimensionais que representam sites na *Web*. Nesta técnica foi utilizado o mapeamento segundo o modelo Klein. Nela, a estrutura é exibida dentro de um espaço delimitado por uma superfície esférica, onde os nodos folhas são representados por pequenas pirâmides como pode ser visto na figura 2.27.

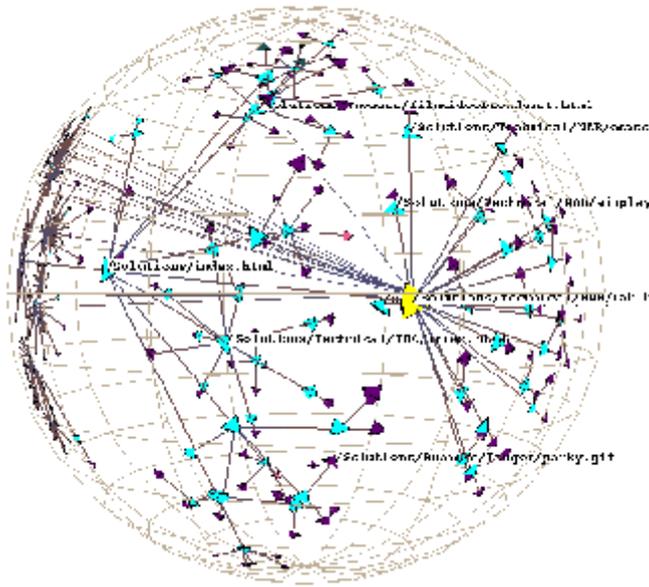


FIGURA 2.27 – Parte do site do grupo de computação gráfica da Universidade de Stanford visualizado com a técnica H3 (extraída de [MUN 97]).

A idéia básica utilizada no layout é semelhante à *Cone Tree* [ROB 91], diferenciando-se pelo uso de um formato de semi-esfera ao invés de cones para a exibição de um nodo com seus respectivos filhos. A cor também é utilizada para identificar o tipo que os nodos representam. Os textos exibidos em cada nodo são ocultados quando eles estão posicionados longe da origem. A seleção de um nodo através do clique do mouse faz com que ele seja transladado para o centro da esfera através de uma transição animada, semelhante ao mecanismo existente no *Hyperbolic Browser*. Junto com esse deslocamento está associado um componente de rotação que faz com que ao ser alcançada a origem, os nodos ancestrais estejam localizados à esquerda e os descendentes à direita. Essa disposição horizontal tem como objetivo orientar o usuário e minimizar a oclusão.

2.3 Comentários finais

Técnicas de visualização de informações hierárquicas provêm, em geral, uma representação onde os relacionamentos existentes entre os elementos de informação podem ser percebidos de forma intuitiva. Em alguns casos, porém esses relacionamentos não são explicitamente representados dificultando tarefas onde essa evidência seja necessária. Frequentemente existe o compromisso entre representar o conjunto de dados de forma mais geral e representá-lo de forma mais detalhada. Esse compromisso é explorado de diversas maneiras nas técnicas de visualização, como será visto no próximo capítulo.

3 Visualização de informações de contexto e detalhe

Este capítulo tem como objetivo apresentar características encontradas em técnicas de visualização através de uma classificação diferenciada em relação a que foi utilizada até o momento. Serão discutidos aqui os principais aspectos existentes nas abordagens denominadas Visão Geral+Detalhe e Foco+Contexto. Esses aspectos serão ilustrados através da apresentação de algumas técnicas que implementam esses conceitos.

Quando a estrutura a ser visualizada possui um grande número de elementos, os mecanismos de navegação incorporados à representação ganham uma maior importância. Dentre esses mecanismos, os mais tradicionais são o *pan*, através do qual o usuário pode deslocar uma janela virtual sobre o desenho a fim de visualizar um novo trecho da estrutura, e o *zoom* que permite o ajuste da escala de exibição proporcionando uma forma de ampliação para a exploração de detalhes. Entretanto, uma desvantagem encontrada nesses mecanismos é que, ao serem utilizados para focalizar a exibição em um determinado trecho, a visão geral da estrutura é perdida, podendo gerar problemas de usabilidade [HER 2000].

A apresentação de uma visão geral é muito importante. Ela reduz a procura, permite a detecção de padrões globais e auxilia o usuário na escolha do próximo movimento. Assim, uma heurística utilizada em um projeto de visualização é começar com uma visão geral. Mas é igualmente necessário que o usuário possa acessar os detalhes rapidamente [CAR 99a].

Dentro da idéia geral de manter uma visão do todo enquanto se analisa em detalhe uma parte dele, duas abordagens alternativas podem ser utilizadas para a exibição das estruturas. A primeira, denominada de *Visão Geral+Detalhe* [CAR 99a] possui a característica de que a região de interesse ocupa o campo de visão principal e o restante do conjunto de dados é mantido em área à parte. Já na *Foco+Contexto* [CAR 99b] a região de interesse é exibida em maior detalhe, de forma integrada e concomitante à visão geral.

3.1 Abordagem *Visão Geral+Detalhe*

A abordagem *Visão Geral+Detalhe* fornece duas ou mais visões da estrutura: uma visão geral para orientação, em geral simplificada, e uma visão detalhada da parte onde o usuário vai concentrar seu trabalho [CAR 99a]. Visões intermediárias com graus de detalhamento sucessivos eventualmente podem estar disponíveis.

As várias visões podem ser exibidas uma de cada vez (multiplexadas no tempo) ou ao mesmo tempo, em diferentes regiões da tela (multiplexadas no espaço). Se elas são exibidas ao mesmo tempo, há necessidade de estabelecer um equilíbrio no uso relativo da quantidade de espaço global disponibilizada para a exibição, o que por sua vez afetará o fator de *zoom*. O projeto de visualizações que utilizam a abordagem *visão geral + detalhe* envolve cuidadoso balanceamento na distribuição do uso do espaço [CAR 99a].

O processo de zoom para obtenção da visão da parte da estrutura que se quer ver em detalhes pode ser conseguido através da utilização de um *zoom* espacial ou *zoom* semântico. No *zoom* espacial a visão de detalhe é apenas uma versão ampliada do trecho da visão geral que interessa. Já no *zoom* semântico, o conteúdo mantém-se o mesmo mas a aparência é alterada. Esse processo é comum quando se trabalha com mapas. Por exemplo, em um mapa geral, uma escola poderia ter uma representação simbólica (do contrário ficaria invisível), mas em um mapa de detalhe poderia aparecer naturalmente. Por fim, deve-se considerar ainda os casos em que não só a aparência, mas o conteúdo também pode ser mudado na passagem de uma visão para a outra.

Hornbaek et al. [HOR 2001], citando diversos autores [BER 90, PLA 95, CAR 99], apontam quatro benefícios dessa abordagem: (1) A navegação fica mais eficiente já que se poderia interagir direto na janela da visão geral ao invés de na janela de detalhe; (2) a visão geral ajuda a manter o registro da posição corrente em relação ao todo do espaço de informação; (3) a janela com a visão geral poderia ter relevância direta para certos tipos de tarefa; e (4) a abordagem *Visão Geral + Detalhe* fornece aos usuários um sentimento de controle.

Um problema com essa abordagem refere-se à relação espacial indireta entre a janela de visão geral e a de detalhe. Essa falta de uma relação mais orgânica e menos arbitrária no posicionamento das vistas pode sobrecarregar a memória além de aumentar o tempo da busca visual [CAR 99a]. É justamente essa questão que a abordagem *Foco+Contexto*, discutida posteriormente, procura tratar.

Até aqui foram apresentadas as características básicas da abordagem *Visão Geral+Detalhe*. A seguir serão mostrados exemplos de sistemas que de uma forma ou de outra utilizam os conceitos explorados acima.

3.1.1 Uma interface *Visão Geral+Detalhe* para mapas

Hornbaek et al. [HOR 2001] apresentam uma interface construída para a realização de um experimento de avaliação. Trata-se de uma interface para mapas que utiliza a idéia de *Visão Geral+Contexto* na sua forma mais típica (figura 3.1). Na parte superior direita o usuário dispõe de uma janela de visão geral que permite visualizar todo o mapa. A área acinzentada de formato retangular dentro da janela de visão geral representa a parte do mapa que está sendo mostrada na janela de detalhe. Ainda é possível redimensionar essa área através da manipulação do canto direito inferior que está sombreado. Com isso, o retângulo pode ficar maior ou menor, produzindo os respectivos efeitos de *zoom in* e *zoom out* na janela de detalhe.

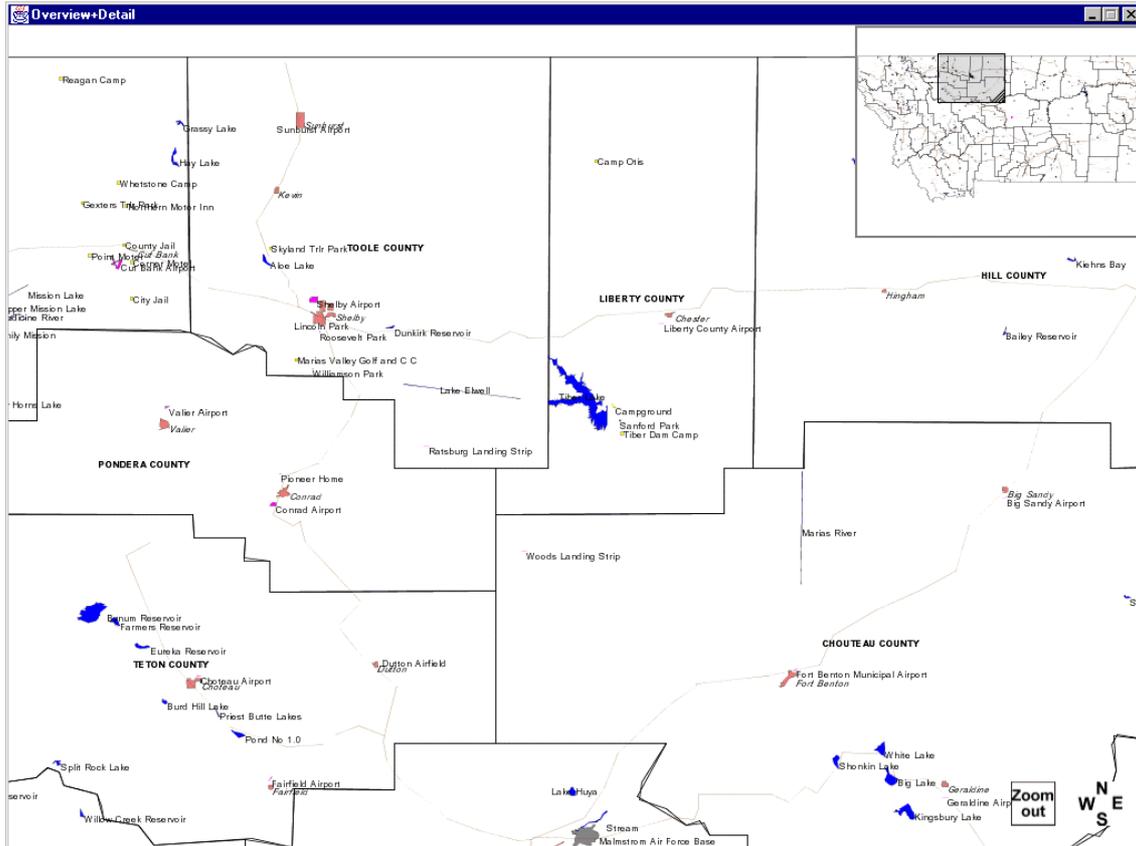


FIGURA 3.1 – Interface Visão Geral+Detalhe utilizada na exibição de mapas (extraída de [HOR 2001])

3.1.2 Life Lines

Life Lines [PLA 96] é uma técnica geral para visualizar sumários de histórias pessoais. Em uma janela com a visão geral (Figura 3.2), múltiplas facetas dos registros são exibidas como regiões distinguidas pela cor de fundo. Histórias e aspectos como condições médicas aparecem como linhas horizontais com mudanças no tamanho ou cor correspondendo a períodos, enquanto ícones indicam eventos discretos como consultas médicas. A cor e a espessura da linha ilustram relacionamentos ou significados dos eventos. *Life Lines* sempre inicia como uma visão geral do registro, e ferramentas permitem ao usuário focalizar uma parte do registro e assim ver mais detalhes. Múltiplas janelas são usadas para exibir relatórios detalhados enquanto se mantém uma visão geral. Arrastando um ícone de evento para a região, espaço é alocado para um relatório formatado de toda a informação associada ao ícone. À medida que mais ícones são colocados na região, ela se divide em segmentos com barras de rolagem independentes, com a posição do novo relatório em relação aos outros, análoga à posição em que o ícone foi colocado na tela.

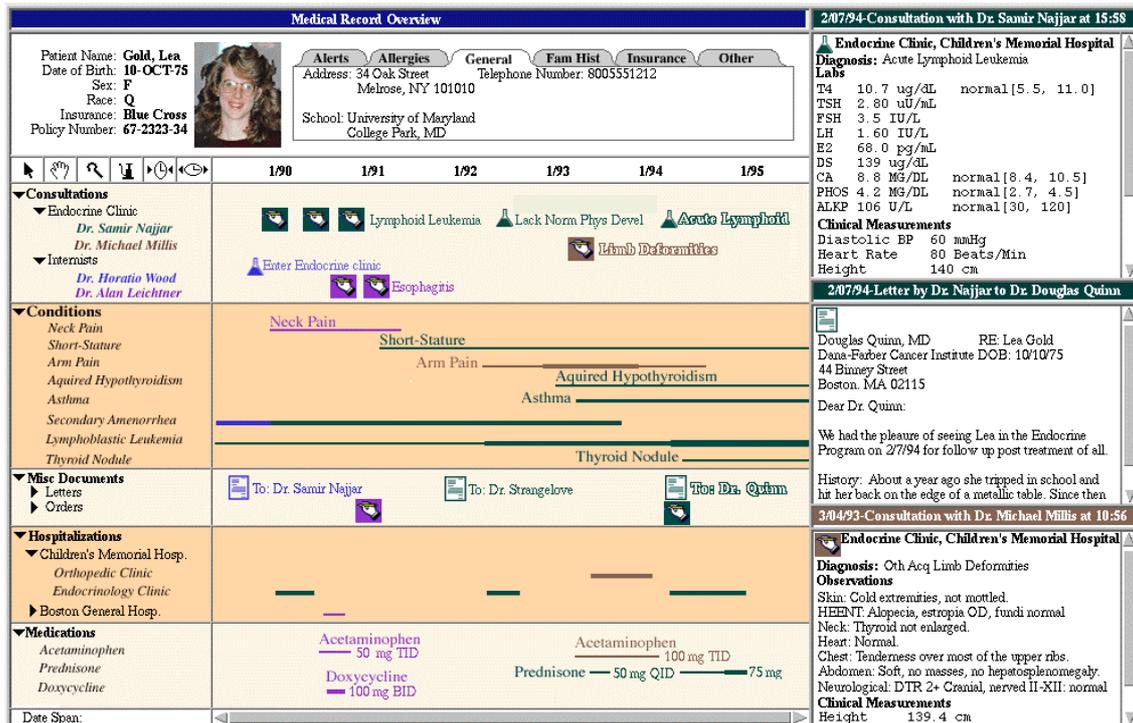


FIGURA 3.2 – *Life Lines* para registros médicos (extraída de [PLA 96])

3.1.3 Microsoft Explorer Browser

Outro sistema em que aparecem algumas características existentes na abordagem *Visão Geral + Detalhe* é o *Microsoft Explorer Browser*. Sua forma de visualização é amplamente utilizada em aplicações que necessitam exibir estruturas de arquivos. Ele possui duas áreas de visualização (figura 3.3). A primeira, localizada à esquerda, tem por objetivo exibir uma visão geral da hierarquia, onde cada nodo é representado por um ícone que denota o seu tipo, seguido por um texto que o identifica. Os nodos que representam arquivos não são exibidos. Eles são posicionados um abaixo do outro, e cada nível é representado a uma certa distância da margem esquerda. Quando o número de nodos a serem exibidos excede as dimensões da janela, são exibidas barras de rolagem, tanto no sentido vertical como no horizontal, para permitir o controle da visualização dos elementos da hierarquia. Nesta situação apenas um trecho da visão geral é exibido.

A janela localizada à direita possui como objetivo a apresentação de todos os filhos do nodo selecionado na janela de contexto, sem a restrição do tipo de nodo. São fornecidas diversas formas de exibição das informações ligadas a esses nodos, que podem ser configuradas pelo usuário. Elas podem variar desde uma representação simplificada, com apenas a relação dos ícones e o texto que identifica cada nodo, até uma listagem com os principais atributos de cada nodo. Cabe destacar que somente os nodos existentes no nível selecionado são exibidos. Se existir uma subárvore cuja raiz pertença a este grupo, seus descendentes não serão exibidos. Caso o número de elementos visuais ocupe uma área maior que a oferecida pela janela, barras de rolagem são anexadas para permitir a navegação.

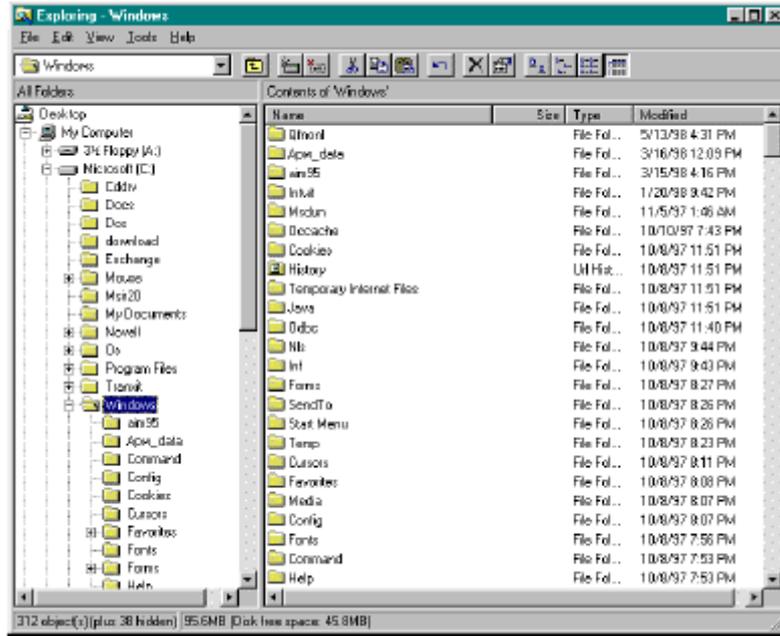


FIGURA 3.3 – Microsoft Explorer Browser

3.2 Abordagem *Foco+Contexto*

A integração da visão de detalhe com a visão geral na mesma área aproveitando a visão periférica é a marca distintiva da abordagem *Foco+Contexto* em relação à anterior. De fato, a própria visão humana se utiliza desse mecanismo. Enquanto enxergamos com nitidez o que está no foco, mantemos, de forma simplificada e/ou distorcida, uma visão periférica que nos permite manter a orientação e aumentar a compreensão do que está em foco, evitando os efeitos negativos da visão de túnel [SCH 96].

Dessa forma, a abordagem *Foco+Contexto* permite dar uma resposta a certas observações críticas que podem ser feitas em relação à abordagem visão geral + detalhe apresentada anteriormente. Primeiro, no caso da variante que utiliza multiplexação no tempo, onde a visão geral e a visão detalhada são exibidas em momentos diferentes, quando se está diante do contexto não há o detalhe para se fazer qualquer tarefa, e por outro lado, quando se visualiza o detalhe, perde-se a orientação. Caso se considere a variante baseada em multiplexação do espaço, onde as visões geral e detalhada compartilham o mesmo espaço de visualização, nota-se que pode ocorrer obstrução visual com a sobreposição de imagens. Além disso, muitas vezes, essa abordagem faz com que o usuário tenha que integrar o detalhe e o contexto mentalmente, já que são exibidos em espaços diferentes [SCH 96]. Esses dois problemas são minimizados na abordagem *Foco+Contexto*, já que neste caso, o contexto está sempre disponível para orientação e a apresentação das duas visões ocorre em uma única área ou janela, evitando assim a possibilidade de sobreposições de visões.

Há outras motivações para se desejar uma abordagem que integre a visão geral àquela de detalhe. Tem sido observado por *designers* que quando a informação é separada em duas áreas de exibição a performance da busca visual e da memória de trabalho degrada [CAR 99b]. Uma analogia que poderia servir de exemplo seria dirigir um carro sem espelho retrovisor, tendo de abandonar temporariamente qualquer contato

com a visão à frente caso se queira ter uma visão do que se passa atrás do veículo. Outra observação que serve de motivação para reforçar a idéia de *Foco+Contexto* é feita por Furnas [FUR 86] de que o interesse dos usuários parece variar de forma sistemática à medida que eles se afastam do objeto principal da atenção. Daí a sugestão de exibir mais informação periférica com um menor número de detalhe em conjunto com a informação completa no foco, admitindo interações que substituam a parte que está no foco para acompanhar as mudanças da atenção do usuário.

A informação periférica usada para formar a área de contexto pode ser alterada ou modificada de várias maneiras. Schaffer et al. [SCH 96] classificam em três as abordagens para efetivar essa tarefa: distorções, filtragem e o uso de representações. Card et al. [CAR 99b] citam pelo menos mais três: agregação seletiva, leituras micro-macro (*micro-macro readings*) e efeitos de destaque (*highlighting*). Estas abordagens são comentadas a seguir:

a) Distorções

Neste caso os itens tem tamanho diminuído à medida que se afastam do ponto de foco. Assim, pode-se dizer, de uma forma genérica, que o número de pixels atribuídos aos objetos é alterado a partir de um processo de distorção que pode ser classificado em três tipos [CAR 99b]: alteração do tamanho dos objetos, alteração do tamanho devido à perspectiva, ou alterações efetuadas no tamanho do próprio espaço.

b) Filtragem

Com a filtragem, trata-se de apresentar visões parciais onde uma função de distância determina que itens devem aparecer ou não na exibição. Essa função poderia incluir outros aspectos além da distância, de acordo com o modelo de interesse desejado.

c) Uso de representações

Aqui se lança mão do uso de abstrações e representações mais simples e menores. Por exemplo, representar um circuito por um ícone. Assemelha-se ao zoom semântico, comum em mapas, citado na seção 3.1.

d) Agregação Seletiva

Trata-se de uma aplicação de “*clustering*” (agrupamento). Itens afastados do foco são agregados numa representação única. À medida que se aproximam do ponto de vista do usuário, seus componentes são separados permitindo análise individual.

e) Leitura micro-macro

Este é o caso em que as informações gráficas em detalhe deixam entrever uma estrutura maior, considerando os itens individuais de uma forma coletiva e cumulativa. Isso é obtido, principalmente, por uma alteração de foco da visão do usuário e não necessariamente através de qualquer interação ou modificação gráfica adicional. Essa estrutura geral emergente possibilita que os itens individuais possam sejam vistos em contexto.

f) Efeitos de destaque

É um caso especial do tipo micro-macro. Os itens são visualmente distinguidos dos demais, de alguma maneira. O conjunto global de itens fornece um ambiente macro onde a leitura micro dos itens destacados individualmente pode ser interpretada.

Convém ressaltar que a essência da abordagem *Foco + Contexto* é a busca da minimização da estrutura de custos da visualização. Desejam-se mapeamentos visuais e suas transformações que minimizem o uso dos limitados recursos seja de espaço de tela para exibição, seja de atenção do usuário. A idéia é aumentar os recursos espaciais disponíveis para processamento detalhado por parte do usuário (processamento serial e mais controlado e que, portanto, demanda mais recursos de atenção) ao mesmo tempo em que mais contexto relevante se torne disponível, dando assim um suporte tanto à busca visual como à memória de trabalho, sem esquecer da melhoria da orientação já que são dadas evidências dos próximos passos disponíveis ao usuário. Dessa forma se amplifica a cognição ao mesmo tempo em que se melhora a estrutura de custo do espaço de informação [CAR 99b].

Algumas técnicas e sistemas que se valem da abordagem *Foco+Contexto* são apresentados a seguir, a título de ilustração das idéias.

3.2.1 *Fisheye Views*

Pode-se destacar Furnas [FUR 81] como um dos pioneiros da abordagem *Foco+Contexto* com a proposta de uma técnica chamada *fisheye view* que recebeu esse nome pelo fato de possuir características análogas às da imagem produzida por uma lente grande angular de uma câmera fotográfica. Nela, os elementos que estão localizados próximos ao local que contém o ponto de interesse do usuário, denominado foco, são exibidos com mais detalhes, que vão diminuindo à medida que se afastam desse ponto. Esse efeito é conseguido através da aplicação de uma função denominada DOI (*Degree-Of-Interest*), que tem por objetivo obter o grau de interesse de um determinado componente fornecendo o nível de detalhe de sua exibição. A função leva em conta a importância do nodo na estrutura e sua respectiva distância em relação ao foco que é determinado pelo usuário. Dessa forma ela é definida como:

$$DOI = API - D$$

onde API (*A Priori Importance*) representa o grau de relevância do elemento na estrutura e D (*Distance*) a distância dele em relação ao foco.

