

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

**O Uso de Ícones na Visualização
de Informações**

por

LUIZ FERNANDO ESTIVALET

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

Profa. Carla Maria Dal Sasso Freitas
Orientadora

Porto Alegre, dezembro de 2000.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Estivalet, Luiz Fernando

O Uso de Ícones na Visualização de Informações / por Luiz Fernando Estivalet. - Porto Alegre: PPGC da UFRGS, 2000.

90f.:il.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2000. Orientadora: Freitas, Carla Maria Dal Sasso.

1. Visualização de Informações. 2. Ícones. 3. Computação Gráfica. 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. Philippe Olivier Alexandre Navaux

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

Coordenadora do PPGC: Profa. Carla Maria Dal Sasso Freitas

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

Agradecimentos

Gostaria de expressar minha gratidão à professora Carla M. Dal Sasso Freitas, pelas sugestões, solicitude e apoio constantes, sem os quais, com certeza, não seria possível a realização deste trabalho. Aos biólogos do Departamento de Genética da UFRGS, prof. Thales Renato O. de Freitas e doutorando Tarik El Jundi pela cessão dos dados e esclarecimentos quando da realização do estudo de caso.

Também devo um agradecimento especial ao apoio financeiro proporcionado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de auxílio para a realização do mestrado.

Um agradecimento muito especial a duas pessoas muito especiais em minha vida: minha mãe, Venila Maria Estivalet, e Márcia Costa Curta pelo apoio, compreensão, paciência e principalmente muito amor, que contribuíram em muito para que eu pudesse alcançar essa realização.

Sumário

Lista de Figuras	6
Lista de Tabelas	8
Resumo	9
Abstract	10
1 Introdução	11
2 Visualização de informações	14
2.1 Conceito	14
2.2 Visualização e mineração de dados	16
2.3 Linhas gerais e requisitos para gerar efetivas representações visuais	19
2.4 Técnicas de visualização de informações	26
2.5 Visualização e banco de dados	29
2.5.1 IVEE (Information Visualization and Exploration Enviroment)	29
2.5.2 VISAGE	31
2.5.3 EXBASE	32
2.5.4 Glyphmaker	33
3 Visualização icônica	36
3.1 Conceitos básicos	36
3.2 Abordagens para visualização icônica	39
4 Geração automática de ícones	49
4.1 Introdução	49
4.2 Proposta para geração de ícones	51
5 Estudo de caso	55
5.1 Descrição da aplicação-exemplo	55
5.2 Protótipo	58
5.2.1 O acesso ao banco de dados	59
5.2.2 Ícones utilizados	63
5.2.3 Técnica de visualização adotada	64
5.2.4 Módulos e funcionamento do protótipo	67
5.2.4.1 Pré-processamento	67
5.2.4.2 Cadastro	70
5.2.4.3 O construtor de consultas interativo	70

5.2.4.4 Mapeador de ícones	73
5.2.4.5 Visualização	76
6 Avaliação	80
7 Conclusões e extensões	85
Bibliografia	87

Lista de Figuras

FIGURA 2.1 - Ilustração esquemática do processo de visualização	14
FIGURA 2.2 - Técnica <i>perspective wall</i>	28
FIGURA 2.3 - Visualização gerada pelo IVEE.....	31
FIGURA 2.4 - Tela do Glyphmaker.....	34
FIGURA 3.1 - Estágios do processo de visualização usando ícones	37
FIGURA 3.2 - Extração de características multi-nível.....	38
FIGURA 3.3 - Exemplo de ícone <i>stick-figure</i>	40
FIGURA 3.4 - Exemplos de ícones de linha no <i>Exvis</i>	40
FIGURA 3.5 - Ícones <i>stick-figures</i>	41
FIGURA 3.6 - Características do ícone velcro.....	41
FIGURA 3.7 - Configuração básica do ícone de cor no <i>Exvis</i>	42
FIGURA 3.8 - Exemplo de ícones de linhas de cor	43
FIGURA 3.9 - Exemplo de ícones de cor.....	43
FIGURA 3.10 - Exemplo de ícones velcro	44
FIGURA 3.11 - <i>Timewheel glyph</i>	45
FIGURA 3.12 - <i>3D wheel glyph</i>	45
FIGURA 3.13 - <i>Infobug glyph</i>	46
FIGURA 3.14 - <i>Weather glyph</i>	47
FIGURA 3.15 - Geometria do glifo para representar o processo de preenchimento de moldes	47
FIGURA 4.1 - Faces de <i>Chernoff</i>	49
FIGURA 4.2 - Tela do <i>GMVT</i>	51
FIGURA 4.3 - Divisões possíveis de um ícone	53
FIGURA 4.4 - Marcações possíveis de um ícone	53
FIGURA 5.1 - Região de estudos dos biólogos do Departamento de Genética da UFRGS	56
FIGURA 5.2 - Arquitetura do projeto <i>IconVis</i>	58
FIGURA 5.3 - Estrutura de um sistema gerenciador de banco de dados	61
FIGURA 5.4 - Acesso a banco de dados usando o BDE.....	62
FIGURA 5.5 - Estrutura interna do Borland Database Engine	63
FIGURA 5.6 - Ícones utilizados no protótipo	63
FIGURA 5.7 - Tela mostrando o universo de pesquisa dos biólogos	65
FIGURA 5.8 - Visualização de uma parte selecionada do universo.	66
FIGURA 5.9 - Esquema para localização do ponto de coleta.....	69
FIGURA 5.10 - Tela de cadastro de novos animais	71
FIGURA 5.11 - Construtor de consultas	71

FIGURA 5.12 - Tela do construtor de consultas	72
FIGURA 5.13 - Tela do mapeador de ícones	74
FIGURA 5.14 - Escala de cores	74
FIGURA 5.15 - Detalhamento da tela principal do <i>IconVis</i>	77
FIGURA 5.16 - Detalhes dos animais	78
FIGURA 5.17 - Visualização após mapeamento.....	79

Lista de Tabelas

TABELA 2.1 - Visualização de informação comparada à visualização científica	15
TABELA 2.2 - Classificação dos dados	20
TABELA 2.3 - Classificação de representações visuais	22

Resumo

A visualização de informações congrega um conjunto de técnicas que têm por objetivo facilitar o entendimento de informações a partir de representações visuais. Nesta área, parte-se do pressuposto de que uma representação visual de dados ou informações proporciona uma forma mais simples e intuitiva de entendê-los e, com isso, inferir mais rapidamente e com maior precisão o seu significado. Normalmente, a tarefa de análise de dados é realizada com o auxílio de ferramentas de acordo com a natureza dos dados e do estudo sendo realizado. As técnicas de visualização de dados e informações não substituem ferramentas de análise específicas. Elas podem ser usadas como primeira aproximação do processo de análise, quando sintetizam de forma visual uma grande massa de dados, permitindo escolher partes do volume de dados para análise mais detalhada, ou como forma de apresentação de dados já reduzidos. Quando o subconjunto dos dados de interesse está disponível, a utilização de uma ferramenta de visualização proporciona maior agilidade nas análises e padrões na massa de dados podem ser descobertos visualmente.

Uma das classes de técnicas de visualização utilizada nesta área é a icônica. Um *ícone* (ou *glifo*) é um objeto com geometria e aparência paramétricas, as quais podem ser arbitrariamente vinculadas a dados. A função de um ícone é agir como uma representação simbólica, que mostra as características essenciais de um domínio de dados ao qual o ícone se refere. Assim é possível obter uma visualização dos dados de uma forma mais clara e compacta. Em geral, ícones são utilizados para representar dados multidimensionais, ou seja, múltiplos atributos associados a uma posição num espaço qualquer, ou a entidades em estudo.

O presente trabalho analisa o uso de ícones na visualização de informações. São discutidos os conceitos fundamentais de visualização de informações e sua relação com a área de mineração de dados. O uso de ícones em diversos trabalhos apontados na literatura é apresentado, sendo abordada a questão de geração automática de ícones, mais flexível do que os conjuntos fixos providos pelos sistemas de visualização estudados. Uma proposta para gerar ícones de forma automática baseada numa especificação paramétrica dos ícones é utilizada em um conjunto de dados característicos de espécimes animais estudados por biólogos do Departamento de Genética da UFRGS.

Palavras-chave : ícones, visualização de informação, visualização científica, mineração de dados.

TITLE: "THE USE OF ICONS IN INFORMATION VISUALIZATION"

Abstract

Information Visualization corresponds to a set of techniques with the goal of making easier to understand information by means of visual representations. In this field, a visual representation of data and other kind of information provides a simple and intuitive way of understanding the data's meaning. Usually, the data analysis task is performed with the aid of tools according to the nature of data and the subject of study. Data visualization techniques do not replace specific data analysis tool. Through the use of a data analysis tool, which reduces the amount of data to a subset of interest to be explored, the use of a visualization tool will improve performance, making it easier to analyse and find patterns in the visualization generated with less data.

One class of these techniques is iconic visualization. An icon, sometimes called glyph, is an object with parametric geometry and appearance, to which data can be mapped to. The role of an icon is to act as a symbolic representation that shows the features of interest of the data under study. Then it is possible to obtain a data visualization in compact and simple form. Usually, icons are used to represent multidimensional or multivariate data, i.e. multiple attributes associated to a location in space or entity in a group.

This work surveys the utilization of icons in information visualisation. Initially, fundamentals concepts of information visualisation are presented as well as their relation with data mining. The use of icons in several works found in literature is presented here, and a prototype is described demonstrating the use of icons in the visualisation of data studied by biologists in Genetics Department at UFRGS.

Keywords : icons, information visualisation, scientific visualization, data mining.

1 Introdução

A visualização une os dois maiores sistemas de processamento de informações conhecidos – a mente humana e o computador moderno. Um processo transforma dados e informações em uma representação visual explorando o potencial natural das pessoas de rápido reconhecimento de padrões visuais [GER 97]. Interfaces visuais efetivas permitem-nos observar, manipular, pesquisar, navegar, explorar, filtrar, descobrir, entender e interagir com grandes volumes de dados muito mais rapidamente e muito mais efetivamente para descobrir padrões ocultos.

A partir de 1989, um novo foco de pesquisa e desenvolvimento estabeleceu-se com a comunidade de visualização para solucionar alguns dos problemas fundamentais associados com novas classes de dados e suas respectivas tarefas de análise e interpretação. Esse foco, chamado de **Visualização de Informações** [ROB 93], combina aspectos da visualização científica, interfaces homem-computador, mineração de dados, processamento de imagens e computação gráfica. Em contraste com a maior parte da visualização científica, a qual tem o foco em dados concretos medidos, amostrados ou calculados de alguma forma, a visualização de informações tem o foco em informação obtida dos dados, sendo esta geralmente abstrata. Informação em muitos casos não é automaticamente mapeada para o mundo físico. Essa diferença fundamental significa que muitas classes interessantes de informação não possuem um representação física óbvia e natural. O problema chave é, então, descobrir novas metáforas visuais para representar informações e para entender quais tarefas de análise elas suportam.

Outra tema chave na visualização de informações envolve a facilidade de uso. Em contraste com a visualização científica, a qual geralmente serve para cientistas altamente treinados, as interfaces criadas para manipulação de informações precisam ser amplamente “usáveis” por uma comunidade diversa e potencialmente não-técnica. Essa comunidade de usuários é diversa, com diferentes níveis de educação, capacidade, e necessidades. É necessário capacitar esse grupo diverso para usar representações visuais adequadas às suas necessidades específicas e ao problema em questão.

Visualização de informações é usada para converter grandes quantidades de informação em uma representação visual interpretável e com significado. Entretanto, produzir uma visualização efetiva não é muito fácil. As representações visuais são projeções em duas ou três dimensões de espaços de informação geralmente multi-dimensionais. Encontrar imagens efetivas de tais estruturas complexas que carregam toda a informação requerida é difícil. Robertson [ROB 98] resume muito bem as questões envolvidas nesta área quando usa a expressão "percepção da informação", já

que a maior parte da pesquisa em visualização de informações tem como objetivo aumentar a percepção humana das informações. Adicionalmente, o mesmo autor menciona que os outros sentidos humanos são utilizados na percepção do mundo real e, portanto, poderiam ser utilizados na percepção de informações representadas visualmente. Em particular, essa tarefa é impossível sem o conhecimento da estrutura e conteúdo do espaço da informação. Assim, uma ferramenta de visualização iria ajudar os projetistas a facilmente desenvolverem representações visuais efetivas do espaço de informação para os usuários finais. E também ajudaria os usuários avançados a realizarem a exploração de dados de que necessitam.

Os aspectos mais importantes nessas técnicas de visualização são a extração de características e o seu mapeamento para representações visuais. Uma *característica* pode ser definida [GER 97] como uma região de interesse para interpretação em uma base de dados. As características são extraídas e representadas como um conjunto de atributos, que são uma representação abstrata dos dados originais. A utilidade e eficiência das imagens geradas são dependentes das representações visuais escolhidas e, por isso, é necessário um estudo aprofundado dos atributos visuais a serem utilizados em relação às informações a serem representadas.

Dentre muitas representações utilizadas estão os ícones ou glifos. Um *ícone* (ou *glifo*) é um objeto com geometria e aparência paramétricas, as quais podem ser arbitrariamente vinculadas a dados. A função de um ícone é agir como uma representação simbólica, que mostra as características essenciais de um domínio de dados ao qual o ícone se refere. Assim é possível obter uma visualização dos dados de uma forma mais clara e compacta. Em geral, ícones são utilizados para representar dados multivariados, ou seja, múltiplos atributos associados a uma posição num espaço qualquer, ou a entidades em estudo. Tais dados são comuns em simulações, nas ciências exatas e em amostragens realizadas no âmbito das ciências naturais e sociais.

Na Biologia, as bases de dados reúnem informações sobre amostras. No estudo de populações animais, por exemplo, tais informações em geral incluem o local de amostragem (latitude e longitude), denominação da espécie do exemplar amostrado, dados morfológicos e, eventualmente, dados genéticos. Dada a diversidade dos dados encontrados nessa área, é interessante o estudo de técnicas de visualização, em especial, aquelas que envolvem a identificação ou extração de características relevantes e seu mapeamento para ícones.

Este trabalho tem por objetivo o estudo de técnicas de visualização baseadas em ícones e a sua experimentação para a exibição de características relevantes de uma base de dados biológicos.

No capítulo 2 serão abordados com mais detalhes os conceitos de visualização de informações, diferenças em relação à visualização científica e sua relação com a área de mineração de dados bem como os requisitos que uma boa ferramenta de visualização deve conter. Também são brevemente abordadas algumas técnicas mais conhecidas de visualização de informações, reunindo-se assim trabalhos anteriores nessa área.

O capítulo 3 destina-se a explicar com maior detalhe a técnica escolhida como alvo dessa dissertação: a visualização de informações através do uso de ícones. Serão abordados temas como o conceito de ícone, tipos de ícones e trabalhos anteriores nessa área.

A geração automática de ícones será abordada no capítulo 4. Será proposto um método para a geração automática de ícones (que será implementado no protótipo descrito no capítulo 5), bem como uma interface para o mapeamento de informações para os parâmetros dos ícones gerados.

No capítulo 5 será descrito um estudo de caso para a utilização da visualização de informações através do uso ícones. Esse estudo de caso será centrado em um conjunto de informações fornecidas por biólogos do Departamento de Genética da Universidade Federal do Rio Grande do Sul. O capítulo 5 descreve também o protótipo implementado, abordando os ícones utilizados, conexão com banco de dados e interação do usuário com a visualização gerada.

O capítulo 6 apresenta um fechamento do estudo de caso, juntamente com uma avaliação do trabalho, enquanto conclusões e extensões do trabalho estão descritas no capítulo 7.

2 Visualização de informações

2.1 Conceito

Como uma disciplina, a visualização de informações ainda está emergindo, seguindo a evolução da Computação. Uma disciplina emergente progride através de quatro estágios. Começa de forma isolada, praticada por poucos usando heurísticas. Depois, pesquisadores formulam princípios para compreender o processo. Posteriormente, engenheiros refinam esses princípios e percepções em regras de produção. Finalmente, a tecnologia se torna amplamente disponível. Para a visualização de informação, entretanto, esses estágios estão acontecendo em paralelo, isto é, ao mesmo tempo em que se observa a proliferação de aplicações nas mais diversas áreas [COX 88, MUK 94, MEI 98, HAN 98, HIG 99], são relatados trabalhos que procuram estabelecer conceitos e certo grau de formalismo no processo de geração de representações visuais de informações [HAB 90, WAT 91, CHI 98, BJO 99].

A visualização de informações tem por objetivo facilitar o entendimento das informações. Assim uma representação visual de dados e de tipos variados de informação proporciona uma forma mais simples e intuitiva de entender melhor o significado dos dados. Um exemplo ilustrativo pode ser visto na Figura 2.1. Fica muito mais fácil para uma pessoa capturar o significado de uma imagem do que vários dados isolados. Considerando-se visualização de informações como uma área se destacando da visualização científica, a tabela 2.1 procura mostrar as diferenças básicas existentes entre a visualização de informações e a visualização científica.

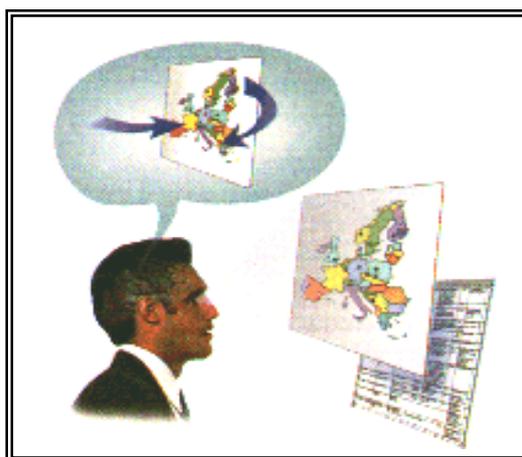


FIGURA 2.1 - Ilustração esquemática do processo de visualização [GER 97]

TABELA 2.1 - Visualização de informação comparada à visualização científica [GER 97]

	<i>Usuário</i>	<i>Tarefa</i>	<i>Entrada</i>	<i>Volume</i>
Visualização científica	Especializado, altamente técnico	Profundo entendimento dos fenômenos científicos	Dados físicos, medidas, resultados de simulação	Pequeno a volumoso
Visualização de informação	Diverso, pode ser menos técnico	Busca, descoberta de relações	Relações, dados não-físicos, informação	Pequeno a volumoso

Além do problema enfrentado por toda tecnologia emergente, a visualização de informações enfrenta outros desafios como segue.

A mídia da computação visual é bastante nova, e suas vantagens e desvantagens não são ainda bem compreendidas. Atualmente, muitos projetistas de software e usuários se relacionam com a nova mídia como se ela fosse uma réplica do papel – com o qual estamos acostumados. Essas novas tecnologias, entretanto, permitem-nos fazer coisas além das que são possíveis com o papel [GER 95], o que dificulta a compreensão dessa nova mídia. Uma preocupação mais recente é a avaliação da efetividade das técnicas de visualização, tanto do ponto de vista de percepção visual [BED 99] como de interação [PIR 00].

Usar técnicas de visualização da informações para resolver problemas do mundo real é um dos maiores desafios para a tecnologia. A maioria das pesquisas e desenvolvimentos feitos nessa área é de interesse acadêmico mais do que para a aplicabilidade. Em muitas aplicações do mundo real, visualização é somente um componente de um sistema complexo mais propriamente do que uma resposta final. É necessário entender o sistema e as necessidades dos usuários de forma a criar representações visuais efetivas, ou seja, é necessário estudar o problema primeiro e depois usar a visualização, se adequado.

Adaptar sistemas de visualização baseando-se na capacidade humana de percepção e sistemas de processamento de informações é um outro desafio. É necessário entender melhor como o ser humano interage com a informação, como ele a percebe visualmente e não-visualmente, como a mente trabalha quando está procurando por informações conhecidas e desconhecidas, como a mente resolve os problemas e como o ser humano compreende os dados apresentados.

Uma boa interação homem-computador é uma necessidade, mas não é o suficiente. Portanto, no projeto de sistemas de visualização, um desafio relacionado envolve criar interfaces flexíveis de modo a fornecer ferramentas de navegação e métodos de pesquisa apropriados para cada um dos tipos de usuários existentes, modelos de dados e tarefas.

Com o desenvolvimento difundido de aceleradores gráficos 3D, os computadores pessoais permitem manipular animações e gráficos 3D muito mais sofisticados. O desafio é como melhor explorar essa capacidade. Atualmente, a tendência é achar que 3D é mais efetivo do que 2D, mas isso nem sempre é válido. É importante saber usar essas capacidades onde elas são apropriadas.

2.2 Visualização e mineração de dados

Os conceitos de visualização de informações e de mineração de dados estão intimamente ligados. Por isso, essa seção é destinada a explicar, de uma forma resumida, as principais características de mineração de dados BJO 99,.

Tecnicamente, mineração de dados é o processo de encontrar correlações ou padrões entre dezenas de campos em grandes bases de dados [FAY 96][RAM 99]. Mineração de dados é, portanto, o processo de análise de dados a partir de diferentes perspectivas e de modo a resumí-los em informação útil. Um software de mineração de dados é uma das várias ferramentas utilizadas para analisar os dados. Ele permite aos usuários analisar os dados de vários ângulos ou dimensões diferentes, categorizá-los e resumir os relacionamentos identificados. Importante nesta área é a diferenciação entre dado, informação e conhecimento.

Dados são qualquer fato, número, ou texto que pode ser processado por um computador. Hoje, as organizações estão acumulando vastas quantidades de dados em formatos diferentes e em diferentes bases de dados. Os padrões, associações, ou relacionamentos entre todos esses dados podem prover **informações**. Por exemplo, a análise de uma transação de vendas de um ponto de varejo pode trazer uma informação sobre quais produtos são vendidos e quando.

Informação pode ser convertida em **conhecimento** sobre padrões anteriores e tendências futuras. Por exemplo, uma informação resumida sobre vendas a varejo de um supermercado pode ser analisada levando em conta esforços promocionais para prover

conhecimento do comportamento de compra dos consumidores. Assim, um fabricante ou um varejista podem determinar quais itens são mais suscetíveis para promoções.

Os grandes avanços na captura de dados, poder de processamento, transmissão de dados e capacidade de armazenamento estão possibilitando às organizações integrar suas várias bases de dados em *data warehouses* [FAY 96]. *Data warehousing* é definida como um processo de gerência e recuperação de dados centralizados. *Data warehousing* representa a visão ideal da manutenção de um repositório central de todos os dados organizacionais. A centralização dos dados é necessária para maximizar a análise e facilitar o acesso do usuário.

Enquanto que a tecnologia de informação de larga escala tem evoluído para separar transações e sistemas analíticos, a mineração de dados provê a união de ambos. Um software de mineração de dados analisa relacionamentos e padrões em dados de transações armazenadas baseadas em consultas dos usuários.

Geralmente, quatro tipos de relacionamentos são procurados :

- **Classes:** dados armazenados são usados para localizar dados em grupos pré-determinados. Por exemplo, uma cadeia de restaurantes poderia minerar dados de compra dos consumidores para determinar quando os consumidores visitaram o restaurante e o que eles tipicamente pediram. Essa informação poderia ser usada para aumentar o movimento do restaurante fazendo ofertas especiais diárias.
- **Agrupamento:** itens de dados são agrupados de acordo com relacionamentos lógicos. Por exemplo, dados podem ser minerados para identificar segmentos de mercado ou afinidades dos consumidores.
- **Associações:** dados podem ser minerados para identificar associações.
- **Padrões seqüenciais:** dados são minerados para antecipar o comportamento determinando padrões e tendências.