Essa técnica foi utilizada para a exibição do código fonte em programas estruturados, onde as linhas que possuem um valor DOI abaixo de um determinado limite são suprimidas na visualização. Por esta razão essa abordagem foi classificada de *logical fisheye* em contraposição à *graphical fisheye view* [SAR 92] apresentada a seguir. A figura 3.4. ilustra a visualização de um programa em C, onde as linhas em negrito indicam as que seriam exibidas em uma visualização *fisheye*. O trecho que está no interior do retângulo apresenta uma visão normal.

```

1  #define DIG 40
2  #include <stdio.h>
3
4  main()
5  {
6  int c,i,x[DIG/4], t[DIG/4], k=DIG/4, noprint = 0;
7
8  while ((c=getchar()) != EOF) {
9      if (c >= '0' && C<= '9') {
10         x[0] = 10 * x[0] + (c-'0');
11         for (i=1; i<k ; i++) {
12             x[i] = 10 * x[i]
13                 + x[i-1]/10000;
14             x[i-1] %= 10000;
15         }
16     } else {
17         switch (c) {
18             case '+':
19                 t[0] = t[0] + x[0];
20                 for (i=1; i<k; i++) {
21                     t[i] = t[i] + x[i]
22                         + t[i-1]/10000;
23                     t[i-1] %= 10000;
24                 }
25                 t[k-1] %= 10000;
26                 break;
27             case '-':
28                 t[0] = (t[0] + 10000)
29                     - x[0];
30                 for (i=1; i<k; i++) {
31                     t[i] = (t[i] + 10000)
32                         - x[i]
33                         - (1 - t[i-1]/10000);
34                     t[i-1] %= 10000;
35                 }
36                 t[k-1] %= 10000;
37                 break;
38             case 'e' :
39                 for (i=0; i<k; i++) t[i] = x[i];
40                 break;
41             case 'q' :
42                 exit(0);
43             default :
44                 noprint = 1;
45                 break;
46         }
47         if (!noprint) {
48             for (i=k - 1; t[i] <= 0 && i>0; i--);
49             printf("%d",t[i]);
50             if (i > 0) {
51                 for (i--; i >= 0; i--) {
52                     printf("%04d",t[i]);
53                 }
54             }
55             putchar ('\n');
56             for (i=0; i>k ; i++) x[i] = 0;
57         }
58     }
59     noprint = 0;
60 }
61 }

```

FIGURA 3.4 – Visualização normal de um programa em C (extraída de [FUR 86])

Na figura 3.5 são apresentadas apenas as linhas que seriam exibidas através de uma visão *fisheye*. Os três pontos que antecedem determinados números indicam a exclusão de linhas da representação.

```

1 #define DIG 40
2 #include <stdio.h>
...4 main()
5 {
6     int c,i,x[ $DIG/4$ ], t[ $DIG/4$ ], k= $DIG/4$ , noprint = 0;
...8     while ((c=getchar()) != EOF) {
9         if (c >= '0' && C<= '9') {
...16         } else {
17             switch (c) {
18                 case '+':
...27                 case '-':
...38                 case 'e' :
39                     for (i=0; i<k; i++) t[i] = x[i];
40                     break;
41                 case 'q':
...43                 default :
...46             }
47             if (!noprint) {
...57             }
58         }
59         noprint = 0;
60     }
61 }

```

FIGURA 3.5 – Visualização do programa em C mostrado na figura 3.4, através da *fisheye view* (extraída de [FUR 86])

3.2.2 Graphical fisheye view

Sarkar e Brown [SAR 92] estenderam a idéia de Furnas fornecendo uma interpretação gráfica para a visualização *fisheye* denominada de *Graphical Fisheye View* (GFV). Eles introduziram considerações sobre o cálculo utilizado na geração do layout possibilitando que a posição, tamanho e nível de detalhes dos objetos exibidos fossem calculados em função da distância de cada um em relação ao foco e de sua importância pré-estabelecida na estrutura global. Enquanto a visualização *fisheye* de Furnas utilizava como distância uma função pré-definida entre dois nodos, a GFV utiliza a distância Euclidiana.

Outro aspecto que diferencia as duas abordagens é que, na proposta de Furnas, um componente está presente na visualização com todos os seus detalhes ou está completamente ausente, não havendo um controle explícito dos atributos que orientam a geração do layout. Isso causa uma certa descontinuidade no espaço de informação, pois os objetos aparecem e desaparecem da área de visualização quando ocorre a alteração do foco por parte do usuário. Essa descontinuidade pode causar desorientação e tornar a navegação mais difícil [MAC 91]. As figuras 3.6, 3.7 e 3.8 exemplificam os dois tipos de abordagens. A figura 3.6 exibe um grafo com 110 vértices e 124 arestas utilizando uma visualização tradicional.



FIGURA 3.6 – Visualização tradicional (extraída de [SAR 92])

A figura 3.7 mostra o mesmo grafo através de uma visualização *fisheye* com o foco situado no nodo 48, onde a distância utilizada na geração do layout foi o comprimento do caminho mínimo entre dois nodos. Nesta figura, alguns nodos foram suprimidos por terem um caminho mínimo maior que o limite estabelecido.

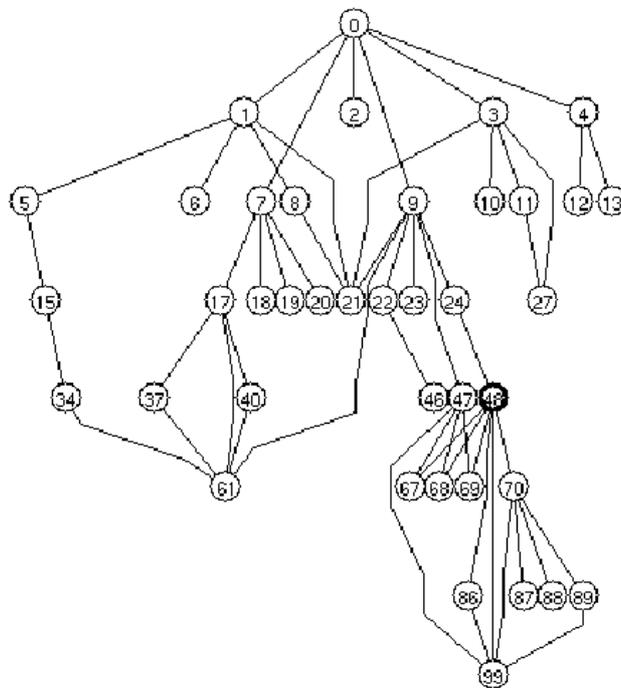


FIGURA 3.7 – Visualização através da *fisheye* de Furnas (extraída de [SAR 92])

A figura 3.8 exibe o grafo da figura 3.6 através de uma GFV com o mesmo nodo 48 situado no foco. Pode-se perceber que alguns nodos também foram excluídos desta

visualização, sendo os próximos ao foco aumentados e os mais distantes diminuídos. Essa característica advém de um parâmetro denominado de custo visual, calculado para cada nodo, que tem por objetivo esconder os menos significativos a fim de acelerar a exibição.

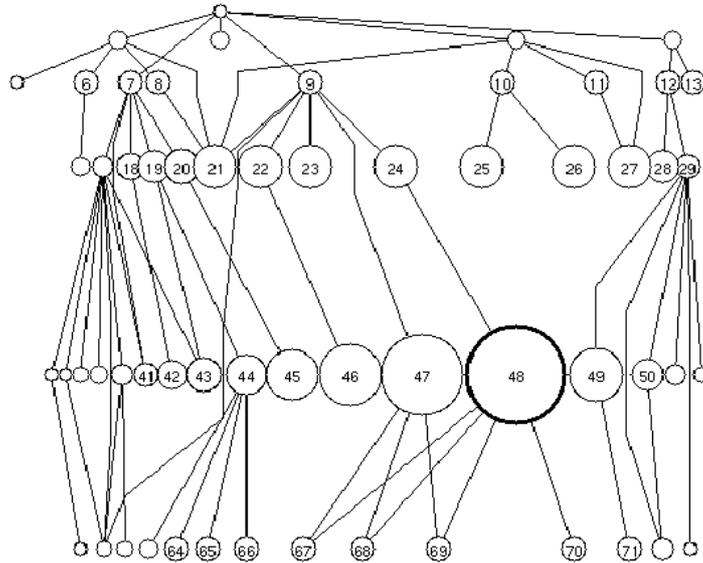


FIGURA 3.8 – Visualização através da GFV (extraída de [SAR 92])

Sarkar e Brown também utilizam o parâmetro API para tratar de forma diferenciada cada nodo conforme o seu grau de importância na estrutura. As figuras 3.9 e 3.10 exemplificam a técnica através da visualização de um grafo que representa o sistema de metrô da cidade de Paris. Os vértices representam as estações e as arestas, conexões entre elas. Neste exemplo foi utilizado como API o número de conexões de cada estação, e o foco foi assinalado na estação *Republique*. A figura 3.9 exibe uma visualização tradicional e a 3.10 uma GFV.

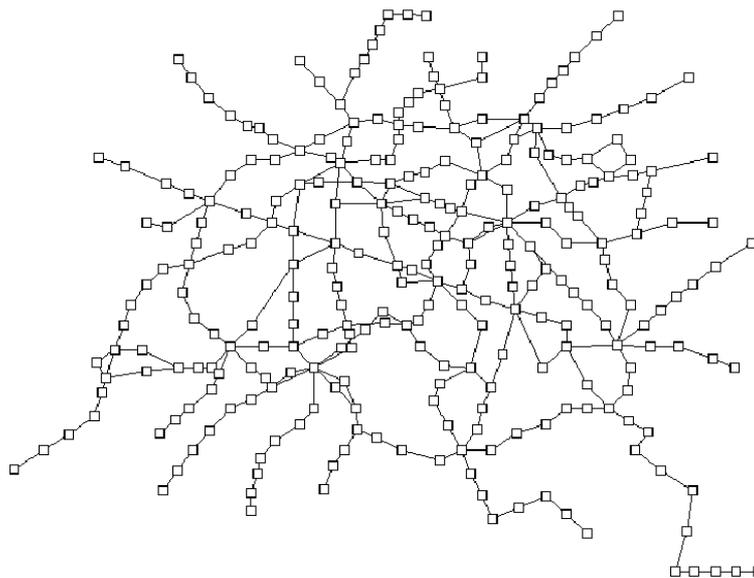


FIGURA 3.9 – Grafo que representa o metrô da cidade de Paris - visualização tradicional (extraída de [SAR 92])

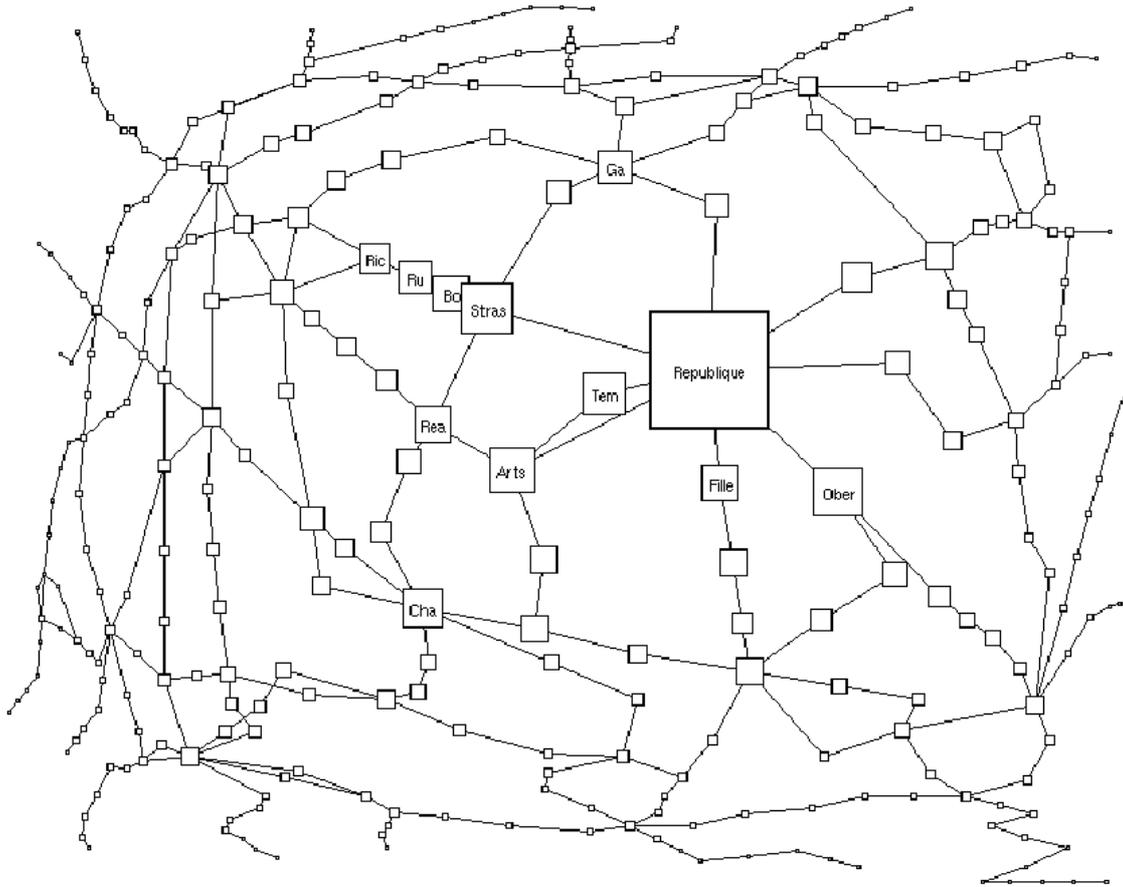


FIGURA 3.10 – Grafo que representa o metrô da cidade de Paris - *Graphical Fisheye View* (extraída de [SAR 92])

3.2.3 *Bifocal Display*

Spence e Apperley, através da técnica *Bifocal Display* [SPE 82], chegaram a resultados similares embora tenham partido de pontos de vista diferentes. Eles investigavam uma forma de melhorar o acesso a grandes espaços de informações existentes em escritórios, normalmente organizados hierarquicamente através de publicações subdivididas em volumes, números e artigos. Para permitir a exibição de um número de informações maior do que seria possível em um sistema tradicional baseado em janelas, a técnica *Bifocal Display* integra detalhe e contexto através de uma área de visualização distorcida. Os itens de informação são posicionados em uma faixa subdividida em três áreas situadas lado a lado. Na parte central situa-se o foco exibindo as informações sem qualquer distorção permitindo, dessa forma, a sua leitura. Nas áreas situadas nas extremidades, cuja função é fornecer o contexto, é aplicada uma distorção na direção X (figura 3.11). Essa distorção é aplicada de forma que todos os objetos contidos na área destinada ao contexto são exibidos de forma idêntica, independente de sua distância em relação ao foco. Através de um mecanismo de *drag-and-drop* o usuário pode mover um objeto situado na área de contexto para o centro a fim de que ele seja exibido de forma detalhada.

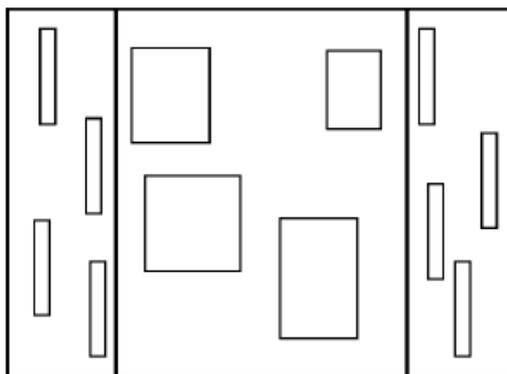


FIGURA 3.11 – Esquema da técnica *Bifocal Display*

Inicialmente a *Bifocal Display* foi implementada para a exibição de texto apresentando cada peça de informação em duas diferentes versões. Por exemplo, enquanto o foco contém uma página de uma determinada publicação, a área destinada ao contexto exibe a data de suas várias edições.

3.2.4 *Perspective Wall*

Segundo Mackinlay et al. [MAC 91] o fato da técnica *Bifocal Display* ser implementada para o espaço bidimensional, faz com que o foco e contexto não sejam integrados de forma suave e intuitiva tornando o relacionamento entre as duas partes pouco óbvio. Além disso, quando o foco é movido, os itens rapidamente se expandem ou se contraem podendo tornar a percepção um pouco confusa. Dessa forma, eles propuseram uma versão alternativa denominada *Perspective Wall*, projetada para o espaço tridimensional. Nela, a interação é executada com o auxílio de animação, promovendo trocas suaves entre as alterações de foco. Essas duas características possuem o objetivo de minimizar os problemas de integração existentes na *Bifocal Display*.

Na técnica *Perspective Wall* as informações são exibidas em uma espécie de parede que possui um painel central para visualização dos detalhes e dois painéis nas extremidades exibidos em perspectiva, onde está localizado o contexto (figura 3.12). São aplicados aos painéis efeitos de sombreamento a fim de destacar a percepção da profundidade.

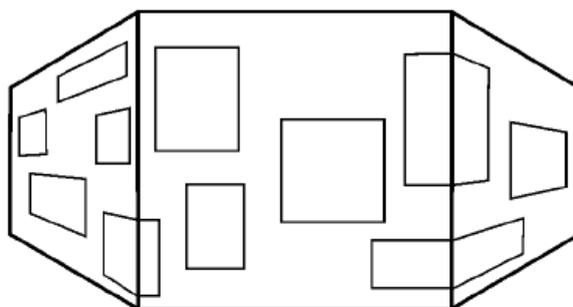


FIGURA 3.12 – Esquema da técnica *Perspective Wall*

O uso da perspectiva faz com que os elementos mais próximos do painel central sejam exibidos em um tamanho maior que aqueles que estão mais distantes. A transição suave de um item selecionado na área de contexto para o painel de detalhe com uso da animação ajuda o usuário a perceber a constância do objeto, deslocando para o sistema de percepção o trabalho que seria requerido do sistema cognitivo para assimilar a nova imagem, após ela ter sido alterada [ROB 89].

3.2.5 Flip Zooming

Outra técnica que possui algumas características encontradas na *Bifocal Display* é a *Flip Zooming* [HOL 97]. Nela as informações são apresentadas em várias áreas de formato retangular. Essas áreas são posicionadas da direita para a esquerda e de cima para baixo de forma a manter a estrutura seqüencial da informação. Uma delas, maior que as demais, é o foco que ocupa a parte central do espaço disponível para a exibição e possui o objetivo de apresentar as informações com maior número de detalhes. As áreas distribuídas em torno deste centro possuem o objetivo de fornecer um contexto ao usuário, e são exibidas na ordem pré-estabelecida para as informações. Ao usuário, é fornecido um mecanismo que permite a seleção da área que ocupará a parte central da janela, que pode ser feita de forma aleatória ou seqüencial permitindo a navegação entre as áreas como se fossem páginas de um livro. A figura 3.13 (a) mostra um esquema da técnica, onde a área 2 está posicionada no centro. Se o usuário selecionar a área 6, o resultado obtido pode ser visto na figura 3.13 (b). Nota-se que o movimento de uma área para a parte central da janela de exibição não altera a seqüência da estrutura visualizada.

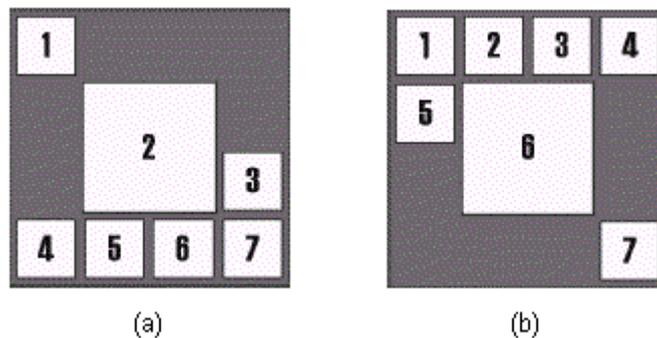


FIGURA 3.13 – Esquema da técnica *Flip Zooming* (extraída de [BJO 2000])

Entre as aplicações que foram desenvolvidas utilizando essa técnica pode-se citar o *Zoom Browser* [HOL 98], que exibe várias páginas WEB simultaneamente distribuídas em um número adequado de pequenos retângulos ligados entre si através de linhas (figura 3.14). Ele oferece ao usuário a possibilidade de seleção de um retângulo a fim de que suas informações possam ser exibidas com uma maior riqueza de detalhes. Aqueles pertencentes ao contexto podem exibir as informações em três maneiras diferentes: como *thumbnails*, como sumário ou uma mistura de ambos. Além disso, o usuário pode optar pela exibição das informações sem o uso de uma área ampliada, criando uma visualização onde todos os dados ocupam uma mesma porção de tela.

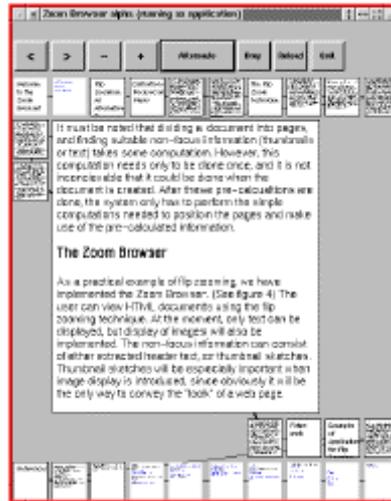


FIGURA 3.14 – *Zoom Browser* (extraído de [HOL 98])

Outro exemplo é o visualizador *The Digital Variants* [BJO 2000], desenvolvido para suportar a pesquisa em literatura sobre as variações de um mesmo texto, como por exemplo comparação de traduções. Cada texto é exibido através da técnica *Flip Zooming* sendo dividido em vários trechos, comparáveis a páginas de um livro. Como pode ser visto na figura 3.15 este sistema suporta múltiplos níveis de contexto além de apresentar simultaneamente duas áreas de detalhe colocadas lado a lado na parte superior da janela de exibição, permitindo a visualização simultânea de dois documentos, a fim de facilitar a sua comparação.

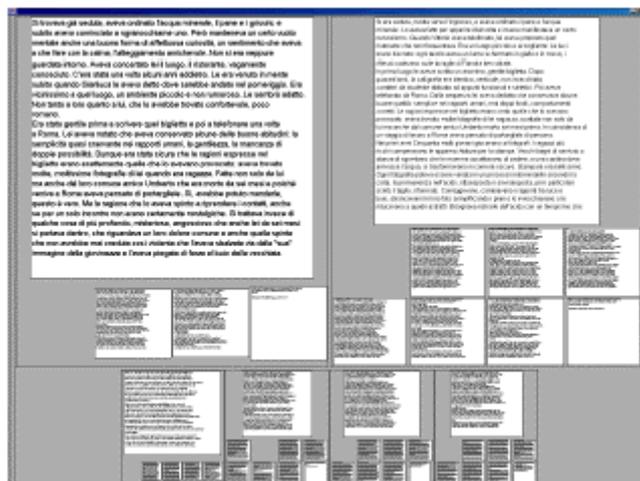


FIGURA 3.15 – Visualizador *The Digital Variants* (extraído de [BJO 2000])

3.3 Informações hierárquicas e visualização de contexto e detalhes

Dentre as técnicas de visualização de informações organizadas hierarquicamente apresentadas no capítulo anterior, percebe-se que em algumas propostas, como *TreeMaps* [SHN 92] e *Time Tube* [CHI 98], não houve a preocupação de proporcionar a exibição de detalhes junto a uma visão geral da estrutura. No entanto, naquelas que utilizam o espaço tridimensional para exibição, como *Cone Tree* [ROB 91], *Reconfigurable Disc Tree* [JEO 98] e *Information Cube* [REK 93], nota-se que o efeito

proporcionado pela projeção perspectiva, onde os elementos mais próximos do observador são exibidos em tamanho maior em relação aos mais distantes, gera uma exibição com características de *Foco+Contexto*. As técnicas *Hiperbolic Tree* [LAM 95] e H3 [MUN 97], foram intencionalmente projetadas levando em conta as características presentes na abordagem *Foco+Contexto*. Técnicas como *Information Slices* [AND 98] e *Sunburst* [STA 2000] podem ser incluídas no grupo daquelas que possuem características da abordagem *Visão + Detalhe*, embora nesta última os detalhes sejam exibidos na mesma área de exibição onde o contexto é apresentado. No próximo capítulo será apresentada a *Bifocal Tree*, que ao utilizar dois focos para a exibição da estrutura integra essas duas abordagens.

4 *Bifocal Tree*

Na última década diversas técnicas foram desenvolvidas para permitir a visualização e navegação de espaços em informação organizados hierarquicamente. Essas técnicas objetivam fornecer ao usuário uma forma de ampliar a percepção das informações existentes através de representações alternativas ou mecanismos de interação que vão além das barras de rolagem existentes nos sistemas tradicionais. Neste capítulo será apresentada a técnica proposta neste trabalho, para visualização de grandes hierarquias, implementada através do *Bifocal Browser*.

Como mencionado no capítulo 2, dentre as técnicas descritas, algumas são baseadas na abordagem de preenchimento de espaço, que é dividido recursivamente de forma que cada subárea represente um nodo da hierarquia, e outras utilizam diagramas de nodos e arestas onde a representação de nodos e arestas assume as mais variadas formas. No capítulo 3 foram discutidas as principais características das técnicas existentes relativamente à maneira como elas utilizam o espaço de exibição. A *Visão Geral+Detalhe* fornece múltiplas visões da estrutura: uma visão geral para orientação, em geral simplificada, e uma visão detalhada da região onde o usuário vai concentrar seu trabalho [CAR 99a]. A abordagem *Foco+Contexto* integra os elementos que estão localizados próximos ao local que contém o ponto de interesse do usuário, denominado foco (exibidos com mais detalhes), com aqueles que tem o objetivo de fornecer o contexto (exibidos com um menor número de detalhes) na mesma área de exibição [CAR 99b].

A técnica *Bifocal Tree* utiliza um diagrama de nodos e arestas e incorpora os conceitos existentes na abordagem *Foco+Contexto* guardando, porém uma divisão mais perceptível da visão de contexto e de detalhe. Ela introduz o uso de um segundo foco para proporcionar ao usuário uma área onde o trecho da estrutura a ser exibido será representado com um maior número de detalhes. Possui ainda alguns mecanismos de interação a fim de facilitar a navegação e a obtenção das informações exibidas na estrutura.

Na seqüência, será descrita a representação visual utilizada, onde serão discutidas as principais características do seu layout. Logo após serão apresentados os principais mecanismos de interação incorporados à técnica. Por último, são detalhados os principais processos envolvidos na implementação e uma descrição do protótipo implementado.

4.1 Representação visual

A técnica *Bifocal Tree* foi desenvolvida com o objetivo de facilitar a percepção das relações hierárquicas existentes entre os elementos da estrutura, apresentando uma visão geral e, ao mesmo tempo, um detalhamento do trecho que contém o ponto de interesse do usuário, mantendo toda a hierarquia visível. Embora não utilize a geometria hiperbólica em seu processo de visualização, ela apresenta algumas características existentes na *Hyperbolic Tree* desenvolvida por Lamping et al. [LAM 95]. Um exemplo de sua representação visual pode ser visto na figura 4.1.