A construção de um sistema de mineração de dados envolve os seguintes passos:

1. Extrair, transformar e carregar dados de transações em um sistema de *data warehouse*;
2. Armazenar e gerenciar os dados em sistema de base de dados multi-dimensional;
3. Prover acesso aos dados para analistas de negócios e profissionais da tecnologia da informação;
4. Analisar os dados por um software aplicativo;
5. Apresentar os dados em um formato útil, como um gráfico ou uma tabela, por exemplo.

A apresentação direta de todos os dados deixa a extração dessas características a cargo do usuário. Mas muitas vezes, o número de características é pequeno comparado à quantidade de dados e, então, é necessário um processo algorítmico para realizar essa tarefa.

Como podemos ver nesta seção, os conceitos de mineração de dados e visualização interseccionam-se. Vemos que um dos elementos principais da mineração de dados é apresentar os dados em um formato útil, de forma que se possa extrair informações e conhecimento a partir da base de dados. Uma forma de apresentar esses dados será através da utilização de *ícones*, o que será descrito nas próximas seções.

Os aspectos mais importantes nessa técnica de visualização são a extração de características e seu mapeamento para ícones. Uma *característica* pode ser definida como uma região de interesse para interpretação em uma base de dados. As características são extraídas e representadas como um conjunto de atributos, que são uma representação abstrata dos dados originais. O próximo passo é vincular os conjuntos de atributos a objetos simbólicos denominados *ícones*.

À medida que as técnicas de mineração de dados se desenvolverem e ficarem mais maduras, será importante integrá-las com as técnicas de visualização. Existem três abordagens para isso. Uma é usar as técnicas de visualização para apresentar os resultados que são obtidos a partir da mineração de dados. Esses resultados podem ser na forma de *clusters* ou podem especificar correlações entre os dados na base de dados. Por exemplo, poderia haver uma correlação entre os preços de casas e apartamentos, assim como bairros que tenham casas de preços elevados e não tenham nenhum edifício

de apartamentos. Assim, quando uma pessoa for consultar os preços das casas em uma região em particular, a ferramenta de mineração de dados poderia também extrair as informações sobre as correlações entre os preços de casas e de apartamentos. Essa informação poderia ser apresentada em uma forma visual mostrando os preços das casas e apresentando graficamente os edifícios de apartamentos em um mapa.

Enquanto que na primeira abordagem aplicou-se a visualização à mineração de dados, a segunda aplica mineração de dados às técnicas de visualização. Assume-se nesse caso que é mais fácil aplicar as ferramentas de mineração de dados aos dados na forma visual. Assim, ao invés de aplicar as ferramentas de mineração de dados a bases de dados grandes e complexas, pode-se capturar alguns dos dados essenciais visualmente e, então, aplicar as ferramentas de mineração de dados. Existem algumas questões a serem levantadas aqui. Uma é que alguma informação pode ser perdida quando da conversão dos dados na base de dados para uma representação visual. Conseqüentemente, as ferramentas de mineração de dados podem ser aplicadas a dados incompletos. Outra questão é o desenvolvimento de ferramentas de mineração de dados apropriadas para serem aplicadas a dados visuais.

Uma terceira abordagem seria usar as técnicas de visualização para complementar as técnicas de mineração de dados. Por exemplo, as técnicas de mineração de dados poderiam ser usadas para obter as correlações entre dados ou detectar padrões. Entretanto, as técnicas de visualização ainda podem ser necessárias para obter um melhor entendimento dos dados na base de dados. A questão aqui é quais dados devem ser minerados e quais dados devem ser simplesmente visualizados.

2.3 Linhas gerais e requisitos para gerar efetivas representações visuais

Para que uma boa representação visual seja gerada é necessário seguir algumas regras que já foram consagradas como as ideais. Mas o que é uma boa representação visual? Para receber essa classificação, a representação visual deve antes de tudo ser intuitiva, ou seja, o observador ao olhar para o que foi gerado deve conseguir entender as informações mostradas sem nenhuma dificuldade.

Um primeiro passo é identificar a que classe pertencem os dados a serem visualizados. Na tabela 2.2, é feita uma classificação dos tipos de dados e informações segundo critérios de: classe da informação, tipos dos valores que podem ser assumidos, natureza do domínio onde a informação está definida e dimensão desse domínio. Esta

classificação foi elaborada considerando aplicações de visualização científica, mas pode ser adotada de forma mais geral, no contexto de visualização de informações.

TABELA 2.2 - Classificação dos dados [FRE 93]

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

Algumas linhas gerais abaixo descrevem como construir imagens eficientes em visualização científica [COX 88] as quais também podem ser aplicadas para a visualização de informações:

- Usar somente informações importantes

O primeiro passo para a visualização é a determinação de quais dados serão exibidos. A apresentação de todos os dados pode não ser possível tecnologicamente ou também pode ser mais do que o sistema visual humano pode registrar em um momento. O que deve ser feito é selecionar as variáveis fundamentais para a descrição do problema, e descrever somente essas variáveis.

- Usar forma para quantidades vetoriais

A forma, o elemento básico dos ícones e glifos, geralmente representa quantidades vetoriais. Linhas convergentes, setas e gráficos de vetores, denotam informação direcional fazendo com que o olho siga para o ponto de intersecção.

- Usar cor para valores escalares

Para valores escalares não associados com uma magnitude vetorial, a sua representação deve ser feita através do uso de cores. Valores escalares são mapeados para uma cor que representa uma faixa de valores numéricos para aquele escalar. Pintando as formas poligonais com essas cores coloca-se mais variáveis na visualização sem ambiguidade de significado.

Mapas de cores tem seguido tradicionalmente a seqüência espectral de violeta, azul, verde, amarelo, laranja e vermelho. Embora a seqüência espectral de cores seja bastante conhecida e dominada pela maioria dos cientistas, como um mapa de cores, o espectro tem desvantagens. Para algumas pessoas, o amarelo tende a atrair mais o olho do que o vermelho e, perceptualmente, assume um valor mais alto apesar do vermelho estar na topo do mapa de cores significando que ele, vermelho, é utilizado para representar os valores mais elevados. Outro ponto fraco na utilização de um mapa espectral vem do fato de que as intensidades das cores são uniformes. O mapa de cores espectral varia apenas em cromo, ignorando o brilho.

A escolha do mapa de cores deve ser feita cuidadosamente de forma que as graduações nas cores tenham correlação com as mudanças nos dados. Cores distribuídas pobremente ou cores não distinguíveis perceptualmente podem obscurecer a informação na visualização. A presença de mais de uma escala de cor em uma imagem (por exemplo, para representar dois escalares diferentes como a pressão e a temperatura) não deve conduzir a ambiguidades. As escalas de cores devem ser mutuamente exclusivas na aparência perceptual. A repetição da mesma cor em duas escalas confunde o observador. Uma abordagem mais ampla e aprofundada sobre o uso de cores em representações visuais e o mapeamento de informações para cores foi realizada por Fortner e Meyer [FOR 97].

- Usar animação para evoluções no tempo

Já que o tempo é percebido como uma evolução linear, a animação provê um excelente método para visualizar simulações ou informações que são tempo-dependentes. A representação visual se torna dinâmica. Geralmente, o tempo para produzir as imagens excede bastante a velocidade ótima de visualização e, portanto, as seqüências de imagens são armazenadas como quadro individuais na memória ou em algum meio de armazenamento, para depois serem reproduzidas.

- Usar técnicas de computação gráfica para representações binárias

Existem outras técnicas da computação gráfica que são úteis para a representação de dados. Por exemplo, transparência, mapeamento de textura, modelos

de sombreamento e iluminação provêem possibilidades de distinguir mudanças de limiares globais na visualização científica. Por exemplo, um limiar nos dados pode ser apresentado permitindo que valores de dados controlem a condição binária de usar *Gouraud* ou *flat shading* em um objeto geométrico. A diferença entre esses modelos de sombreamento provê uma diferença visual em como o observador percebe o objeto.

Finalmente, a escolha de uma representação visual deve ser guiada pelas tarefas que o usuário deve ou quer desempenhar através da mesma. Em função do tipo de informação e do tipo de tarefa, pode-se estabelecer diretrizes para a escolha da representação visual mais adequada [FRE 93]. Tais diretrizes estão resumidas na tabela 2.3.

TABELA 2.3 - Classificação de representações visuais [FRE 93]

Representação visual	Informação que proporciona
<i>Ícone</i>	Identificação do objeto em um contexto
<i>Gráficos e tabelas</i>	Propriedades estáticas dos objetos Evolução das propriedades no tempo
<i>Mapas e diagramas/redes</i>	Estrutura de objetos complexos Propriedades estáticas dos objetos Estado do comportamento dos objetos
<i>Imagens</i>	Modelo geométrico de um objeto Propriedades estáticas dos objetos Propriedades estáticas mapeadas em um modelo geométrico Estado do comportamento de um objeto
<i>Seqüência de mapas, diagramas, redes ou imagens</i>	Comportamento de um objeto

A escolha de uma representação visual deve ser resultado da aplicação de uma metodologia que leve em conta as tarefas de análise de dados que o usuário vai realizar com base na representação visual [FRE 94]. Apesar de na prática serem utilizadas regras informais como as mencionadas anteriormente, freqüentemente, são relatados trabalhos propondo metodologias para maximizar a efetividade das representações visuais [KER 99].

O desenvolvimento de uma ferramenta de visualização deve levar em consideração um conjunto de requisitos além das linhas gerais sugeridas anteriormente. Mukerjea e Stasko [MUK 95] comentam que a ferramenta deve ser o mais completa possível, contemplando:

1. *Manipulação de dados*: tarefas de manipulação de dados envolvem a seleção de porções de dados ou sua transformação em novas formas mais simples. Essas tarefas podem ser utilizadas para reduzir a complexidade do espaço de informação. Já que a visualização e a compreensão de um espaço de informação grande e complexo é difícil, os módulos de manipulação de dados necessitam ser integrados com a visualização de dados. Exemplos de tarefas de manipulação de dados incluem a transformação de uma estrutura de dados complicada, como um grafo, em uma forma mais simples como uma árvore, e consultas dinâmicas, as quais permitem a seleção de somente uma porção dos dados, resultando assim menos informações a serem visualizadas.

2. *Análise de dados*: tarefas de análise de dados envolvem a obtenção de estatísticas sobre porções dos dados, que proporcionem uma boa percepção do espaço de informação. Essa análise é útil não somente para dados numéricos mas para informações mais abstratas também. Por exemplo, o usuário pode querer achar o valor médio de um atributo numérico ou o valor modal de um atributo enumerado para obter um melhor entendimento do espaço de informação.

3. *Visualização dos dados*: tarefas de visualização dos dados envolvem ver os dados através de representações visuais apropriadas. Além dos diagramas de barra e gráficos tradicionais, métodos de visualização alternativos devem ser desenvolvidos.

Portanto, uma ferramenta de visualização deve permitir aos usuários executar várias manipulações de dados e tarefas de análise, além de diferentes tipos de visualização das informações. A integração dessas diferentes funcionalidades deve permitir que, por exemplo, depois de fazer uma análise sobre os dados, o número de informações a serem visualizadas seja reduzido. Assim uma tarefa seria a de filtragem de dados e outra tarefa seria a visualização propriamente dita. Essas tarefas devem interagir uma com a outra.

Outros requisitos importantes são:

1. *Combinar análise estrutural e de conteúdo para manipulação de dados*: o espaço de informação tem uma estrutura de nodos unidos por links. Os nodos e os links também têm conteúdo. Quando os módulos de manipulação de dados tentam reduzir a complexidade do espaço de informação eles devem tentar analisar tanto a estrutura quanto o conteúdo do espaço de informação. Isso fará com que os módulos sejam mais poderosos. Por exemplo, um módulo de consulta deve permitir ao usuário não somente consultar o conteúdo dos nodos e links como também permitir a consulta

sobre a estrutura; por exemplo, o usuário deveria ser capaz de achar todos os nodos de um certo tipo unidos por outros tipos de link.

2. *Integrar consulta e navegação*: existem duas maneiras de acessar uma informação: consultando ou navegando/procurando. Procurar é realizado pela navegação através do espaço de informação. Essa forma é muito apropriada para usuários novos e inexperientes, os quais não possuem muito entendimento do espaço de informação e assim não podem formular uma consulta. Entretanto, se o usuário conhece exatamente qual a informação que é necessária, recuperar essa informação através da navegação irá consumir muito tempo. Dessa forma a consulta será bastante apropriada neste caso. Entretanto, a consulta sofre do fenômeno “tudo ou nada”. Realizando uma consulta geral obtém-se muita quantidade de informação. Nesse caso, o usuário deve ser apto a navegar através das informações retornadas pela consulta. Fazendo uma consulta mais especializada obtém-se muito pouca informação e o usuário deve ser apto a procurar através de outras informações relacionadas. Depois de um melhor entendimento sobre o espaço de informação, uma consulta mais apropriada pode ser formulada. Esse método de especificar consultas mais apropriadas sucessivamente, conhecido como *consultando através de refinamentos sucessivos*, é uma estratégia efetiva. De fato, foi sugerido [WAT 91] que unir consulta e navegação irá prover sistemas poderosos de exploração de informações. Assim, ferramentas de visualização de informação devem permitir ao usuário combinar navegação com consulta.

3. *Permitir o controle de usuários (experientes) sobre as propriedades visuais*: para ser efetiva, as representações visuais devem usar propriedades visuais para representar as informações. Essa estratégia tem sido utilizada efetivamente nos sistemas de visualização científica e de software como o *Glyphmaker* [RIB 94] e *Lens* [MUK 94]. A quantidade de informação geralmente é muito maior do que a quantidade de propriedades visuais que podem ser usadas. Já que o projetista pode não saber quais informações serão úteis para os usuários, ele não deve fixar as ligações entre as informações e as propriedades visuais *a-priori*. Os usuários devem ser aptos a controlar as ligações dinamicamente e adequar as representações visuais de acordo com suas necessidades.

4. *Visualizações simultâneas*: produzir uma visualização efetiva de espaços de informação grandes e complicados que seja compreensível para o usuário é difícil. Diferentes representações visuais do mesmo espaço de informação, dando diferentes perspectivas, podem ser mais úteis. Já que essas vistas destacam somente certos aspectos do espaço, elas devem ser fáceis de serem produzidas e mais compreensíveis. Neuwirth et al. [NEU 87] também observou que a habilidade para ver o conhecimento de diferentes perspectivas é importante. Múltiplas representações visuais foram usadas

efetivamente em vários sistemas de visualização de informação, como IVEE [AHL 95] e Harmony [AND 95].

As ações do usuário em uma vista devem resultar em modificações nas outras vistas. Por exemplo, se o usuário selecionou um nodo de um espaço de informação em uma vista, as demais vistas devem também destacar o nodo selecionado. Isso irá permitir que o usuário compreenda a posição de qualquer nodo particular em diferentes perspectivas. Para melhores efeitos, animação suave deve ser usada; assim, as mudanças nas vistas não serão abruptas e o usuário poderá compreendê-las.

5. *Prover detalhes de contexto*: geralmente, o espaço de informação que é visualizado é muito grande em relação ao espaço na tela do computador. Se o tamanho é reduzido para caber na tela, os detalhes se tornam muito pequenos para serem vistos. Uma alternativa seria usar barras de rolagem para visualizar. Entretanto, isso tende a obscurecer a estrutura global. O objetivo deveria ser mostrar ambos os detalhes e o contexto integrados num único espaço. Provendo detalhes com o contexto tem sido reconhecido como sendo um dos maiores requisitos para uma visualização de informação efetiva. Por exemplo, “*perspective wall*” [MAC 91] e “*table lens*” [RAO 94] usam técnicas “*fish-eye*” [FUR 86] de forma que a informação que o usuário focaliza é mostrada em detalhes, suavemente integrada com o contexto da informação.

6. *Facilidade de uso*: assim como ferramentas de interface, um dos maiores requisitos de ferramentas de visualização deve ser a facilidade de uso. Dependendo dos objetivos do usuário e do espaço de informação, diferentes combinações de operações de exploração de dados serão necessárias. O usuário deve ser apto a experimentar várias opções para decidir qual é a melhor visualização que se adequa para seus interesses. Conseqüentemente, se é realmente difícil ou se consome muito tempo para experimentar diferentes operações de exploração de dados, a ferramenta vai ter seu uso limitado. Assim, facilidade de uso da ferramenta é muito importante.

7. *Salvamento do estado*: a ferramenta de visualização de informações deve manter o *estado corrente* do sistema. Tarefas de exploração de dados irão conduzir a mudanças de estados. Deve ser possível salvar o estado corrente. Isso irá permitir aos projetistas salvar representações visuais úteis assim como manipulação de dados úteis e resultados de análise de dados para os usuários finais. Mais ainda, os resultados de uma sessão podem ser usados em seção posterior e os usuários não são precisar partir do início em cada nova sessão.

8. *Extensibilidade*: não é possível prever todas as explorações de dados necessárias para o usuário. Assim, mesmo que vários módulos de exploração de dados

sejam oferecidos, para alguma tarefa imprevista alguns outros módulos podem vir a ser necessários. Por exemplo, uma tarefa de exploração de dados pode requerer um novo modo de visualização de dados ou uma nova operação de manipulação de dados. Se a ferramenta não puder ser utilizada para essas tarefas, ela vai ter seu uso limitado. Por isso, a ferramenta deve ser extensível. Ela deve permitir a adição de código apropriado para realizar as funções requeridas. Observa-se que uma das maiores razões para a popularização das ferramentas de visualização científica como AVS[UPS 89] e Iris Explorer [SIL 95] é que elas permitem a adição de novos módulos além de prover vários outros para muitas tarefas comuns de visualização científica.

9. *Generalidade com respeito ao tipo de informação*: um sistema de gerenciamento de banco de dados permite ao usuário facilmente construir bancos de dados para diferentes aplicações. Similarmente, seria desejável que a ferramenta pudesse ser feita para trabalhar com diferentes espaços de informações com um mínimo esforço. Se for preciso muito esforço para adequar a ferramenta para diferentes espaços de informações, o usuário vai ficar desencorajado para usá-la para outro sistema e a utilidade da ferramenta ficará reduzida.

2.4 Técnicas de visualização de informações

As técnicas mencionadas nesta seção são basicamente utilizadas para representação de informações hierárquicas como documentos, diretórios e *sites*. Embora não estejam diretamente ligadas ao tópico central deste trabalho, elas são apresentadas como forma de situar o contexto de visualização de informações. A técnica de visualização icônica está detalhada em um capítulo a seguir.

A técnica de visualização *fish-eye view* [FUR 86] é utilizada como método de aproximação provendo detalhes locais e contexto global (ou estrutura) simultaneamente. A técnica de *fish-eye view* vence o problema comum de aproximação (*zooming*), o qual causa a perda do contexto quando uma figura está sendo aumentada. Isso é obtido pela ênfase da região de interesse e desênfase dos detalhes remanescentes. Essa técnica de visualização é útil quando a pessoa quer uma visão mais detalhada de uma determinada região mas não deseja perder o contexto da visualização dos arredores dessa região de interesse.

Dada uma região de interesse, uma visualização típica usando a técnica *fish-eye view* tem a taxa de ampliação mais alta no centro da região, e essa taxa vai decrescendo proporcionalmente à distância a partir do centro da região de interesse. Áreas que estão

fora da região de interesse podem ser alteradas uniformemente de modo que a região não obscureça os detalhes da vizinhança.

Variações dessa técnica incluem suporte para múltiplas regiões de interesse, diferentes técnicas para redução do tamanho dos objetos, e métodos para unir as fronteiras entre a região de interesse e as demais regiões da imagem. Quando a técnica de *fish-eye view* é utilizada para visualizar estruturas gráficas, a região de interesse pode ser trocada por uma função de grau de interesse. Esta função define um valor de “importância” para cada nodo da estrutura, e os nodos são então mostrados baseados na sua relação a um limiar de importância.

A técnica de visualização *fish-eye views* tem sido utilizada em várias áreas para a visualização de informações [GRE 95]. Furnas [FUR 86] criou sistemas para visualizar e filtrar código de programa estruturado, taxonomia biológicas e calendários. Egan, Remde, Landauer [GRE 95] usaram um tipo de *fish-eye view* no *Superbook*, um livro eletrônico baseado em texto, para prover a noção familiar de uma tabela de conteúdo (índice) expansível. Sarkar e Brown [SAR 94] implementaram uma representação *fish-eye* para visualizar os nodos de redes de computadores assim como cidades em um mapa.

Vários métodos estão disponíveis para implementação dos efeitos de lente tipo olho de peixe através de transformações geométricas simples. Geometria hiperbólica e projeções perspectivas de superfícies curvas 3D são exemplos de uma classe de transformações não-lineares que podem simular tal visão. Uma discussão detalhada sobre esse tópico foi desenvolvida por Kealy e Robertson [KEA 96].

A técnica de visualização ***Information Cube*** [REK 93] é uma técnica aplicada em três dimensões. Essa técnica faz uso de cubos aninhados e translúcidos para representar informação hierárquica. O cubo que está posicionado mais “para fora” corresponde ao nível superior da hierarquia, e os cubos que estão dentro desse cubo representam o próximo nível de dados abaixo do topo. Cada cubo de segundo nível é composto de cubos do terceiro nível da hierarquia e assim sucessivamente. Os cubos possuem texturas descritivas em suas superfícies, enquanto que os dados terminais (representando o nível mais baixo da hierarquia) são representados por rótulos.

Na sua proposição original [REK 93], os cubos são mostrados em três dimensões. Eles podem ser visualizados diretamente no monitor de vídeo ou através de *head-mounted displays (HMD)*. O usuário tem a liberdade de rotacionar e mover os cubos de forma a obter a melhor vista para poder visualizar e analisar as informações.

Depois de feita a seleção de um dos cubos, o usuário tem a possibilidade de navegar até o cubo selecionado para poder visualizar maiores detalhes sobre o mesmo.

A forma de apresentação aplicada aos cubos é semi-transparente, permitindo assim, que o usuário possa visualizar o interior dos cubos próximos sem ser atrapalhado por muitos detalhes. Pela variação do nível de transparência, o sistema pode controlar a complexidade da visualização. A estrutura hierárquica dos cubos é facilmente distingüível, e as superfícies semi-transparentes separam claramente os diferentes níveis de informação. Os autores comentam que a representação através de cubos aninhados é intuitiva pois nós somos familiares com o conceito de uma caixa como sendo um *container*.

A técnica de visualização *Perspective Wall* [MAC 91] foi criada como parte de um protótipo de sistema de visualização desenvolvido no Xerox PARC (Palo Alto Research Center). Esse protótipo ficou conhecido como *Information Visualizer*. A *perspective wall* é uma metáfora para a visualização 3D de dados abstratos lineares. Essa técnica integra o detalhe e o contexto de uma estrutura linear em uma única visualização, um conceito bastante similar ao da técnica *fish-eye view*. A diferença está na forma em que o *zooming* é aplicado.

Nessa técnica, a área exibida é representada como um retângulo horizontal longo. A idéia é desenhar a informação no retângulo e, então, dobrar o retângulo para trás nos limites à direita e à esquerda (definido pela área de interesse). Uma perspectiva 3D do resultado é gerada. A quantidade visual do contexto das informações é controlada pelo ajuste do ângulo das dobras. Animação interativa também é provida para ajudar a explorar as mudanças dinâmicas das vistas da estrutura de informação. Na Figura 2.2 arquivos de documentos são visualizados usando essa técnica. Os documentos estão organizados por autor no eixo *Y* e cronologicamente no eixo *X*.

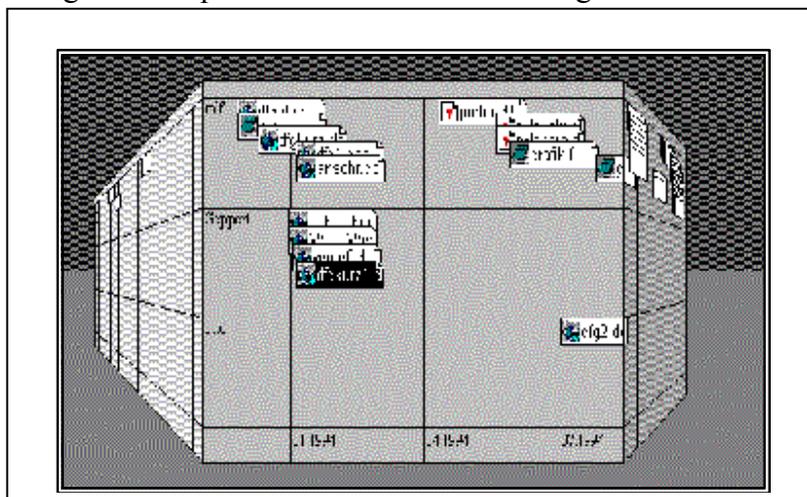


FIGURA 2.2 - Técnica de *Perspective Wall* [MAK 91]

Quando um usuário seleciona um objeto da “parede”, o ítem é movido para o centro da parede usando animação. O usuário tem a liberdade de ajustar a relação *contexto X detalhe* conforme a sua necessidade.

O conceito de *perspective wall* é bastante similar à técnica *fish-eye view* vista anteriormente: a criação de uma visualização que provê uma vista detalhada de uma região de interesse enquanto o contexto global das estruturas adjacentes é mantido. A principal diferença é que a *perspective wall* aplica um aumento uniforme na região retangular de interesse, enquanto que a técnica de *fish-eye* utiliza uma taxa variável de aumento em uma região circular de interesse. A técnica *perspective wall* com essas características é bastante adequada para a visualização de informação textual. Todo o texto dentro de uma região de interesse é ampliado na mesma proporção, fazendo com que este texto seja legível na tela como um texto comum. A técnica de *fish-eye* não é adequada para leitura de texto, já que a taxa variável de ampliação faz com que o texto se torne ilegível à medida que a distância do centro da região de interesse aumenta. Outro aspecto a ser mencionado é a forma circular da *fish-eye view*, conflitante com a forma retangular adequada para leitura de textos, provida pela *perspective wall*.

2.5 Visualização e banco de dados

Nesta seção são resumidas informações sobre sistemas que estão centrados em criar representações visuais de dados obtidos a partir de consultas feitas a bancos de dados. Tendo em vista que a proposta aqui apresentada diz respeito à visualização de dados mantidos em banco de dados, esta seção fornece um referencial para o protótipo apresentado posteriormente.

2.5.1 IVEE (*Information Visualization and Exploration Environment*)

O IVEE [AHL 95] é um sistema para criação automática de aplicações de consultas dinâmicas. O objetivo do IVEE é demonstrar como um sistema de visualização pode ser projetado para automatizar a tarefa da criação de ambas visualização e manipulação de objetos (chamados de “*sliders*”), permitindo aos usuários se concentrarem diretamente na tarefa exploratória. O IVEE requer que os usuários especifiquem relacionamentos (por exemplo, oportunidades de emprego, preços de casas, índices de criminalidade) e atributos que participam desses relacionamentos (por exemplo, ano, estado, população) de uma maneira direta usando

formato texto. Os relacionamentos podem ser originados de um sistema de banco de dados ou planilha eletrônica.

A interface do usuário do IVEE (fig. 2.3) tem dois componentes principais, uma área de consulta contendo uma quantidade de dispositivos de consultas, e uma área de visualização contendo uma quantidade de representações visuais. Uma característica importante do IVEE é que ele pode manter várias instâncias de representações visuais, banco de dados (especificações de relacionamentos) e áreas de consultas.

A abordagem seguida no IVEE para representações visuais é flexível em consideração às técnicas de visualização e aos elementos gráficos utilizados para representar os objetos em um relacionamento da base de dados. Ele provê uma coleção rica de elementos gráficos e representações visuais que os usuários podem escolher. Uma configuração padrão é definida para ambos os elementos gráficos e representações visuais de forma a fazer com que o sistema seja facilmente iniciado e, assim, os usuários não são forçados a especificar nada sobre a visualização ou elementos gráficos para que o sistema seja iniciado. Um único elemento gráfico pode ser especificado para representar cada objeto de dado em um relacionamento. Usuários também podem especificar um elemento gráfico para cada objeto na base de dados. Também é possível especificar um pequeno número de objetos para representar objetos na base de dados. Nesse último caso, objetos e seus mapeamentos correspondentes para elementos da base de dados devem ser especificados. O padrão utilizado no IVEE é um quadrado, que é utilizado para representar cada objeto da base de dados, mas os usuários também podem especificar um único ponto, glifo, ou um polígono definido em duas ou três dimensões. O uso de glifos é limitado ao conjunto de glifos pré-definidos na biblioteca que o IVEE provê. Objetos gráficos podem ser codificados por cor, forma, brilho, etc.

A visualização básica do IVEE é o *starfield*. Um *starfield* é similar a um diagrama de dispersão mas tem características adicionais para *panning*, *zooming*, detalhamento, rotações, etc. O IVEE estendeu a visualização *starfield* permitindo que os usuários especifiquem objetos de fundo que são adicionados ao contexto. Por exemplo, representações visuais geográficas podem ser criadas pela especificação de um mapa como fundo (figura 2.3). Além de mapas, outros objetos mais estruturados podem ser especificados. Outras representações visuais também estão disponíveis: *cone-tree*¹, *tree-maps*², e representações visuais usando objetos tri-dimensionais complexos.

A área do dispositivo de consulta pode conter três dispositivos lógicos diferentes: 1) *rangesliders*, usados para selecionar uma faixa de valores para atributos

¹ *Cone Tree* é uma representação tridimensional de informação hierárquica usada para representar hierarquias tais como *web sites* e diretório de arquivos.

² *Tree Maps* é uma técnica utilizada para representação de informação hierárquica.

inteiros ou outros atributos com relação de ordem; 2) *alphasliders*, usados para seleção de itens (geralmente não-numéricos ou atributos de categoria) de uma longa lista de *strings*; e 3) *toggles*, usados quando somente poucas alternativas existem para um atributo. Os componentes de consulta suportam uma combinação conjuntiva. O efeito combinado de todos os dispositivos de consulta (conjunção) afeta toda a visualização.

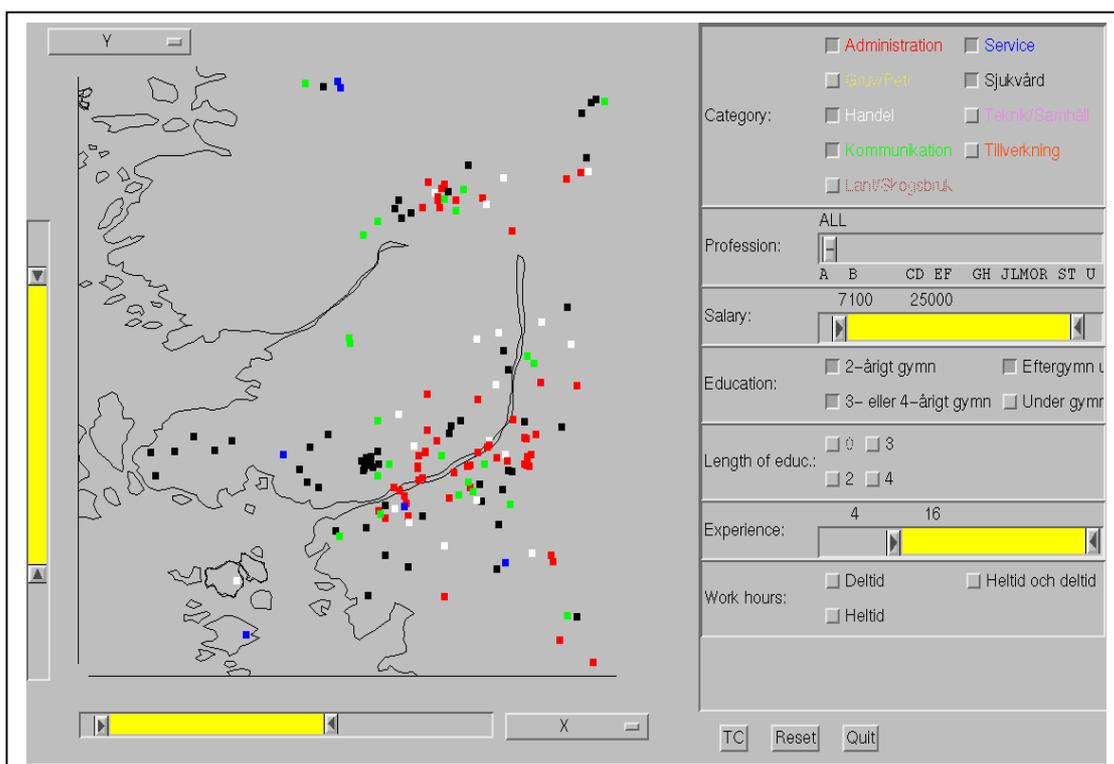


FIGURA 2.3 - Visualização gerada pelo IVEE [AHL 95]

Uma abordagem interessante usada para implementar o mecanismo de dispositivos de consulta no IVEE é que os dispositivos de consulta são fortemente acoplados, o que significa que a manipulação de um dispositivo de consulta não afeta somente toda a visualização, mas também os outros dispositivos de consulta. O mecanismo força todos os dispositivos de consulta a incluir nas suas faixas respectivas somente critérios existentes dentro dos elementos restantes da base de dados.

2.5.2 VISAGE

VISAGE [ROT 96] é um sistema para exploração de informações cuja interface com o usuário é baseada em um modelo que os autores chamaram de paradigma centrado na informação. Objetos de informação, representados como objetos de interface, podem ser manipulados usando um conjunto de operações básicas aplicadas através do ambiente. Operações tais como *drill-down* e *roll-up*, movimentos *drag-and-*

drop, cópia e mudança de escala dinâmica podem ser aplicados aos objetos de dados se esses objetos aparecem em uma tabela hierárquica, um mapa, um *slide show*, uma consulta, em outra interface de aplicação do usuário, ou mesmo no ambiente da área de trabalho.

O VISAGE é construído sobre o SAGE [ROT 94] que é um sistema baseado em conhecimento para automatizar o *design* de representações visuais. Mais recentemente, VISAGE foi empregado para visualização de elementos da WWW, como um browser [HIG 99].

O VISAGE renderiza *designs* gráficos criados pelo SAGE, e cada elemento nesses *designs* é sujeito às operações de manipulação direta mencionadas anteriormente. VISAGE também inclui várias operações de exploração que são relacionadas a consultas. Por exemplo, é possível agregar objetos da base de dados em um novo objeto que irá conter as propriedades derivadas desses elementos. Essas operações são especificadas proceduralmente, usando uma seqüência de operações de manipulação direta.

VISAGE foi estendido com um linguagem de consulta visual - VQE [DER 97] - a qual permite aos usuários expressar consultas mais complicadas do que aquelas possíveis usando a abordagem procedural. Essa extensão do VISAGE com o VQE permite ao usuário criar representações visuais a partir de consultas e consultas a partir de visualizações. A integração do VQE no VISAGE habilita a representação de consultas de uma forma explícita. Usuários podem construir consultas complexas e reusá-las posteriormente, uma capacidade não possível usando a abordagem procedural. As consultas também podem expressar relacionamentos entre conjuntos de objetos, suportando navegação entre eles, e definir agregações. Uma interface de consultas dinâmicas é usada para seleção de faixas, e todos esses componentes são totalmente integrados no VISAGE e, portanto, podem ser submetidas todas as operações básicas disponíveis no VISAGE. Por exemplo, o resultado de uma consulta representada visualmente pode ser arrastado para qualquer visualização ou dispositivo de consulta. Visualizações e dispositivos de consulta no VISAGE são ligados; portanto, mudanças em qualquer um desses componentes são imediatamente refletidas nos demais.

2.5.3 EXBASE

EXBASE [LEE 94] é um sistema onde visualização e base de dados são integrados, permitindo uma exploração visual da primeira. O EXBASE usa um modelo de dados baseado no conceito de visão. Representações visuais e base de dados podem

ser consultadas. A consulta em uma base de dados refere-se à seleção de um conjunto de dados em particular, o qual constitui uma visão local da base de dados. A consulta em uma visualização refere-se a obter um subconjunto da visão local da base de dados. Isso constitui o que é chamado de visão da visualização. EXBASE oferece a capacidade de seleção de dados tanto através do uso de uma interface textual, baseada em SQL, como através de ferramentas gráficas, no nível da visualização.

EXBASE integra um sistema de gerenciamento de banco de dados com várias técnicas de visualização e usa uma abordagem orientada a objetos para prover suporte necessário para integrar outras técnicas de visualização. Cada técnica de visualização é modelada como uma classe em uma hierarquia de abstração. Essa hierarquia inclui a classe abstrata básica *Visualization*, a qual contém o código comum da interface gráfica com o usuário para todas as subclasses concretas. O EXBASE inclui também um classe abstrata básica para a visualização baseada, a *CartesianVis*, que provê suporte para a seleção de dados espaciais na visualização. No caso particular do uso de técnicas usando o sistema de coordenadas cartesianas, subclasses concretas para cada técnica de visualização necessitam somente especificar os controles de *display*, mapeamento de funções e funções de rendering.

Um aspecto interessante do EXBASE é que ele pode monitorar todas as interações do usuário com a base de dados e a visualização. Pela representação explícita das interações dos usuários com a base de dados e as representações visuais, o EXBASE é capaz de manter um histórico de tais interações de forma a reusá-las imediatamente ou para chamá-las no futuro para análises.

2.5.4 Glyphmaker

O *Glyphmaker* [RIB 94] é uma ferramenta exploratória desenvolvida no *Georgia Institute of Technology*, que fornece um ambiente flexível no qual mesmo usuários não experientes podem criar representações visuais adequadas ao usuário usando suas próprias representações gráficas, os glifos (figura 2.4). Três componentes principais provêm a funcionalidade requerida para esse propósito: o módulo de leitura, o módulo editor de glifos, e o módulo de associação dos glifos.

O módulo de leitura é capaz de ler uma variedade de formatos de dados de entrada e convertê-los para o formato requerido para os outros módulos. O *Glyphmaker* usa uma estrutura de dados própria, com metadados de forma a suportar diferentes formatos de entrada. Essa estrutura têm dois componentes um cabeçalho, o qual descreve os dados e seu formato, e os dados propriamente ditos. Essa estrutura simples

permite aos usuários especificar seus próprios formatos de uma maneira flexível. Por exemplo, cabeçalhos e dados podem ser colocados em qualquer parte do arquivo. Mais de um cabeçalho e seus dados correspondentes podem ser colocados no mesmo arquivo.

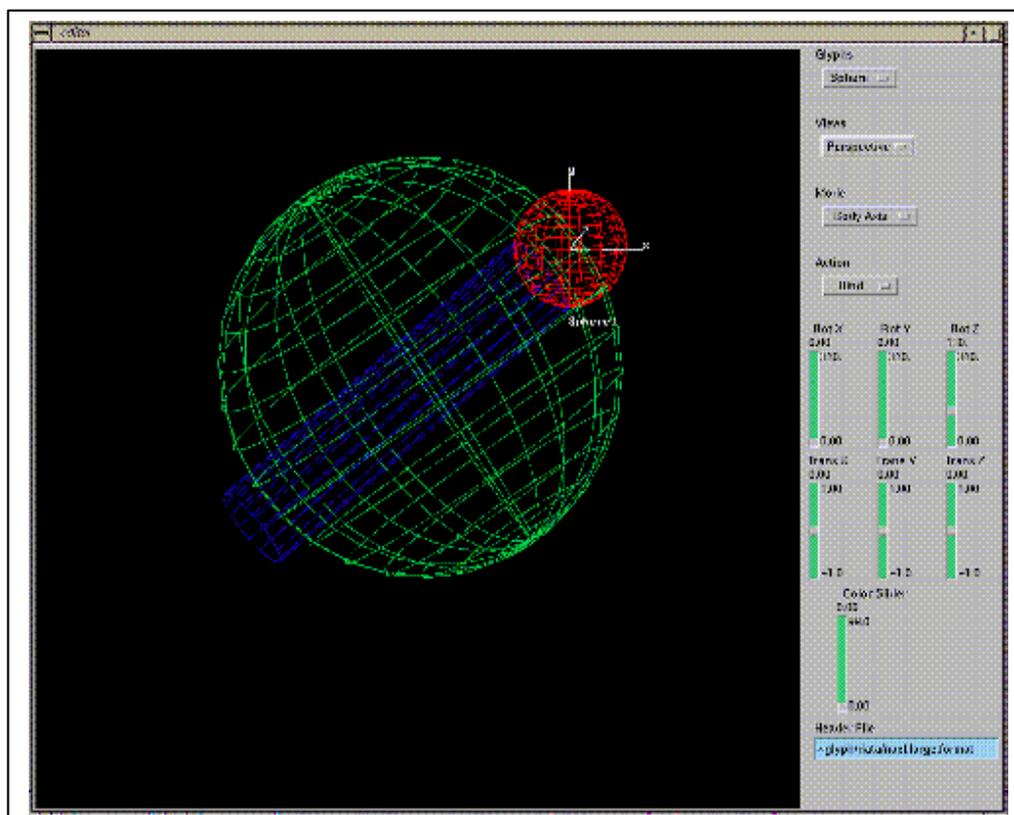


FIGURA 2.4 - Tela do *Glyphmaker* [RIB 94]

Usuários do *Glyphmaker* podem usar uma variedade de elementos gráficos para representar os dados. Usuários podem selecionar elementos gráficos de um conjunto de elementos gráficos pré-definidos ou também podem desenvolver seus próprios elementos gráficos através da utilização do módulo editor de glifos. O módulo editor de glifos provê uma biblioteca de glifos básicos (pontos, linhas, esferas, cubóides, cilindros, cones, e flechas) que podem ser usados em combinação para desenvolver ainda outras variações. O editor de glifos provê um ambiente tri-dimensional no qual usuários não somente podem desenvolver novos glifos, mas também ver como eles irão ficar na visualização final. Ferramentas para manipular esses glifos no espaço tri-dimensional (seleção de diferentes vistas, incluindo perspectiva, visão superior, lateral, frontal e todas vistas simultaneamente), e comandos para agrupamento, desagrupamento, carregar e salvar glifos também estão disponíveis.

Os atributos visuais dos glifos são associados aos atributos de dados. O módulo de associação é uma interface que permite mapeamento interativo e dinâmico entre

atributos dos elementos gráficos e os atributos dos dados. O módulo de associação é mais do que uma interface simples para permitir associação dinâmica. Ele também provê o mecanismo para restrição da quantidade de dados a ser visualizado. Cada atributo tem um valor mínimo e máximo que é inicialmente atribuído com os valores especificados no cabeçalho, o qual basicamente significa selecionar todo um conjunto de registros. Usuários podem mudar esses valores (mínimo e máximo) para um atributo particular de forma a selecionar um subconjunto de registros de acordo com o atributo. Vários atributos podem ser usados para definir o subconjunto de valores para serem visualizados. O subconjunto é determinado pela conjunção das faixas de todos os atributos ativos.

O *Glyphmaker* também inclui o que é chamado de “*conditional box*”. Usando o *conditional box*, usuários são capazes de focar em regiões específicas dos dados sendo visualizados.

3 Visualização icônica

3.1 Conceitos básicos

Um *ícone* (ou *glifo*) é um objeto com geometria e aparência paramétricas, as quais podem ser arbitrariamente vinculadas a dados. A função de um ícone é agir como uma representação simbólica que mostra as características essenciais de um domínio de dados ao qual o ícone se refere. Assim é possível obter uma visualização dos dados de uma forma mais clara e compacta. Em geral, ícones são utilizados para representar dados multivariados, ou seja, múltiplos atributos associados a uma posição num espaço qualquer, ou a entidades em estudo.

Existem alguns aspectos e características dos ícones que são importantes e, por isso, serão abordados a seguir [WAL 96].

Os *parâmetros (graus de liberdade)* podem ser atribuídos ao ícone e separadamente associados aos dados. Os parâmetros podem ser divididos em três grupos: *espaciais*, os quais dizem respeito à posição e orientação do ícone; *geométricos*, os quais controlam a forma do ícone; e *descritivos* os quais dizem respeito à cor, textura, transparência, som, etc.

Quanto à forma, existem basicamente dois *tipos de ícones*: o ícone de *modelo-fixo* e o de *modelo-amorfo*. O ícone *modelo-fixo* tem a forma básica pré-definida. Possui um conjunto de parâmetros de definição que são utilizados para variar a forma básica. Por exemplo, um ícone “seta” tem sempre a mesma forma básica, podendo variar o comprimento, a posição e a direção que são os parâmetros de definição desse ícone. A forma básica de um ícone de modelo fixo precisa ser projetada. O segundo tipo de ícone é o de *modelo amorfo* onde a forma não é pré-definida, mas é totalmente determinada pelas propriedades locais de um campo de dados.

O *domínio de referência* de um ícone é a área que é representada por um único ícone. Os domínios de referência de um ícone podem ser de três tipos: *local*, que significa um ponto e seu ambiente imediato; *global* que significa todo o domínio; *intermediário* que é o nível de um subvolume, em uma escala entre o local e o global, o qual pode ser aplicada a características ou seleções [WAL 96].

A *dimensão* de um ícone depende da dimensão do espaço-objeto no qual ele existe. Tempo é considerado como sendo uma dimensão em separado. Variações tempo-dependentes são mostradas como animações, deformações, ou mudanças de cor de um ícone.

A *escala* é o tamanho e a distinção de um ícone. Micro-ícones são discretos mas não são objetos individuais. Eles só podem existir em grandes quantidades, como elementos de pontos ou padrões de linhas ou como elementos de textura. Na macro-escala, ícones são objetos individuais representando atributos de dados de um determinado domínio de referência.