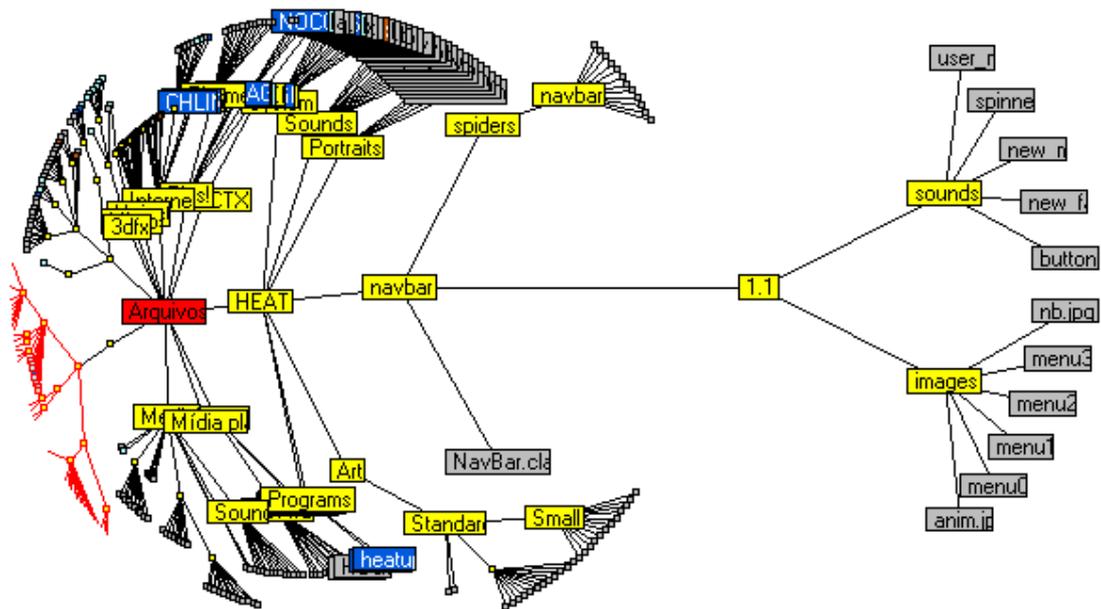


FIGURA 4.1 – *Bifocal Tree* – Uma hierarquia com aproximadamente 370 nodos

Na *Bifocal Tree*, o diagrama de nodos e arestas que representa a estrutura é dividido em dois subdiagramas que definem duas áreas distintas, apresentados de forma radial. A primeira, situada no lado direito da janela de visualização é denominada área de detalhe, pois possui o objetivo de exibir de forma detalhada a subárvore cujo nodo raiz foi selecionado pelo usuário. A segunda, situada no lado esquerdo e denominada área de contexto fornece uma visão global da estrutura. A abordagem *Foco+Contexto* é aplicada a cada área de forma distinta de forma que cada uma possua seu próprio foco, denominados respectivamente de foco de detalhe e foco de contexto. Esses pontos estão localizados lado a lado separados por uma distância arbitrária que pode ser modificada pelo usuário. Uma linha conecta os dois focos com o objetivo de diminuir a sensação de separação entre as duas áreas, ao contrário do que acontece em outros *browsers* como por exemplo o *Microsoft Windows Explorer*, que exibe as informações em duas janelas distintas.

As duas áreas onde a estrutura é visualizada possuem um formato de setor circular com uma medida angular pré-determinada que pode ser alterada pelo usuário. O espaço delimitado pela parte externa dos setores é necessário para minimizar a possibilidade de sobreposição entre as linhas que ligam os nodos aos que estão localizados em cada foco. A figura 4.2 mostra um esquema dos dois setores com o ângulo máximo. Os retângulos localizados em cada centro indicam respectivamente o foco de contexto e o foco de detalhe.

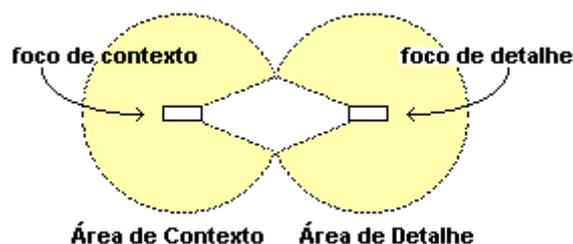


FIGURA 4.2 – Setores circulares utilizados para delimitar as áreas de contexto e detalhe

Os nodos que compõem os dois trechos da estrutura (contexto e detalhe) são distribuídos em cada área segundo um conjunto de regras semelhantes, entretanto com pequenas diferenças, como será descrito posteriormente. Para o posicionamento de cada elemento é utilizado um layout radial, onde o nodo raiz situa-se no centro e cada um de seus filhos distribuídos em torno de círculos concêntricos. O círculo de maior raio representa o local onde serão exibidos aqueles que estão no nível mais baixo da estrutura.

Para obter o efeito *fisheye* [FUR 81] característico da abordagem foco+contexto, o tamanho dos retângulos que representam os nodos, bem como suas coordenadas são obtidos levando-se em conta a distância destes em relação ao foco. Os nodos mais distantes são exibidos com dimensões menores localizando-se relativamente mais próximos entre si do que aqueles que se situam nas proximidades do foco. Além disso, aqueles situados após uma distância arbitrária são excluídos da representação. Essa estratégia foi adotada para evitar a exibição e manipulação dos elementos que estão muito distantes do ponto de interesse, baseado na idéia de que eles seriam pouco identificáveis no contexto geral da hierarquia devido à pequena dimensão das linhas que os unem, além da redução do custo computacional na geração da imagem. Entretanto, caso seja necessário, eles serão recuperados para exibição através da navegação na estrutura como será discutido adiante.

Como mencionado anteriormente, os nodos são exibidos na forma de retângulos com tamanho dependente da sua localização em relação ao foco. A função que determina a sua largura é a mesma usada para calcular a distância de cada nodo ao foco. A altura dos retângulos é fixada em um tamanho adequado a exibição de texto. A partir de uma determinada distância limite, a representação dos nodos passa a ser alterada para um pequeno quadrado de tamanho constante. Essa estratégia foi adotada para minimizar a oclusão que ocorre devido à manutenção da altura dos retângulos inalterada, reduzindo o esforço cognitivo necessário para distinguir um nodo específico. Embora o texto não seja exibido nos pequenos retângulos, as operações de interação podem ser executadas sobre eles. Quando o nodo representa a raiz de um subárvore que foi excluída da representação através de uma operação de poda, um pequeno retângulo é adicionado na extremidade esquerda para sinalizar este fato. A figura 4.3 mostra um trecho de uma estrutura que ilustra cada tipo de representação, onde os nodos **Font** e **Reader** identificam as raízes de subárvores podadas da estrutura.

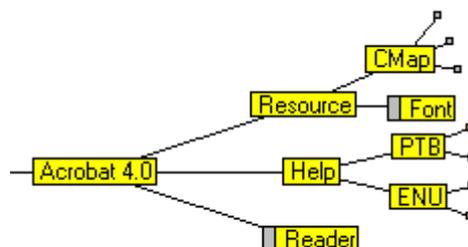


FIGURA 4.3 – Tipos de representação de nodos

Cores são utilizadas para identificar os tipos de nodos. O nodo raiz é exibido na cor vermelha para fornecer um ponto de referência ao usuário, já que através da navegação ele pode ser deslocado para qualquer lugar da área de exibição. Outros

atributos, como por exemplo o tipo de arquivo, podem ser identificados com o uso da cor, que também destaca, na área de contexto, a última subárvore que ocupou a área de detalhe. Isso minimiza a possível desorientação que pode ocorrer durante as rotações e translações aplicadas à subárvore quando um novo nodo é selecionado para o foco de detalhe, já que essas transformações podem alterar de forma significativa a distribuição dos elementos em todo o diagrama. A figura 4.4 ilustra um exemplo da representação após a seleção do nodo **Sounds**. A subárvore que estava na área de detalhe antes da seleção (nodo **navbar**) é exibida com as suas arestas na cor vermelha.

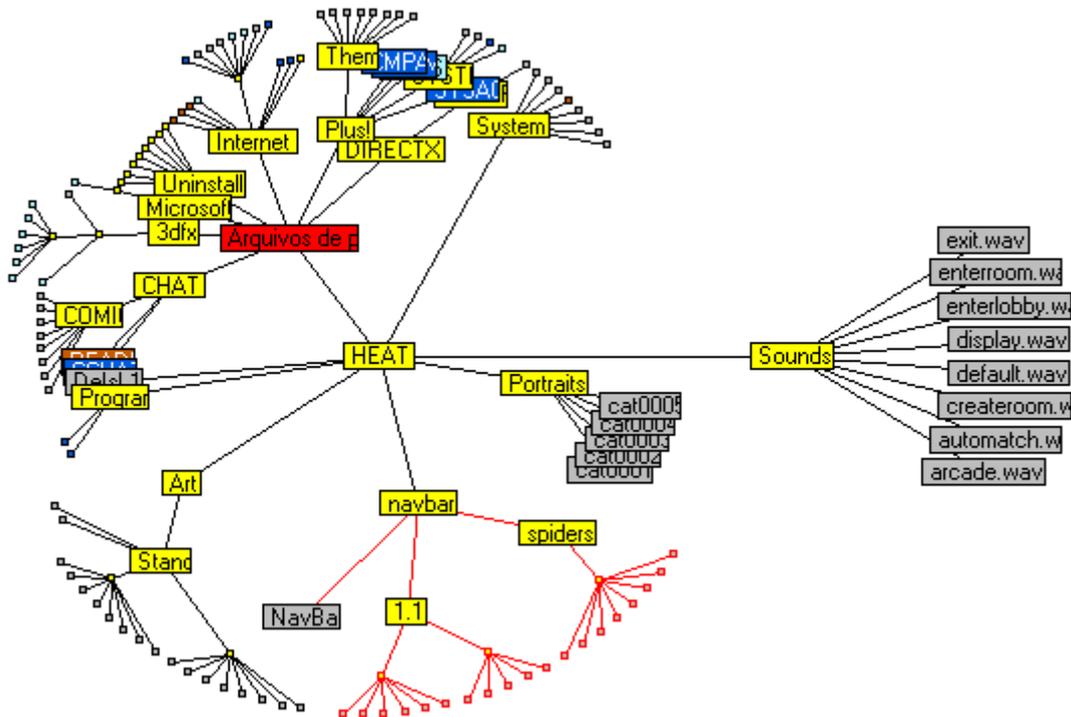


FIGURA 4.4 – Destaque da última subárvore que ocupou a área de detalhe

Para proporcionar um melhor aproveitamento do espaço da área de contexto, a distribuição dos nodos em torno do nodo raiz é feita levando-se em conta o número de nodos-folha existentes em cada subárvore. Dessa forma, a subárvore com um maior número de folhas ocupará um setor com uma maior porção angular. Essa regra é aplicada recursivamente a cada subárvore presente na estrutura. No entanto, na área de detalhe, o objetivo é fornecer uma melhor percepção dos relacionamentos hierárquicos representados pela árvore que possui como nodo raiz o ponto de interesse do usuário. Para atingir esse objetivo, os nodos localizados no primeiro nível são uniformemente distribuídos levando-se em conta nesta divisão apenas o número de nodos filhos daquele localizado no foco. Dessa forma, cada um receberá uma mesma porção angular mantendo o mesmo nível de importância na representação. A partir do segundo nível a divisão é executada conforme a regra descrita para a área de contexto. Essa diferença no layout é mais evidente em árvores não-balanceadas. A figura 4.5 mostra uma estrutura utilizando as regras adotadas, e a figura 4.6 o layout gerado caso fossem utilizadas as mesmas regras aplicadas à área de contexto.

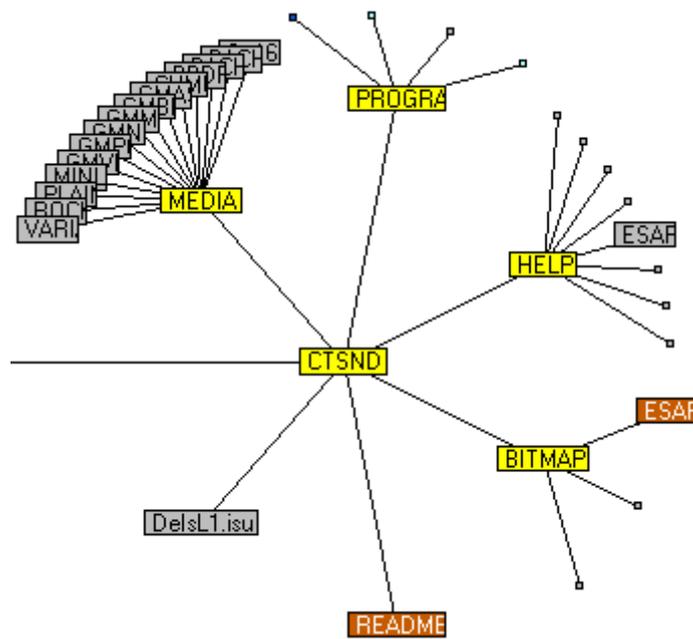


FIGURA 4.5 – Porção angular ocupada pelos nodos do primeiro nível inversamente proporcional ao **número de filhos** da raiz da subárvore (nodo **CTSND**)

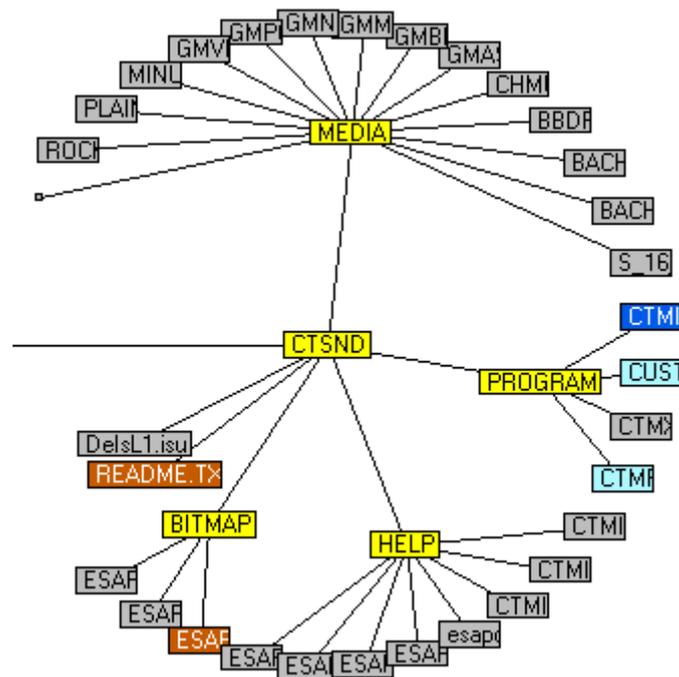


FIGURA 4.6 – Porção angular ocupada pelos nodos do primeiro nível proporcional ao **número de nodos folhas** existentes na subárvore (nodo **CTSND**)

O valor angular inicial do setor onde os nodos são distribuídos na área de detalhe é de 315° (figura 4.7a), mas pode ser reduzido dependendo do número de nodos contidos na subárvore (figura 4.7b). Essa redução é aplicada para permitir que trechos pequenos sejam exibidos sem o espalhamento causado por uma medida angular muito grande.

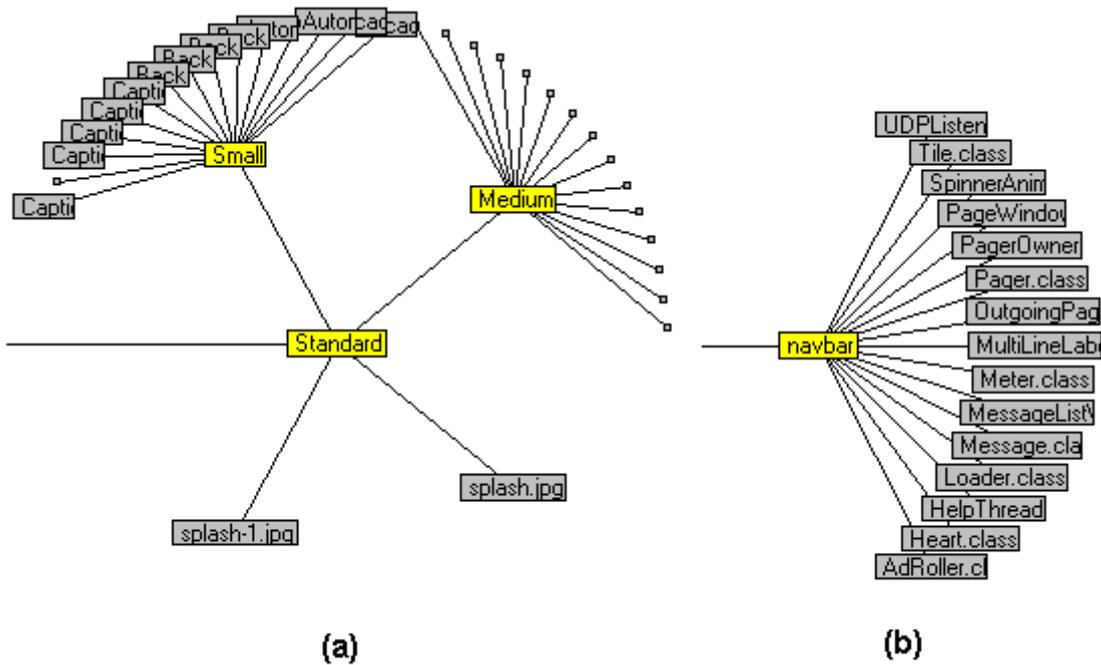


FIGURA 4.7 – (a) Setor da área de detalhe com 315° (b) setor da área de detalhe com a medida angular proporcional ao número de nodos folhas

No entanto, quando o nodo raiz estiver posicionado no foco de detalhe, o círculo inteiro será utilizado na distribuição dos nodos, já que não haverá nodo algum na área de contexto (figura 4.8).

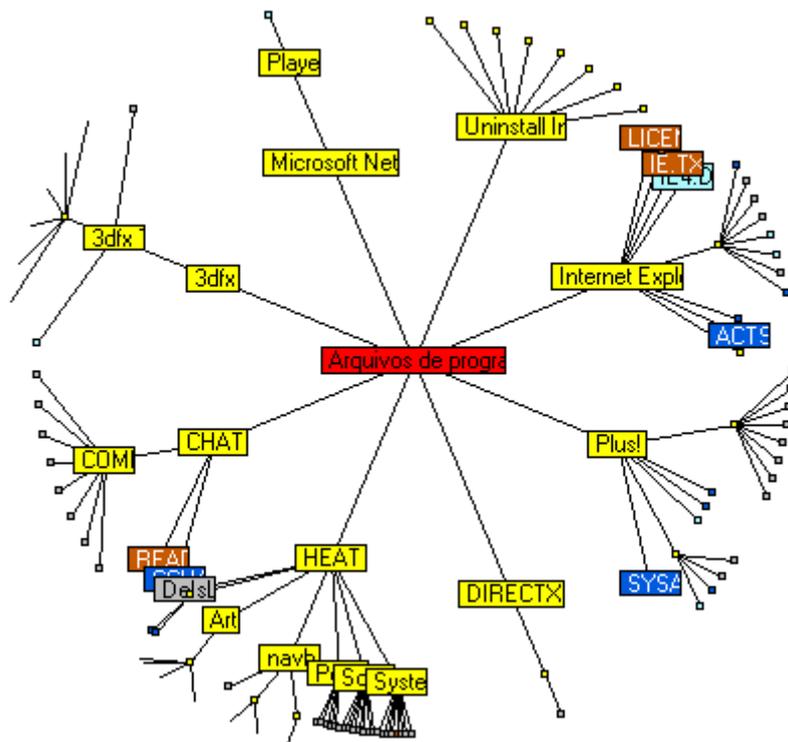


FIGURA 4.8 – Estrutura com o nodo raiz localizado na área de detalhe

Além da medida angular do setor, outra característica geométrica importante refere-se à distância do foco aos nodos localizados no primeiro nível. Ao invés de

mantê-la constante, ela é incrementada suavemente para permitir que os retângulos que representam os nodos sejam exibidos com o mínimo de sobreposição. Considerando como referência o círculo trigonométrico, essa regra é aplicada aos nodos situados no quadrante I e IV; nos demais, a distância é mantida constante. Sem essa particularidade, conforme eles se aproximassem do ponto mais alto (ou baixo) do círculo, a sobreposição seria inevitável. A figura 4.9 ilustra a diferença entre a distribuição com as distâncias variáveis (figura 4.9b) e a que utiliza o círculo como base na distribuição dos nodos (figura 4.9a).

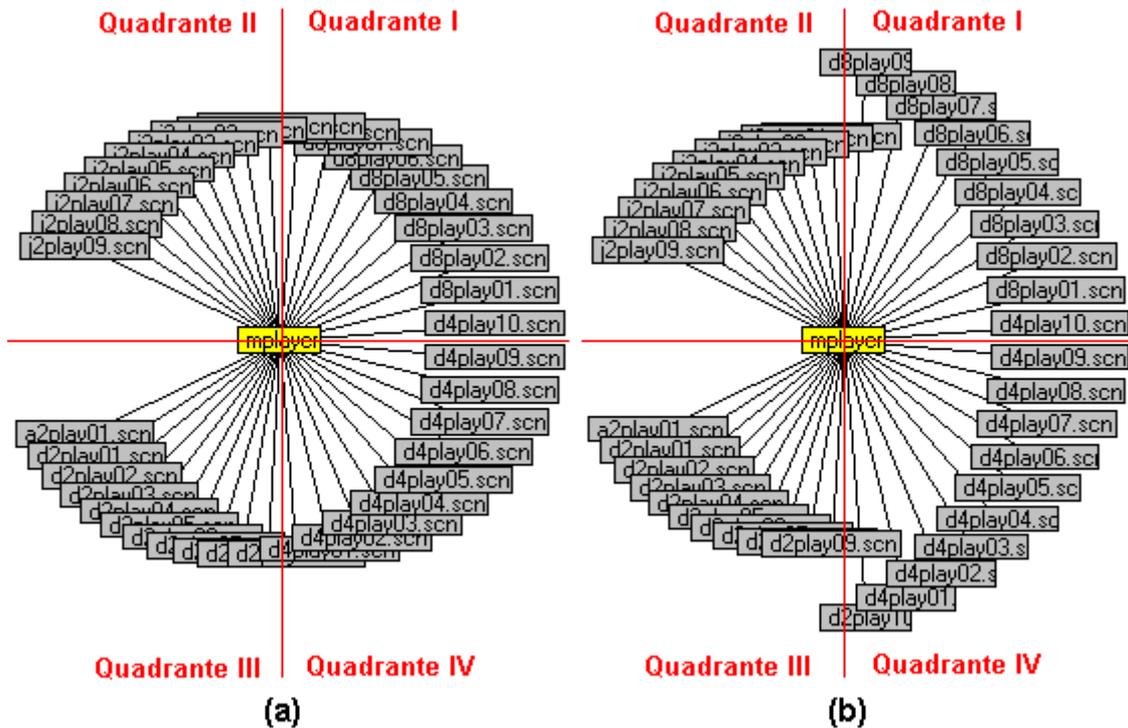


FIGURA 4.9 – Distribuição dos nodos com distância fixa (a) e com distância variável em relação ao foco (b)

4.2 Mecanismos de interação

Na visualização de uma grande estrutura, freqüentemente uma representação visual estática não é suficiente para permitir a sua perfeita compreensão. Dessa forma é necessário agregar à representação um conjunto de funções que provoquem alterações de forma que novos aspectos da estrutura possam ser observados. Dessa forma foram incorporados alguns mecanismos de interação ao *Bifocal Tree*, os quais serão descritos a seguir.

A seleção é o principal mecanismo fornecido ao usuário. Ela pode ser aplicada a qualquer nodo e tem como função movê-lo automaticamente para o foco da área de detalhe juntamente com toda a sua subárvore, permitindo que os seus nodos filhos possam ser exibidos com mais detalhes. Essa operação modifica também a área de contexto, já que pai do nodo selecionado é transladado para o foco da área de contexto. Além da translação, a aplicação de uma rotação na estrutura visualizada na área de contexto também é necessária a fim de permitir que a conexão entre as duas áreas possa ser estabelecida sem que ocorram sobreposições. Caso o novo nodo selecionado seja irmão daquele que está no foco da área de detalhe, a rotação somente é aplicada à área

de contexto. As figuras 4.10 e 4.11 apresentam dois instantes de uma seleção. A figura 4.10 mostra o estado anterior e a figura 4.11 ilustra a situação da representação após a seleção do nodo **navbar**.