Haber et al. [HAB 90] dividem o processo de visualização usando ícones em:

1. *melhoramento e realce dos dados*: os dados são pré-processados para serem mais satisfatórios para visualização
2. *mapeamento de visualização*: as quantidades de dados são mapeados para quantidades visuais
3. *exibição*: uma imagem dessas primitivas visuais é gerada e exibida na tela de acordo com o mapeamento anterior

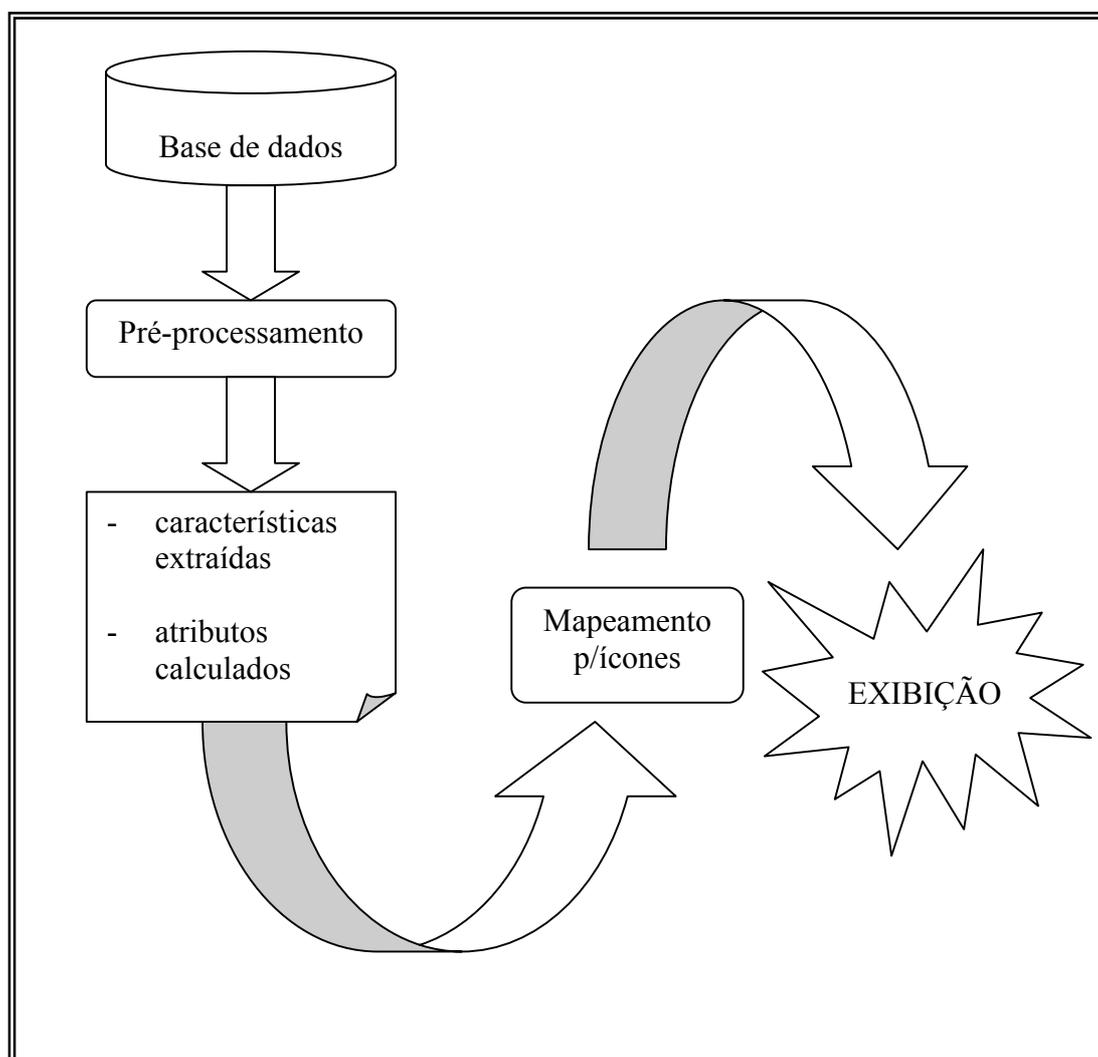


FIGURA 3.1 - Estágios do processo de visualização usando ícones

O objetivo da representação através de ícones é obter uma visualização “resumida”. Assim, o primeiro passo é encontrar regiões de interesse para interpretação (*características*) na base de dados. Simultaneamente, ou no próximo passo, valores para os parâmetros de características (*atributos*) são calculados. Isso é chamado de *extração de características* e *cálculo de atributos*. Como resultado, tem-se um conjunto de valores de atributos que definem aquela característica. Esse conjunto é chamado de *conjunto de atributos*, e cada valor é um atributo para a característica.

O processo de extração de características e cálculo de atributos na visualização através de ícones é um processo de abstração de dados, já que o conjunto de atributos representa os dados em nível mais alto e abstrato. O processo de abstração pode ser aplicado recursivamente, usando os atributos calculados no estágio anterior como entrada para o próximo. Isso pode resultar em vários conjuntos de atributos, cada um representando os dados em níveis diferentes de abstração. Os dados em cada nível podem ser visualizados com um ícone correspondente. O nível de abstração da representação simbólica tende a aumentar se o nível de abstração dos dados aumenta; uma representação de dados mais abstrata gera uma representação visual mais abstrata (figura 3.2).

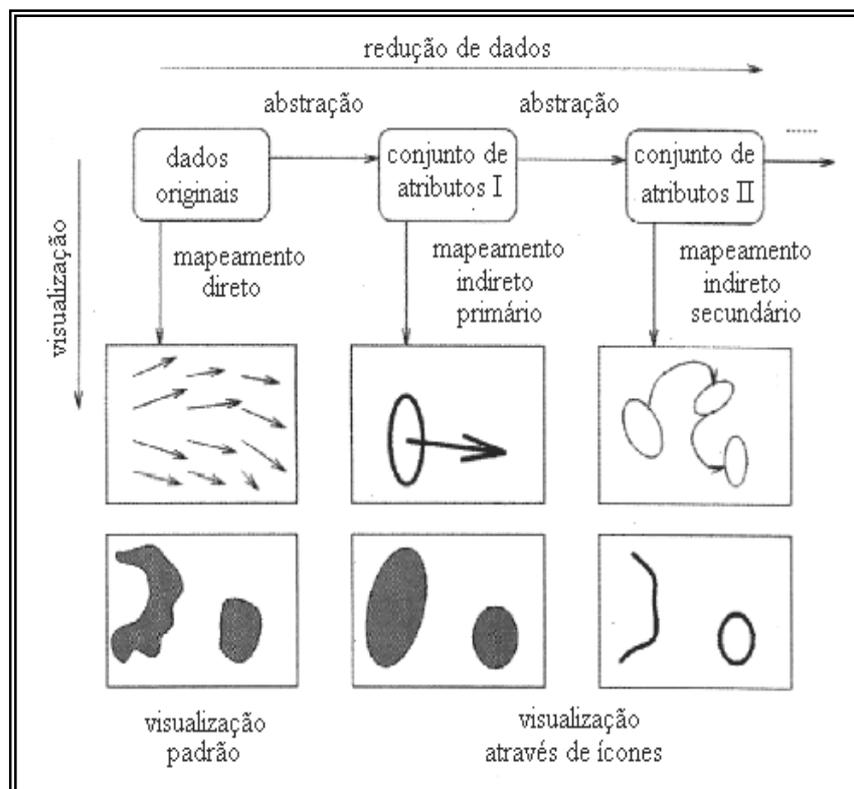


FIGURA 3.2 - Extração de características multi-nível (adaptada de [WAL 96])

Na figura 3.2, a linha superior mostra os multi-níveis do processo de abstração, do qual a redução de dados é importante efeito colateral. A cada nível de abstração, os conjuntos de atributos podem ser mapeados em ícones. O mapeamento para uma representação visual, sem abstração, é chamado de mapeamento direto. Mapeamentos com um ou mais níveis de abstração são chamados mapeamentos indiretos

O estágio de extração de características e cálculo de atributos pode ser implementado de diversas formas, e tem relação estreita com técnicas de mineração de dados. No estágio de mapeamento para ícones, os atributos são mapeados nos parâmetros dos *ícones*. A forma com que os atributos são mapeados vai determinar a aparência do ícone.

Uma importante consequência desse processo é uma redução substancial de dados. O campo de dados original é substituído usualmente por um conjunto menor de atributos, visualizados através de objetos geométricos simples. Isso provê uma excelente oportunidade para distribuição de processamento. Toda a primeira parte do processo de visualização pode ser executada em um computador remoto de alto desempenho, e os resultados podem ser facilmente transferidos para uma estação responsável pela visualização dos mesmos.

3.2 Abordagens para visualização icônica

As técnicas de visualização de informações através do uso de ícones ou glifos têm sido aplicadas em várias áreas. A seguir serão descritos alguns tipos de ícones mais comumente encontrados em sistemas de visualização.

A figura 3.3 descreve a configuração básica de um ícone *stick-figure* [ERB 95]. Os segmentos de linhas são denominados de “ramos”, e existe um ramo “principal” especial o qual serve como referência. O ícone *stick-figure* consiste de cinco segmentos conectados, denominados de ramos (*limbs*, em inglês). Um desses segmentos denominado de corpo (*body*, no original), serve como referência para várias transformações geométricas que um ícone pode sofrer. Cada ramo tem três parâmetros que podem ser mapeados para dados: *ângulo*, *intensidade* e *comprimento*. A maneira pela qual os ramos são ligados ao corpo ou uns aos outros define uma família de ícones *stick-figure*.

No ambiente de visualização *Exvis* [ERB 95] existem doze possibilidades desses ícones. A figura 3.3 mostra um membro da família de ícones *stick-figure*. A figura mostra o ícone na sua configuração básica (ícone teórico, nenhum dado mapeado para

os parâmetros dos ramos) e também um ícone de exemplo (com dados mapeados para os parâmetros dos ramos). Nesse ícone, em particular, todos os quatro ramos estão ligados diretamente no corpo.

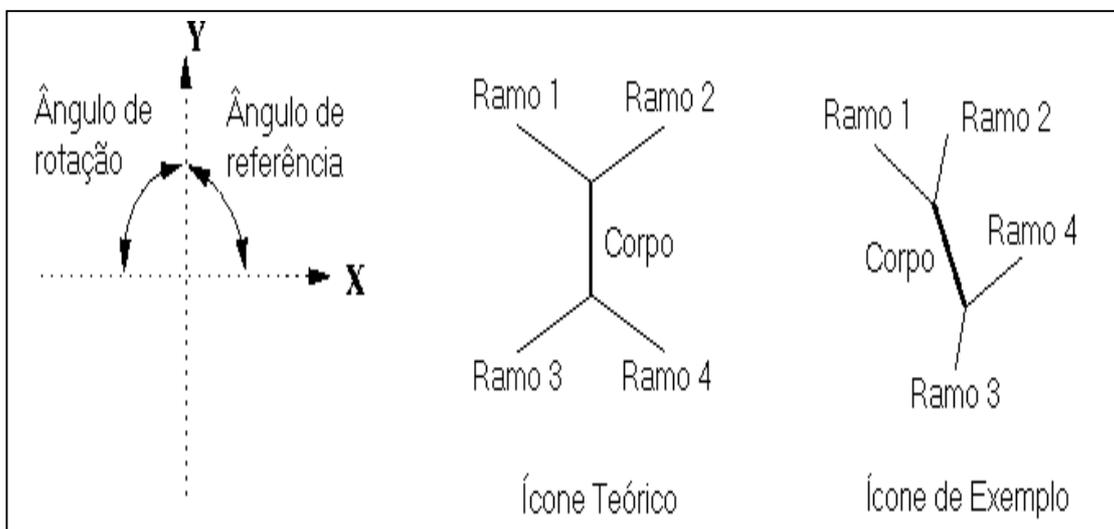


FIGURA 3.3 - Exemplo de ícone *stick-figure* [ERB 95]

Na figura 3.4 temos seis dos tipos de ícones de linhas encontrados no sistema *Exvis*:

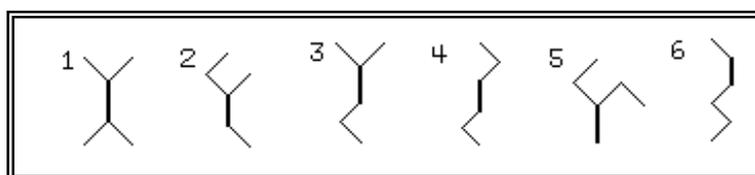


FIGURA 3.4 - Exemplos de ícones de linha no *Exvis* [ERB 95]

Os ícones *stick-figures* são bastante adequados para geração a partir de gramáticas [THE 97]. Essas gramáticas de ícones podem ser estruturadas de forma que cada iteração da derivação da gramática produza ramos adicionais. Isso pode ser bastante útil para o aumento da capacidade da representação do ícone, ou para adicionar ou reforçar detalhes na visualização. Um exemplo de como esse crescimento de ramos pode afetar o ícone é mostrado na figura 3.5. Na figura 3.5(a) temos o ícone original enquanto que nas imagens 3.5(b) e 3.5(c) temos os ícones resultantes após algumas iterações da gramática que é aplicada ao ícone. Na imagem 3.5(d) temos outro ícone e nas imagens 3.5(e) e 3.5(f) os ícones resultantes da aplicação de uma outra gramática.

O **ícone de linha colorida** é uma versão do ícone de linha *stick-figure* o qual substitue o parâmetro da intensidade do ramo com uma cor e também é implementado no ambiente *Exvis*.

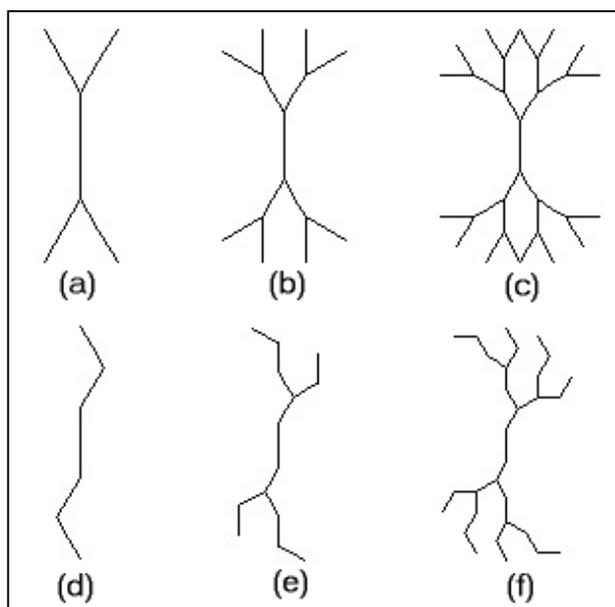


FIGURA 3.5 - Ícones *stick-figures* [THE 97]

O *ícone velcro* foi projetado para estender os efeitos de ícone de linha para três dimensões. Esse ícone é utilizado no ambiente Exvis. O ícone velcro se comporta de forma similar ao caule da grama, e produz texturas tipo grama efetivamente em 3D. A figura 3.6 abaixo descreve a configuração básica de um ícone velcro.

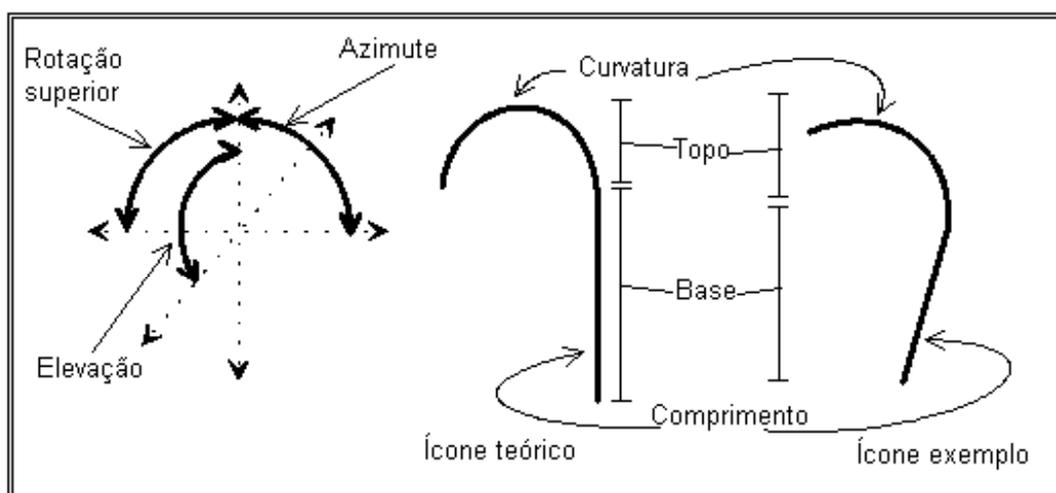


FIGURA 3.6 - Características do ícone velcro [ERB 95]

Se os ícones mostrados têm densidade suficiente, eles formam uma superfície texturizada, e estruturas nos dados são reveladas como gradientes, riscos, ou ilhas de textura contrastante.

Aplicando os conceitos introduzidos com o ícone *stick-figure* para cores, Levkowitz [LEV 91] desenvolveu o *ícone de cor 2D*. Esses ícones também são utilizados no ambiente *Exvis*. Com essa técnica, um *pixel* nos dados de entrada é representado por uma caixa $M \times N$ de *pixels*. A configuração comum mapeia os dados dependentes da cor para os quatro cantos da caixa que contém o ícone, e o interior dessa caixa é preenchido pela interpolação das cores através da caixa. A cor é especificada pelo mapeamento dos dados para os parâmetros do modelo de cor selecionado. Essa configuração permite que 12 parâmetros de dados sejam mapeados (por exemplo, *RGB* para cada um dos quatro cantos e modelo de cor *RGB*). A figura 3.7 abaixo descreve a configuração básica do ícone de cor.

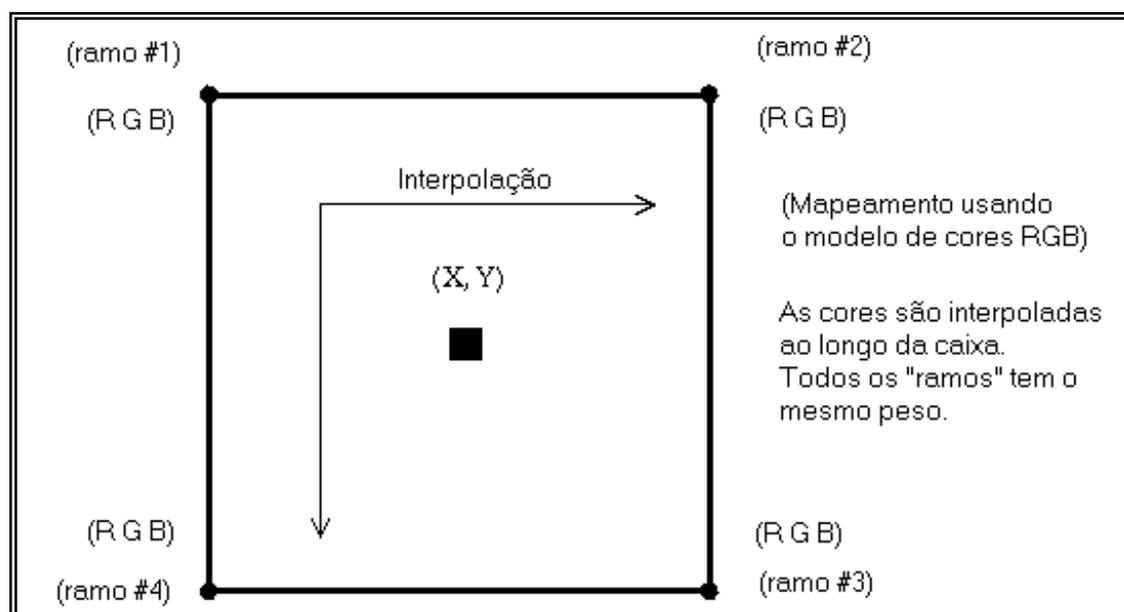


FIGURA 3.7 - Configuração básica do ícone de cor no *Exvis* [LEV 91]

Uma versão 3D do ícone de cor foi desenvolvida permitindo que os dados sejam mapeados para a altura do ícone.

Para ilustrar os conceitos acima, as figuras 3.8, 3.9, 3.10 mostram exemplos de ícones de linhas coloridas, de cor e de velcro respectivamente. A figura 3.8 representa uma visualização de um conjunto de dados do FBI sobre a quantidade de homicídios de um determinado ano usando ícones de linhas de cor. A figura 3.9 mostra uma visualização de um grande tumor usando ícones de cor, e a figura 3.10 mostra uma visualização usando ícones de velcro para o mesmo conjunto de dados da figura 3.8.

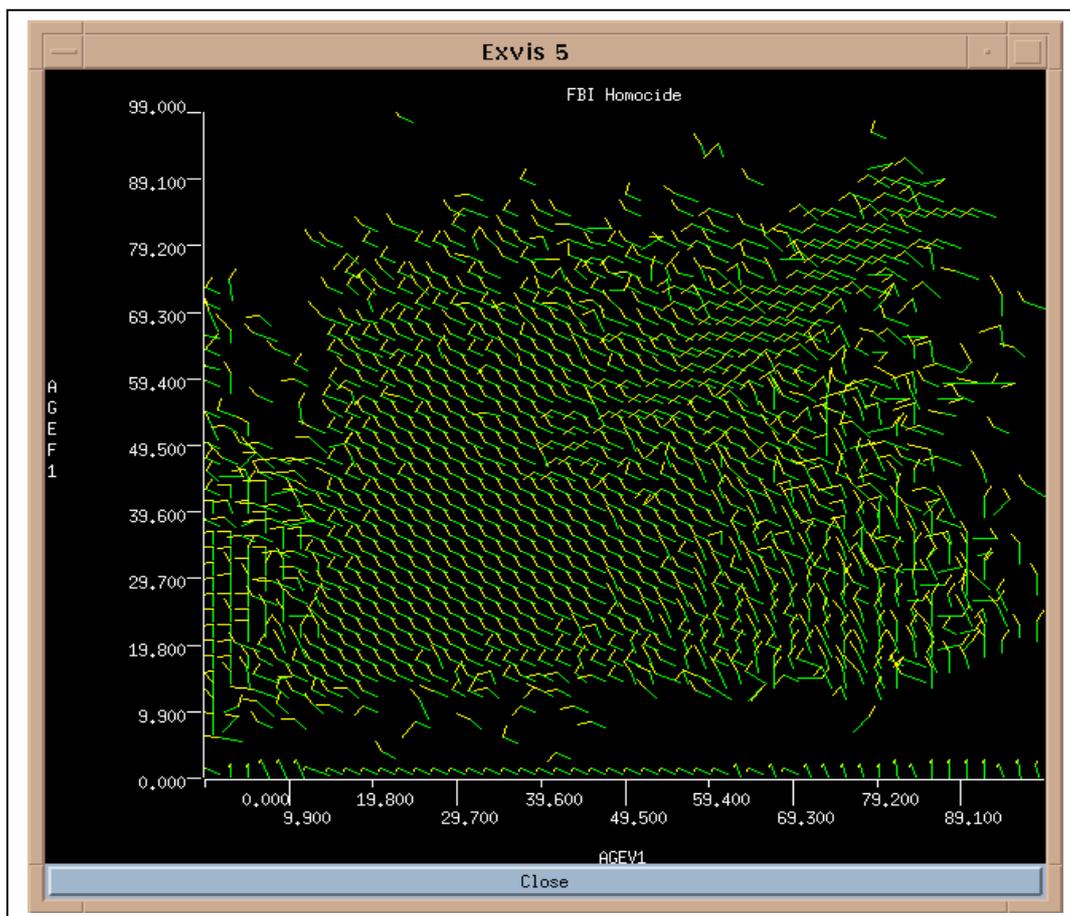


FIGURA 3.8 - Exemplo de ícones de linhas de cor

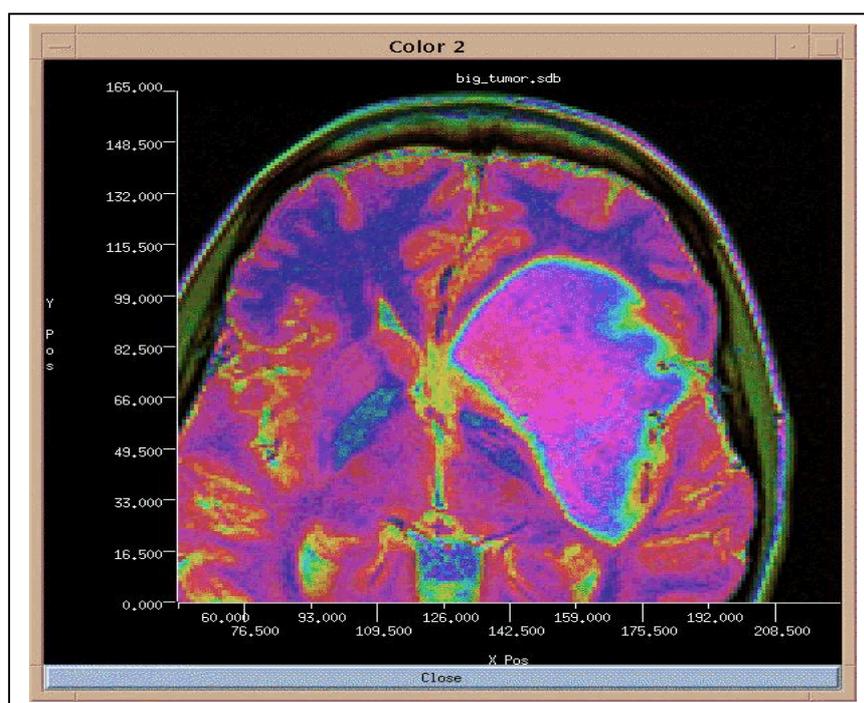


FIGURA 3.9 - Exemplo de ícones de cor

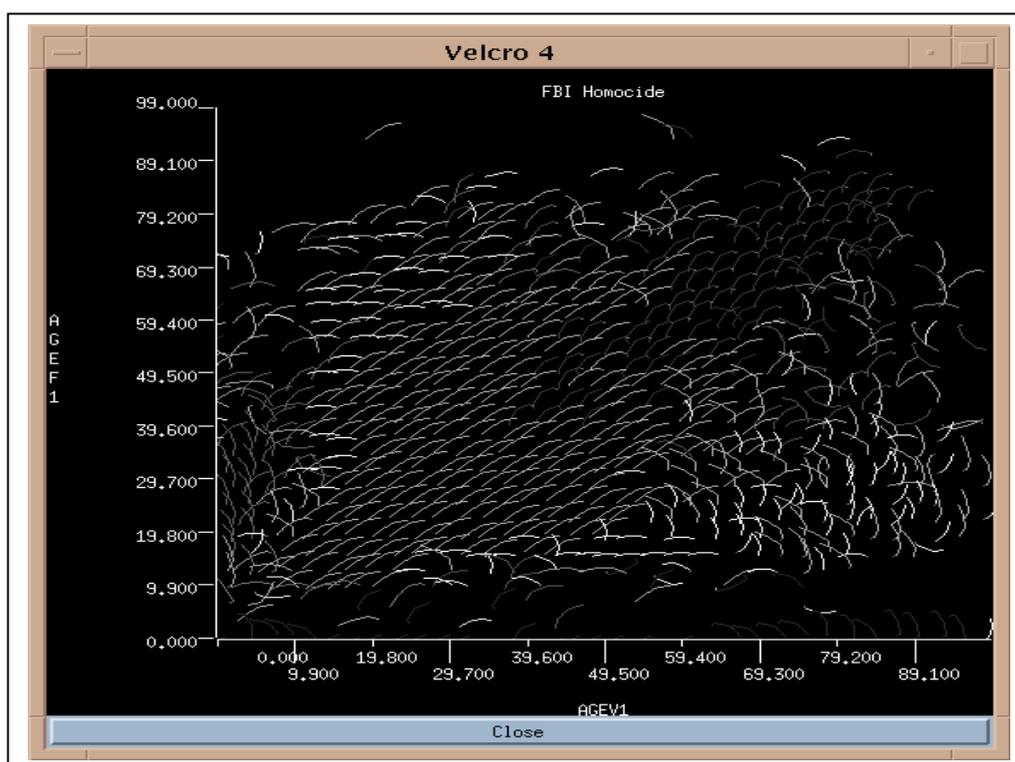


FIGURA 3.10 - Exemplo de ícones velcro

Os ícones *timewheel glyph*, *3D wheel glyph*, *infobug glyph* foram utilizados e criados por [MEI 98]. Esse ícones são utilizados para visualizar informações sobre versões de software, para apoiar o processo de manutenção. Informações quantitativas como número de linhas de código adicionadas, número de linhas de código removidas, número de pessoas na equipe de desenvolvimento, etc. fornecem uma base de dados que permite análise da evolução do software. O que Meier [MEI 95] propõe é dar suporte e melhorar o entendimento dessas informações através do uso da visualização. Os glifos são projetados de forma a expor padrões e diferenças entre versões de software.

Um dos ícones desenvolvidos para a visualização dessas informações é o *Timewheel Glyph*. Ele é utilizado para visualizar informações que variam no tempo. O modo convencional de visualizar informações é através da utilização da plotagem de séries temporais – um gráfico de linhas onde o tempo é colocado no eixo *X* e a variável no eixo *Y*. O *timewheel glyph* mostra múltiplas séries temporais, com cada série rotacionada em volta de um círculo conforme mostra a figura 3.11. Para simplificar a identificação dos atributos dos dados, cada atributo é colorido de acordo com um mapa de cores seguindo o espectro.

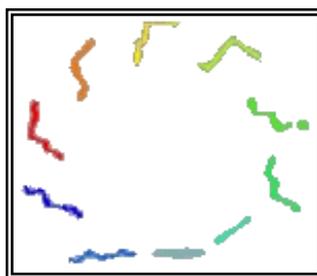


FIGURA 3.11 - *Timewheel glyph* [MEI 98]

Outro glifo utilizado é o *3D wheel glyph*. Esse glifo codifica os mesmos atributos de dados do glifo anterior (*timewheel*), mas agora neste, a terceira dimensão (a altura) é usada para codificar o tempo. Cada variável é codificada como pedaços iguais da base do círculo, e o raio do pedaço codifica o valor da variável. Cada variável também é colorida com a sua própria cor. A figura 3.12 ilustra o *3D wheel glyph*.



FIGURA 3.12 - *3D wheel glyph* [MEI 98]

Existe ainda um terceiro glifo utilizado na visualização de informações para a manutenção de software: *Infobug Glyph*. Como o próprio nome indica, ele se parece muito com um inseto. Esse glifo representa quatro importantes classes de dados sobre projeto de software, sendo dividido em 4 partes: asa, cabeça, cauda e corpo, conforme mostra a figura 3.13.

Cada asa representa uma série temporal com o tempo variando de cima para baixo. O eixo X da asa da esquerda codifica o número de linhas de código, e o eixo X da asa da direita codifica o número de erros. Geralmente, o aumento de código traz consigo um aumento do número de erros. Isso resulta em asas simétricas. A posição da asa (localizada a partir do começo no topo em direção a base) indica o tempo no qual o projeto iniciou. Através de animação o *infobug glyph* pode mostrar diferentes informações de diferentes tempos do projeto. Clicando sobre a asa seleciona-se um fatia de tempo, causando uma modificação na cabeça, corpo e cauda de acordo com a informação dessa fatia de tempo.

Um componente de software em particular pode consistir de vários tipos diferentes de código, por exemplo, C, C++, Java ou VRML, sendo o tipo de código codificado por uma cor. A cabeça representa o tamanho do código relativo, por tipo, numa dada fatia de tempo.

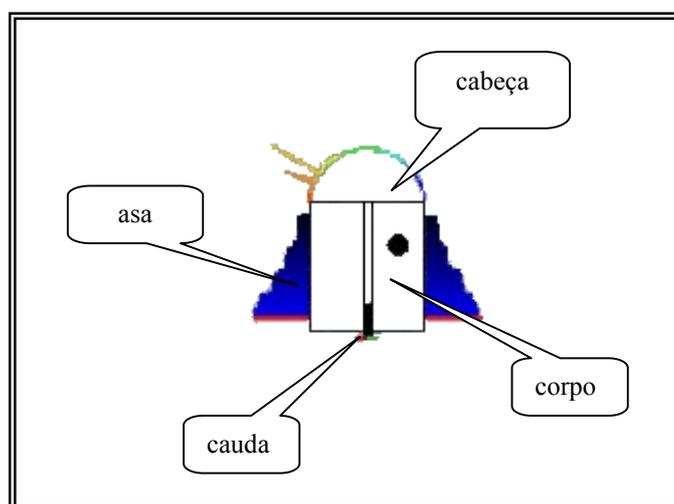


FIGURA 3.13 - *Infobug glyph* [MEI 98]

A cauda do ícone é representada por uma forma triangular. A base codifica o número de linhas de código adicionadas e a altura, o número de linhas apagadas. Além disso, a base da cauda do ícone é dividida em duas partes: código adicionado devido a correção de erros (representado em vermelho) e código adicionado para novas funcionalidades (representado em verde).

Geralmente importante, o tamanho do componente reflete a amplitude que um componente afeta o projeto. Isso é codificado de duas formas: através do número de arquivos alterados e através do número de objetos filhos que um componente de software contém. A barra no meio do corpo do *infobug glyph* mostra o número absoluto de mudanças de arquivos. O tamanho dos círculos escuros no corpo do *infobug glyph* codifica o número de componentes contidos dentro dos objetos correntes.

O glifo normalmente usado para representar condições do tempo em meteorologia (*weather glyph*) é citado em [COX 88]. O glifo do tempo (figura 3.14) indica a velocidade e a direção do vento, assim como a quantidade de nuvens. O valor escalar, a fração da quantidade de nuvens, vem da quantidade de preto no símbolo. Na aplicação descrita por Cox [COX 88], é mostrada a utilização de glifos para visualização do processo de injeção em moldes. Este é um método econômico para a fabricação de estruturas com paredes finas com formas complicadas, como em vários

ítems familiares : aparelhos de telefone, caixas para fitas de áudio e vídeo, comportas para cubos de gelo, etc.

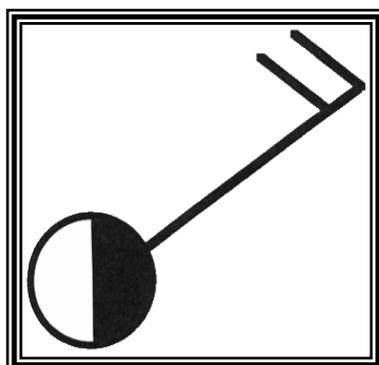


FIGURA 3.14 - Glifo de tempo em meteorologia [COX 88]

As variáveis chaves no processo de injeção foram identificadas como pressão, temperatura e velocidade, com um valor de pressão para cada elemento finito (a simulação é realizada através da análise de elementos finitos), e um vetor velocidade e uma temperatura para cada diferença finita de cada elemento.

A forma do glifo é derivada a partir do vetor velocidade. Assim como em um *weather glyph*, o comprimento e a orientação da forma do glifo indica a magnitude da velocidade e a direção do fluxo. A geometria desse glifo (figura 3.15) é mais complexa do que a do glifo do tempo pois ele representa um perfil da velocidade ao invés de uma velocidade única. Assim, ao longo da distância vertical do glifo, a qual corresponde a distância do centro da seção até as paredes do molde, a magnitude da velocidade varia. Cada glifo está sobre uma base triangular, o domínio do elemento finito. A malha de elementos finitos tem um glifo por elemento. Cada glifo indica a direção do fluxo pela rotação na base triangular do elemento finito.

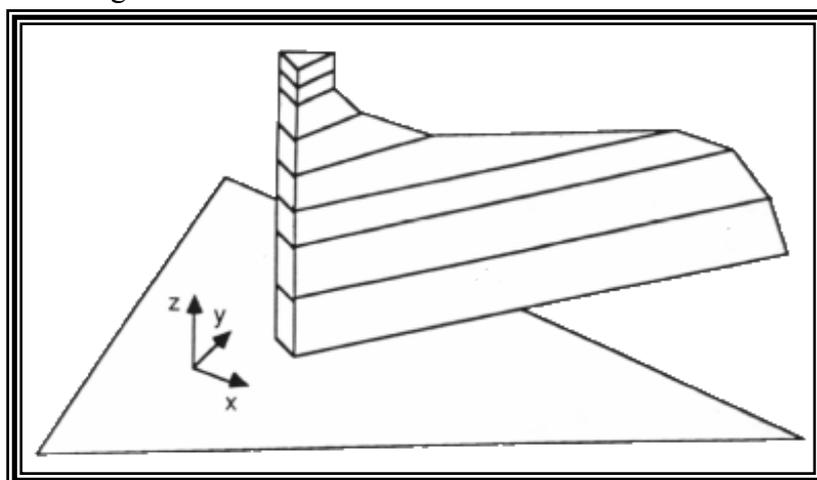


FIGURA 3.15 - Geometria do glifo para representar o processo de preenchimento de moldes [COX 88]

Os valores para a pressão e temperatura são colocados em escalas de cores separadas. O glifo foi pintado com as cores da temperatura para indicar a variação da temperatura sobre o perfil vertical. A pressão foi indicada por um valor de cor preenchendo o elemento finito no qual o glifo está localizado.

4 Geração automática de ícones

4.1 Introdução

Por geração automática de ícones entende-se a possibilidade de obter-se ícones diferenciados automaticamente conforme a necessidade em termos de visualização. Numa situação ideal, a ferramenta de visualização analisaria os dados e determinaria que representação visual é a mais conveniente considerando as entidades e atributos que devem ser apresentados e a tarefa que será realizada a partir da representação visual.

Normalmente, como visto no capítulo anterior, o processo de visualização icônica corresponde ao mapeamento de atributos dos dados para atributos visuais dos ícones ou glifos.

Um exemplo típico são as faces de *Chernoff* (figura 4.1) [CHE 73]. Neste método, faces humanas foram utilizadas para individualizar pontos em um espaço k -dimensional. Características da face humana foram escolhidas para representar cada uma das k dimensões desse espaço. Assim, olhos, nariz, sobrancelhas, distância entre sobrancelhas, orelha, formato da face, etc foram utilizados como meio de apresentação dos valores de cada uma das k dimensões. No trabalho original, o método foi utilizado inicialmente com dados financeiros e populacionais.

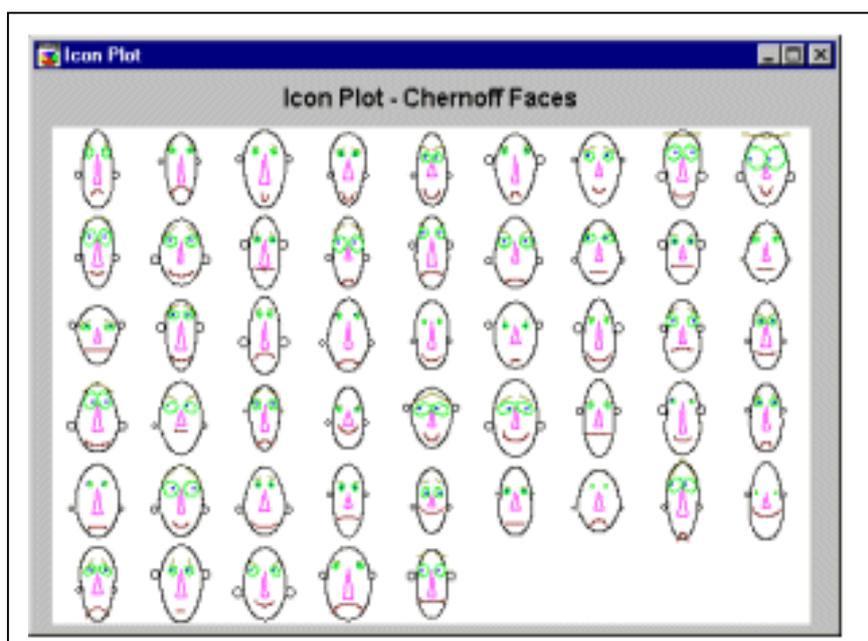


FIGURA 4.1 - Faces de *Chernoff*

Cumprе ressaltar que os ícones não são gerados automaticamente, ou seja, uma vez escolhida a forma visual "face de Chernoff", o mapeamento atributo - característica visual precisa ser programado.

Como vimos no capítulo anterior, a maioria das propostas para visualização icônica apresenta um conjunto fixo de ícones ou glifos para um determinado sistema de visualização. Esse conjunto de ícones geralmente está descrito em uma biblioteca pré-definida para o sistema e não pode ser modificado ou expandido. Assim, o usuário fica restrito à utilização das entidades gráficas definidas para o sistema.

Um conjunto fixo de ícones (em uma biblioteca ou não) faz com que o usuário fique bastante limitado, podendo apenas mapear seus dados e informações para as entidades existentes. A adição de novos ícones em um sistema que trabalha dessa forma seria bastante custosa, implicando na modificação do código fonte para dar suporte aos novos ícones. O programa deve ser recompilado, podendo surgir novos erros (pois foi adicionado mais código para dar suporte aos novos ícones) ocasionando maior perda de tempo para o desenvolvedor e para o usuário final.

Um exemplo desse tipo de sistema é o *PV-WAVE: Glyph Multivariable Visualization Toolkit (GMVT)* [JOS 94] é uma ferramenta criada para o software de visualização denominado *PV-WAVE*. Esse *toolkit* é constituído de duas partes: uma é a biblioteca de rotinas de glifos escritas em uma linguagem própria da ferramenta *PV-WAVE* e a outra é uma interface gráfica para a utilização dessa biblioteca.

A biblioteca de rotinas do *PV-WAVE: GMVT* é um conjunto de procedimentos e funções desenvolvidos para desenhar os vários tipos de glifos bem como ler, escrever e manipular o conjunto de dados dos usuários. O *PV-WAVE* é uma arquitetura aberta e, assim, é possível criar um novo conjunto de procedimentos e funções de acordo com as necessidades do usuário.

A figura 4.2 mostra a interface gráfica do *toolkit*. Através do uso dessa interface é possível criar e manipular os ícones e glifos gerados, bem como fazer o mapeamento de dados e informações dos mesmos. Para criar um glifo basta escolher um dos tipos de glifos disponíveis no programa de visualização *PV-WAVE*. Como podemos ver na figura, estão disponíveis sete tipos de glifos, inclusive o *Exvis* discutido anteriormente, e as faces de Chernoff. Um ícone proposto pelo autor é o *Joslyn Glyph* [JOS 94] que é bastante simples como pode-se ver na mesma figura. Esse ícone tem a forma básica representada por um cubo e os dados mapeados para esse ícone controlam os vários parâmetros dessa forma básica: largura, comprimento, profundidade, ângulo de rotação em torno do eixo x,y,z e cor.

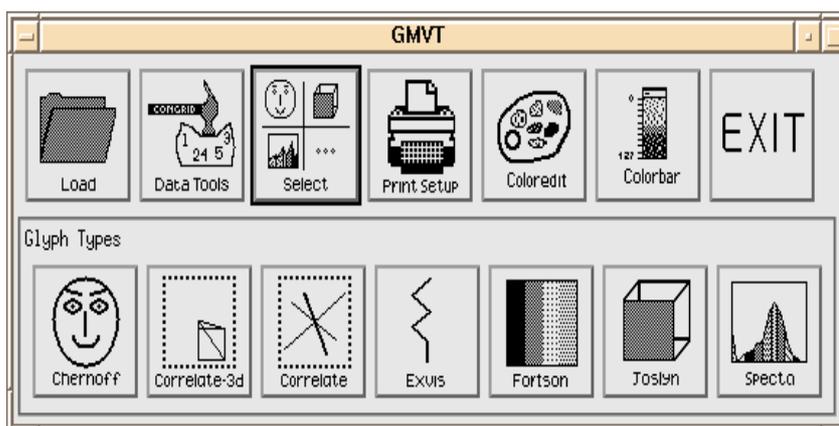


FIGURA 4.2 - Interface gráfica do *GMVT*

Mas como discutido anteriormente, o usuário fica limitado a usar somente os tipos de glifos que estão disponíveis nesse programa. Se o usuário necessitar ou quiser utilizar uma outra representação gráfica para seus dados, ele terá que desenvolver as rotinas para desenhá-las para depois adicionar esses novos procedimentos à biblioteca. O ideal seria que o usuário apenas fornecesse uma especificação da sua representação gráfica (ícone ou glifo) e que o programa de visualização interpretasse essa especificação e gerasse os ícones automaticamente.

A geração automática de ícones é uma solução para esse tipo de problema. Esse tema é muito pouco abordado na literatura. A idéia central da geração automática de ícones é ter uma forma de representar os ícones que não implique na alteração do programa de visualização.

4.2 Proposta para geração de ícones

A geração automática de ícones pode ser obtida de duas formas: a primeira delas seria através de um programa inteligente o suficiente para determinar o tipo de representação visual mais adequado para um conjunto de dados; a segunda seria a especificação dos ícones pelo usuário.

A proposta dessa dissertação é criar uma forma de especificação de representações visuais que possa ser adequada pelo usuário, dependendo da natureza dos dados. O que se propõe é a utilização de um arquivo texto para descrever os ícones criados pelo usuário. Esse arquivo texto tem um formato padrão e irá servir de entrada para o programa de visualização. Através da descrição de ícones fazendo uso dessa representação o usuário não vai ficar limitado a um conjunto de ícones disponíveis em uma biblioteca de ícones. O usuário terá a liberdade de criar e escolher os ícones que

melhor se adaptem aos seus dados. Além desses ícones personalizáveis o usuário ainda pode fazer uso dos ícones pré-definidos pelo programa de visualização.

O esqueleto do arquivo de descrição de ícones, com alguns campos já previamente definidos segue abaixo :

```

!ICONS
TOTAL_ICONS      = n
PRIMITIVES      = [SQUARE|CIRCLE|TRIANGLE]
DIVISION_TYPE   = [0|1|2|3]
STAMP           = [0|1|2|3|4|5]
POSITION        = [LEFT|RIGHT|CENTER|TOP|BOTTOM]

<ICON_i
  NAME = ICON_NAME
  BODY = {
    BASE_FORM = PRIMITIVES
    BASE_PROPERTIES : NULL
    BASE_SIZE = W x L
    PRIMITIVE_1 (PARAMETERS)
    PROPERTIES : NULL

    ...

    PRIMITIVE_N (PARAMETERS)
    PROPERTIES : [DIV(X), STA(Y,Z)]
  }
>
ICONS!

```

O arquivo de definição de ícones poderá ter n ícones. Todo arquivo de descrição de ícones deve conter a marcação *!ICONS* no início do arquivo e *ICONS!* para indicar o fim de arquivo de descrição de ícones. A segunda marcação *TOTAL_ICONS* indica o número total de ícones dentro do arquivo. *PRIMITIVES* é uma lista de formas primitivas das quais o ícone pode ser constituído. O programa de visualização deve conter as rotinas para desenhar essas formas primitivas. A marcação *DIVISION_TYPE* indica se a forma primitiva será subdividida ou não. Um valor zero (0) indica que a forma primitiva não será dividida; um valor diferente de zero a primitiva pode ser dividida (figura 4.3).

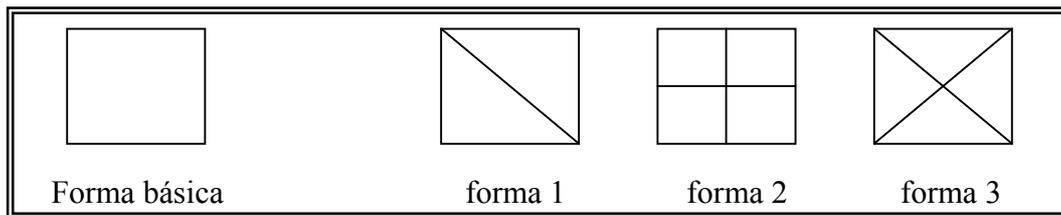


FIGURA 4.3 - Divisões possíveis de um ícone

O exemplo acima aplica-se à forma primitiva *SQUARE*, para as demais formas primitivas o procedimento de divisão é análogo. A divisão da forma primitiva é utilizada para que mais valores possam ser mapeados para o ícone. Por exemplo, a forma 2 da figura 4.3 está dividida em 4 partes. Assim, cada uma dessas partes poderia ser atribuída a uma característica dos dados. Se o usuário está trabalhando com dados de uma população de animais, ele poderia escolher mapear o peso para uma das partes, o comprimento para outra, a idade para outra e a raça do animal para outra, além da forma (nesse exemplo, um quadrado) poder representar o sexo do animal. Assim teríamos cinco atributos mapeados para esse ícone, possibilitando ao usuário visualizar um conjunto maior de informações simultaneamente na tela. Utilizando apenas a forma básica da figura 4.3, um usuário poderia ter apenas dois atributos do ícone para representar dois atributos de dados. Por exemplo, a forma básica representa o sexo e o seu preenchimento poderia ser utilizado para representar o peso dos animais, segundo uma escala de cores pré-definidas ou escolhida pelo usuário.