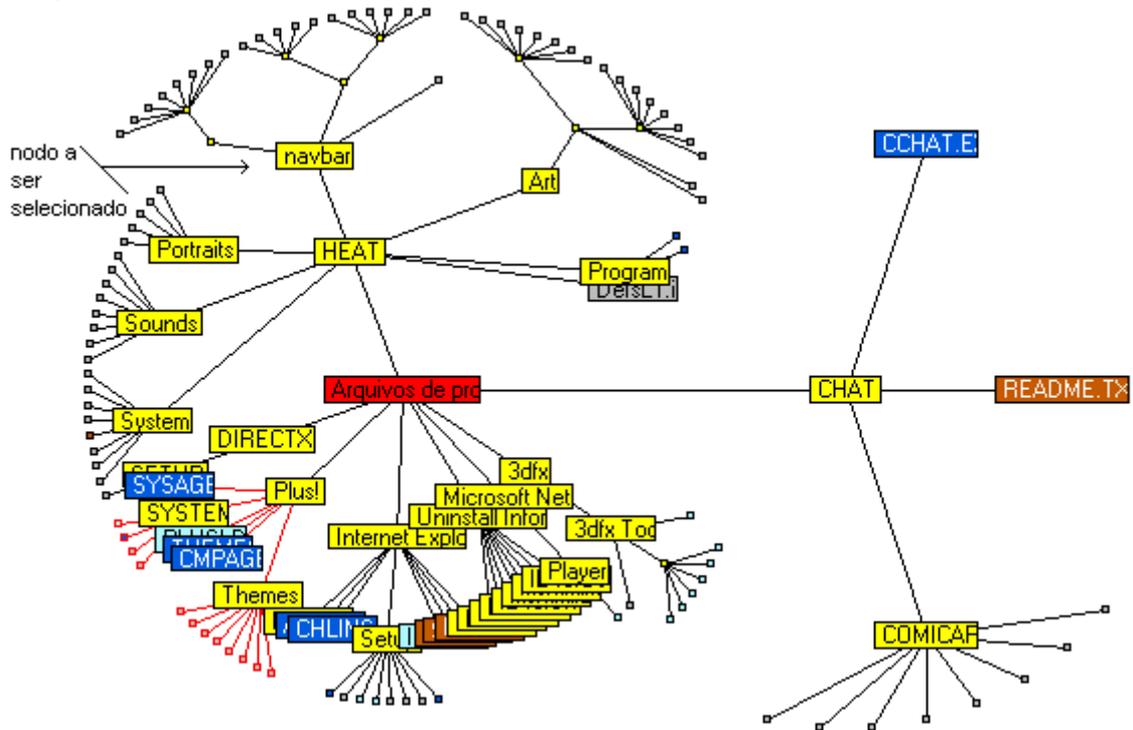


FIGURA 4.10 – Estado da representação antes da seleção do nodo **navbar**

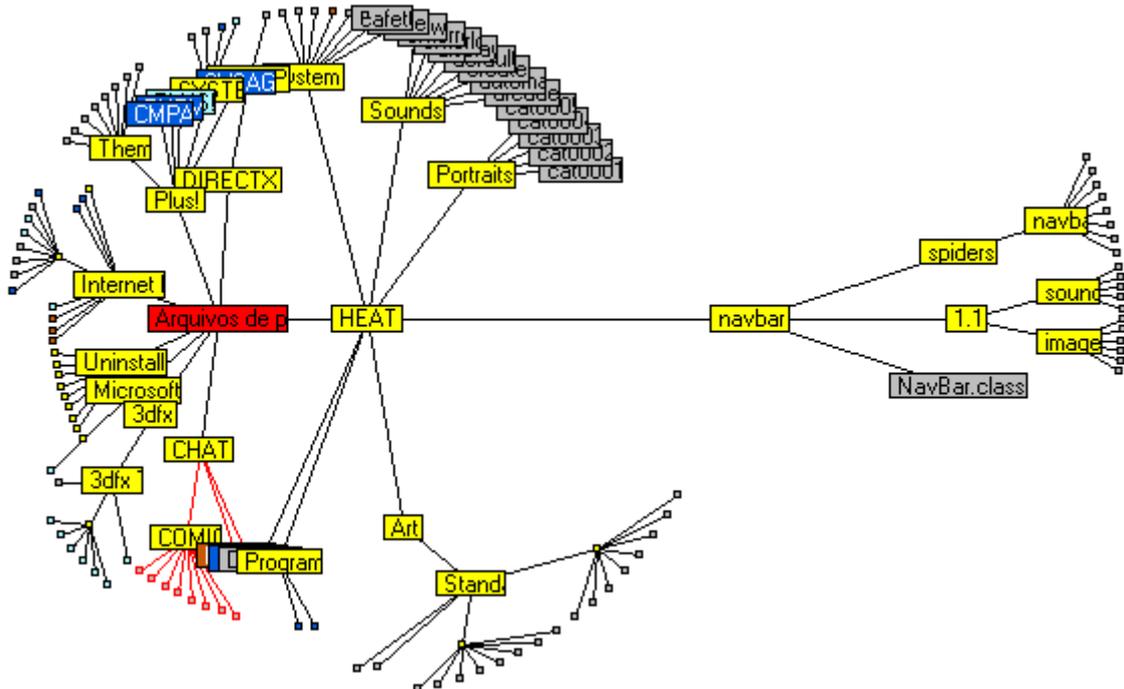


FIGURA 4.11 – Estado da representação após a seleção do nodo **navbar**

A existência de muitos filhos (acima de 20) no primeiro nível da área de detalhe produz sobreposição nos nodos situados nos quadrantes II e III, já que aqueles localizados nos quadrantes I e IV são distribuídos de forma a evitá-la. Dessa forma foi incluído um mecanismo para rotacionar a subárvore exibida na área de detalhe nos sentidos horário e anti-horário permitindo que os nodos sobrepostos sejam exibidos no

local onde eles possam ser perfeitamente identificados. A figura 4.12 exibe dois instantes dessa operação.

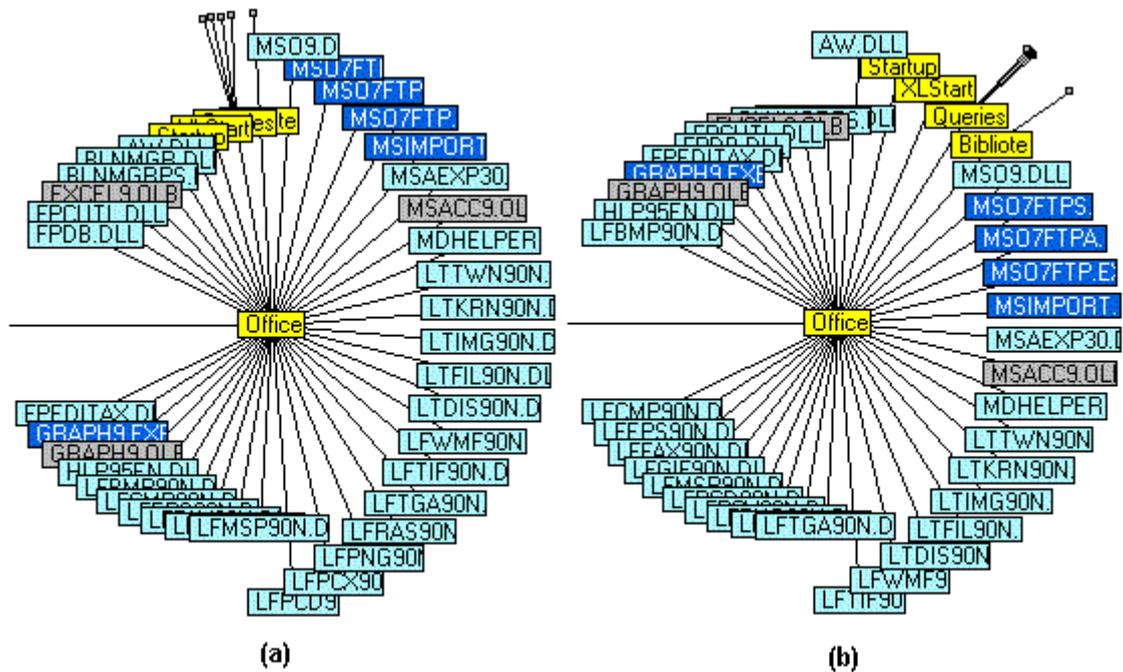


FIGURA 4.12 – Área de detalhe antes (a) e depois da rotação (b)

Outra característica apropriada para a navegação em hierarquias que possuem nodos com muitos filhos é a capacidade de exibição, em uma janela separada, dos nodos irmãos daquele selecionado. Esse mecanismo permite que, a partir desta relação, seja possível executar o deslocamento de um nodo para o foco da área de detalhe. Ele torna-se indispensável quando o nodo desejado está localizado em algum ponto em que ocorrem sobreposições que dificultam a sua seleção a partir da representação. A figura 4.13 ilustra essa situação.

Mecanismos de poda e expansão, naturalmente presentes em sistemas de visualização de hierarquias, também foram definidos. Através da poda, uma subárvore inteira pode ser excluída da representação visual diminuindo a sobrecarga cognitiva causada pela exibição de um grande número de nodos, enquanto a expansão permite restaurar a visualização completa. Os nodos que representam a raiz de uma subárvore podada são diferenciados através do acréscimo de um pequeno retângulo na extremidade esquerda de cada nodo. A figura 4.14 mostra a representação de um trecho da estrutura antes e após a poda da subárvore abaixo do nodo 1.1.

Outro mecanismo presente é o de *undo/redo*, que permite que o usuário navegue pelas seleções já executadas. Ele é obtido através do armazenamento de uma lista que contém a ordem dos nodos que foram posicionados no foco de detalhe.

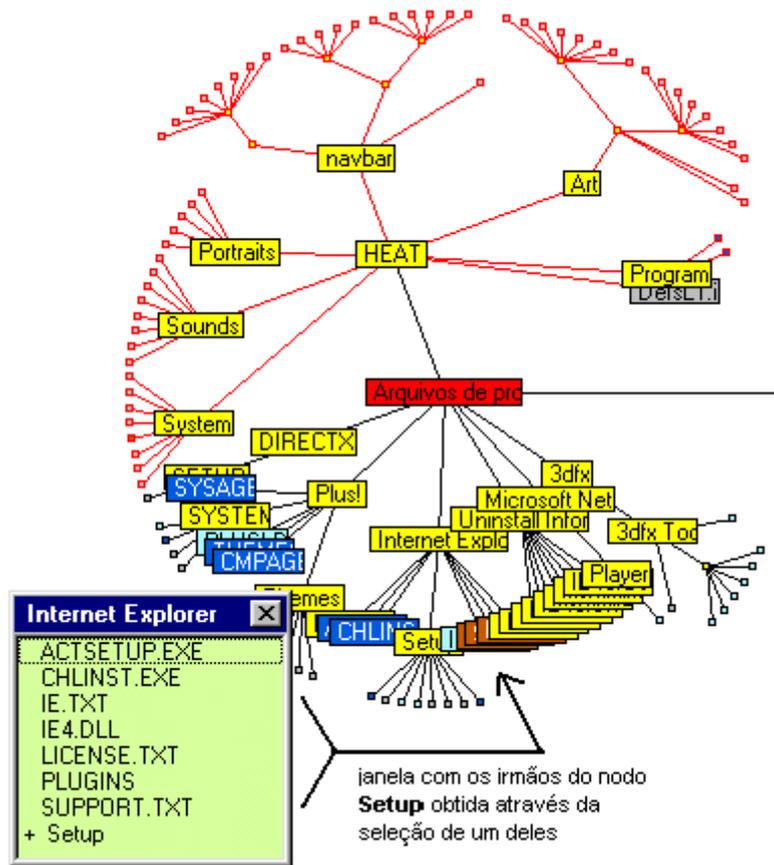


FIGURA 4.13 – Janela para seleção de nodos irmãos sobrepostos

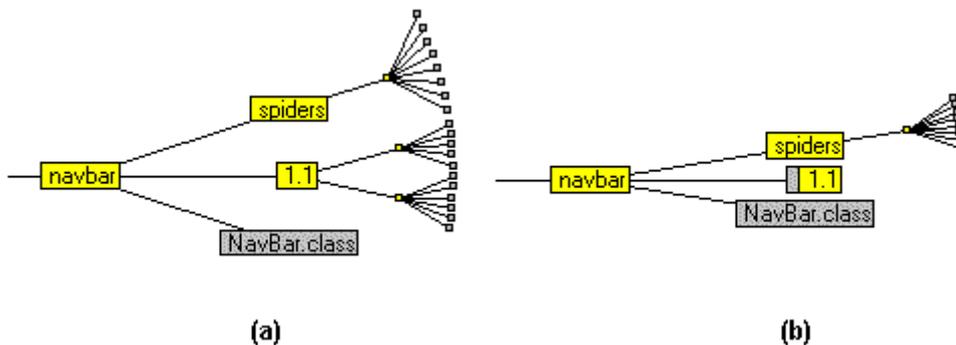


FIGURA 4.14 – Estado da representação antes (a) e depois da operação de poda (b)

4.3 Implementação

Nesta seção serão apresentados os principais algoritmos envolvidos na implementação da representação visual da *Bifocal Tree*. Inicialmente será descrito o processo responsável pela obtenção das coordenadas dos nodos localizados em cada uma das áreas de exibição. Elas são obtidas utilizando-se como base um sistema de coordenadas polares com o nodo raiz localizado na origem. Posteriormente, será apresentado o processo de mapeamento que gera a visão *fisheye* dentro dos setores de círculos onde a estrutura é exibida. A fim de facilitar a compreensão, o processo será

ilustrado através de figuras que exibem uma hierarquia (figura 4.15) de apenas 18 nodos.

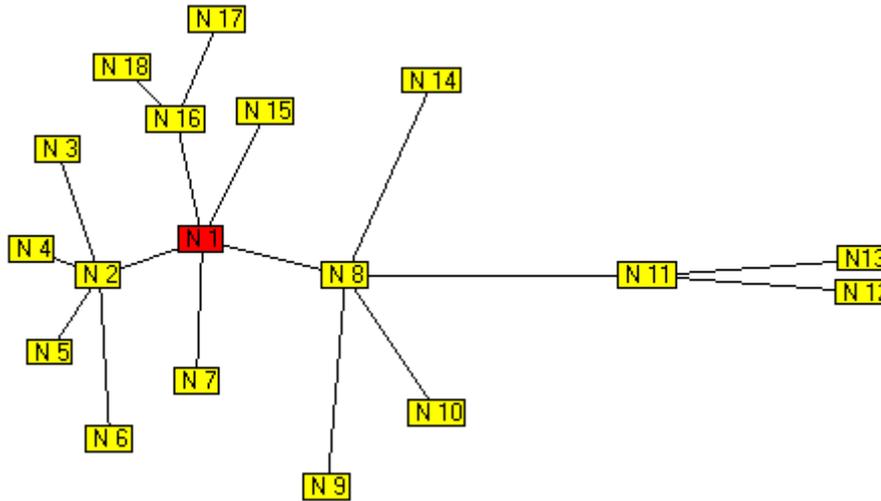


FIGURA 4.15 – Hierarquia com o total de 18 nodos

4.3.1 Layout da área de contexto

Como mencionado anteriormente, os nodos de cada nível da hierarquia são distribuídos em círculos concêntricos em cujo centro está localizado o nodo raiz. O algoritmo utilizado baseia-se no *Leaf-Based Layout* [WIL 99] onde cada nodo folha recebe uma mesma porção do espaço da área de visualização. Dessa forma deve-se obter, para cada nodo, o número de folhas existentes na subárvore na qual ele é o nodo raiz. Nessa contagem não são consideradas as folhas que fazem parte de subárvores que foram podadas da estrutura. A tabela 4.1 apresenta o número de nodos folhas de cada subárvore contida na hierarquia da figura 4.15.

TABELA 4.1 – Número de nodos folhas contidos em cada subárvore

| Raiz de cada subárvore | | | | | | | | | | | | | | | | | |
|------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 13 | 4 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 |

Para obter a medida angular do setor ocupado por cada subárvore, inicialmente é necessário calcular o ângulo necessário para a exibição de cada nodo folha. Ele é obtido dividindo-se a medida angular total utilizada para a área de contexto pelo número de nodos folhas que efetivamente serão exibidos neste local (Eq. 4.1). Dessa forma exclui-se do total, aqueles que serão visualizados na área de detalhe.

$$AngNodoFolha = \frac{AngAreaContexto}{[NFo(raiz) - NFo(AreaDetalhe)]} \quad (4.1)$$

onde:

AngNodoFolha é a medida angular do setor ocupado por um nodo folha;

AngAreaContexto é a medida angular total do setor da área de contexto ocupado pela estrutura;

$NFo(\text{raiz})$ é o número de nodos folhas existente na estrutura;
 $NFo(\text{AreaDetalhe})$ é o número de nodos folhas existente na subárvore que será exibida na área de detalhe.

Para a árvore da figura 4.15, considerando-se um ângulo de 315° (*default*) para o setor utilizado na área de contexto e o nodo **N11** posicionado na área de detalhe, cada nodo folha ocupará um setor de aproximadamente $28,63^\circ$ (figura 4.16) obtido através da equação 4.1.

O valor da medida angular de cada setor ocupado pelas subárvores é obtido por dois processos distintos conforme a situação de seus componentes em relação à parte da estrutura que será exibida na área de detalhe. Considerando X o nodo que será posicionado no foco da área de detalhe, pode-se classificar as subárvores em dois tipos: aquelas que não contém o pai de X e que o contém. No primeiro caso, a medida angular do setor necessário para exibí-la é diretamente proporcional ao número de folhas existentes na subárvore e pode ser calculada da seguinte forma:

$$AngSub\grave{a}rvore(i) = NFo(i) * AngNodoFolha \quad (4.2)$$

onde:

$AngSub\grave{a}rvore(i)$ é o ângulo do setor ocupado pela subárvore i;
 $NFo(i)$ é o número de nodos folhas existente na subárvore i;
 $AngNodoFolha$ é a medida angular do setor ocupado por um nodo folha.

Quando a subárvore contém o pai do nodo que está localizado na área de detalhe, o cálculo da medida angular leva em conta apenas os nodos que serão efetivamente exibidos neste setor. Portanto, é necessário excluir o número de nodos folhas que estão na área de detalhe. Entretanto, deve-se acrescentar ao valor obtido, a medida angular do setor localizado na área de contexto, que não será ocupado pela estrutura cujo valor é obtido pela diferença entre a medida total do círculo e o ângulo atribuído ao setor ocupado pela área de contexto (Eq. 4.3).

$$AngSub\grave{a}rvore(i) = [NFo(i) - NFo(\text{AreaDetalhe})] * AngNodoFolha + 360 - AngAreaContexto \quad (4.3)$$

onde:

$AngSub\grave{a}rvore(i)$ é o ângulo do setor ocupado pela subárvore i;
 $NFo(i)$ é o número de nodos folhas existente na subárvore i;
 $NFo(\text{AreaDetalhe})$ é o número total de nodos folhas existente na subárvore que será exibida na área de detalhe;
 $AngNodoFolha$ é a medida angular do setor ocupado por um nodo folha;
 $AngAreaContexto$ é a medida angular do setor da área de contexto ocupado pela estrutura.

As tabelas 4.2 e 4.3 apresentam a medida angular obtida para cada nodo da hierarquia utilizada como exemplo, considerando que o nodo **N11** está localizado no foco da área de detalhe.

TABELA 4.2 – Medida angular dos setores ocupados pelos nodos de 1 a 9 da hierarquia utilizada como exemplo.

| N 1 | N 2 | N 3 | N 4 | N 5 | N 6 | N 7 | N 8 | N 9 |
|--------|---------------|-------|-------|-------|-------|-------|---------------|-------|
| 315,00 | 114,52 | 28,63 | 28,63 | 28,63 | 28,63 | 28,63 | 130,89 | 28,63 |

TABELA 4.3 – Medida angular dos setores ocupados pelos nodos de 10 a 18 da hierarquia utilizada como exemplo.

| N 10 | N 11 | N 12 | N 13 | N 14 | N 15 | N 16 | N 17 | N 18 |
|-------|---------|---------|---------|-------|-------|-------|-------|-------|
| 28,63 | --- (*) | --- (*) | --- (*) | 28,63 | 28,63 | 57,27 | 28,63 | 28,63 |

(*) Nodos que não são exibidos na área de contexto

A figura 4.16 apresenta os setores ocupados pelos dois tipos de subárvores mencionadas anteriormente que estão representados pelas áreas preenchidas. Aquela cuja raiz é o nodo **N2** (não contém o pai do nodo **N11**) é exibida em um setor medindo $114,54^\circ$ obtido pela Eq. 4.2, enquanto que a subárvore cuja raiz é o nodo **N8** (contém o pai do nodo **N11**) possui um setor de $130,89^\circ$ resultado da Eq.4.3. A subárvore situada na área de detalhe está representada através de linhas tracejadas.

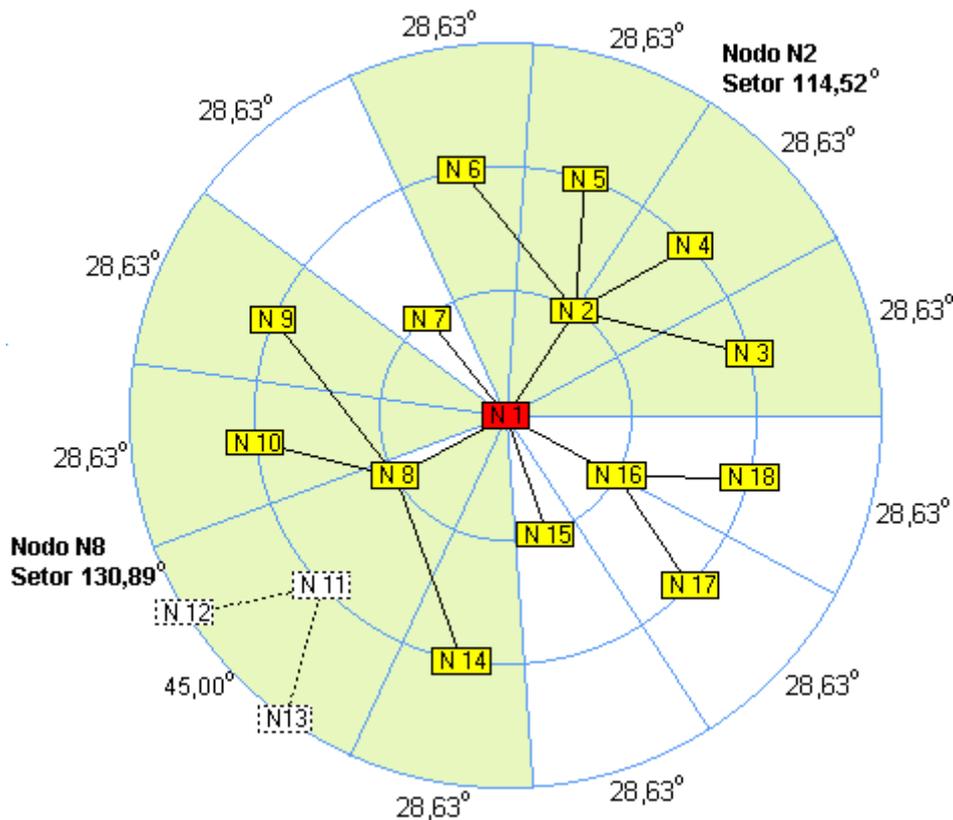


FIGURA 4.16 – Distribuição das subárvores nos setores

Após a determinação da medida angular do setor ocupado em cada subárvore, torna-se possível calcular as coordenadas polares do nodo que representa a sua

respectiva raiz. A componente angular é obtida acumulando-se em cada nível as medidas dos setores vizinhos com o acréscimo da metade do ângulo do seu setor para que o nodo seja posicionado no respectivo centro. A componente relacionada à distância é obtida através da multiplicação do número do nível onde o nodo se encontra por uma medida previamente estabelecida que representa a distância entre os níveis. Esse processo é aplicado recursivamente para todas as subárvores que compõem a estrutura. A figura 4.17 ilustra esse procedimento utilizando o nodo **N4** como exemplo. As tabelas 4.4 e 4.5 apresentam as coordenadas polares de cada nodo.

TABELA 4.4 – Coordenadas polares dos nodos de 1 a 9 da hierarquia utilizada como exemplo.

| | N 1 | N 2 | N 3 | N 4 | N 5 | N 6 | N 7 | N 8 | N 9 |
|----------------|------|-------|-------|-------|-------|--------|--------|-------|--------|
| Ângulo (graus) | 0,00 | 57,26 | 14,31 | 42,94 | 71,57 | 100,20 | 128,83 | 208,6 | 157,46 |
| Distância | 0,00 | 0,50 | 1,00 | 1,00 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 |

TABELA 4:5 – Coordenadas polares dos nodos de 10 a 18 da hierarquia utilizada como exemplo.

| | N 10 | N 11 | N 12 | N 13 | N 14 | N 15 | N 16 | N 17 | N 18 |
|----------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Ângulo (graus) | 186,09 | ---(*) | ---(*) | ---(*) | 259,72 | 288,75 | 331,30 | 316,98 | 345,61 |
| Distância | 1,00 | ---(*) | ---(*) | ---(*) | 1,00 | 0,50 | 0,50 | 1,00 | 1,00 |

(*) Nodos que não são exibidos na área de contexto

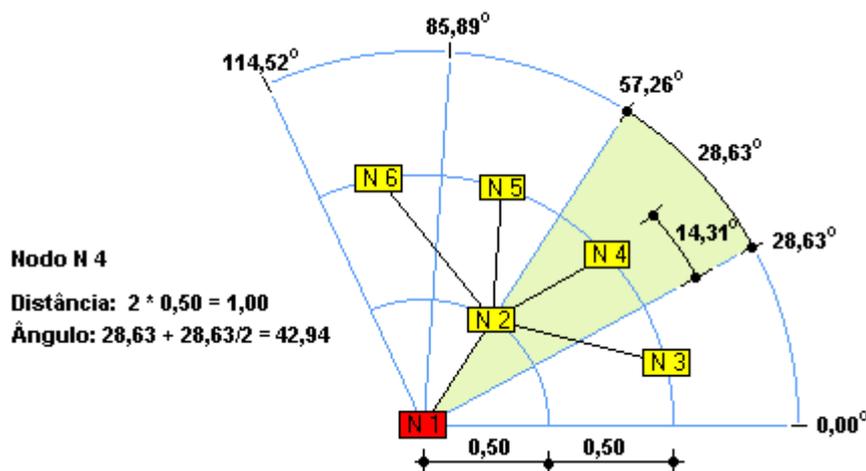


FIGURA 4.17 – Cálculo das coordenadas polares do nodo N 4

4.3.2 Layout da área de detalhe

O cálculo das coordenadas dos nodos que serão exibidos na área de detalhe é bastante semelhante ao anterior. Dessa forma, serão descritas apenas as diferenças existentes entre os dois processos. A primeira diferença reside na forma como é determinada a medida angular do setor referente a área de detalhe. Enquanto na área de contexto esse ângulo possui um valor *default* que pode ser alterado pelo usuário, na área de detalhe ele é determinado automaticamente segundo os seguintes critérios:

1. Se o nodo raiz está localizado no foco da área de detalhe, a medida angular utilizada para o setor será de 360° , já que nesta situação não é necessário prever a conexão com a área de contexto.