A marcação *STAMP* indica se o ícone fará uso de uma marcação. Essa marcação pode ser utilizada para fazer o mapeamento de valores enumerados. O valor 0 (zero) em *STAMP* indica que o ícone não terá uma marcação. Para o exemplo do arquivo descritor acima, foram definidos 5 (cinco) tipos de marcações possíveis para os ícones que podem ser vistas na figura abaixo:

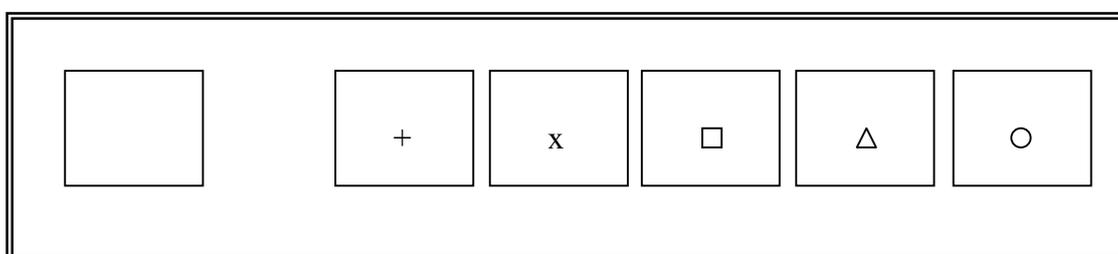


FIGURA 4.4 - Marcações possíveis de um ícone

Novamente, o exemplo da figura aplica-se ao *SQUARE* podendo facilmente ser estendido para as demais primitivas básicas. *POSITION* indica a posição em que será

colocada a marcação. No exemplo da figura 4.3 as marcações foram colocadas em *CENTER*.

A definição do ícone propriamente dito é indicado pelas marcações $\langle \text{ICON}_i \rangle$ onde i é um número inteiro para identificação do ícone. A marcação *NAME* é opcional e serve para dar um nome ao ícone. A marcação *BODY* irá definir a forma do ícone. A marcação *BASE_SIZE* serve para definir o tamanho (largura x comprimento, pois as primitivas são em duas dimensões) da forma primitiva usada como base para a construção do ícone. A marcação *PROPERTIES* serve para informar as propriedades da forma básica. Um valor *NULL* indica que a primitiva manterá sua forma original. A propriedade *DIV(X)* indica qual o tipo de divisão da forma primitiva. O valor de X será um dos valores possíveis para o *DIVISION_TYPE*. Na propriedade *STA(Y,Z)* o valor para Y será um dos valores possíveis para *STAMPS*, indicando qual tipo de marcação que o ícone irá utilizar e Z terá um dos valores da propriedade *POSITION* indicando a posição em que essa marcação deve ser colocada no ícone.

Esse arquivo texto pode ser gerado manualmente (que é como o protótipo está implementando atualmente) ou poderia ser criada uma interface gráfica para orientar o usuário a gerar esse arquivo. Assim, à medida que o usuário vai fazendo modificações no seu ícone, essas modificações vão sendo visualizadas simultaneamente à criação do ícone.

O recurso de dividir as formas primitivas e o uso de marcações foi idealizado com o objetivo de aumentar o número de parâmetros utilizados para mapeamento dos ícones. Adicionalmente a esses recursos, o uso de cores permite representar mais informações. No caso de grandezas escalares, os valores podem ser mapeados para cores. Em uma aplicação de visualização de populações animais as cores poderiam ser utilizadas para representar o peso, comprimento, largura ou outra característica cujos valores sejam escalares. O usuário pode escolher uma cor para o valor inicial (menor valor, por exemplo) e uma cor para o maior valor, e o programa então gera uma variação de cores partindo da cor inicial até a cor final. No caso do peso, por exemplo, o usuário poderia escolher um tom de amarelo com pouco intensidade para os animais que tenham os menores pesos e um tom de vermelho mais intenso para os animais cujos pesos sejam maiores.

Para desenvolver um mapeador de características para parâmetros dos ícones, este deve ter as informações sobre essas características, ou seja, quais correspondem a valores escalares ou vetoriais, por exemplo, ou quais serão enumerações. De posse dessas informações, o mapeador irá construir uma tabela que representa o mapeamento das diversas características para os parâmetros dos ícones.

5 Estudo de caso

5.1 Descrição da aplicação-exemplo

A título de experimentação de mapeamento de dados para ícones, foi implementado um protótipo denominado *IconVis (Iconic Visualization System)* para a visualização de informações sobre populações animais estudadas por biólogos.

O projeto dos biólogos do Laboratório de Citogenética Animal do Departamento de Genética da UFRGS tem como objetivo o estudo da dinâmica populacional e fluxo gênico entre populações da espécie de roedores fossoriais *Ctenomys* no Parque Estadual de Itapuã (tucu-tuco). Os animais são capturados, marcados com tatuagem, suas posições na região de estudo são registradas, e são retiradas amostras de sangue e pele. Medidas morfológicas são tomadas e anotadas, assim como dados característicos como sexo, coloração e outros aspectos morfológicos. Em laboratório, dados genéticos são obtidos a partir do processamento das amostras de sangue e pele. O estudo da dinâmica populacional envolve a análise dos diversos dados registrados em relação à localização espacial dos pontos de captura e recaptura dos animais.

Em função dos objetivos do trabalho é importante a localização dos animais e, assim, a visualização básica é geo-referenciada. O sistema de coordenadas natural neste caso seria um sistema de coordenadas cartesianas envolvendo latitude e longitude. Entretanto, tendo em vista que a localização dos animais por *GPS (Global Positioning System)* é impossível devido à proximidade dos pontos de coleta e por falta de resolução do GPS há necessidade de se utilizar outra forma alternativa de localizar os pontos de coleta dos animais. Isto, em geral, é feito tomando algum item da região como referencial. Nesta região específica, os pontos de coleta utilizam como referência uma cerca já existente no local, cuja representação esquemática pode ser vista na figura 5.1. Inicialmente, a cerca foi dividida em marcos de referência. Esses marcos são os pontos *A, B, C, D, E, F* e *G* que estão representados na figura como bandeiras. Os pontos servem de base para o cálculo da posição dos animais capturados.

A necessidade de estabelecer diferentes referenciais, dependendo da região de estudo, induz à necessidade de registrar também na base de dados as características geométricas que definem o referencial, o que é pouco usual na maioria dos sistemas de visualização. Como mencionado anteriormente, em geral, quando dados são geo-referenciados é utilizado um sistema de coordenadas latitude-longitude. Na situação descrita neste estudo de caso, os pontos de coleta são determinados indicando dois pares (marco, ângulo tomado com a bússola), onde marco corresponde a um dos pontos A-G da figura 5.1.

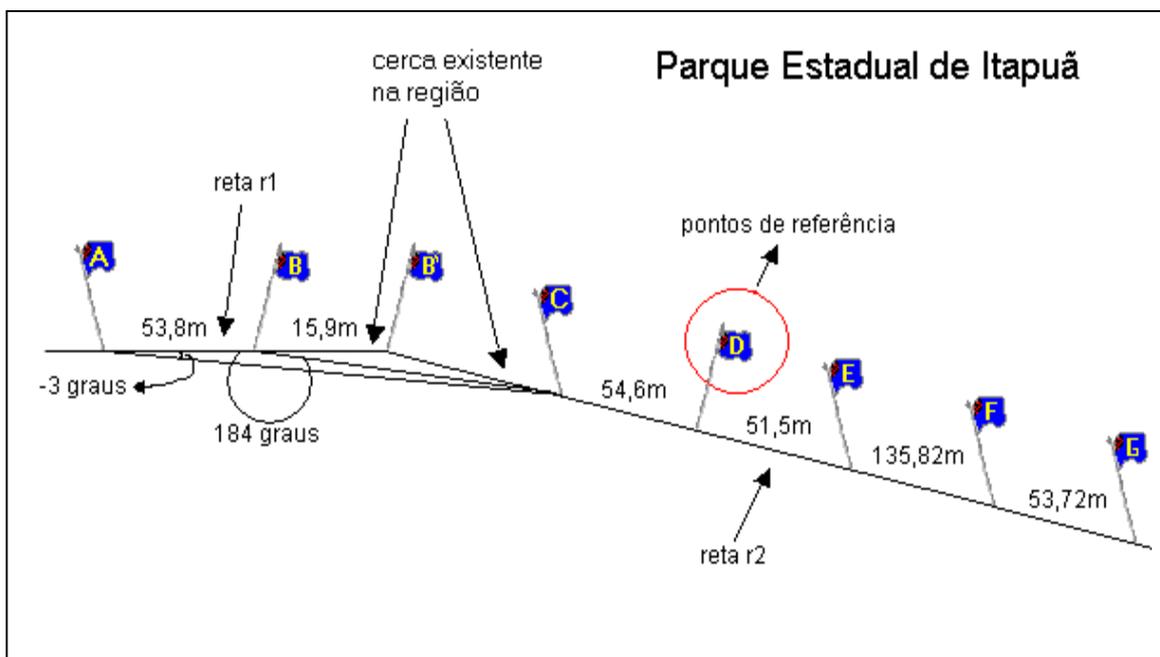


FIGURA 5.1 - Região de estudos dos biólogos do Departamento de Genética da UFRGS

Os dados coletados pelos biólogos são enumerados a seguir. *IconVis* usa uma tabela do banco de dados *Microsoft Access*. O banco de dados *Microsoft Access* foi o escolhido pois os biólogos já vinham utilizando este software para armazenar as informações coletadas sobre os animais.

- **tuco_ID**: chave primária da tabela; é um campo especial do Microsoft Access geralmente utilizado para chaves primárias e funciona basicamente como um contador. À medida que novos registros são inseridos na base de dados, esse contador é aumentado em uma unidade. Essa atualização é feita automaticamente e o usuário não pode editar esse campo.
- **número**: é a identificação atribuída aos animais pelos biólogos. Cada animal é identificado por um prefixo correspondente às iniciais do biólogo que coletou o exemplar mais um número sequencial. Esse identificador é único para cada animal. A base de dados irá conter *XXI* (o primeiro animal a ser capturado) até o *XXnn* onde *nn* é o último animal capturado pelo biólogo *XX*.
- **recaptura_1**: um campo do tipo *Sim/Não* para indicar se o animal foi recapturado após a primeira coleta.
- **recaptura_2**: um campo do tipo *Sim/Não* para indicar se o animal foi recapturado duas vezes. Como pode ser deduzido esta informação permitindo analisar a dinâmica dos animais dentro da região de estudo. Isto é importante, uma vez que os animais habitam túneis e poucos dados existem sobre a sua mobilidade na região.

- **data:** indica a data em que o animal foi capturado.
- **ponto_A:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como A.
- **ponto_B:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como B.
- **ponto_C:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como C.
- **ponto_D:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como D.
- **ponto_E:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como E.
- **ponto_F:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como F.
- **ponto_G:** indica a informação angular do ponto de coleta em relação ao marco de localização definido como G.
- **sexo:** para indicar se o animal é fêmea ou macho.
- **peso:** o peso do animal no momento da captura em gramas.
- **comprimento_total:** para armazenar o comprimento total do animal (corpo mais a cauda) no momento da captura, em milímetros.
- **pata_sem_unha:** comprimento da pata do animal sem considerar o comprimento da unha, em milímetros.
- **pata_com_unha:** comprimento da pata do animal mais o comprimento da unha, em milímetros.
- **dente:** contém a largura do dente incisivo do animal, em milímetros.
- **hora:** hora em que o animal foi capturado.
- **características_sexuais:** contém informações sobre as características sexuais dos animais. Essa informação se refere apenas aos indivíduos do sexo fêmea. Valores possíveis: *cicatrizada*, *perfurada*, *perfurada recentemente*, *imperfurada*, *cicatrizada (c/secreção)*
- **tatuagem:** ao capturar o animal, os biólogos fazem uma tatuagem de identificação do animal, correspondente a um número. Este campo serve para armazenar o valor dessa tatuagem. É um valor único para cada animal.
- **cauda:** armazena o comprimento da cauda do animal
- **observação:** contém informações relevantes sobre o animal no momento de sua captura, por exemplo se a fêmea está grávida, se o animal está com a pata roída, etc.

5.2 Protótipo

Dadas as características dos dados coletados, a forma de representação visual natural é a icônica. Assim, o *IconVis* utiliza as informações coletadas pelos biólogos sobre os animais como fonte primária de dados para a visualização através de ícones localizados espacialmente em função dos pontos de coleta. A maneira pela qual o cálculo da posição de captura dos animais bem como as técnicas utilizadas para fazer a visualização desses dados serão descritas a seguir.

O protótipo foi inteiramente desenvolvido usando a linguagem de programação *Delphi* versão 4 do fabricante *Inprise Corp.*

A figura 5.2 mostra uma visão geral da arquitetura do programa. Como os dados que são utilizados pelo programa estão em um banco de dados é necessário usar uma camada intermediária, o *BDE* (*Borland Database Engine*). As informações a serem visualizadas estão contidas em uma tabela do banco de dados *Microsoft Access*.

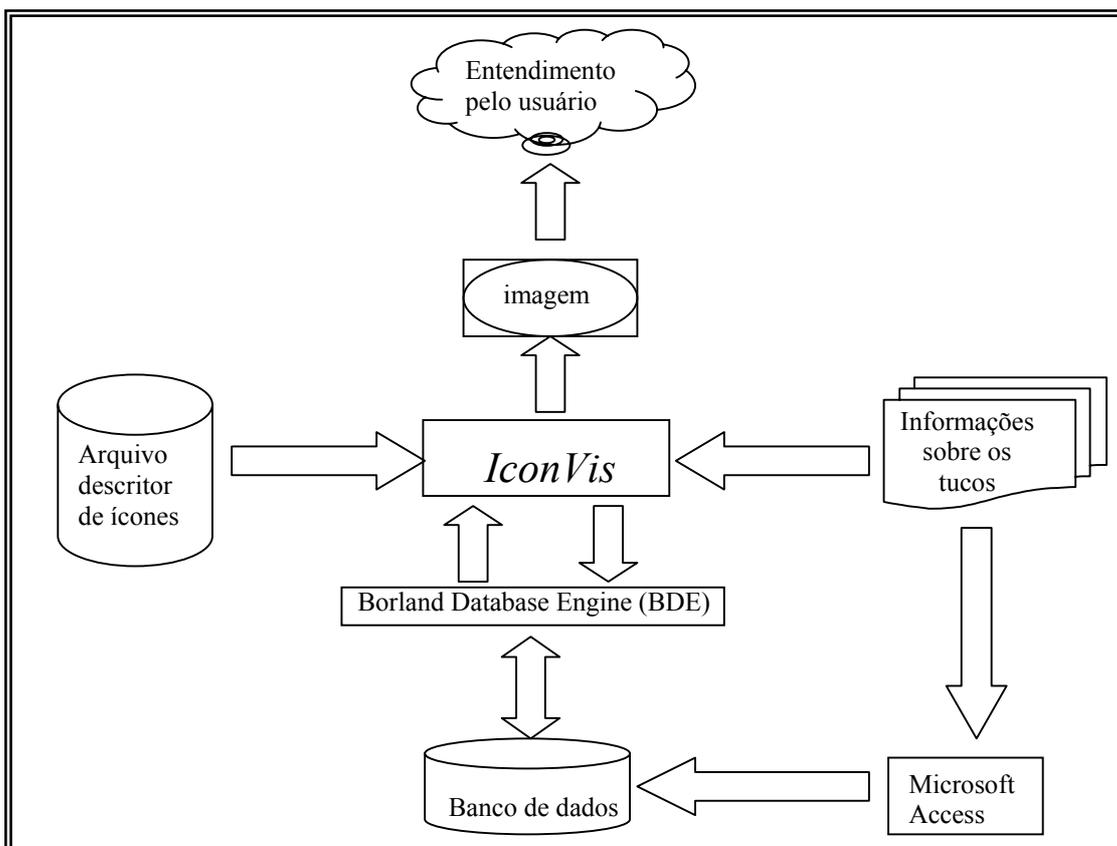


FIGURA 5.2 - Arquitetura do *IconVis*

O *BDE* que acompanha o *Delphi* fornece um conjunto completo de ferramentas de programação para muitos sistemas de programação de bancos de dados, inclusive para o *Microsoft Access*. Os componentes de banco de dados do *Delphi* proporcionam um aspecto orientado a objetos ao desenvolvimento de aplicativos de bancos de dados. Na próxima seção segue uma descrição mais detalhada do funcionamento do *Borland Database Engine*.

5.2.1 O acesso ao banco de dados

Essa seção tem por objetivo explicar o funcionamento do acesso ao banco de dados. Primeiramente será abordada uma descrição mais genérica de um sistema de gerenciamento de banco de dados (SGBD) e, por final, será explicado como esses conceitos genéricos foram implementados no protótipo *IconVis*.

Um sistema de bancos de dados é dividido em módulos que tratam de cada uma das responsabilidades do sistema geral. Na maioria dos casos, o sistema operacional do computador fornece apenas os serviços mais básicos, e o sistema de banco de dados precisa ser construído sobre essa base. Então, o projeto do sistema de banco de dados precisa incluir considerações sobre a interface entre o sistema de bancos de dados e o sistema operacional.

Os componentes funcionais de um sistema de banco de dados incluem:

- **Gerenciador de arquivos**, que gerencia a alocação do espaço de armazenagem do disco e as estruturas de dados usadas para representar a informação armazenada no disco.
- **Gerenciador de banco de dados** que fornece a interface entre os dados de baixo nível armazenados no disco e os programas aplicativos e de consulta submetidos ao sistema.
- **Processador de consultas** que traduz comandos numa linguagem de consulta em instruções de baixo nível que o gerenciador do banco de dados entende. Além disso, o processador de consultas tenta transformar uma requisição do usuário em uma forma compatível e mais eficiente com respeito ao banco de dados, encontrando uma boa estratégia para executar a consulta.
- **Pré-compilador DML (linguagem de manipulação de dados)** que converte comandos DML embutidos em um aplicativo para chamadas de procedimento normal na linguagem hospedeira. O pré-compilador precisa interagir com o processador de consultas para gerar o código apropriado.

- **Compilador DDL (linguagem de definição de dados)** que converte comandos DDL em um conjunto de tabelas contendo *metadados* ou “dados sobre dados”.

Adicionalmente, diversas estruturas de dados são requisitadas como parte da implementação do sistema físico, incluindo:

- **Arquivos de dados**, que armazenam banco de dados por si mesmos.
- **Dicionário de dados**, que armazenam metadados sobre a estrutura do banco de dados. O dicionário de dados é bastante usado. Assim, deve-se dar uma grande ênfase ao desenvolvimento de um bom projeto e eficiente implementação ao dicionário.
- **Índices**, que fornecem acesso rápido aos itens de dados guardando dados particulares.

De forma resumida, um sistema de gerenciamento de banco de dados (SGBD) consiste em uma coleção de dados inter-relacionados e em uma coleção de programas para acessar esses dados. O objetivo principal de um sistema de banco de dados é prover um ambiente que seja *conveniente* e *eficiente* para uso na recuperação e no armazenamento da informação. A figura 5.3 abaixo mostra a estrutura de um SGBD genérico.

No protótipo *IconVis*, o SGBD utilizado foi o Microsoft Access. A aplicação propriamente dita foi desenvolvida usando o Borland Delphi. O Delphi contém uma API para acesso a banco de dados. Essa API é o Borland Database Engine (BDE). O acesso a essa API pode ser feito via programação ou também através do uso dos componentes já criados para essa finalidade. Esses componentes implementam as principais funções de acesso a banco de dados, recuperação de informações e todas as demais operações que envolvem um banco de dados. Esses componentes são por exemplo: *TDatabase*, *TTable*, *TQuery* (que serão explicados mais adiante). Através do uso desses componentes já desenvolvidos a programação para o acesso ao banco de dados é facilitada.

O BDE fornece a capacidade de acesso padronizado a banco de dados para Delphi, C++ Builder e outros ambientes de programação da Borland, oferecendo um grande conjunto de funções para auxiliar no desenvolvimento de aplicações Desktop e Cliente/Servidor. Os controladores da BDE podem ser usados para acessar bases de dados dBase, Paradox, Access, FoxPro, Interbase, Oracle, Sybase e MS-SQL Server,

DB2, Informix, além de um controlador de acesso a arquivos texto. Pode-se utilizar também fontes de dados ODBC, podendo acessar qualquer base de dados compatível.

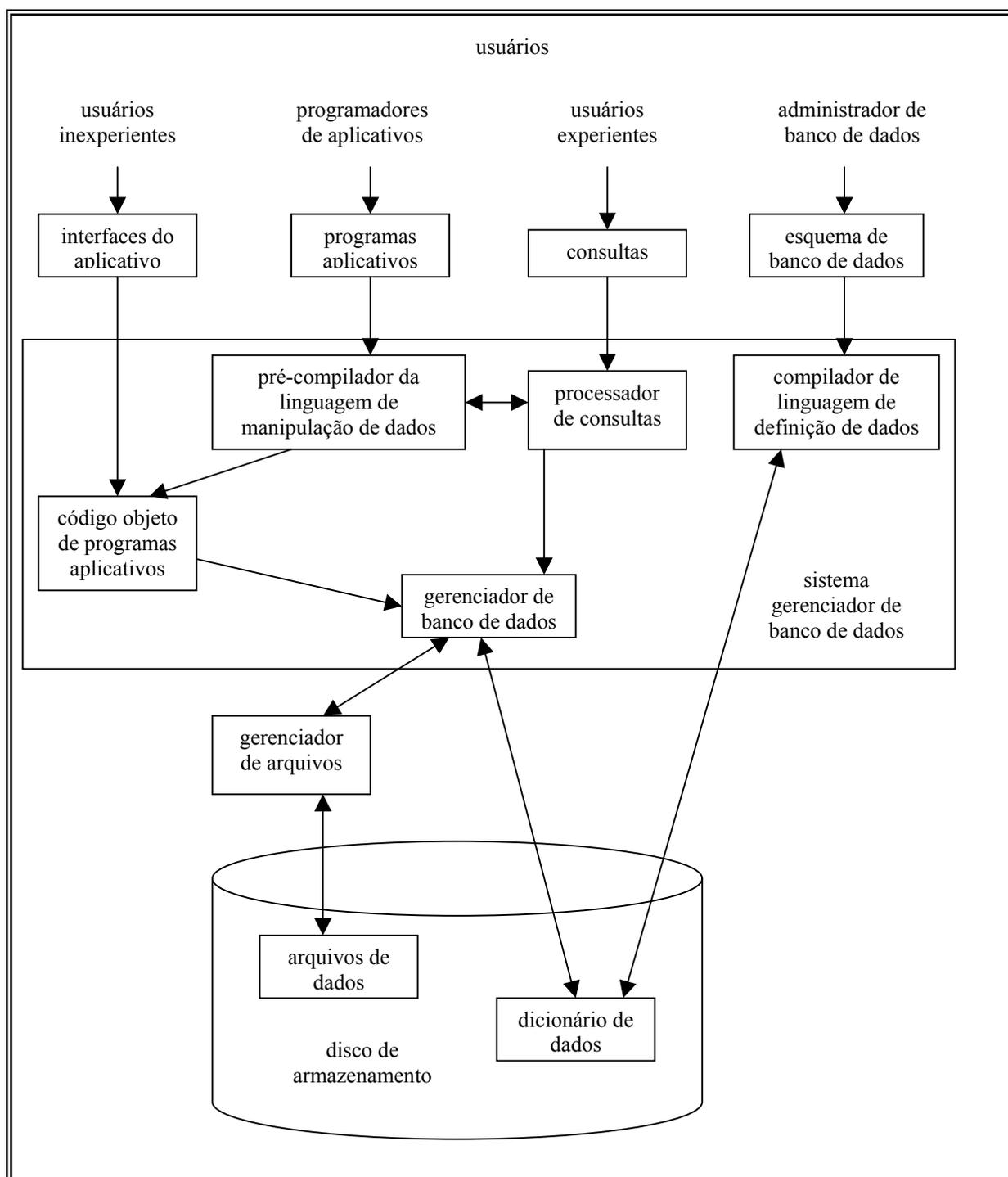


FIGURA 5.3 - Estrutura de um sistema gerenciador de banco de dados

As funções que compõem uma API da BDE são usadas internamente pelos componentes de acesso a dados do Delphi e muito raramente são utilizadas diretamente via programação, mas isso é totalmente possível. O acesso e manipulação de um banco

de dados por um programa Delphi é realizado como mostra a figura 5.4. A aplicação não acessa os dados diretamente, mas utiliza sempre a BDE e, para tanto, são necessários os componentes de acesso e os controles visuais que fazem parte do Delphi. Os componentes de acesso são utilizados para acessar banco de dados, enquanto que os controles visuais são utilizados para visualizar as informações contidas nesse banco de dados.

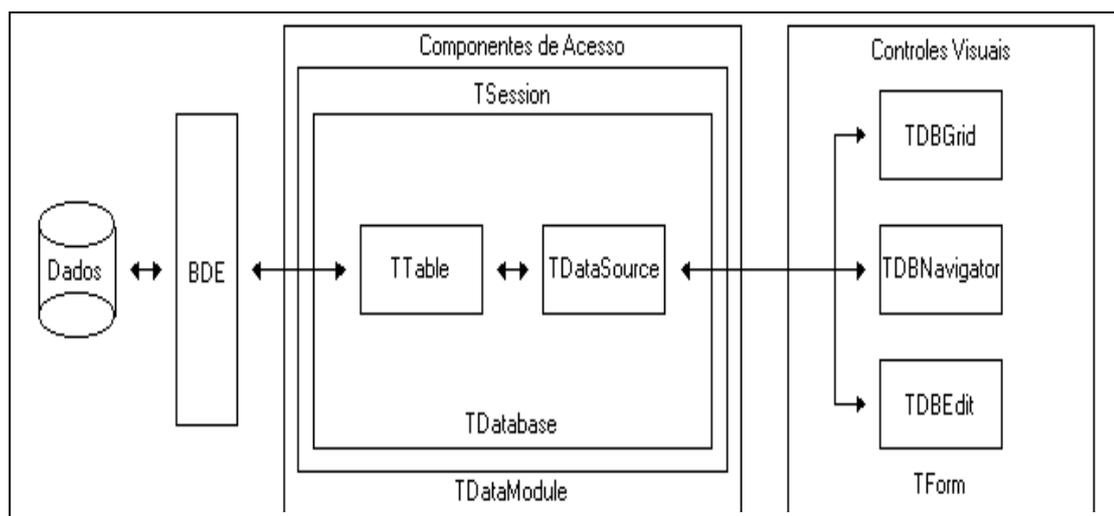


FIGURA 5.4 - Acesso a banco de dados em Delphi usando o BDE

O componente de acesso *TDatabase* permite que seja escolhido um banco de dados e o *TTable* faz o acesso às tabelas desse banco de dados escolhido. O componente *TDataSource* é utilizado como uma interface entre os componentes de acesso e os componentes visuais. Os componentes visuais *TDBGrid* (componente utilizado para mostrar os dados em uma forma tabular), *TDBNavigator* (componente utilizado para navegação entre os registros do banco de dados), *TDBEdit* (componente utilizado para exibição ou entrada de informações em formulários) devem estar associados a um *TDataSource* para que as informações do banco de dados sejam exibidas. Assim, para uma aplicação de bancos de dados funcionar, é preciso que a BDE esteja instalada na máquina, não bastando apenas o arquivo executável.

Na figura 5.5 podemos ver a estrutura interna do BDE.

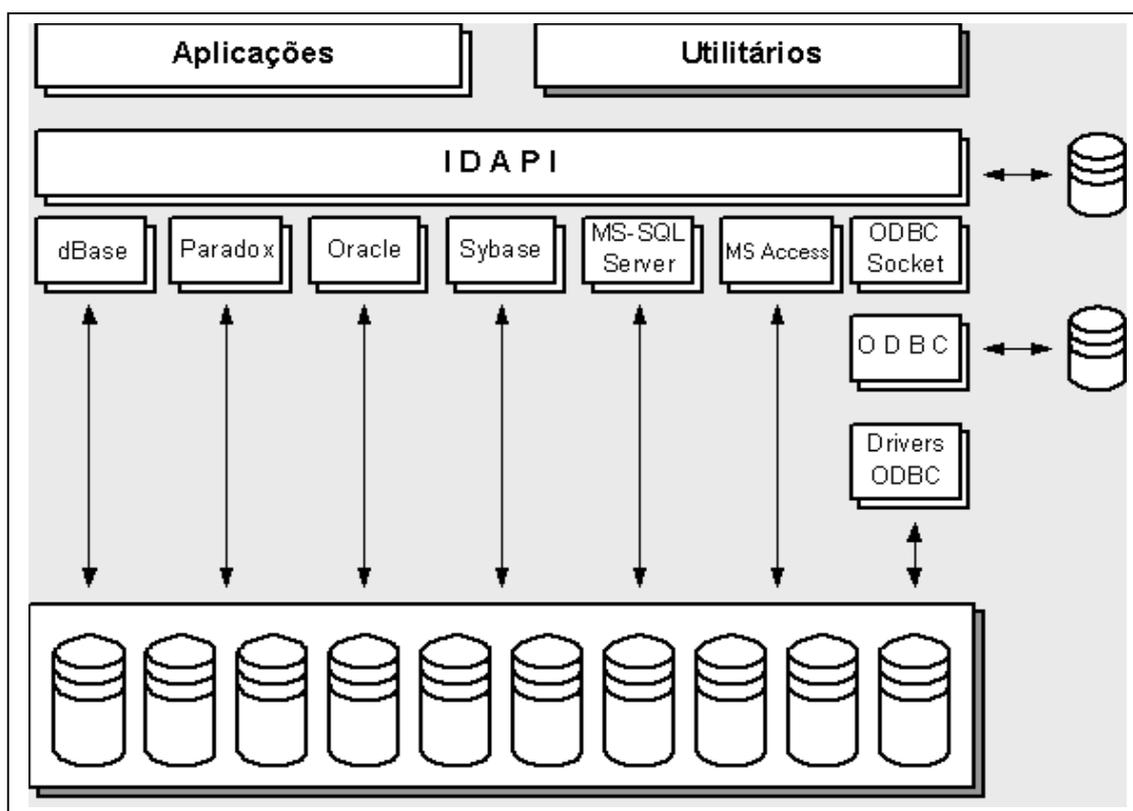


FIGURA 5.5 - Estrutura interna do Borland Database Engine

5.2.2 Ícones utilizados

Os ícones que são utilizados no protótipo, os quais podem ser vistos na figura 5.6, são descritos por um arquivo texto que é um subconjunto daquele proposto no capítulo 4. Os ícones utilizados são baseados nas primitivas básicas *SQUARE* e *CIRCLE* podendo ou não ser divididos e ter marcas.

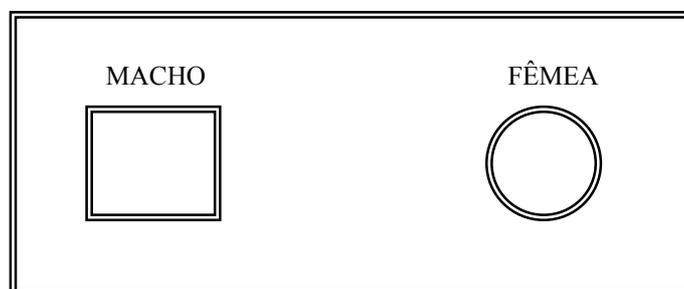


FIGURA 5.6 - Ícones utilizados no protótipo

Cabe salientar que a escolha dessas formas para os ícones para representação de animais machos e fêmeas é baseada em notação clássica usada por biólogos na representação do gênero em diagramas.

O arquivo descritor de ícones utilizado como entrada no *IconVis* é apresentado a seguir:

```
! ICONS 2
PRIM = SQUARE
BASE = 16
ALTU = 16
DIVI = N
STAM = N
PRIM = CIRCLE
BASE = 16
ALTU = 16
DIVI = N
STAM = N
ICONS!
```

Esse arquivo define dois tipos de ícones que serão utilizados pelo programa. Um da forma *SQUARE*, utilizado para representar os indivíduos machos, de tamanho inicial 16x16 e outro da forma *CIRCLE*, utilizado para representar os indivíduos fêmeas, também de tamanho inicial 16x16. Ambos inicialmente não são divididos nem possuem marca (marcações *DIVI* e *MARC*). Os ícones terão a aparência inicial conforme a figura 5.6. Posteriormente, através da utilização do mapeador de ícones, essas características dos ícones podem ser alteradas pelo usuário para que mais informações possam ser agregadas às entidades gráficas.

5.2.3 Técnica de visualização adotada

A visualização de todos os animais registrados no banco de dados através de ícones em uma única tela fica bastante prejudicada. O principal fator que agrava a visualização dos ícones é a sobreposição de ícones.

Essa sobreposição é ocasionada pelo fato de que muitos animais são capturados em posições relativamente muito próximas umas das outras. Dessa forma, foi criado um *universo* e uma *janela* atuando sobre esse *universo*. Com um universo e uma janela de seleção definidos, somente os animais que estiverem contidos dentro dessa janela de seleção serão visualizados num dado momento. Para observar os ícones que representam os demais animais basta deslocar essa janela de seleção sobre o universo de maneira a colocá-la em uma posição que inclua os animais objetos do estudo.

A vantagem da janela de seleção é que pode-se fazer uma análise de uma determinada região onde estão concentrados alguns animais. O problema da sobreposição dos ícones também fica diminuído, pois ao se aplicar a janela de seleção, automaticamente estamos realizando uma operação de *zoom* sobre aquela determinada região de interesse. A única restrição é que não é possível visualizar todos os animais em uma única tela, até por que a tela ficaria bastante “poluída” e confusa de se entender.

A figura 5.7 representa a região de pesquisa dos biólogos. Ela é muito semelhante à figura 5.1. Como explicado anteriormente, as linhas retas representam a cerca existente no local utilizada como referencial para a captura dos animais. As bandeiras da figura, identificadas por letras: *A*, *B*, *C*, *D*, *E*, *F* e *G*, são também utilizadas como pontos de referência. E as pequenas formas circulares representam os animais capturados pelos biólogos. Ainda nessa figura, temos um pequeno retângulo entre os pontos de referência *A* e *B*. Esse retângulo representa a janela de seleção sobre o universo de pesquisa.

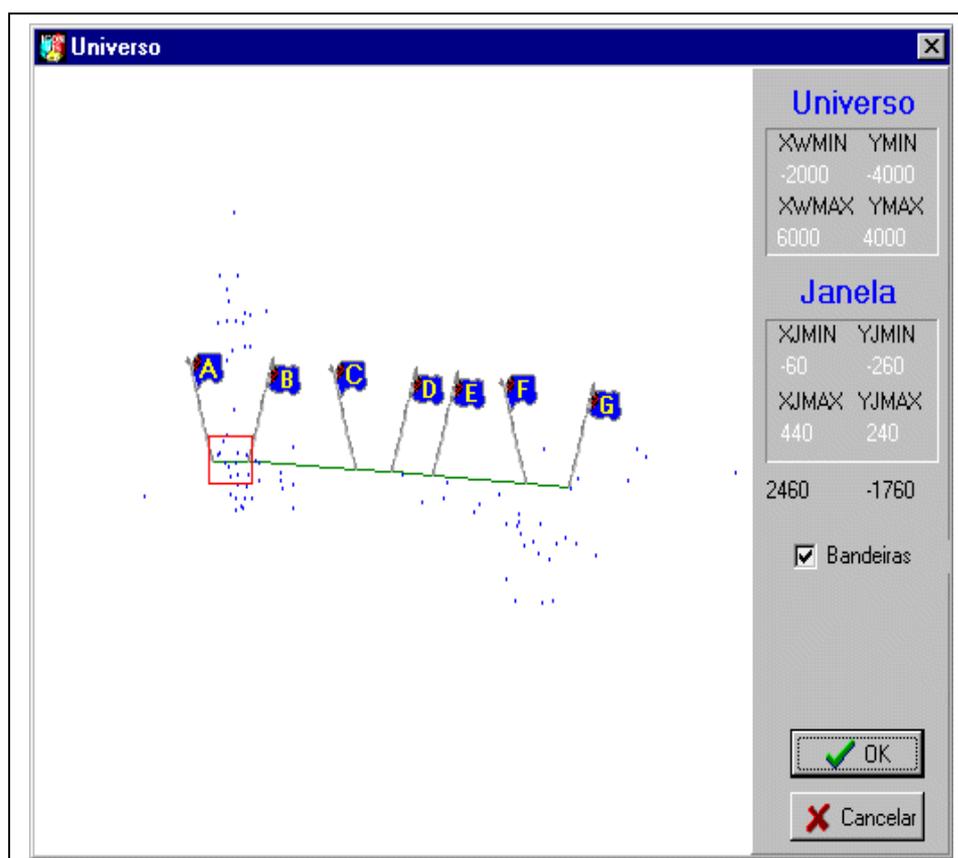


FIGURA 5.7 - Tela mostrando a região de pesquisa dos biólogos

A figura 5.8 abaixo mostra a visualização de todos os animais capturados que estão dentro da janela de seleção atual.

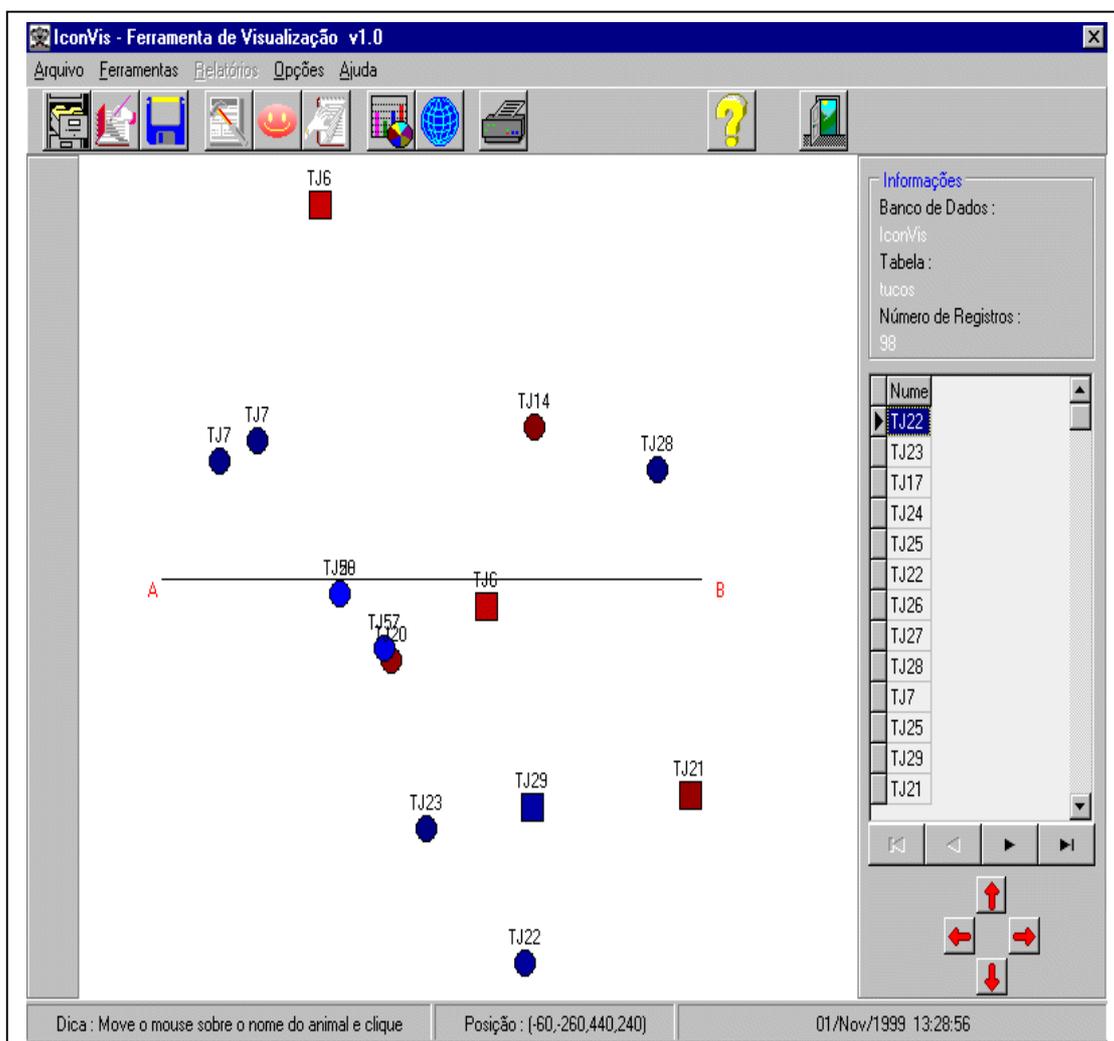


FIGURA 5.8 - Visualização de uma parte selecionada do universo

Para visualizar outras regiões, basta acessar a janela que representa a região de estudo dos biólogos e deslocar com o mouse a janela de seleção para a posição desejada. Ao fazer isso e clicando-se no botão *OK*, a tela principal é automaticamente atualizada para mostrar somente os animais contidos dentro da nova região de interesse.

Outra forma de se obter o mesmo efeito é através do uso das setas direcionais da tela principal. Ao clicarmos nas setas com direção *para cima*, *para baixo*, *para esquerda* e *para direita*, haverá deslocamento da janela de seleção no universo para a direção correspondente. Essa opção é menos versátil, pois são necessários vários cliques do mouse para deslocar a janela de seleção para outra posição. Assim esse método é

mais indicado para deslocamentos curtos, enquanto que o outro método é mais utilizado para deslocamentos maiores.

Apesar da simplicidade dessa técnica de obtenção de *zoom*, ela foi escolhida em detrimento da qualquer técnica *fish-eye* porque julgou-se inconveniente a introdução de técnica que incluísse distorção da imagem final. Na realidade o que os biólogos necessitam é de uma técnica que lhes permita observar a relação espacial entre os diversos pontos de coleta.

5.2.4 Módulos e funcionamento do protótipo

O programa está dividido em vários módulos distintos mas complementares. A seguir veremos cada um desses módulos em maior detalhe. O *IconVis* pode ser dividido nos seguintes módulos :

- *pré-processamento*: necessário para adaptar os dados sobre os tucos para o programa
- *cadastro*: permite que novas informações sobre os animais sejam inseridas, apagadas ou modificadas
- *construtor de consultas*: possibilita ao usuário criar consultas sobre o banco de dados de forma interativa e simples, retornando apenas as informações desejadas, reduzindo assim o volume de dados
- *mapeador de ícones*: o usuário tem a sua disposição um conjunto de ícones e parâmetros que podem ser associados às informações que serão visualizadas
- *visualização*: responsável pela definição do universo de visualização dos animais.

5.2.4.1 Pré-processamento

Essa fase é necessária devido a natureza dos dados contidos no banco de dados com as informações sobre os animais capturados. Como não existe uma informação direta sobre o ponto de captura dos animais (conforme explicação na seção 1.4), é necessária uma fase preliminar para a determinação desses pontos. No banco de dados existe apenas a informação sobre os ângulos relativos aos pontos de referência para a determinação do cálculo da posição de captura do animal. Por esse motivo, a fase de pré-processamento é não só importante, mas necessária. Aproveitando a necessidade dessa fase, esse módulo também faz diversos cálculos adicionais que serão utilizados posteriormente na visualização dos dados. A seguir segue uma breve descrição do que é feito nessa fase de pré-processamento.

Etapas seguidas na fase de pré-processamento :

1) *Verificação dos dados* : é feita uma consistência dos dados de entrada. Isso é necessário para que aqueles dados que não estiverem de acordo com o esperado não sejam visualizados erroneamente. Nesse momento verifica-se se todas as informações para um determinado animal no banco de dados estão completas. Verifica-se, por exemplo, se existe intersecção entre os valores obtidos a partir dos ângulos fornecidos, se não existem números negativos para o peso e comprimento do animal, verificação de ausência de informação, verificação da consistência de dados em relação ao seu tipo (no peso por exemplo, só são permitidos valores numéricos).

2) *Cálculos de estatísticas*: as informações da média de alguns dados são de importância para os biólogos bem como para a visualização dos dados. Então, são calculados por exemplo, o peso mínimo, máximo e médio dos animais (com distinção para indivíduos machos e fêmeas). De forma análoga, o mesmo é feito para o comprimento total dos animais. Em função dos resultados calculados, novas informações podem ser obtidas que não estão contidas diretamente no banco de dados e que são de suma importância para a visualização dos animais. Por exemplo, a partir do peso do animal podemos determinar a sua idade, cujo cálculo é explicado logo a seguir.

Todas essas informações calculadas são colocadas em uma tabela temporária auxiliar que serão utilizadas no momento da visualização. Toda vez que o programa é carregado ou quando um novo animal é cadastrado no banco de dados esse fase de pré-processamento é executada novamente.

3) *Classificação*: os animais podem ser enquadrados em quatro grupos distintos com relação a idade: *jovem*, *subadulto fêmea*, *subadulto macho* e *adulto*. O seguinte método é utilizado para a determinação da faixa etária dos animais :

a) Limite superior para jovens = a média entre o maior peso de uma fêmea e o menor peso de todos os animais.

b) Limite superior para subadulto fêmea = menor peso de fêmea perfurada/cicatrizada.

c) Limite superior para subadulto macho = média entre o menor e o maior peso encontrado.

d) Limite inferior dos adultos = limite superior do subadulto do mesmo sexo.

4) *Cálculo*: como no banco de dados não existe a informação de local de captura dos animais diretamente, é necessário um procedimento preliminar para obter esse ponto de captura. De posse dessa informação poderemos plotar os ícones. Para a

determinação do local de captura de um determinado animal iremos usar a figura 5.1 (representando a região de estudos dos biólogos em Itapuã) como base.

A distância do ponto A ao ponto B corresponde a 53,8 metros. A distância do ponto C ao ponto D corresponde a 54,6 metros. A distância do ponto D ao ponto E corresponde a 51,5 metros. A distância do ponto E ao ponto F corresponde a 135,82 metros. E a distância do ponto F ao ponto G corresponde a 53,72 metros. Os pontos A e B estão contidos na mesma reta, que no desenho corresponde a reta $r1$. Já os pontos C, D, E, F e G estão contidos em outra reta denominada de $r2$.

Existe um ponto adicional, denominado de B' , que fica distante do ponto B 15,9 metros, e é utilizado como uma referência para determinação do ponto C. O ângulo α formado entre a reta $r1$ (ligando os pontos A-B) e a reta “imaginária” que liga o ponto A ao ponto C é de -3 graus. O ângulo β formado entre a reta $r1$ e a reta “imaginária” que liga o ponto B ao ponto C é de 184 graus. Todas essas informações são necessárias para o cálculo que vai ser desenvolvido para a determinação desses pontos para utilização no protótipo.

O procedimento de localização do ponto onde cada roedor é capturado é função dos ângulos de duas retas imaginárias tomadas com bússola a partir do ponto de coleta e dos pontos A e B, que são utilizados como exemplo na figura 5.1. Esse exemplo mostra como seria o cálculo de um ponto de captura de um animal partindo dos pontos A e B da cerca. O procedimento é análogo para os demais pontos da cerca.

A figura 5.9 ilustra esse procedimento.