2. Caso o nodo localizado no foco da área de detalhe possua filhos, a medida angular utilizada para o setor é calculada de forma a minimizar a sobreposição de nodos no primeiro nível da estrutura (Eq. 4.4). Dessa forma foi definida uma constante que representa o ângulo necessário para que o nodo possa ser visualizado sem extrapolar os limites do setor onde ele está inserido. Essa constante (*AngMinimo*) foi definida em função da altura do texto utilizado no interior dos retângulos que representam os nodos e de sua respectiva distância em relação ao foco da área de detalhe (figura 4.18). No protótipo implementado foi utilizado um ângulo medindo 9° .

(4.4)

$$AngAreaDetalhe = NFi(NodoFocoDet) * AngMinimo * MaiorNumNetos(NodoFocoDet)$$

onde:

AngAreaDetalhe é a medida angular do setor da área de detalhe ocupado pela estrutura;
NFi(NodoFocoDet) é o número de filhos do nodo que está no foco da área de detalhe;
AngMinimo é o ângulo mínimo descrito anteriormente;
MaiorNumNetos(NodoFocoDet) é o maior número de netos do nodo que está no foco da área de detalhe de um mesmo pai. Caso o nodo que está no foco não possua netos, será utilizado o valor unitário.

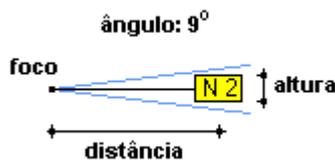


FIGURA 4.18 – Ângulo mínimo necessário para exibir os nodos sem sobreposição no primeiro nível

A figura 4.19(a) ilustra o caso da não existência de “netos”. Nesta situação a medida angular do setor utilizado para a exibição da estrutura na área de detalhe é 135° . A figura 4.19(b) mostra uma estrutura onde a maior quantidade de netos é 6. Neste caso, cada subárvore do nodo denominado **Textos** receberá um setor de 54° para ser exibida totalizando 108° para o setor da área de detalhe.

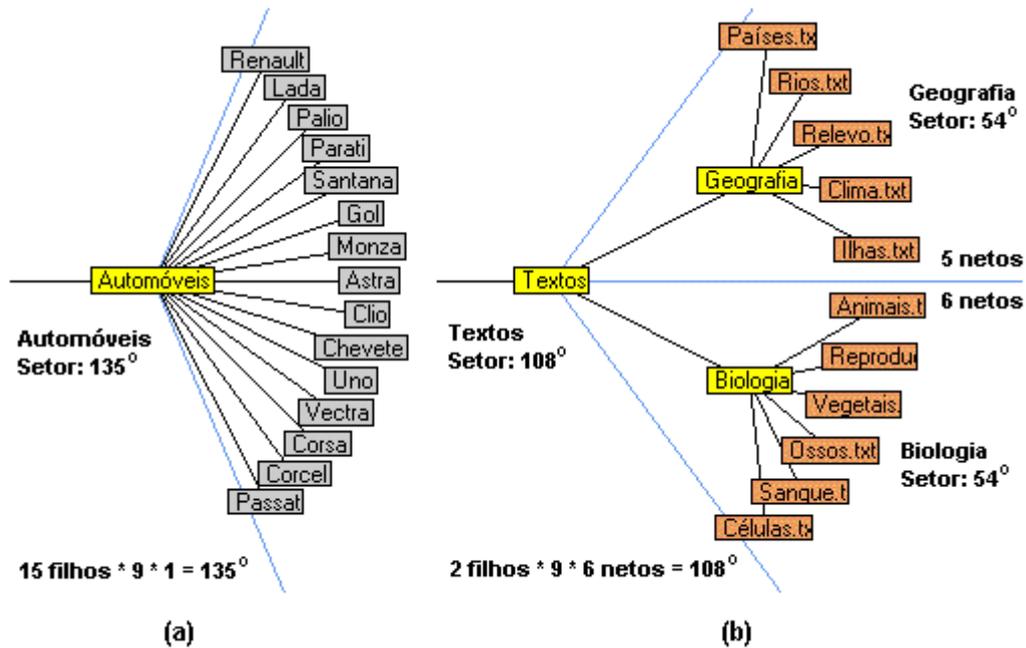


FIGURA 4.19 – Medida angular do setor da área de contexto sem (a) e com a existência de “netos” (b)

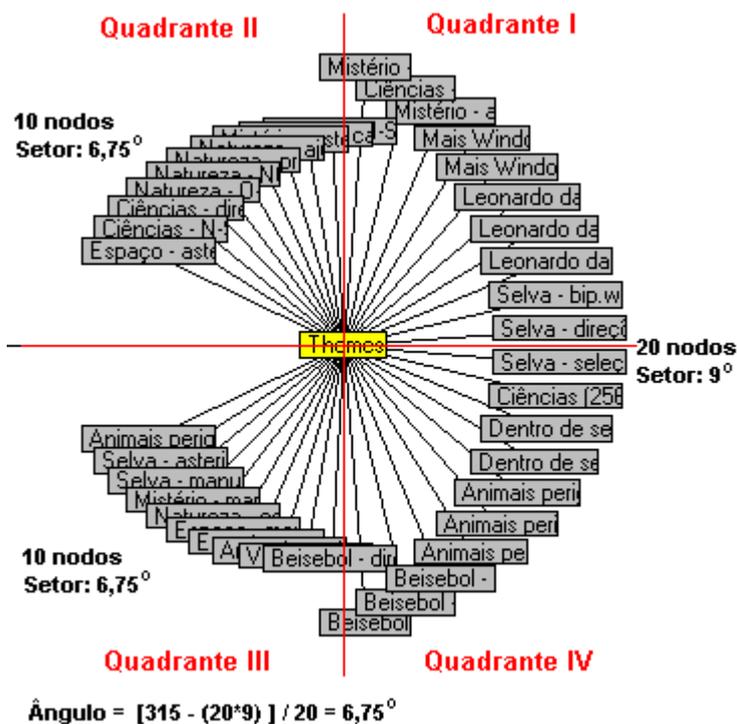


FIGURA 4.20 – Hierarquia com setor angular inferior a 9°

Cabe salientar que, se o valor angular obtido ultrapassar o máximo permitido, será tomando como base um limite máximo que, no protótipo implementado, foi 315° . Entretanto, nesta situação pode ocorrer que o ângulo do setor ocupado pelas subárvores, cujas raízes estão localizadas no primeiro nível (obtido pela equação 4.5), seja menor que o ângulo mínimo aceitável (9°) o que neste caso geraria sobreposição dos nodos localizados nos quadrantes I e IV. Para contornar tal situação atribui-se ao valor angular do setor ocupado por cada um dos 20 nodos centrais (localizados nos quadrantes I e IV)

a medida do ângulo mínimo e divide-se o valor angular restante (135°) entre aqueles localizados nos quadrantes II e III. A figura 4.20 ilustra essa situação exibindo uma subárvore cujo nodo raiz possui 40 filhos. Os nodos localizados nos quadrantes I e IV ocupam setores de 9° enquanto os restantes recebem um ângulo de $6,75^\circ$.

3. A última situação ocorre quando o nodo que está no foco da área de detalhe não possui filhos. Neste caso é utilizado como medida angular, o valor definido por *AngMinimo*.

Outra diferença existente no processo de geração do layout da estrutura visualizada na área de detalhe está na forma como a medida angular de cada setor ocupado pelas subárvores do primeiro nível é obtida. Neste caso é utilizado um algoritmo baseado no *Subnode-Based Layout* [WIL 99], onde o espaço é dividido igualmente entre os filhos. A partir do segundo nível utiliza-se o mesmo processo descrito anteriormente, usado na área de contexto. O tratamento do primeiro nível de forma diferenciada dos demais é feito para que usuário tenha uma melhor percepção do relacionamento dos nodos irmãos, já que este é o seu ponto de interesse. Caso fosse utilizado o algoritmo anterior, em árvores desbalanceadas, os nodos irmãos ficariam separados com distâncias distintas, dificultando a leitura. A figura 4.21 ilustra a distribuição dos nodos no primeiro nível através do *Subnode-Based Layout* [WIL 99] enquanto que a figura 4.22 apresenta o resultado caso fosse utilizado o *Leaf-Based Layout* [WIL 99].

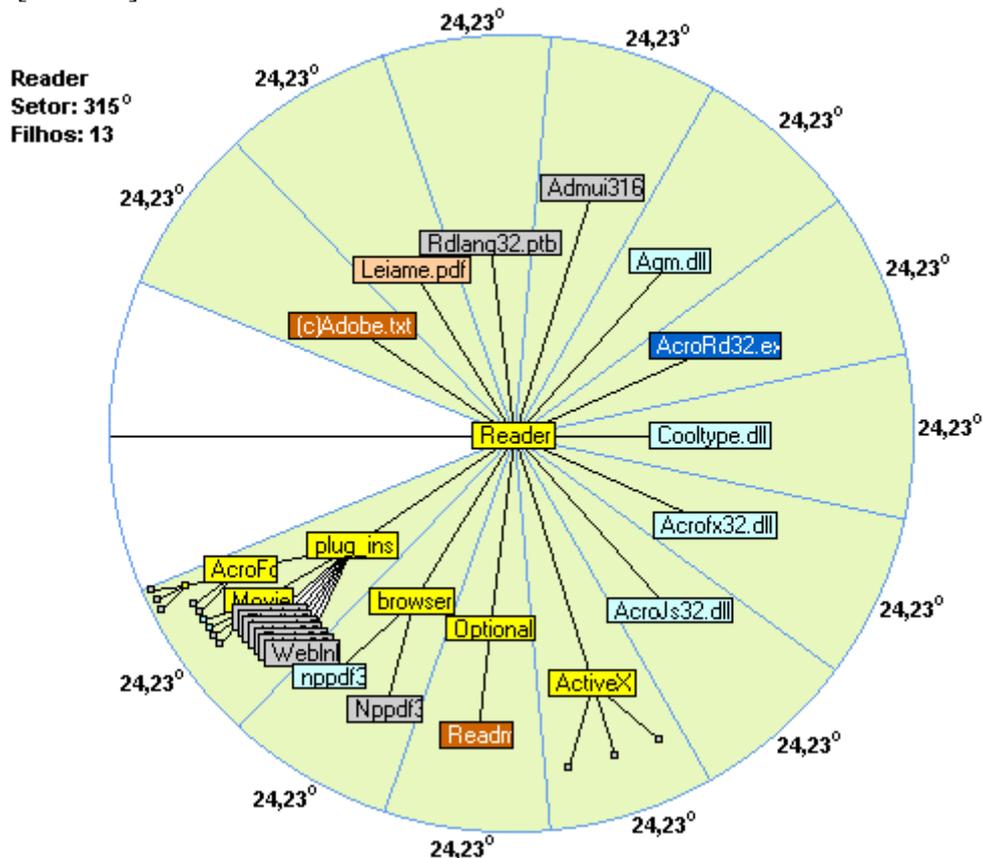


FIGURA 4.21 – Distribuição dos nodos no primeiro nível através do *Subnode-Based Layout*

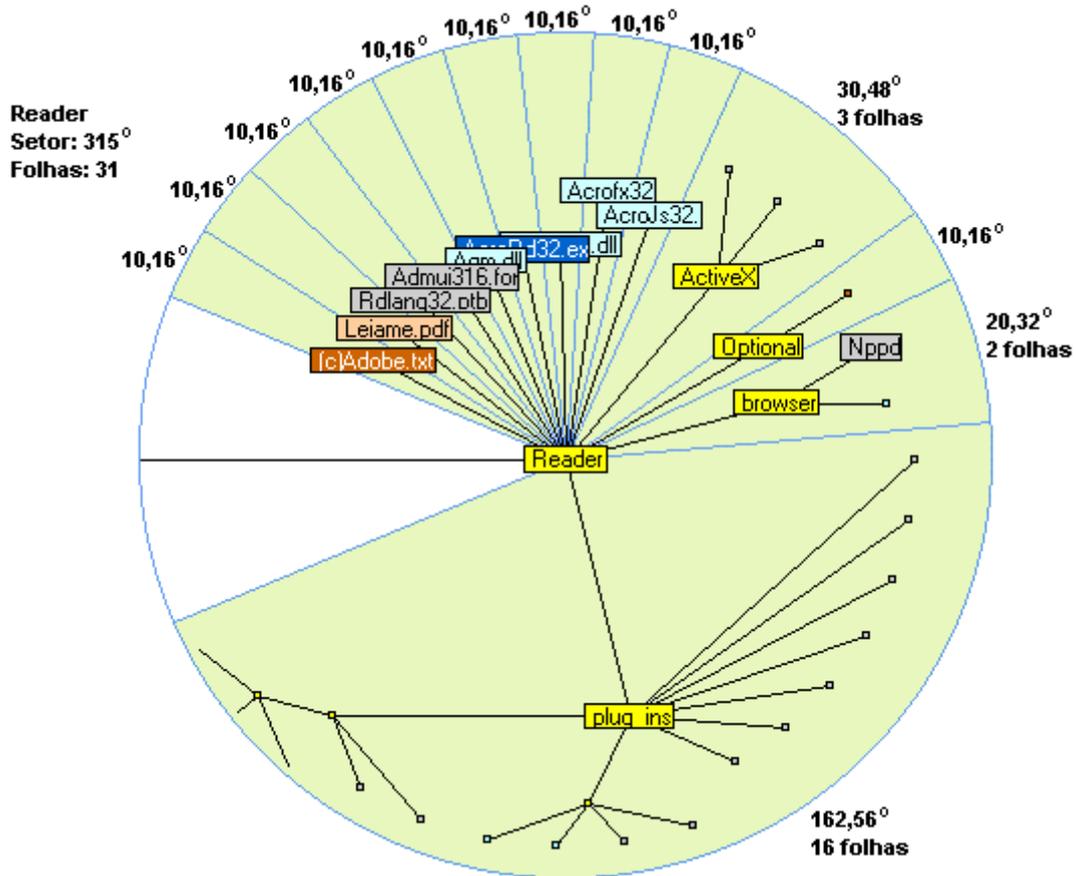


FIGURA 4.22 – Distribuição dos nodos no primeiro nível através do *Leaf-Based Layout*

Dessa forma o cálculo utilizado para obter a medida angular das subárvores do primeiro nível é dado pela equação 4.5.

$$AngSub\grave{a}rvore(i) = \frac{AngAreaDetalhe}{NFi(NodoFocoDetalhe)} \quad (4.5)$$

onde:

$AngSub\grave{a}rvore(i)$ é o ângulo do setor ocupado pela subárvore i do primeiro nível;
 $NFi(NodoFocoDetalhe)$ é o número de filhos do nodo que está no foco da área de detalhe.

A componente angular é calculada de forma a variar entre $\frac{- AngAreaDetalhe}{2}$ e $\frac{+ AngAreaDetalhe}{2}$, sendo obtida de forma similar ao utilizado para a área de contexto.

O cálculo da distância para os nodos do primeiro nível também sofre uma alteração em relação ao processo utilizado para os nodos localizados nos outros níveis (Eq. 4.6). Conforme a componente angular se aproxima dos limites descritos anteriormente ela sofre um acréscimo para compensar a curvatura, com o objetivo de minimizar a sobreposição entre nodos.

$$Dist\grave{a}ncia(i) = \frac{AlturaNodo * CompAngular(i)}{\frac{AngMinimo}{\text{sen}(CompAngular(i))}} \quad (4.6)$$

onde:

AlturaNodo é a altura do retângulo que representa um nodo;

CompAngular(i) é a componente angular das coordenadas polares do nodo *i*;

AngMinimo é o ângulo mínimo descrito anteriormente.

Cabe salientar que esse cálculo é aplicado somente se o nodo estiver localizado nos quadrantes I e IV tomando-se o círculo trigonométrico como referência, portanto com a componente angular variando entre -90° e $+90^\circ$. A figura 4.23 mostra um esquema das duas formas de cálculo utilizadas para obter as distâncias.

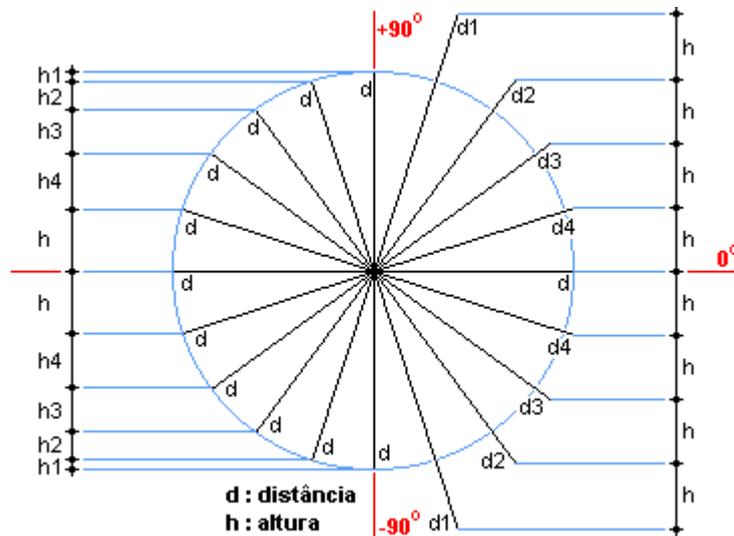


FIGURA 4.23 – Esquema do cálculo das distâncias no primeiro nível da área de detalhe

4.3.3 Efeito *fisheye*

Após o cálculo das coordenadas polares dos nodos que compõem a estrutura exibida na área de contexto, são aplicadas duas operações (translação e rotação) a fim de ajustar seus nodos nas coordenadas definitivas. A translação é aplicada para mover o nodo pai daquele que será posicionado no foco da área de detalhe para o foco da área de contexto (coordenadas 0,0), já que originalmente o nodo raiz ocupa essa posição. A rotação é aplicada para fazer com o nodo seja posicionado na direção da coordenada cujo ângulo é igual a 0° . Isso garante a perfeita conexão do nodo localizado no foco da área de contexto com o localizado na área de detalhe.

Na geração do layout, as coordenadas dos nodos são calculadas levando-se em conta um círculo sem limite de raio. Porém antes de obter as coordenadas de dispositivo, é feita uma normalização de forma que as coordenadas se encaixem em um círculo de raio unitário, onde o 1 corresponde a um nodo no infinito. Nesse mapeamento, foi utilizada a função arco tangente que, para um parâmetro variando entre 0 e infinito, gera valores entre 0 e 90° . Dessa forma a normalização pode ser executada de acordo com a equação 4.7.

$$DistanciaNormal = \frac{\arctan(CompDistancia)}{\frac{\pi}{2}} \quad (4.7)$$

onde:

DistanciaNormal é a distância normalizada com valores gerados entre 0 e 1;
CompDistancia é a componente distância da coordenada polar do nodo.

Além disso, a função arco-tangente atende ao requisito de produzir o efeito *fisheye*, visto que ela comporta-se de forma aproximadamente linear com uma inclinação de 45° próxima da origem reduzindo, de forma mínima, as distâncias próximas ao foco, enquanto incrementa suavemente a redução aplicada para os nodos mais distantes.

Como mencionado anteriormente, os nodos são exibidos na forma de retângulos, pequenos quadrados ou simplesmente são omitidos da representação. O processo de escolha da alternativa utilizada, bem como a determinação das coordenadas do retângulo que os representa será descrita a seguir.

A seleção daqueles que serão exibidos é feita a partir da análise da componente distância de sua coordenada polar. Se ela for maior que um limite pré-estabelecido (configurável) ele não será exibido. Se o nodo for selecionado para exibição, a largura do retângulo que o representa é obtida acrescentando-se uma medida padrão (configurável) de um nodo à componente distância de sua coordenada. A diferença das duas distâncias, após a aplicação da função de mapeamento entre os dois pontos, fornecerá a largura do retângulo (figura 4.24).

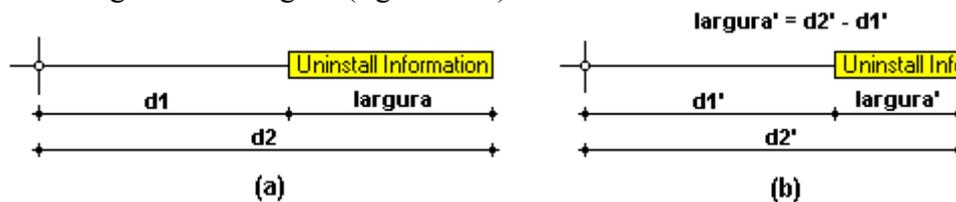


FIGURA 4.24 – Largura do retângulo antes (a) e depois do mapeamento (b)

Caso a largura obtida fique abaixo de um limite pré-estabelecido (configurável) as coordenadas do retângulo serão substituídas pelas de um pequeno quadrado.

4.4 O protótipo *Bifocal Browser*

O *Bifocal Browser* foi projetado com o objetivo de exibir estruturas de arquivos através da *Bifocal Tree*, embora possa ser adaptado para visualizar qualquer tipo de hierarquia. No entanto, cabe salientar, que foi dada ênfase somente às operações relacionadas à visualização, não existindo, por exemplo, mecanismos para criação de diretórios e cópia de arquivos. A figura 4.25 mostra a janela principal do sistema, que possui uma barra de ferramentas onde os botões são agrupados segundo sua funcionalidade.

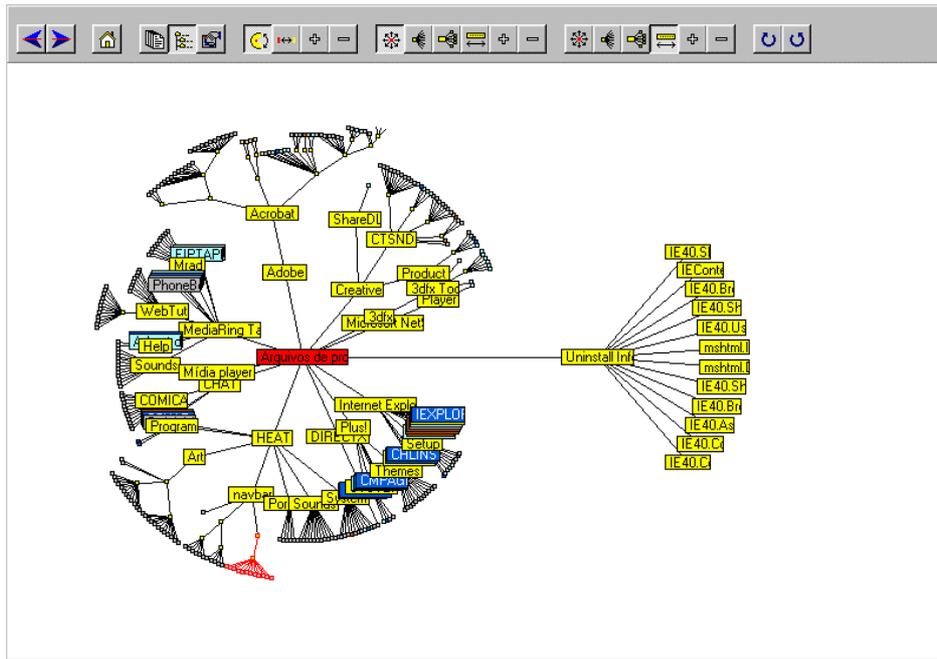


FIGURA 4.25 – Janela principal do *Bifocal Browser*

O grupo localizado à esquerda permite a seleção das ferramentas de poda/expansão, consulta de nodos irmãos e de atributos. Após a escolha da operação, sua execução é efetivada através do pressionamento do botão direito do mouse cujo cursor deve estar localizado sobre o nodo desejado.

Os grupos que possuem os botões de incremento (+) e decremento (-) têm o objetivo de alterar os parâmetros de visualização, após a seleção do parâmetro desejado. Os demais grupos possuem botões que tem sua atuação efetivada imediatamente após o pressionamento. As tabelas 4.6 e 4.7 descrevem a funcionalidade de cada botão existente no sistema:

TABELA 4.6 – Botões da barra de ferramentas de uso geral

| Botões | Funcionalidade |
|---|---|
|  | Redo |
|  | Undo |
|  | Exibe os nodos irmãos de um nodo |
|  | Exibe os atributos (nome, data, hora e tamanho) de um nodo |
|  | Ativa as operações de poda/expansão |
|  | Restaura os parâmetros default do sistema posicionando a estrutura em sua localização inicial (nodo raiz situado no foco da área de detalhe). |
|  | Rotaciona a subárvore localizada na área de detalhe no sentido horário. |
|  | Rotaciona a subárvore localizada na área de detalhe no sentido anti-horário. |

TABELA 4.7 – Botões da barra de ferramentas relacionados com alteração do layout

| Botões | Funcionalidade |
|---|---|
|  | Altera o ângulo do setor utilizado para exibição da estrutura na área de contexto. |
|  | Altera a distância existente entre os dois focos. |
|  | Altera a distância entre os níveis da hierarquia. |
|  | Altera a distância a partir da qual um nodo é podado (retirado da representação). |
|  | Altera a largura utilizada como limite para que o nodo passe a ser representado por um pequeno retângulo. |
|  | Altera a largura máxima do retângulo que representa um nodo. |

5 Avaliação da *Bifocal Tree*

Com o objetivo de verificar se a técnica proposta atende ao propósito de oferecer uma forma eficaz para a visualização de hierarquias, foi realizada uma avaliação baseada na execução de tarefas. O resultado dessa avaliação foi obtido a partir da comparação do desempenho de um grupo de usuários na utilização de outros dois sistemas de visualização. Neste capítulo será apresentada uma descrição do processo de avaliação utilizado, bem como os resultados obtidos.

Em geral novas técnicas de visualização são apresentadas na literatura com avaliações experimentais pouco sistemáticas. Mais recentemente, alguns trabalhos tem buscado estratégias que permitam avaliar técnicas com critérios mais objetivos [BRA 97][JUA 2000][LUZ 2001]. Outros realizam comparações entre diferentes técnicas experimentalmente.

Wiss et al. [WIS 98] avaliaram três técnicas de visualização de informações que possuem como característica comum utilizar o espaço tridimensional para exibição de seus elementos. São elas: Cam Tree [ROB 91], *Information Cube* [REK 93] e *Information Landscape* [AND 95]. Elas foram implementadas e utilizadas na visualização de dois conjuntos diferentes de dados. O primeiro refere-se a uma tabela de conteúdos de um jornal eletrônico e o segundo a um sistema de arquivos. A avaliação foi obtida a partir da observação dos problemas encontrados durante o uso dos sistemas e pela enumeração das sete das tarefas de usuário definidas por Shneiderman [SHN 96] suportadas em cada técnica. Em suas conclusões são relatados os comportamentos diferenciados que cada sistema possui ao ser utilizado para visualizar diferentes conjuntos de dados.

Uma análise comparativa entre duas outras técnicas de visualização (*Hyperbolic browser* da Inxight¹ e o *Microsoft Windows Explorer*®), foi apresentada por Pirolli et al. [PIR 2000] através de dois experimentos com usuários. Inspirado em uma competição para avaliar a performance entre *browsers* (The Great CHI'97 Browser-Off), seus desempenhos foram estudados em condições controladas de laboratório.

Os experimentos basearam-se na execução de 128 tarefas distribuídas em 4 tipos: tarefas que executam consultas simples e complexas e tarefas que exigem a identificação de relacionamentos locais e globais. Foi levantada a hipótese de que a familiaridade com o conteúdo da estrutura utilizada no teste influenciaria o desempenho. Dessa forma, as tarefas foram classificadas em função de um índice denominado *information scent* que mede o conhecimento prévio da estrutura que cada usuário possui.

O primeiro experimento buscou compreender o efeito do *information scent* sobre a performance. Para isso foram utilizadas tarefas distribuídas de maneira uniforme ao longo de toda faixa do índice. Os resultados do experimento não indicaram diferença significativa de desempenho entre os *browsers*, conclusão que contraria o resultado da competição realizado no CHI 97 onde o *Hyperbolic Browser* foi “vitorioso”. No entanto, a análise realizada nos dados esclareceu terem sido as diferenças individuais dos usuários que contribuíram para esse resultado. No que se refere ao efeito do índice

¹ www.inxight.com – Inxight é uma companhia derivada da Xerox.

information scent, ambos *browsers* melhoraram seus desempenhos com valores mais altos, com uma superioridade significativa ao *Hyperbolic Browser*. Entretanto, sob condições de baixo *information scent* o Explorer foi superior.

Dessa forma, o segundo experimento procurou fazer uma análise mais detalhada dos desempenhos em tarefas com diferenças mais significativas de *information scent*. Os resultados confirmaram a análise anterior, indicando que o tempo de execução das tarefas com o *Hyperbolic Browser* é superior quando o índice *information scent* é alto e inferior quando baixo.

Neste trabalho, a fim de avaliar a técnica *Bifocal Tree*, foi realizado um experimento comparando o seu desempenho com o *Windows Explorer*, brevemente descrito no capítulo 3, e o *MagniFind*, cujas características serão apresentadas a seguir.

5.1 MagniFind

O *MagniFind* é um *browser* de arquivos desenvolvido pela Inxight Software Inc. com o objetivo de demonstrar a utilização da técnica *Hyperbolic Tree* proposta por Lamping et al. [LAM 95]. Como pode ser visto na figura 5.1, o *MagniFind* utiliza a abordagem foco+contexto e representa os nodos da estrutura através de um ícone acrescido de um texto que o identifica. Cada ícone tem seu tamanho diminuído à medida que se afasta do foco que está situado no centro da janela de exibição.

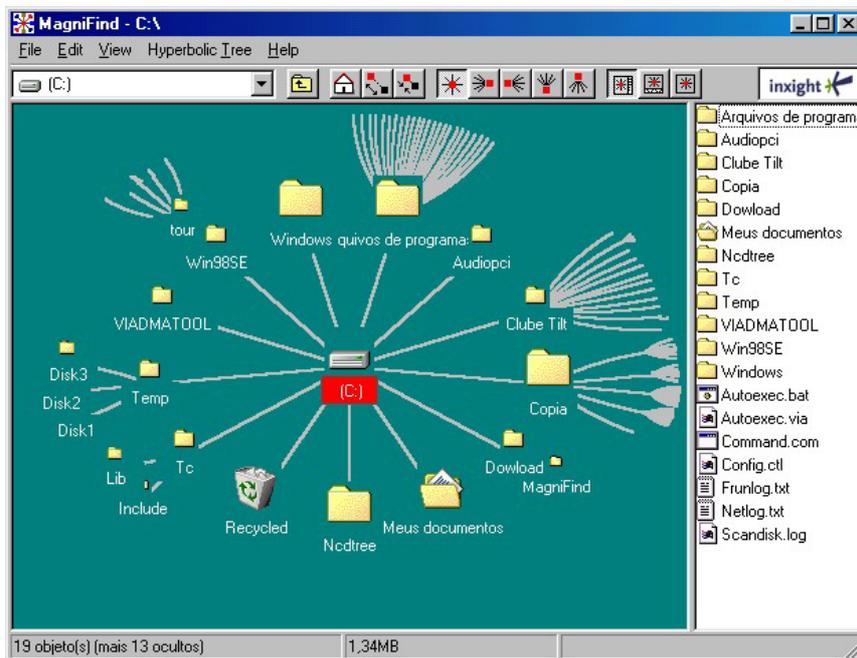


FIGURA 5.1 – *Browser* de arquivos *MagniFind* da Inxight

O nodo raiz é identificado através de um retângulo vermelho que o destaca dos demais. Somente os textos localizados próximos ao foco são exibidos por inteiro. Os demais são truncados à medida que se afastam do foco até o ponto onde não são mais exibidos. De forma similar ao *Windows Explorer*, nesta janela não são exibidos os nodos que representam arquivos.

A navegação na estrutura pode ser executada de duas formas. Na primeira, o nodo que recebe o clique do mouse é deslocado para o foco, transladando a estrutura inteira para as novas coordenadas. Na segunda, o nodo pode ser arrastado para outro local qualquer dentro da janela de exibição. Esse mecanismo é utilizado para permitir que um nodo oculto, por estar muito distante do foco, possa ser exibido com detalhes suficientes para a consulta do usuário. As diferentes transições geradas por esse sistema de navegação são exibidas com animação.

Da mesma forma, as distâncias entre os níveis da hierarquia podem ser alteradas dependendo da necessidade de se obter uma aproximação ou afastamentos dos nodos. Embora não existam mecanismos de poda ou expansão, o *MagniFind* permite que um nodo selecionado torne-se o novo raiz, retirando aqueles que não pertençam a sua subárvore da representação. A qualquer momento é possível exibir uma janela de detalhe similar à existente no *Windows Explorer*, localizada à direita ou abaixo da principal, com o objetivo de exibir os nodos filhos daquele selecionado, incluindo os que representam arquivos.

5.2 Metodologia de avaliação

Um ensaio de interação foi realizado com um grupo de 12 usuários que cumpriram um conjunto de 22 tarefas utilizando cada um dos três *browsers* com hierarquias pré-definidas. O desempenho na realização dessas tarefas foi medido levando-se em consideração o tempo gasto para executá-las, bem como a respectiva taxa de acertos. A seguir será descrito como foi realizado o experimento.

5.2.1. Hierarquias

Na avaliação de cada *browser* foram utilizadas duas hierarquias de tamanhos diferentes a fim de verificar se essa característica resultaria em uma alteração significativa em seus desempenhos. Foi construída uma hierarquia pequena com 292 nodos, um grau máximo igual a 9 e altura igual a 6, e uma cerca de três vezes maior possuindo 1055 nodos, um grau máximo igual a 13 e altura igual a 8.

As hierarquias utilizadas no teste de cada *browser* possuem exatamente a mesma estrutura, diferenciando-se entre si apenas pelos nomes dos nodos. Essa similaridade foi mantida para permitir a comparação dos tempos de execução de cada tarefa nos diferentes *browsers*. A troca dos nomes evita uma memorização, por parte dos sujeitos, das respostas fornecidas a uma mesma tarefa quando executada em cada *browser*. Quanto ao conteúdo representado pela estrutura, foi utilizada uma relação de nomes de ruas sem qualquer critério de classificação. Essa característica evita que o conhecimento prévio das relações hierárquicas representadas possam influenciar no desempenho da execução das tarefas. Também foram incluídos alguns nodos com nomes iguais, mas em níveis diferentes, para minimizar a influência do fator sorte na sua busca. Dessa forma, caso o sujeito encontrasse o nodo aleatoriamente, teria que se certificar de que ele realmente era o procurado forçando a consulta de sua relação com os demais. Praticamente todos os nodos que compõem a hierarquia representam diretórios. Essa decisão foi tomada em razão de que apenas a *Bifocal Tree* representa todos os tipos, fato que poderia influenciar de forma significativa as tomadas de tempo. Entretanto, para confirmar essa influência, foram incluídos arquivos de outros tipos em um pequeno trecho que foi objeto de consulta em uma tarefa projetada para esse fim.

5.2.2. Sessões de teste

O experimento com cada sujeito foi dividido em três sessões distintas realizadas em dias diferentes para testar cada um dos *browsers* separadamente. Essa divisão foi feita para não tornar o teste cansativo ao usuário, já que havia uma previsão de que cada sessão duraria aproximadamente 50 minutos, fato que foi confirmado durante o experimento.

Cada sessão seguiu a mesma estrutura básica. Inicialmente o sujeito recebeu uma orientação sobre as características do *browser* a ser avaliado. Foram descritos os elementos que compõem o layout adotado e a forma como ele representa os relacionamentos. Além disso, foram apresentados os mecanismos de interação existentes no *browser* que poderiam ser usados na execução das tarefas. A seguir, utilizando uma hierarquia como exemplo, foi explicado ao sujeito o tipo de tarefa a ser executada, com o objetivo de fazer com que ele se familiarizasse com o experimento. Nesta fase, foi dada a oportunidade de experimentar o *browser* até o momento em que se sentisse à vontade para iniciar o teste. O tempo máximo permitido para esta etapa foi de 15 minutos, sendo que todos os usuários sentiram-se aptos dentro do prazo estabelecido.

Após a fase de adaptação, a sessão de avaliação foi efetivamente iniciada. As tarefas foram executadas primeiramente sobre a hierarquia pequena e, logo em seguida, sobre a grande. A sessão foi assistida por um avaliador que cumpriu as funções de anotar as observações importantes e cronometrar o tempo de execução de cada uma das tarefas.

5.2.3. Execução das tarefas

Cada tarefa foi entregue por escrito ao sujeito. Ela é constituída de um enunciado e um desenho que ilustra o trecho da hierarquia que sofrerá a respectiva ação conforme mostra a figura 5.2. Para facilitar a execução das tarefas, os níveis da hierarquia foram numerados considerando o nodo raiz localizado no nível 1.

Tarefa 6: Informe o nome de cada filho do nodo **Nunes Machado** Considere que está localizado no nível 4 no final do seguinte caminho: **Ruas\Nilo Torres\Flores**. Para cada nodo informado cite sua respectiva quantidade de filhos.

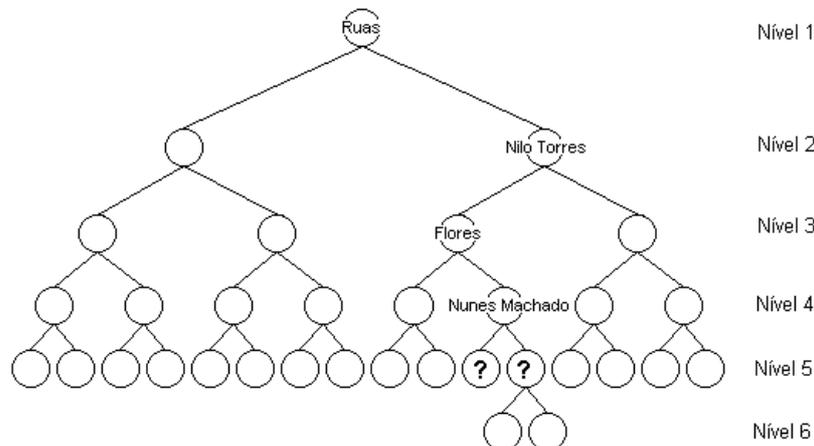


FIGURA 5.2 – Um exemplo de tarefa fornecida ao sujeito

Ao término da leitura, foi dada a oportunidade ao usuário de perguntar sobre algum ponto que não tenha sido compreendido. O avaliador forneceu explicações verbais sobre o que deveria ser feito utilizando o desenho como ilustração. Após o sujeito sinalizar estar apto a executá-la, o cronômetro foi disparado para medir o tempo até que a resposta tenha sido fornecida verbalmente. Foi permitido ao sujeito interromper a execução da tarefa, caso tenha surgido alguma dúvida sobre o seu enunciado ou sobre os mecanismos de interação existentes no *browser*. Nessa situação a contagem do tempo foi interrompida temporariamente e retomada após a dúvida ser esclarecida. O objetivo de admitir essa interrupção foi fazer com que a falta de compreensão da tarefa ou inexperiência com o uso do *browser* influenciassessem o mínimo possível o resultado. Após o cumprimento da tarefa foi informado ao sujeito sobre o sucesso da resposta fornecida. Em caso de erro, após o registro do insucesso, foi dada a oportunidade para que a execução fosse retomada até a obtenção da resposta correta. O tempo considerado nesta situação engloba todo o período de execução da tarefa. Entre as tarefas não foi permitido alterar a configuração do browser ou a seleção anterior. O tempo utilizado para as mudanças desejadas antes de seu início foi computado na própria tarefa.

Como mencionado anteriormente, após o término do teste com a hierarquia pequena, foi iniciado, sem interrupção, o teste com a hierarquia grande. Foram utilizadas tarefas bastante similares nas duas hierarquias, diferenciando-se apenas pelo fato de que a segunda possibilita a navegação em uma profundidade maior. Para ilustrar a similaridade a figura 5.3 mostra a tarefa utilizada na segunda hierarquia equivalente ao exemplo da figura 5.2.

Tarefa 6: Informe o nome de cada filho do nodo *Nazaré*. Considere que está localizado no nível 6 no final do seguinte caminho: *Ruas\Machado de Assis\Morro da Primavera\Múcio Teixeira\Murilo Furtado*. Para cada nodo informado cite sua respectiva quantidade de filhos.

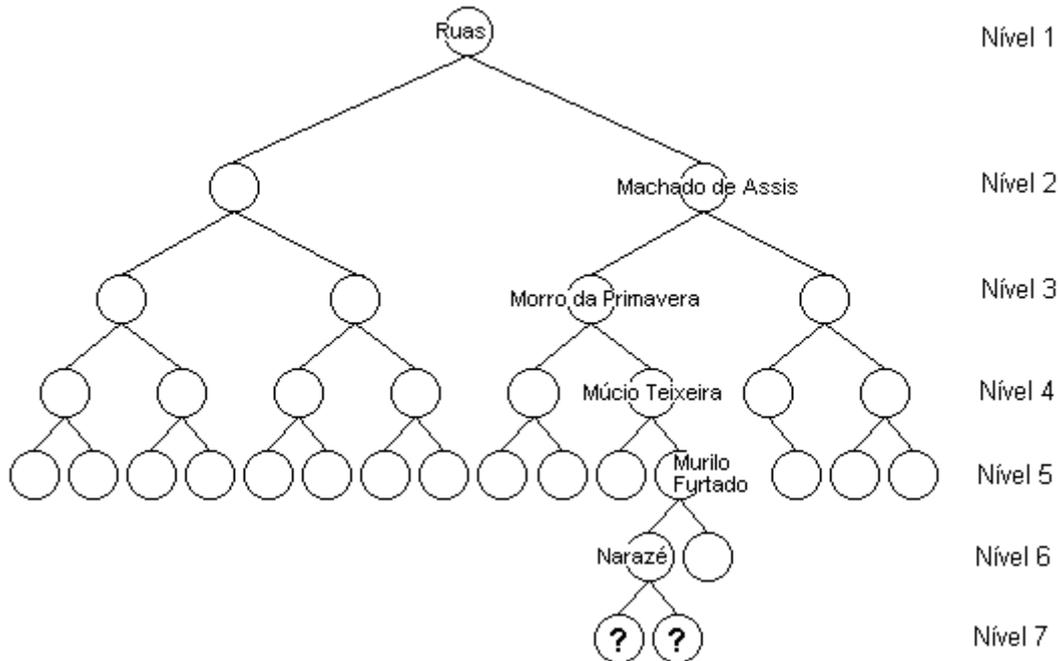


FIGURA 5.3 – Um exemplo de tarefa utilizada com a hierarquia grande

Ao término das três sessões, um questionário foi fornecido ao sujeito a fim de coletar a sua opinião em relação à utilização de cada *browser*. Ele é constituído de três

questões referentes à representação visual e três relacionadas com os mecanismos de interação. As questões são respondidas, para os três *browsers*, através da marcação em quadro de um valor compreendido entre 1 e 5 onde os significados dos limites são fornecidos. A figura 5.5 mostra uma das questões.

1.O desenho que representa a hierarquia torna a execução das tarefas
(1.difícil ... 5.fácil)

| | 1 | 2 | 3 | 4 | 5 |
|-----------------|---|---|---|---|---|
| Explorer | | | | | |
| Bifocal Browser | | | | | |
| MagniFind | | | | | |

FIGURA 5.4 – Um exemplo de uma questão utilizada no teste de satisfação do usuário

5.2.4. Definição da ordem das sessões

Considerando que o mesmo conjunto de questões seria executado seis vezes, existia a possibilidade de que a experiência adquirida na execução das tarefas pudesse ocasionar um melhor desempenho com o último *browser*. Para minimizar essa influência, foi estabelecido que as sessões fossem executadas em todas as ordens possíveis, que neste caso, considerando os três *browser* totaliza um conjunto de seis possibilidades. Com um total de 12 sujeitos, cada combinação de ordens foi executada duas vezes. A escolha da ordem usada em um determinado conjunto de três sessões foi estabelecida por sorteio entre as que ainda não tinham sido testadas.

5.3 Descrição e objetivo das tarefas

Cada hierarquia foi avaliada a partir da execução de um conjunto de 11 tarefas. Cada tarefa foi projetada para avaliar cada um dos seguintes aspectos referentes a utilização de um *browser* de arquivos.

- Busca de um determinado nodo na hierarquia.
- Navegação através de um caminho determinado.
- Identificação dos relacionamentos pai/filhos.
- Comparação entre dois trechos da hierarquia.
- Desorientação do usuário causada pela alteração do layout durante a navegação.
- Representação de nodos que não são diretórios.

5.3.1. Busca de um nodo

Consiste em localizar um nodo específico dentro da hierarquia utilizando apenas os mecanismos de interação fornecidos pelos *browsers*. As tarefas que avaliam essa funcionalidade foram projetadas de forma que o nodo possa ser localizado a partir da informação de uma pista que auxilie na sua localização. Caso a busca fosse efetuada apenas pelo nome do nodo, o aspecto sorte poderia ser um fator determinante no resultado obtido. As tarefas 1 e 2 avaliam essa funcionalidade e possuem os seguintes objetivos específicos e respectivos enunciados.

Tarefa 1

Objetivo: Avaliar a execução de uma busca de nodos localizados próximos ao nodo raiz, envolvendo os níveis 2 e 3 da hierarquia pequena e 2,3 e 4 da hierarquia grande. Para as duas hierarquias o nodo referência para o começo da procura foi posicionado no nível 1. No entanto, o nodo procurado foi posicionado no nível 2 para a hierarquia pequena e no nível 3 para a hierarquia grande.

Enunciado: Considerando a subárvore, cuja raiz é o nodo **João Teles** (localizado no nível 2 da estrutura), informe a quantidade de filhos do nodo **Santa Rita** que está localizado em algum ponto da referida subárvore.

Tarefa 2

Objetivo: Avaliar a execução de uma busca de nodos localizados mais distantes do nodo raiz, envolvendo os níveis 3,4 e 5 da hierarquia pequena e 3,4,5 e 6 da hierarquia grande. Para as duas hierarquias o nodo referência para o começo da procura foi posicionado no nível 3. No entanto, o nodo procurado foi posicionado no nível 5 para a hierarquia pequena e no nível 6 para a hierarquia grande.

Enunciado: Considerando a subárvore, cuja raiz é o nodo **Humberto de Campos** (localizado no nível 3 da estrutura), informe a quantidade de filhos do nodo **João Pessoa** que está localizado em algum ponto da referida subárvore.

5.3.2. Navegação através de um caminho determinado

Consiste em consultar informações de um nodo localizado em um ponto da hierarquia cujo caminho é previamente conhecido, tarefa esta bastante comum na utilização de browser de arquivos. A tarefa 3 foi criada com o objetivo de avaliar a navegação no sentido do nodo raiz para o nodo procurado e a tarefa 4 contempla a situação inversa, onde a navegação é executada de um nível mais profundo em direção ao nodo raiz.

Tarefa 3

Objetivo: Avaliar a execução de uma navegação através de um caminho pré-definido, que inicia no nodo raiz e segue em direção aos níveis hierarquicamente inferiores, onde está localizado o nodo destino. Na hierarquia pequena o caminho atravessa 4 níveis estando o nodo procurado no quinto. Na grande, o caminho percorre 6 níveis encontrando-se o nodo procurado no sétimo.

Enunciado: Informe o nome completo do nodo cujo nome começa com a sílaba “**Mar**” localizado no nível 5 (nível 7 para hierarquia grande) que tem como ancestrais os nodos **Ruas**, **Pinto Bandeira**, **Hercílio Graça** e **Farnese** em ordem decrescente de hierarquia.

Tarefa 4

Objetivo: Avaliar a execução de uma navegação através de um caminho pré-definido, que inicia em um nodo localizado em um nível mais profundo da hierarquia

em direção ao nodo raiz. Na hierarquia pequena o nodo que determina o ponto inicial localiza-se no nível 5 e na grande no nível 7.

Enunciado: A partir do nodo cujo nome é **Martim Afonso** localizado na tarefa anterior, cite o nome completo e a respectiva quantidade de filhos de cada um de seus ancestrais até o nodo **Pinto Bandeira**.

5.3.3. Identificação dos relacionamentos

Consiste em verificar se a representação visual fornece uma forma eficiente de identificação dos relacionamentos hierárquicos existentes na estrutura. As tarefas 5 e 6 foram projetadas para avaliar os relacionamentos entre pai e filhos, e a 7 para verificar a facilidade de identificação dos níveis onde os nodos estão localizados.

Tarefa 5

Objetivo: Avaliar se os relacionamentos entre um nodo pai e seus filhos localizados próximos ao nodo raiz são facilmente identificáveis. Nesta tarefa são referenciados os níveis 2 e 3 da estrutura.

Enunciado: Informe o nome de cada filho do nodo **Evaristo da Veiga** (localizado no nível 2). Para cada nodo informado cite sua respectiva quantidade de filhos.

Tarefa 6

Objetivo: Avaliar se os relacionamentos entre um nodo pai e seus filhos localizados em um ponto mais distante do nodo raiz são facilmente identificáveis. Nesta tarefa são referenciados os níveis 4 e 5 (6 e 7 para hierarquia grande) da estrutura.

Enunciado: Informe o nome de cada filho do nodo **Nunes Machado**. Considere que está localizado no nível 4 (nível 6 para a hierarquia grande) no final do seguinte caminho: **Ruas\Nilo Torres\Flores**. Para cada nodo informado cite sua respectiva quantidade de filhos.

Tarefa 7

Objetivo: Avaliar se os níveis da hierarquia são facilmente identificáveis. Nesta tarefa são consultados nodos localizados nos níveis 3,4 e 5 (5,6 e 7 para a hierarquia grande) da estrutura.

Enunciado: Informe o nome de 3 nodos localizados respectivamente nos níveis 3,4 e 5 (5,6 e 7 para hierarquia grande) de forma que não estejam hierarquicamente subordinados entre si (não pertençam a mesma subárvore).

5.3.4. Comparação entre dois trechos da hierarquia.

Consiste em verificar a facilidade de comparação entre duas informações localizadas em áreas distintas da estrutura. A tarefa 8 permite a comparação entre dois trechos localizados no mesmo nível da hierarquia e a tarefa 9 compara duas áreas

situadas em níveis diferentes. A primeira localizada próxima ao nodo raiz e a segunda em um nível mais profundo.

Tarefa 8

Objetivo: Avaliar a facilidade de comparação entre duas informações localizadas em duas áreas relativamente próximas entre si situadas no mesmo nível, mas em subárvores diferentes. Nesta tarefa, as informações que serão comparadas situam-se no nível 4.

Enunciado: Considerando os filhos do nodo **Farrapos** (localizado no caminho **Ruas\São Jacó\Farrapos**) e do nodo **Hercílio Graça** (localizado no caminho **Ruas\Pinto Bandeira\Hercílio Graça**), informe o nome e o número de filhos daquele nodo que possui a maior quantidade de filhos.

Tarefa 9

Objetivo: Avaliar a facilidade de comparação entre duas informações localizadas em duas áreas relativamente distantes entre si situadas em níveis e subárvores diferentes. Nesta tarefa, as informações que serão comparadas situam-se nos níveis 3 e 5 (3 e 8 para a hierarquia grande).

Enunciado: Considerando os filhos do nodo **Goiás** (localizado no caminho **Ruas\São Jacó\Encantado\Goiás**) e de **Eça de Queirós** (localizado no caminho **Ruas\Eça de Queirós**), informe o nome e o número de filhos daquele nodo que possui a maior quantidade de filhos.

5.3.5. Desorientação do usuário causada pela alteração do layout durante a navegação

Consiste em verificar a influência da desorientação do usuário causada pela alteração do layout da representação visual durante a navegação na execução das tarefas. Para forçar a navegação em pontos distintos da hierarquia, a tarefa foi dividida em 4 estágios que foram realizados em seqüência. Os nodos consultados situam-se no nível 4 da estrutura. Foi considerado apenas o tempo total utilizado para a realização da tarefa.

Tarefa 10

Objetivo: Avaliar a desorientação do usuário causada por uma seqüência de seleções executadas em pontos diferentes da hierarquia.

Enunciado: Esta tarefa é subdividida em 4 estágios. Execute cada parte informando os resultados solicitados.

- a) Informe o número de filhos do nodo **Jaime Teles** localizado no final do caminho **Ruas\João Teles\Santa Rita\Jaime Teles**.
- b) Informe o número de filhos do nodo **Campo do Brito** localizado no final do caminho **Ruas\Gaspar Martins\Rafael Saadi\Campo do Brito**.
- c) Informe o nome de um nodo irmão de **Jaime Teles** referenciado na letra (a).

- d) Informe o nome de um nodo irmão do nodo **Campo do Brito** referenciado na letra (b).

5.3.6. Representação de nodos que não são diretórios

Dentre os três *browsers* avaliados, o *MagniFind* inclui em sua representação visual apenas nodos que identificam diretórios. O *Windows Explorer* também inclui na janela que exhibe uma visão geral da estrutura apenas esse tipo de nodo. Por essa razão, a hierarquia foi construída quase que exclusivamente com nodos que representam diretórios. Entretanto, a fim de avaliar se a falta de representação dos nodos que não são diretórios poderia influenciar no desempenho do usuário, foi incluída uma tarefa que executa uma consulta em um trecho da estrutura que possui nodos que não são diretórios.

Tarefa 11

Objetivo: Avaliar se uma consulta em um trecho da estrutura que contém nodos que não identificam diretórios sofre influência da falta de exibição destes na representação visual. Nesta tarefa, a consulta é executada em nodos situados no nível 4.

Enunciado: Informe o nome dos filhos do nodo **Benedito Guedes**. Considere que ele está localizado no nível 3 no final do seguinte caminho: **Ruas\Érico Veríssimo**. Para cada nodo informado cite a sua respectiva quantidade de filhos.

5.4 Descrição dos sujeitos e das medidas utilizadas

Inicialmente foi prevista a utilização de dois grupos de usuários para a execução do experimento. Eles seriam constituídos respectivamente por pessoas com e sem experiência no uso de *browsers* de arquivos. Entretanto, devido à quantidade de tempo exigida para a realização do teste com um grande número de sujeitos, optou-se apenas pelo grupo de usuários mais experientes. Desse modo, foi selecionado um conjunto de 12 usuários que, além de possuir experiência no uso deste tipo de ferramenta, conheciam a nomenclatura utilizada na denominação dos componentes de uma hierarquia. Na relação dos selecionados constaram dois professores e dez alunos do curso de Informática da UCPel.

Na escolha dos alunos, foi levada em conta a presença das seguintes características: já ter cursado a disciplina de Estruturas de Dados e ser bolsista de iniciação científica. Esse segundo critério foi utilizado em razão de que, nesta situação, além de serem usuários experientes na área da computação, eles seriam selecionados dentre os melhores alunos do curso garantindo uma maior qualidade nas opiniões fornecidas no questionário de satisfação respondido no final do experimento. Além disso, o grupo mais homogêneo evitaria que características individuais influenciassem a performance na realização das tarefas.

A medida do tempo na realização de cada tarefa foi obtida manualmente através da leitura de um cronômetro manipulado pelo avaliador a fim de que o sujeito pudesse concentrar-se apenas na utilização do *browser*.

5.5 Resultados obtidos

Após a realização do experimento, os resultados foram tabulados obtendo-se a média aritmética dos tempos de realização das tarefas em cada um dos *browsers*. O desempenho do *Bifocal Browser* foi medido através da comparação de suas médias com as dos outros dois. Para verificar a significância estatística das diferenças entre as três técnicas foi utilizado a Análise da Variância (ANOVA), já que é mais adequada para situações onde existem mais do que duas categorias em análise. Os resultados obtidos são discutidos na seção 5.6.

A tabela 5.1 apresenta a relação das médias obtidas em cada tarefa, onde cada par de médias assinalado em negrito identifica as que possuem uma diferença significativa estatisticamente. A cor preta identifica diferenças a favor do *Bifocal Browser* e a vermelha onde ele foi mais ineficiente. Optamos por grifar na tabela 5.1, apenas os resultados relativos a comparação da *Bifocal* em relação às outras duas, o que corresponde aos objetivos dos experimentos realizados.

TABELA 5.1 – Relação das médias de tempo obtidas na execução das tarefas

| Tarefas | Hierarquia pequena | | | | Hierarquia grande | | | |
|---------|--------------------|--------------|---------------------|--------------|--------------------|--------------|---------------------|--------------|
| | Bifocal x Explorer | | Bifocal x MagniFind | | Bifocal x Explorer | | Bifocal x MagniFind | |
| | (B) | (E) | (B) | (M) | (B) | (E) | (B) | (M) |
| 1 | 9,00 | 8,25 | 9,00 | 9,83 | 25,25 | 21,33 | 25,25 | 17,67 |
| 2 | 55,08 | 42,16 | 55,08 | 40,75 | 64,17 | 87,42 | 64,17 | 98,17 |
| 3 | 12,58 | 13,50 | 12,58 | 16,33 | 20,75 | 21,25 | 20,75 | 24,92 |
| 4 | 13,08 | 17,50 | 13,08 | 16,92 | 22,83 | 26,50 | 22,83 | 21,50 |
| 5 | 21,50 | 22,58 | 21,50 | 22,08 | 26,33 | 21,00 | 26,33 | 26,00 |
| 6 | 22,00 | 22,41 | 22,00 | 25,17 | 27,25 | 26,58 | 27,25 | 24,08 |
| 7 | 32,67 | 32,75 | 32,67 | 28,25 | 107,41 | 124,00 | 107,41 | 111,00 |
| 8 | 31,58 | 48,50 | 31,58 | 42,58 | 43,08 | 71,83 | 43,08 | 63,92 |
| 9 | 34,25 | 47,33 | 34,25 | 44,17 | 59,84 | 61,17 | 59,84 | 63,83 |
| 10 | 60,00 | 49,25 | 60,00 | 59,92 | 63,33 | 60,00 | 63,33 | 60,33 |
| 11 | 21,75 | 29,83 | 21,75 | 33,67 | 37,83 | 33,00 | 37,83 | 42,33 |

TABELA 5.2 – Percentual de erros cometidos na execução das tarefas

| Tarefas | Hierarquia pequena | | | Hierarquia Grande | | |
|---------|--------------------|--------------|--------------|-------------------|--------------|--------------|
| | (B) % erros | (E) %erros | (M)%erros | (B)%erros | (E)%erros | (M)%erros |
| 1 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| 2 | 0,00 | 0,00 | 0,00 | 0,00 | 8,33 | 25,00 |
| 3 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| 4 | 8,33 | 25,00 | 0,00 | 8,33 | 33,33 | 0,00 |
| 5 | 0,00 | 8,33 | 8,33 | 0,00 | 0,00 | 0,00 |
| 6 | 0,00 | 0,00 | 8,33 | 0,00 | 0,00 | 0,00 |
| 7 | 16,67 | 33,33 | 8,33 | 25,00 | 66,67 | 41,67 |
| 8 | 0,00 | 0,00 | 8,33 | 16,67 | 0,00 | 0,00 |
| 9 | 8,33 | 8,33 | 8,33 | 0,00 | 0,00 | 8,33 |
| 10 | 8,33 | 8,33 | 0,00 | 0,00 | 0,00 | 0,00 |
| 11 | 0,00 | 25,00 | 50,00 | 0,00 | 25,00 | 16,67 |
| Total | 3,79 | 9,85 | 8,33 | 4,55 | 12,12 | 8,33 |

A quantidade de erros cometidos pelos usuários na realização das tarefas também foi contabilizada (tabela 5.2). Nota-se que o *Bifocal* foi o *browser* que apresentou a menor taxa de erros tanto na hierarquia pequena quanto na grande.

5.6 Análise dos resultados

A análise dos resultados obtidos foi elaborada através da comparação dos tempos médio em cada *browser*, em razão de não haver um padrão de eficiência estabelecido. Em uma rápida análise da tabela 5.1 nota-se que para grande parte das tarefas não há uma diferença significativa entre o desempenho com o *Bifocal* em relação aos demais. Entretanto, considerando as situações que as diferenças ocorreram o *Bifocal* obteve um resultado a seu favor na maioria delas. A análise a seguir será detalhada através da apresentação de considerações baseadas em cada aspecto descrito na seção 5.2.

5.6.1 Busca de um nodo

A busca de um determinado nodo na hierarquia foi avaliada através da execução de duas tarefas. Na primeira, a procura inicia no nível 2 com o nodo alvo localizado no nível 3 para a hierarquia pequena e no nível 4 para a grande. Na segunda tarefa, a busca inicia no nível 3 com o nodo procurado localizado nos níveis 5 e 6, respectivamente. Para as médias de tempos exibidas na figura 5.5, a Análise da Variância evidenciou diferenças significativas entre os tempos dos três *browsers* ($F=5,73$; $p=0,007$) apenas quando considerada a busca realizada em níveis mais próximos da raiz (tarefa 1) na hierarquia grande. Para essa situação, o teste de Kruskal-Wallis de múltiplas comparações revelou que a média de tempo do *Bifocal* é diferente da obtida para o *MagniFind* ($Z=3,02 > 1,960$).

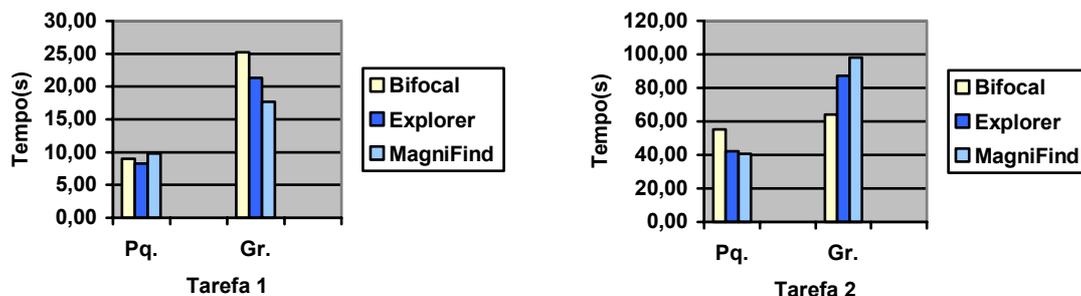


FIGURA 5.5 – Comparação das médias de tempo obtidas na execução das tarefas 1 e 2

Embora o *Bifocal* tenha obtido uma média de tempo menor (figura 5.5) para a busca em níveis mais profundos na hierarquia grande (tarefa 2), estatisticamente essa diferença não foi considerada significativa. Entretanto, cabe destacar que nesta situação 25% dos usuários que testaram o *MagniFind* cometeram erros ou não conseguiram realizar a tarefa no tempo máximo de 3 minutos, o que demonstra uma considerável ineficiência na busca em níveis mais profundos.

5.6.2 Navegação através de um caminho determinado

A navegação através de um caminho determinado foi avaliada no dois sentidos. A tarefa 3 avalia a navegação em direção aos níveis mais profundos e a 4, na direção

contrária. No primeiro tipo de navegação, embora o *Bifocal* tenha apresentado tempos médios inferiores (figura 5.6), principalmente em relação ao *MagniFind*, essa diferença não foi considerada estatisticamente significativa. Entretanto, analisando o resultado obtido pela Análise da Variância ($F=3,23$; $p=0,052$) da tarefa 3 realizada sobre a hierarquia pequena, observa-se que o valor de p encontra-se quase no limite que define a aceitação da diferença ($p<0,05$). Um experimento realizado com um maior número de sujeitos deve levar a um resultado mais confiável. Mas se mesmo assim, nesta situação particular, analisarmos o resultado obtido pelo teste de Kruskal-Wallis de múltiplas comparações observamos que as médias de tempo entre o *Bifocal* e *MagniFind* podem ser consideradas diferentes ($Z=2,274>1,960$). Esse pior desempenho do *Magnifind* pode ter sido causado pela mudança de layout produzida na seleção dos nodos. Embora o *Bifocal* também possua o mesmo problema, agravado pela falta de animação, a navegação é facilitada em função de que os nodos filhos daquele selecionado são distribuídos sempre no mesmo local e com a mesma orientação.

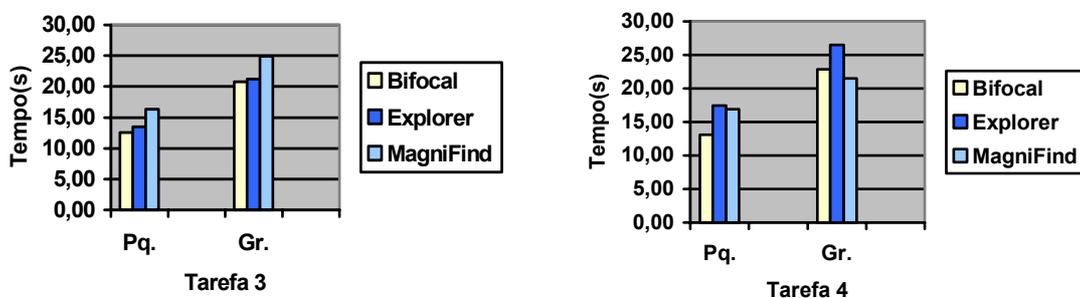


FIGURA 5.6 – Comparação das médias de tempo obtidas na execução das tarefas 3 e 4

Para a navegação em direção ao nodo raiz (tarefa 4) realizada na hierarquia pequena, o resultado obtido pela Análise de Variância também ficou quase no limite de aceitação da diferença ($F=3,15$, $p=0,055$). Se analisarmos os resultados obtidos pelo teste de Kruskal-Wallis de múltiplas comparações, como feito na tarefa anterior, observamos que as médias de tempo entre o *Bifocal* e o *Explorer* podem ser consideradas diferentes ($Z=2,365>1,960$) da forma que entre o *Bifocal* e o *MagniFind* ($Z=2,073>1,960$). Atribui-se esse resultado ao fato de que o nodo pai está sempre posicionado no foco da área de contexto e o nodo filho no foco da área de detalhe, o que torna a sua localização para a seleção bastante direta. Entretanto, esse desempenho não foi confirmado na hierarquia grande, provavelmente em razão de ocorrer uma perda de tempo na contagem dos filhos de cada nodo selecionado, informação solicitada na execução da tarefa. É possível que a distribuição angular utilizada na representação dos nodos filhos possa ter prejudicado a contagem.

Cabe ressaltar ainda, que a quantidade de erros cometidos neste tipo de navegação foi bastante significativa no *Explorer* (25% para hierarquia pequena e 33% para a hierarquia grande). Atribui-se a esse resultado ao fato de que alguns sujeitos tentaram fazer a contagem sem fechar as pastas abertas na tarefa anterior o que dificulta a sua realização. O atalho de teclado que o sistema possui com o objetivo de selecionar um nível acima foi utilizado apenas por um sujeito, em razão dos demais desconhecerem esse mecanismo.

5.6.3 Identificação dos relacionamentos pai/filhos

A identificação de relacionamentos existentes em uma hierarquia foi avaliada através de uma consulta da quantidade de filhos de nodos localizados próximos ao nodo raiz (tarefa 5) e em níveis mais profundos (tarefa 6). A avaliação da facilidade de identificar os níveis em que os nodos se encontram foi contemplada pela tarefa 7. Para as tarefas 5 e 6 (figura 5.7) a Análise da Variância não detectou diferenças significativas no desempenho das três técnicas tanto na execução na hierarquia grande quanto na pequena.

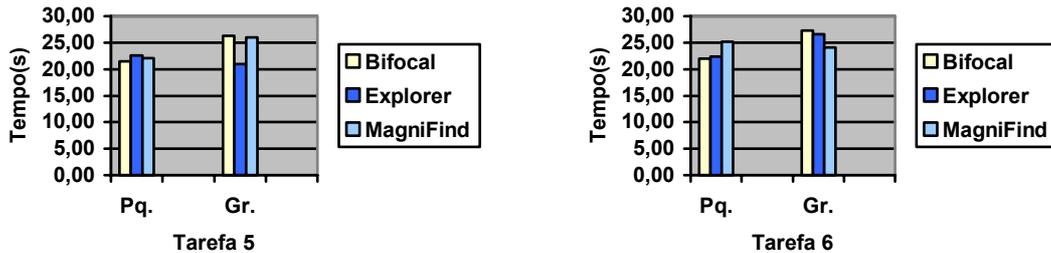


FIGURA 5.7 – Comparação das médias de tempo obtidas na execução das tarefas 5 e 6

A avaliação da tarefa 7 foi obtida através da análise da quantidade de erros cometidos pelos usuários em razão de seu grande número. Além disso, foi a única tarefa em que o sujeito não recebeu um retorno imediato de que ela tenha sido realizada com sucesso. Os erros foram identificados após a realização do experimento. Dessa forma, embora não tenham sido detectadas diferenças significativas (em ambos os testes estatísticos) entre os tempos médios obtidos, eles não foram levados em conta na avaliação. Como mostra a figura 5.8 33,33% dos usuários que testaram o *Explorer* cometeram erros na hierarquia pequena e 66,67% na hierarquia grande o que comprova a dificuldade de identificação dos níveis com esse *browser*.

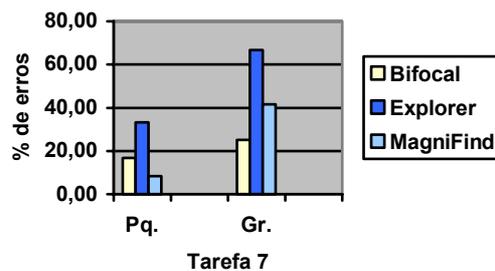


FIGURA 5.8 – Percentual de erros cometidos na execução da tarefa 7

Além disso, a tarefa exigiu a necessidade de uma visão geral da estrutura para orientar a navegação, característica prejudicada no *Explorer* quando uma hierarquia grande é exibida. Embora os 3 *browsers* tenham proporcionado uma taxa de erros significativa para a estrutura grande nota-se que a menor foi obtida com o *Bifocal*.

Cabe observar que houve algumas dificuldades na realização dessa tarefa, pois a estratégia adotada pelos sujeitos influenciou de forma significativa no resultado. Alguns não conseguiram completar a tarefa no limite de tempo estabelecido.

5.6.4 Comparação entre dois trechos da hierarquia

A eficiência dos *browsers*, em relação a facilitar a comparação de informações localizadas em trechos diferentes da hierarquia, foi avaliada através de uma consulta em nodos situados no mesmo nível (tarefa 8) e em dois níveis diferentes (tarefa 9).

Observando a figura 5.9 verifica-se que o tempo médio obtido com o *Bifocal* é menor quando a comparação é feita entre nodos localizados no mesmo nível (tarefa 8). Esta diferença mostrou-se significativa entre os três *browsers* apenas na hierarquia grande ($F=5,88$; $p=0,006$). Analisando os resultados obtidos pelo teste de Kruskal-Wallis de múltiplas comparações observa-se que as médias de tempo entre o *Bifocal* e o *Explorer* ($Z=2,975 > 1,960$) podem ser consideradas diferentes assim com entre o *Bifocal* e o *MagniFind* ($Z=2,665 > 1,960$).

Em relação a consulta efetuada em níveis distintos (tarefa 9), nota-se que existe uma vantagem do *Bifocal* em relação aos demais quando a hierarquia é pequena ($F=6,1$; $p=0,0055$). Analisando os resultados obtidos pelo teste de Kruskal-Wallis de múltiplas comparações observa-se que as médias de tempo entre o *Bifocal* e o *Explorer* ($Z=2,967 > 1,960$) podem ser consideradas diferentes assim com entre o *Bifocal* e o *MagniFind* ($Z=2,182 > 1,960$). No entanto, na hierarquia grande seu desempenho cai em relação ao resultado obtido na tarefa 8, enquanto os demais mantêm suas médias de tempo praticamente equivalentes.

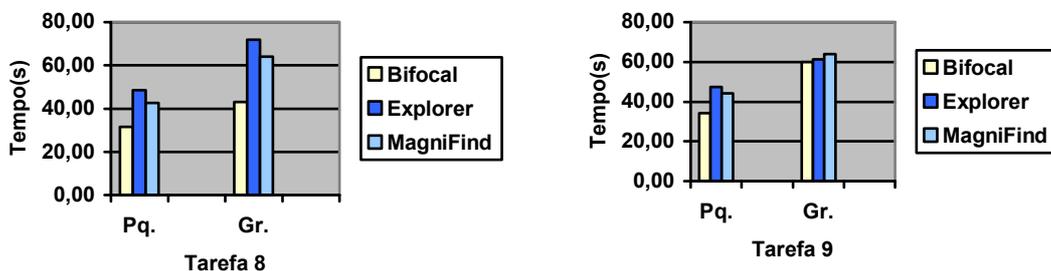


FIGURA 5.9 – Comparação das médias de tempo obtidas na execução das tarefas 8 e 9

5.6.5 Desorientação do usuário causada pela alteração do layout durante a navegação

Esse aspecto foi avaliado através da tarefa 10 em que o usuário deveria realizar consultas alternadas em dois pontos da hierarquia. Havia uma expectativa de que o *Bifocal* obtivesse um desempenho pior, em razão de que ele provoca uma alteração brusca do *layout* sem a utilização de animação. A Análise da Variância dos tempos médios (figura 5.10) para a hierarquia pequena confirmou esse fato ($F=4,12$; $p=0,023$). Entretanto, analisando os resultados obtidos pelo teste de Kruskal-Wallis de múltiplas comparações observa-se que apenas as médias de tempo entre o *Bifocal* e o *Explorer* ($Z=2,21 > 1,960$) foram consideradas diferentes. Para consultas realizadas na hierarquia grande não foram detectadas diferenças significativas entre os três *browsers*.

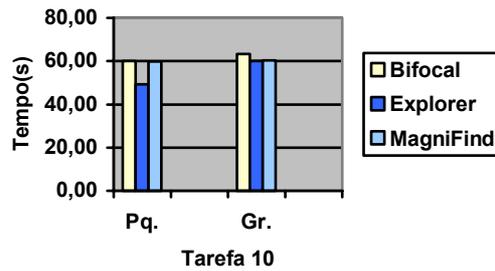


FIGURA 5.10 – Comparação das médias de tempo obtidas na execução da tarefa 10

Entretanto, ao observar a estratégia adotada pelos usuários no cumprimento dessa tarefa, registrou-se o fato de que praticamente todos eles executaram a consulta a cada subárvore iniciando a navegação sempre pelo nodo raiz, sem sequer analisar a localização do trecho selecionado anteriormente. Somente com o *Explorer*, alguns usuários utilizaram-se da estratégia de deixar a pasta dos trechos selecionados abertas, para posteriormente através do *scroll* da janela localizá-las novamente. Dessa forma, conclui-se que a tarefa não cumpriu de forma desejada a análise do efeito da desorientação provocada pelas alterações de layout. Entretanto, pode-se supor que a desorientação existente tenha motivado a estratégia adotada pelos sujeitos.

5.6.6 Representação de nodos que não são diretórios

A tarefa 11 foi criada com o objetivo de avaliar se a inexistência de representação de nodos que não são diretórios em dois dos *browsers* causaria alguma diferença nos tempos médios das consultas. Analisando o gráfico da figura 5.11 nota-se que praticamente não há diferença grande entre os tempos (fato comprovado pela Análise de Variância). Analisando a tabela dos erros cometidos, percebe-se que o *MagniFind* possui um desempenho pior. Como previsto, não foram cometidos erros com o *Bifocal* já que é o único *browser* a exibir todos os tipos de nodos na janela principal.

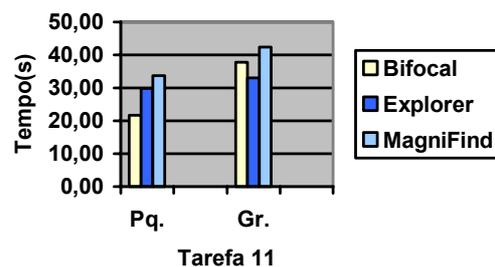


FIGURA 5.11 – Comparação das médias de tempo obtidas na execução da tarefa 11

O *Explorer* também proporcionou alguns erros, já que ele exibe todos os tipos de nodos apenas na área de detalhe. Atribui-se a diminuição da taxa de erros para a hierarquia grande ao fato de que o usuário provavelmente tenha levado em consideração os seus erros cometidos anteriormente na execução da tarefa com a hierarquia pequena.

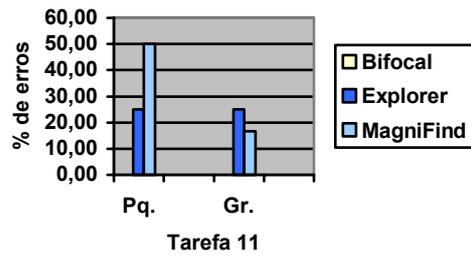


FIGURA 5.12 – Percentual de erros cometidos na execução da tarefa 11

5.7 Avaliação subjetiva

Como mencionado anteriormente, ao final da realização das tarefas, cada sujeito respondeu um questionário a fim de coletar a sua opinião em relação à utilização de cada *browser*. A seguir serão apresentados os resultados obtidos.

5.7.1 Representação visual.

A satisfação do usuário em relação à representação visual foi avaliada a partir das questões abaixo que foram respondidas atribuindo-se um grau de 1 a 5 cujo significado encontra-se entre parênteses. Analisando-se os gráficos apresentados na figura 5.13 nota-se que o *Bifocal* obteve a preferência dos usuários nas três questões, embora não havendo uma grande diferença em relação ao *MagniFind*. Ao *Explorer* são atribuídos os piores resultados.

Questões:

- 1.O desenho que representa a hierarquia torna a execução das tarefas (1.difícil ... 5.fácil)
- 2.A quantidade de informações exibidas são (1.inadequadas ... 5.adequadas)
- 3.As relações existentes nas estruturas são percebidas de forma (1.difícil ... 5.fácil)

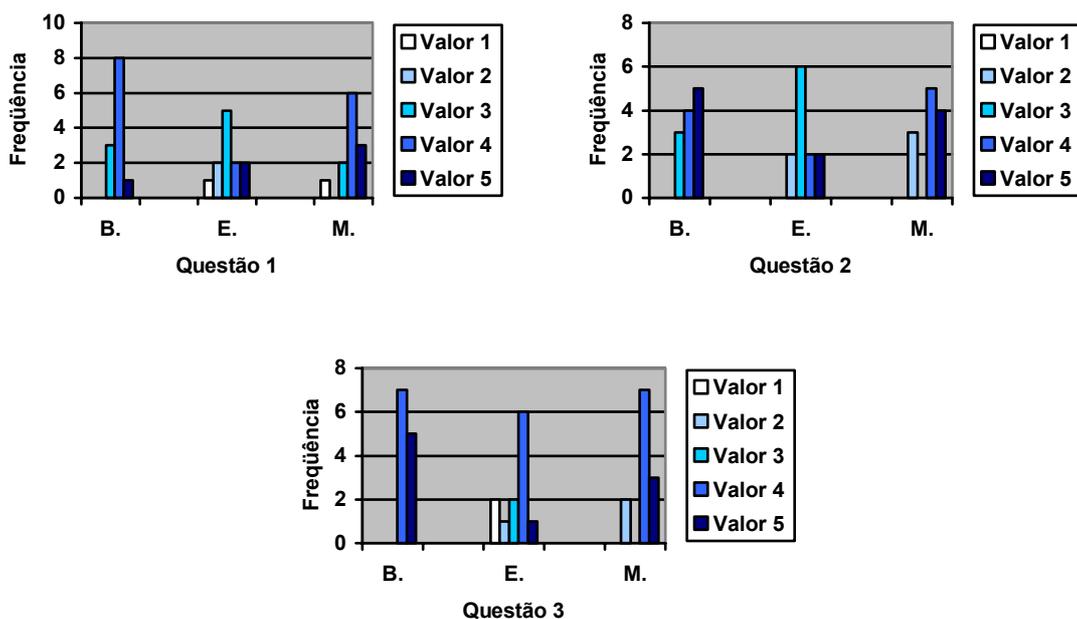


FIGURA 5.13 – Frequência de valores atribuídos às questões sobre a representação visual

5.7.2 Mecanismos de interação

Abaixo estão relacionadas as questões referentes aos mecanismos de interação oferecidos pelos *browsers*. Através da análise dos gráficos apresentados na figura 5.14 percebe-se que novamente o *Bifocal* obteve a preferência dos usuários, seguido pelo *MagniFind*. O *Explorer* recebeu a pior pontuação.

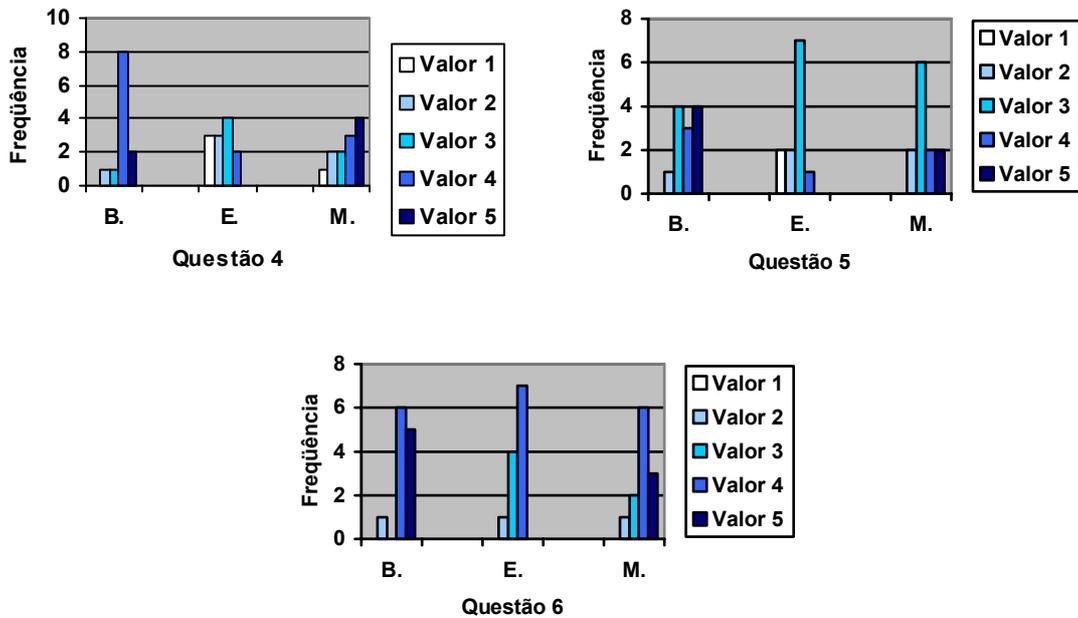


FIGURA 5.14 – Frequência de valores atribuídos às questões sobre mecanismos de interação

Questões

- 4.Desorientação causada pela mudança do layout durante a interação
(1.muita ... 5.pouca).
- 5.O número de interações por tarefa é (1.demais ... 5.suficiente)
- 6.As ferramentas utilizadas para a interação na execução das tarefas são
(1.inadequadas ... 5.adequadas)

6 Conclusões

As principais contribuições do trabalho são relacionadas à investigação de técnicas de visualização de estruturas hierárquicas. Foi proposta a *Bifocal Tree*, uma técnica alternativa para a exploração de informações hierárquicas que utiliza diagramas de nodos e arestas em sua representação e incorpora características existentes nas abordagens *Foco+Contexto* e *Visão Geral+Detalhe*. Embora, na representação adotada, os nodos mais próximos do foco sejam exibidos com um maior número de detalhes, característica da abordagem *Foco+Contexto*, a utilização de um segundo foco introduz uma divisão da área de exibição em dois espaços com objetivos distintos. O primeiro garante a visualização global da estrutura, enquanto o outro tem a função de exibi-la de forma a facilitar a percepção de seus detalhes, característica que define a abordagem *Visão Geral+Detalhe*.

A *Bifocal Tree* foi implementada através de um protótipo de browser (o *Bifocal Browser*) que oferece possibilidades de interação com a representação, embora não seja ainda um browser completo com filtros de busca, por exemplo. A técnica (e o *browser*) foram avaliados através de experimentação com usuários. Os resultados do trabalho até o momento foram publicados [CAV 2001a, CAV 2001b, CAV 2002], e sua utilização no contexto da tese de doutorado de Luzzardi [LUZ 2002] deverá subsidiar a definição de outros melhoramentos.

As próximas seções fazem, respectivamente, uma avaliação crítica do trabalho e delinham possibilidades de continuidade.

6.1 Avaliação geral

A integração entre as duas abordagens mencionadas anteriormente busca oferecer, além da visão geral da estrutura (área de contexto), um espaço para a visualização do ponto de interesse do usuário (área de detalhe). Neste espaço, as relações hierárquicas existentes na estrutura são percebidas com maior facilidade em razão do layout utilizado na construção da representação, que posiciona o nodo selecionado sempre no mesmo ponto (foco de detalhe). Além disso, a distribuição dos nodos pertencentes aos níveis inferiores é realizada no formato de círculos concêntricos em torno do nodo raiz, tornando mais previsível a sua localização, o que contribui para a facilidade de percepção da estrutura.

Além dos aspectos de disposição espacial dos nodos, a utilização das duas áreas (contexto e detalhe) situadas lado a lado contribui para o uso mais eficiente do espaço retangular da tela quando comparado às técnicas que se baseiam em layouts puramente radiais.

Outro aspecto positivo incorporado à *Bifocal Tree* é a oferta de mecanismos de interação que permite ao usuário configurar o layout a fim de favorecer a exibição da área de interesse. Essa característica ganha importância pelo fato de que em algumas situações, dependendo da hierarquia, a representação obtida pode não permitir a identificação dos relacionamentos com o mesmo grau de clareza, tornado-se desejável um ajuste manual.

No entanto, a área destinada ao contexto oferece uma certa dificuldade de navegação quando a estrutura é composta por um grande número de nodos em razão da ocorrência de sobreposição dos retângulos que os representam. Ainda, quando o nodo selecionado pelo usuário situa-se em níveis mais profundos, a identificação dos relacionamentos entre os nodos localizados em torno do foco da área de contexto não é tão imediata em razão da aplicação da operação de translação, que desloca o nodo raiz para um local próximo dos limites da área de exibição. Esse fato foi comprovado pela observação dos usuários durante a realização do experimento de avaliação. Todos eles efetuaram a navegação quase sempre na área de detalhe.

Um outro ponto crítico existente na técnica proposta é a desorientação do usuário provocada pela mudança brusca de layout quando um determinado nodo é selecionado. Na área de contexto, esse fato é agravado em função da rotação aplicada à estrutura, com o objetivo de permitir a sua ligação com a área de detalhe, que dificulta a previsão de onde cada nodo será posicionado após a seleção. Na tentativa de minimizar esse problema o trecho da estrutura anteriormente selecionado é exibido na área de contexto em uma cor diferenciada. Já na área de detalhe, esse fato não causa grande prejuízo, pois no momento da seleção a atenção do usuário está voltada para o trecho de interesse que será exibido sempre da mesma maneira. Entretanto, é importante salientar que diferenças existentes na estrutura, como por exemplo a quantidade de filhos e descendentes, afetam de forma significativa o layout obtido podendo dificultar, por consequência, a leitura dos relacionamentos.

Em relação ao experimento realizado com os usuários para avaliar a técnica proposta, pode-se considerar que de uma forma geral o resultado obtido foi bastante positivo. Os vários aspectos levantados no processo subsidiarão futuros trabalhos de avaliação. Por exemplo, embora o experimento tenha sido preparado com o cuidado de que fatores externos influenciassem o mínimo possível, observou-se que as diferenças entre os usuários causaram uma certa influência. A partir da observação da execução de cada tarefa notou-se que a estratégia que cada sujeito utilizou para realizá-las foi a responsável pela grande variação nos tempos medidos. Outro fator importante foi o conhecimento prévio da ferramenta. Em razão do pequeno tempo fornecido para o treinamento no uso do *Bifocal Browser* e do *MagniFind* observou-se que a maioria dos sujeitos não utilizou os mecanismos oferecidos para o ajuste do layout. Entretanto, os sujeitos demonstraram grande habilidade no uso do *Windows Explorer*, como era esperado.

6.2 Trabalhos futuros

Como mencionado anteriormente, a técnica proposta possui algumas limitações que poderiam ser investigadas em novos trabalhos. Uma alternativa é estudar uma forma de realizar as transições entre duas seleções através de animação a fim de minimizar a desorientação do usuário.

A forma de representação utilizada na exibição da estrutura na área de contexto também poderia ser objeto de estudo. Em grandes hierarquias, com nodos que possuem muitos filhos, a representação torna-se muito densa, exibindo muito próximas entre si as linhas que ligam os nodos que a compõem. Uma alternativa para suavizar o diagrama seria investigar a utilização de agrupamentos para representar os subconjuntos da

hierarquia onde ocorre essa situação. Trabalho nessa linha foi iniciado por Chubachi [CHU 2002].

Observando o desenho gerado para diferentes tipos de hierarquias, nota-se que nem sempre o layout obtido representa a estrutura com a máxima clareza, o que muitas vezes exige um ajuste manual dos parâmetros utilizados na sua geração. Uma alternativa de estudo é verificar a possibilidade de gerar automaticamente o valor desses parâmetros. O algoritmo para geração do layout também poderia levar em conta a forma da estrutura a ser visualizada a fim de obter um melhor efeito estético.

O processo de avaliação também pode ser objeto de estudo, analisando os resultados de um experimento realizado com um maior número de sujeitos divididos em grupos com características diferentes da que foi utilizada neste trabalho, como por exemplo, usuários com pouca experiência no uso de computadores. Além disso, a avaliação poderia ser realizada com hierarquias que possuem características diferentes entre si. Por exemplo, estruturas onde o grau dos nodos possua uma grande variação, estruturas totalmente desbalanceadas, estruturas com grande variação de altura de suas subárvores. Por fim, este trabalho pode ser usado como ferramenta de teste no desenvolvimento de critérios para avaliação de técnicas de visualização de informações por Luzzardi [LUZ 2001, FRE 2001, LUZ 2002].

Anexo 1 Principais algoritmos

Este anexo contém uma descrição simplificada dos principais algoritmos relacionados com a determinação das coordenadas dos nodos exibidos nas áreas de contexto e detalhe.

```

/*-----*/
| Calcula coordenadas dos nodos localizados na área de contexto
*/
inicio
  Determinar as coordenadas do nodo raiz
  Para cada nodo localizado no primeiro nível da árvore faça
    Se o nodo não está localizado no foco da área de detalhe então
      Calcular as Coordenadas do Nodo para área de contexto
    FimSe
  FimPara
  Gerar as arestas que irão representar a árvore na área de contexto
Fim.

/*-----*/
| Calcular as Coordenadas do Nodo para área de contexto
*/
inicio
  Determinar as coordenadas do nodo
  Se o nodo possui filhos então
    Para cada filho faça
      Se o filho não está localizado no foco da área de detalhe então
        Se o filho é ancestral do nodo localizado no foco da área de detalhe então
          Calcular o ângulo do setor ocupado pelo nodo (Eq. 4.3)
        Senão
          Calcular o ângulo do setor ocupado pelo nodo (Eq. 4.2)
        FimSe
      Chamar recursivamente: Calcular as Coordenadas do Nodo para área de contexto
    FimSe
  FimPara
Fimse
Fim.

/*-----*/
| Calcula coordenadas dos nodos localizados na área de detalhe
*/
inicio
  Calcular as coordenadas do Nodo para área de detalhe no primeiro nível
  Gerar as arestas que irão representar a árvore na área de detalhe
Fim.

/*-----*/
| Calcular as Coordenadas do Nodo para área de detalhe no primeiro nível
*/
inicio
  Determinar as coordenadas do nodo raiz
  Se o nodo raiz possui filhos então
    Se o n. de filhos  $\leq 20$  ou setor angular/n. de filhos  $>$  ângulo mínimo então
      Para cada filho faça
        Calcular o ângulo do setor ocupado pelo nodo (Eq. 4.2)
        Se nodo está localizado nos quadrantes I ou IV então
          Calcular a distância através da Eq. 4.6
        FimSe
        Calcular as Coordenadas do Nodo Filho p/ área de detalhe
      FimPara
    Senão
      Determinar quais serão os 20 nodos exibidos nos quadrantes I e III
      Para cada filho faça
        Calcular o ângulo do setor ocupado pelo nodo (Eq. 4.2)
        Se nodo está localizado nos quadrantes I ou IV então
          Calcular a distância através da Eq. 4.6
        FimSe
        Calcular as Coordenadas do Nodo Filho p/ área de detalhe
      FimPara
    FimSe
  Fimse
Fim.

```

```
/*-----*/  
| Calcular as Coordenadas do Nodo Filho p/ área de detalhe  
*/  
inicio  
  Determinar as coordenadas do nodo  
  Se o nodo possui filhos então  
    Para cada filho faça  
      Calcular o ângulo do setor ocupado pelo nodo (Eq. 4.2)  
      Chamar recursivamente: Calcular as Coordenadas do Nodo Filho p/área de detalhe  
    FimPara  
  Fimse  
Fim.
```

Bibliografia

- [AND 95] ANDREWS, Keith. Visualizing Cyberspace: Information Visualization in the Harmony Internet Browser. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1995, Atlanta. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1995. p.97-104.
- [AND 98] ANDREWS, Keith; HEIDEGGER, Helmut. Information Slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1998, Raleigh Durham, North Carolina. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.9-12
- [BEA 96] BEAUDOIN, Luc; PARENT Marc-Antoine; VROOMEN, Louis C. Cheops: A compact explorer for complex hierarchies. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1996. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1996. p.87-92.
- [BER 90] BEARD, D. B.; WALKER, J. Q. Navigational Techniques to Improve the Display of Large Two-Dimensional Spaces. **Behavior and Information Technology**, New York, v.9, n.6, p.451-466, 1990.
- [BJO 2000] BJÖRK, S. Hierarchical Flip Zooming: Enabling Parallel Explorations of Hierarchical Visualizations. In: CONFERENCE ON ADVANCED VISUAL INTERFACES, 2000, Italy, Palermo. **Proceedings...** New York:ACM Press, 2000. p.232-237.
- [BRA 97] BRATH, Richard. Concept Demonstration: Metrics for Effective Information Visualization. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1997. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1997. p.108-111.
- [CAR 91] CARD, Stuart K.; ROBERTSON, George G.; MACKINLAY, Jock D. The Information Visualizer, an information workspace. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS CHI, 1991, New Orleans. **Proceedings...** New York: ACM Press, 1991. p.181-188.
- [CAR 95] CARRIÈRE, J.; KAZMAN,R. Interacting with Hierarchies: Beyond Cone Trees. In: IEEE CONFERENCE ON INFORMATION VISUALIZATION, 1995. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1995. p.74-81.
- [CAR 99] CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. Information Visualization. In: CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. **Readings in Information Visualization: Using Vision to Think**. San Francisco, California: Morgan Kaufmann Publishers, 1999. p.1-34.

- [CAR 99a] CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. Overview + Detail. In: CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. **Readings in Information Visualization: Using Vision to Think**. San Francisco, California: Morgan Kaufmann Publishers, 1999. p.285-286.
- [CAR 99b] CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. Focus + Context. In: CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. **Readings in Information Visualization: Using Vision to Think**. San Francisco, California: Morgan Kaufmann Publishers, 1999. p.307-309.
- [CAV 2001a] CAVA, R.A.; FREITAS, C.M.D.S. Visualizing Hierarchies Using a Modified Focus+Context Technique (interactive poster) In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2001, San Diego, California. **Late Break Hot Topics: Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001.
- [CAV 2001b] CAVA, R.A.; FREITAS, C.M.D.S. Node-Edge Diagram Layout for Displaying Hierachies. In: BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, 14., 2001, Florianópolis, SC. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2001. p. 373
- [CAV 2002] CAVA, R.A.; LUZZARDI, P.R.G.; FREITAS, C.M.D.S. The Bifocal Tree: a Technique for the Visualization of Hierarchical Information Structures. In: WORKSHOP ON HUMAN FACTORS IN COMPUTER SYSTEMS, IHC, 2002, Fortaleza, CE. **Anais...** Fortaleza: SBC, 2002.
- [CHI 98] CHI, Ed H. et al. Visualizing the evolution of web ecologies. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1998, Los Angeles. **Proceedings...** New York: ACM Press 1998.
- [CHU 2002] CHUBACHI, O.M. **Mecanismo de Agrupamento Baseado em Estrutura para uma Técnica de Visualização Foco+Contexto**. Dissertação de mestrado em andamento (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre
- [FRE 2002] FREITAS, C.M.D.S.; LUZZARDI, P.R.G.; CAVA, R.A.; WINCKLER, M.A.A.; PIMENTA, M.S.; NEDEL, L.P. Evaluating Usability of Information Visualization Techniques. Aceito para publicação e apresentação no IHC 2002 - Workshop sobre Fatores Humanos em Sistemas Computacionais, Fortaleza, CE, outubro 2002.
- [FUR 81] FURNAS, George W. The FISHEYE View: A New Look at Structured Files. In: CARD, Stuart K.; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. **Readings in Information Visualization: Using Vision to Think**. San Francisco, California: Morgan Kaufmann Publishers, 1999. p.312-330.

- [FUR 86] FURNAS, George W. Generalized fisheye views. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1986. **Proceedings...** New York: ACM Press 1986. p.16-23.
- [GER 97] GERSHON, N. e EICK,S. Information Visualization. **IEEE Computer Graphics and Applications**, New York, v.17,n.4,p.29-31, July/Aug. 1997
- [HER 2000] HERMAN, Ivan; MELANÇON, Guy; MARSHALL, M. Graph Visualization and Navigation in Information Visualization: A Survey. **IEEE Transactions on Visualization and Computer Graphics**, New York, v.6, n.1,p.24-43, January/March. 2000.
- [HOL 97] HOLMQUIST, L.E. Focus+Context Visualization with Flip Zooming and the Zoom Browser. In: EXTENDED ABSTRACT OF CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1997. **Proceedings...** New York: ACM Press, 1997. p.263-264.
- [HOL 98] HOLMQUIST, L.E. The Zoom Browser Showing Simultaneous Detail and Overview in Large Documents. **Human IT**, Boras, Sweden, v.2, n.3,p.131-150,1998.
- [HOR 2001] HORNBAEK, K.; BEDERSON, B.; PLAISANT, C. Navigation Patterns and Usability of Overview+Detail and Zoomable User Interfaces for Maps. Maryland: University of Maryland, Department of Computer Science, 2001. (Technical Report 2001-11).
- [JEO 98] JEONG, C.; PANG, A. Reconfigurable disc trees for visualizing large hierarchical information space. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1998, Raleigh Durham, North Carolina. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p.19-25
- [JOH 91] JOHNSON, Brian; SHNEIDERMAN, Ben. Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In: IEEE VISUALIZATION, 1991, San Diego. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1991. p.284-291.
- [JUA 2000] JUAREZ, O. et al. Evaluating Visualizations Based on the Performed Task. In: INTERNATIONAL CONFERENCE ON INFORMATION VISUALISATION, 2000, London, England. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2000. p.135-144.
- [LAM 95] LAMPING, John; RAO, Ramana; PIROLI, Peter. A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1995. **Proceedings...** New York: ACM Press, 1995. p.401-408.

- [LUZ 2001] LUZZARDI, Paulo R.G. **Avaliação de Técnicas de Visualização de Informações**. 2001. Proposta de Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [LUZ 2002] LUZZARDI, Paulo R.G.; WINCKLER, M.A.A.; CAVA, R.A.; PIMENTA, M.S.; NEDEL, L.P.; FREITAS, C.M.D.S. Usability of Information Visualization Techniques. Submetido ao CODATA Journal, 2002.
- [MAC 91] MACKINLAY, Jock D.; ROBERTSON, George G.; CARD, Stuart K. The Perspective wall: Detail and context smoothly integrated. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1991, New Orleans. **Proceedings...** New York: ACM Press, 1991. p.173-179.
- [MUN 95] MUNZNER, Tamara; BURCHARD, Paul. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. In: SYMPOSIUM ON THE VRML MODELING LANGUAGE, 1995, San Diego, CA. **Proceedings...** New York: ACM Press, 1995. p.33-38.
- [MUN 97] MUNZNER, Tamara. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1997. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1997. p.2-10.
- [PIR 2000] PIROLI, Peter; CARD, Stuart K.; WEGE, Mija M.V.D. The effect of information scent on searching information visualizations of large tree structures. In: CONFERENCE ON ADVANCED VISUAL INTERFACES, 2000, Palermo, Italy. **Proceedings...** New York: ACM Press, 2000.
- [PLA 95] PLAISANT, C.; CARR, D.; SHNEIDERMAN, B. Image browsers Taxonomy, guidelines and informal specifications. **IEEE Software**, Los Alamitos, CA, v.12, n.2, p.21-32, 1995.
- [PLA 96] PLAISANT, Catherine et al. LifeLines: Visualizing Personal Histories. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS CHI, 1996. **Proceedings...** New York: ACM Press, 1996. p.221-227.
- [REI 81] REINGOLD, Edward M.; TILFORD, John S. Tidier Drawings of Trees. **IEEE Transactions on Software Engineering**, New York, v.7, n.2, p.223-228, Mar. 1981.
- [REK 93] REKIMOTO, Jun; GREEN, Mark. The Information Cube: Using Transparency in 3D Information Visualization. In: ANNUAL WORKSHOP ON INFORMATION TECHNOLOGIES AND SYSTEMS, WITS, 3., 1993. **Proceedings...** [S.l.:s.n.], 1993. p.125-132.

- [ROB 89] ROBERTSON, George G.; CARD, Stuart K.; MACKINLAY, Jock D. The cognitive coprocessor architecture for interactive user interfaces. In: SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, UIST, Williamsburg, 1989, Virginia. **Proceedings...** New York: ACM Press, 1989. p. 10-18.
- [ROB 91] ROBERTSON, George G.; CARD, Stuart K.; MACKINLAY, Jock D. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS CHI, 1991, New Orleans. **Proceedings...** New York: ACM Press, 1991. p.189-194.
- [SAR 92] SARKAR, Manojit; BROWN, Marc H. Graphical fisheye views of graphs. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1992. **Proceedings...** New York: ACM Press, 1992. p.83-91.
- [SCH 96] SCHAFFER, Doug et al. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. **ACM Transactions on Computer-Human Interaction**, New York, v.3, n.2, p.162-188, June 1996.
- [SHN 92] SHNEIDERMAN, Ben. Tree Visualization with Treemaps: A 2d space filling approach. **ACM Transactions on Graphics**, New York, v.11, n.1, p.92-99, Jan. 1992.
- [SHN 96] SHNEIDERMAN, Ben. The Eyes Have It: A Task Data Type Taxonomy for Information Visualization. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES, 12., 1996, Boulder, CO. **Visual languages and innovation: proceedings**. Los Alamitos: IEEE Computer Society, 1996. p.326-343.
- [SPE 82] SPENCE, R.; APPERLEY, M. Data base navigation: an office environment for the professional. **Behavior and Information Technology**, New York, v.1, n.1, p.43-54, 1982.
- [STA 2000] STASKO, J.; ZHANG, E. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 2000, San Francisco, California. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2000. p.57-65.
- [TUK 77] TUKEY, J.W. **Exploratory data analysis**. Reading, Mass.:Addison Wesley, 1977.
- [TUR 92] TURO, David; JOHNSON, Brian. Improving the Visualization of Hierarchies with Treemaps: Design Issues and Experimentation. In: IEEE VISUALIZATION, 1992. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1992. p.2-9.

- [TVE 93] TVERSKY, Oren J.; SNIBBE, Scott S.; ZELEZNIK, Robert. **Cone Tree in the UGA Graphics System: Suggestions of a more Robust Visualization Tool**. Providence, Brown University, Department of Computer Science, 1993. (Technical Report CS-93-07).
- [WIJ 99] WIJK, Jarke J. V.; WETERING, Huub V. Cushion Treemaps: Visualization of Hierarchical Information. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1999, San Francisco, California. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.73-78.
- [WIS 98] WISS, Ulrika; CARR, David; JONSSON, Hakan. **Evaluation 3-Dimensional Information Visualization Designs**. Lulea, Sweden: Lulea University of Technology, Department of Computer Science and Electrical Engineering, 1998. Technical Report.
- [WIL 99] WILSON, Richard M.; BERGERON, Daniel R. Dynamic Hierarchy Specification and Visualization. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, 1999, San Francisco, California. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1999. p.65-72.