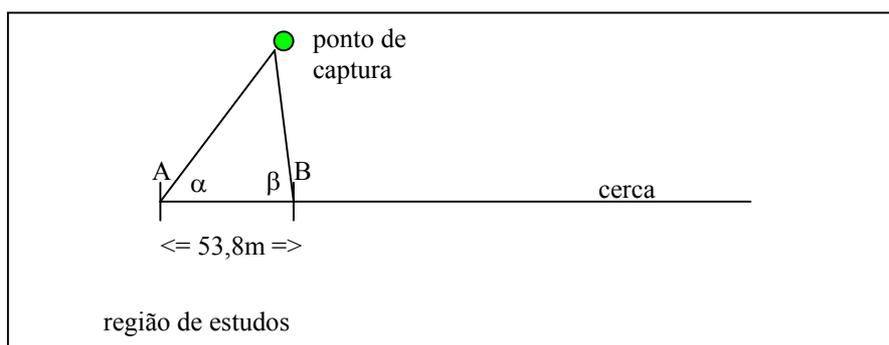


FIGURA 5.9 - Esquema para localização do ponto de coleta

O ponto de captura do animal identificado na figura 5.8 é determinado da seguinte forma: um biólogo marca esse ponto de captura, enquanto que um segundo biólogo se desloca para o ponto A da cerca. No ponto A, com o auxílio de um bússola, é determinado o ângulo α que é formado entre a reta que contém os pontos A e B e a reta “imaginária” que vai do ponto A até o ponto de captura. Da mesma forma, o biólogo se

desloca para o ponto B da cerca, e o ângulo β é determinado de forma análoga.. O *ponto de intersecção* dessas duas retas é o ponto $P(x,y)$, correspondente ao ponto de coleta. Maiores detalhes sobre o o cálculo do ponto de intersecção implementado no protótipo pode ser encontrado no Anexo A.

5.2.4.2 Cadastro

Esse módulo foi desenvolvido com o objetivo de ser um facilitador para os biólogos. Através de uma interface simples (figura 5.10), os biólogos podem fazer alterações sobre as informações do banco de dados. Basicamente o que é mantido é um cadastro de todos os animais capturados por eles.

As principais funções desse módulo são:

- *cadastro*: novos animais podem ser adicionados ao banco de dados
- *alteração*: as informações já existentes no banco de dados podem ser alteradas a qualquer momento
- *exclusão*: pode-se excluir um determinado animal
- *pesquisa*: através da interface pode-se percorrer o banco de dados registro a registro

5.2.4.3 O construtor de consultas interativo

Para auxiliar o trabalho dos usuários do *IconVis* foi desenvolvido um módulo destinado a executar consultas no banco de dados de pesquisa. Com essa ferramenta, é possível escolher um subconjunto dos dados para fazer a visualização. Através da utilização de uma interface bastante simples e intuitiva, o usuário com o mínimo de dificuldade consegue realizar consultas sobre o banco de dados. Essas consultas podem ser simples ou complexas dependendo da necessidade do usuário.

Após a execução da consulta, somente os dados retornados por ela estarão disponíveis para a visualização. Caso o usuário necessite ter acesso a todos os registros do banco de dados novamente, basta entrar no construtor de consultas e realizar uma consulta que retorne todas as informações disponíveis no banco de dados.

Manutenção do cadastro dos animais

Identificação

Número Sexo Característica Sexual

Tatuagem

Recaptura 1

Recaptura 2

Localização

Data Hora

Ponto A Ponto C Ponto E Ponto G

Ponto B Ponto D Ponto F

Medidas

Comprimento Total

Comprimento Cauda

Largura Dente

Pata Com Unha

Pata Sem Unha

Comentários

Observações

FIGURA 5.10 - Tela de cadastro de novos animais

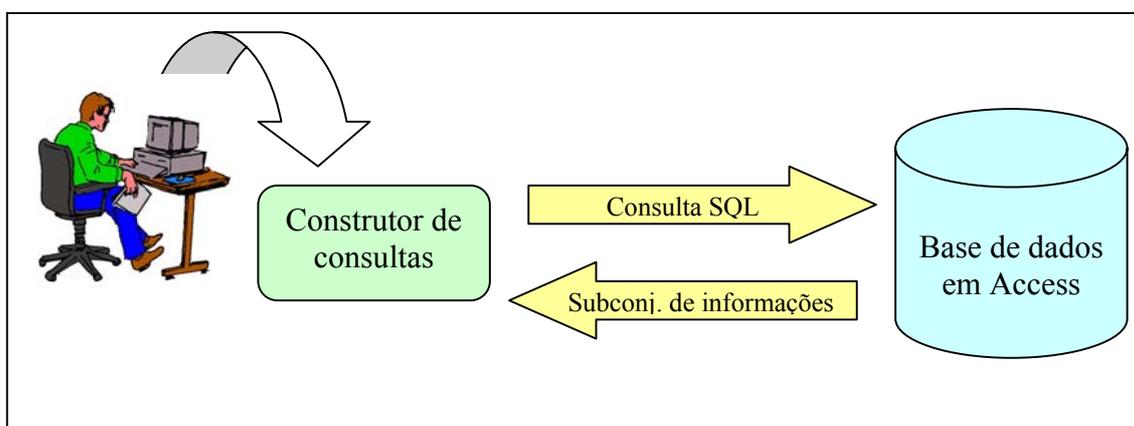


FIGURA 5.11 - Construtor de consultas

À medida que o usuário vai criando a sua consulta, o módulo construtor de consultas vai montando um cláusula SQL para ser enviada ao banco de dados. Quando o usuário clica no botão *Executar*, essa cláusula SQL é executada, ou seja, é enviada através do BDE (explicado anteriormente) diretamente ao banco de dados. A consulta é então processada, e os dados resultantes dessa consulta serão enviados pelo banco de dados através do BDE de volta ao programa.

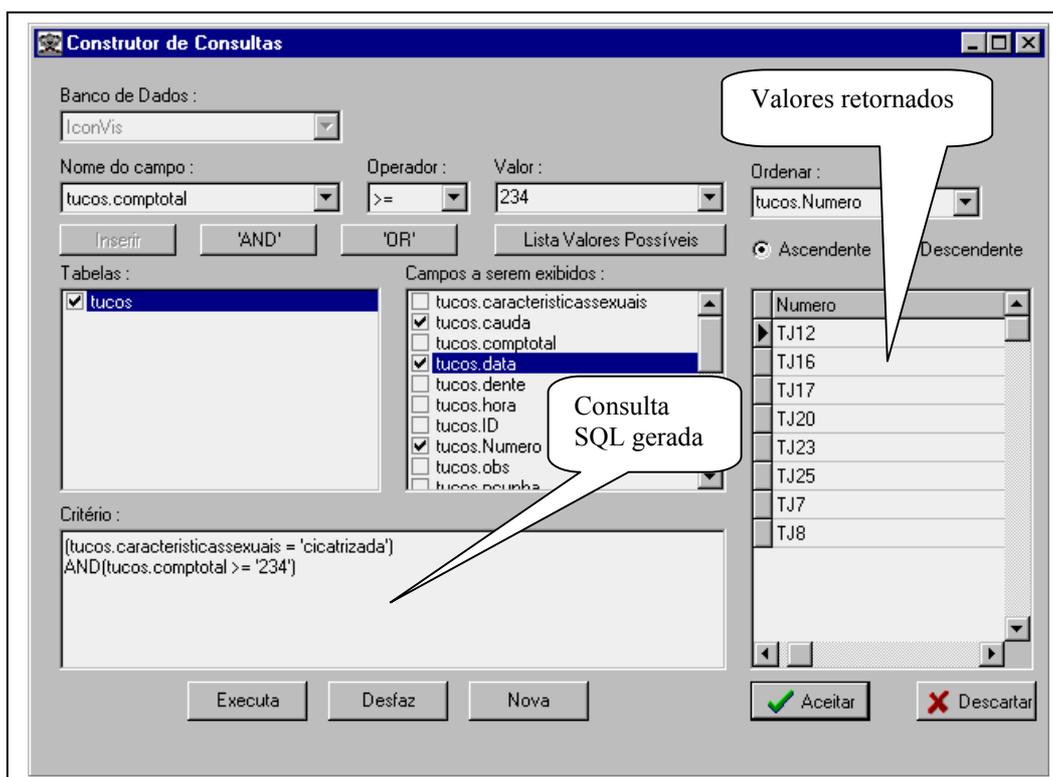


FIGURA 5.12 - Tela do construtor de consultas

Na lista de seleção denominada de *Banco de Dados*, podemos escolher qualquer banco de dados presentes no sistema. Vale ressaltar que o módulo de construtor de consultas é totalmente genérico, podendo ser utilizado para criar consultas em qualquer banco de dados. Depois de termos escolhido o banco de dados, temos que escolher a tabela sobre a qual queremos realizar a consulta. Para isso basta marcar a *checkbox* corresponde à tabela desejada. Feita a escolha a lista de seleção dos nomes dos campos se torna ativa e podemos então escolher o campo da tabela que desejamos fazer a consulta. A utilização do *BDE* contribuiu para que esse módulo ficasse totalmente genérico, podendo ser facilmente adaptado em outros sistemas. Através dos componentes do *BDE*, é possível ler qualquer banco de dados que o usuário desejar (componente *TDatabase*). Depois de ler a estrutura do banco de dados, através do uso do componente *TTable* é possível ler todas as tabelas e os campos correspondentes que estão registradas nesse banco de dados escolhido. Através dos componentes *TDataSource* e *TDBGrid* é possível criar uma consulta *SQL* para ser executada nesse

banco de dados, com os resultados retornando para o componente *TDBGrid*. Quando o usuário estiver satisfeito com os valores retornados por sua consulta, clicando o botão *OK*, esses valores serão passados a aplicação principal (no caso do *IconVis*, para o módulo de visualização).

Pode-se escolher um operador para limitar o número de registros retornados pela consulta. À medida que vai se clicando nos botões denominados de *Inserir*, *And* e *Or*, o usuário vai construindo a sua consulta. Existe uma opção de fazer o ordenamento dos registros retornados. Nessa opção pode-se escolher o campo que se deseja ordenar, bem como se essa ordenação será de forma ascendente ou descendente de acordo com os valores do campo da tabela.

5.2.4.4 Mapeador de ícones

Esse é o módulo responsável por fazer o mapeamento dos dados do banco de dados (*características*) para os parâmetros dos ícones. A interface com o usuário para fazer o mapeamento pode ser vista na figura 5.13. O usuário têm a liberdade de escolher quais informações deseja mapear para quais parâmetros dos ícones. Após ter sido feita a escolha, a visualização é automaticamente refeita para contemplar as novas configurações. Essas configurações permanecerão como padrões para as novas representações visuais.

Através da interface, o usuário poderá mudar os parâmetros *dividido* e *marcação*. Na figura 5.13 temos um exemplo do mapeador de ícones em execução. Nesse exemplo, o ícone em questão é o círculo, utilizado para representar os animais fêmeas. O ícone original está sendo dividido segundo o *padrão 3*, ou seja, o ícone terá divisões. Foram escolhidas as *características* idade do animal, peso e comprimento total para ser mapeado para este ícone. Como pode ser visto na figura, a divisão 1 está sendo utilizada para mapear a idade do animal. Cores serão utilizadas para diferenciar as idades dos animais. A cor 1 (verde) é utilizada para representar os animais jovens. A cor 2 (amarelo) é utilizada para representar os animais subadultos machos e a cor 3 (marrom) é utilizada para representar os animais adultos. A divisão 2 é utilizada para representar o peso dos animais. Como essa é uma característica com valores escalares, o usuário pode escolher uma cor para o menor peso (o padrão é o azul) e outra cor para representar o maior peso (o padrão é vermelho). Os valores de peso intermediários serão mapeados para cores entre a cor representando o menor peso e a cor representado o maior peso através do uso de interpolação entre essas duas cores. A divisão 3 é usada para representar o comprimento total dos animais. Da mesma forma que para representar o peso, o usuário pode escolher duas cores para representar o menor e o maior comprimento total e os valores intermediários serão mapeados segundo a

interpolação dessas cores. O uso das marcas só faz sentido para características cujos domínios de valores seja enumerado, como, por exemplo, a idade ou característica sexual. Como pode ser visto na figura 5.13, as marcas estão sendo utilizadas para representar a característica sexual dos animais (essa característica só faz sentido para os indivíduos fêmeas representados pelo ícone na forma de círculo). A marca 1 representa a fêmea imperfurada, as perfuradas são representadas pela marca 2, a marca 3 representa as fêmeas perfuradas recentemente, e marca 4 a cicatrizada com secreção e marca 5 a cicatrizada. Com o mapeamento acima, conseguimos visualizar diversas características dos animais ao mesmo tempo.

FIGURA 5.13 - Tela do mapeador de ícones



FIGURA 5.14 - Escala de cores

A seguir seguem as definições de estruturas de dados utilizadas no *IconVis* (são mostradas apenas as mais importantes). Primeiramente são definidas algumas constantes que são utilizadas para representar as características principais do banco de dados e que são utilizadas pelo mapeador de ícones. Logo a seguir segue a definição da estrutura *stcIcône* que é utilizado para armazenar as informações que foram lidas do arquivo descritor de ícones. O campo *strLabel* é utilizado para guardar o nome do ícone que será exibido a tela. O campo *bytForma* guarda a informação sobre qual primitiva o ícone é baseado. Os campos *intPosX* e *intPosY* indicam onde o ícone será plotado na tela. Os campos *intAltura* e *intBase* correspondem ao tamanho que o ícone terá. Os campos *cor* são utilizados para guardar a informação de qual cor será utilizada para preencher o ícone. O campo *bytMarca* e *bytDividido* são utilizados para indicar se o ícone terá marca e se será dividido, respectivamente. A estrutura *stcMapea* é utilizada para armazenar as informações sobre o mapeamento propriamente dito. Por exemplo, o campo *strFundo* é utilizado para guardar a característica que foi atribuída ao fundo do ícone. É criada uma variável dinâmica denominada *ícones* cujo tamanho desse *array* dinâmico será definido no momento da leitura do arquivo descritor de ícones. O número de elementos desse array será igual ao número de ícones definidos no arquivo descritor. No rotina de leitura do arquivo, esse array será preenchido com os valores definidos no arquivo descritor. As modificações realizadas no mapeador de ícones serão refletidas para essas estruturas de dados.

```
const
    NENHUM                = 0;
    PESO                  = 1;
    COMPRIMENTO_TOTAL    = 2;
    PERFURADA             = 3;
    PERFURADA_RECENTE    = 4;
    IMPERFURADA          = 5;
    CICATRIZADA_SECRECAO = 6;
    CICATRIZADA          = 7;
    JOVEM                 = 8;
    SUBADULTO_FEMEA      = 9;
    SUBADULTO_MACHO      = 10;
    ADULTO                = 11;
    RECAPTURA_1         = 12;
    RECAPTURA_2         = 13;

type
    stcIcône = record
        strLabel      : string;
```

```

    bytForma      : byte;
    intPosX      : integer;
    intPosY      : integer;
    intBase      : integer;
    intAltura    : integer;
    bytDividido  : byte;
    CorLinha     : TColor;
    CorFundo     : TColor;
    Cor1         : TColor;
    Cor2         : TColor;
    Cor3         : TColor;
    Cor4         : TColor;
    bytMarca     : byte;
    bytPosMarca  : byte;
    Mapea       : stcMapea;
end;

```

```

type
    stcMapea = record
        fundo : string;
        tamanho : string;
        cor1 : string;
        cor2 : string;
        cor3 : string;
        cor4 : string;
        marca1 : string;
        marca2 : string;
        marca3 : string;
        marca4 : string;
        marca5 : string;
    end;

```

```

var icones : array of stcIcône;

```

5.2.4.5 Visualização

A visualização das informações corresponde à tela principal do protótipo. Após terem sido executadas consultas sobre a base de dados usando o construtor interativo de consultas e realizar o mapeamento de ícones utilizando o mapeador de ícones, o

protótipo irá usando essas informações passadas pelo usuário realizar a visualização. Se o usuário não tiver executado nenhuma consulta ou não tiver feito nenhum mapeamento, o protótipo assume a sua configuração padrão e irá exibir todos os animais da base de dados usando o mapeamento padrão de ícones, que é o fundo do ícone representar o peso do animal, usando a escala de cores da figura 5.14.

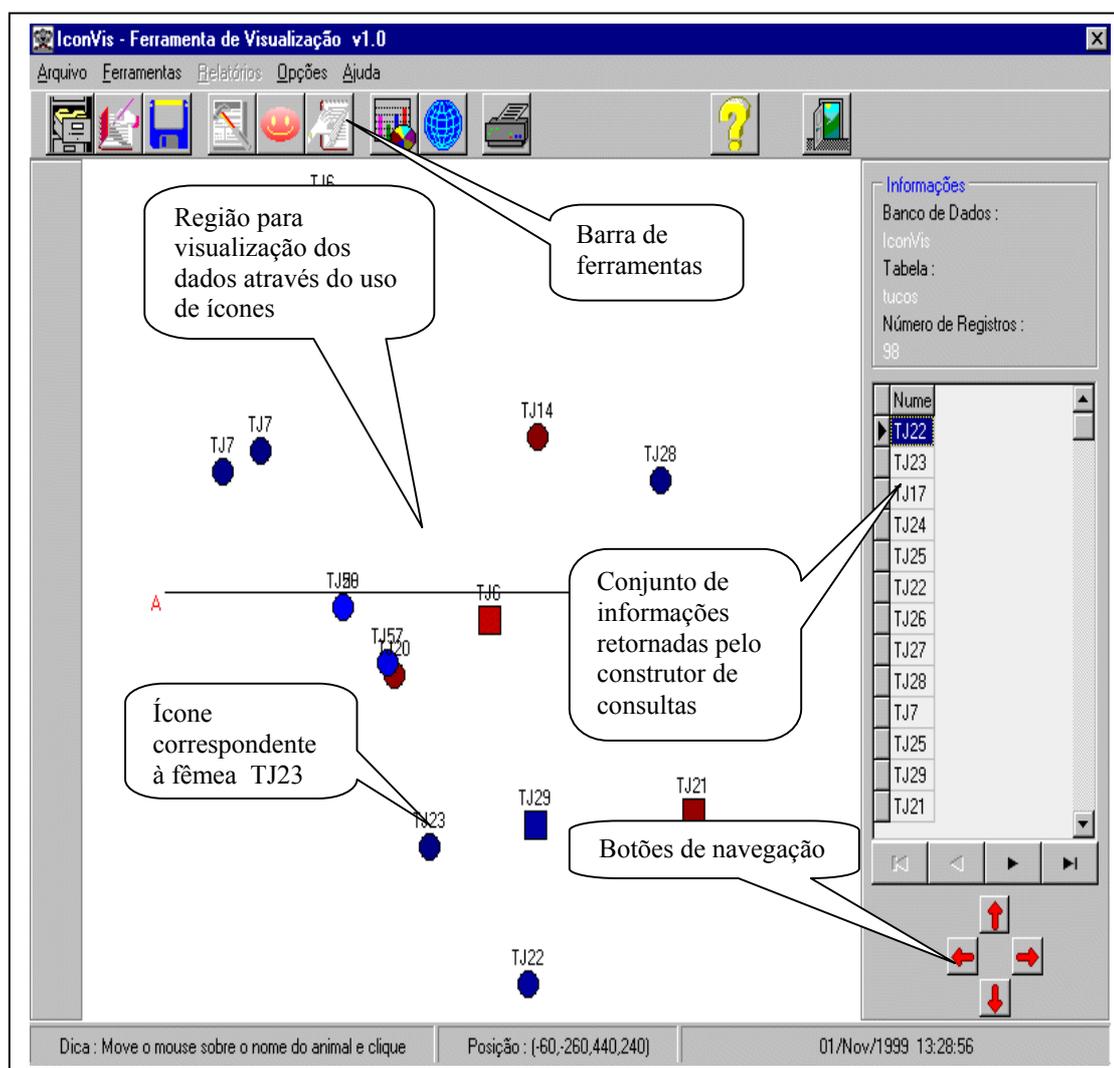


FIGURA 5.15 - Detalhamento da tela principal do *IconVis*

O usuário pode navegar através da utilização do *mouse*, selecionando se quer ou não visualizar os rótulos dos ícones. Essa opção é útil quando ícones se encontram muito próximos uns dos outros. Assim, retirando os rótulos dos ícones menos informação é apresentada tornando a visualização mais clara e “limpa”. Ainda assim o usuário terá a condição de identificar cada um dos animais, mesmo sem os rótulos estáticos: para isso basta deslocar o *mouse* sobre um dos ícones e um rótulo dinâmico será exibido por alguns segundos mostrando a identificação do animal representado por esse ícone.

Ao clicar sobre um determinado ícone, uma nova janela se abre (figura 5.16) mostrando maiores informações sobre a entidade.

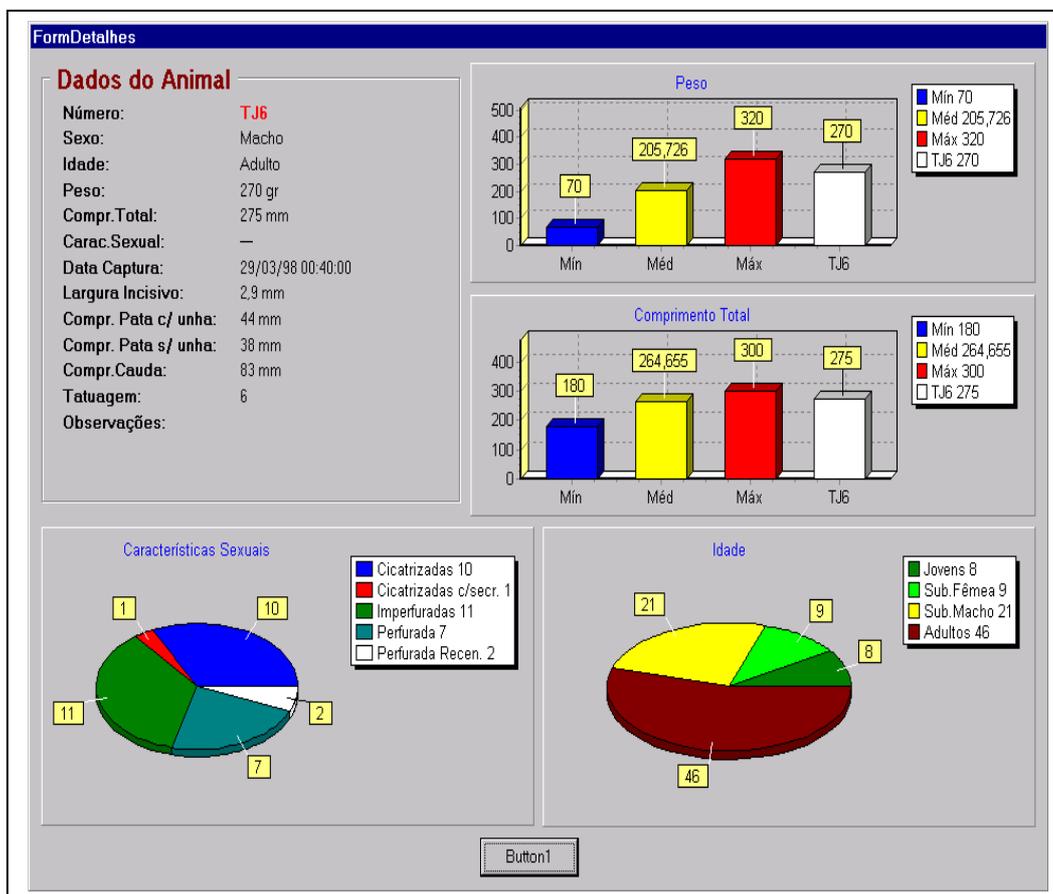


FIGURA 5.16 - Detalhes dos animais

Do lado esquerdo da figura temos um detalhamento completo do animal selecionado. Do lado direito da figura temos gráficos comparando o *peso* e o *comprimento total* desse animal com os demais. Na parte inferior da figura temos algumas estatísticas que foram calculadas em função dos dados disponíveis na base de dados. São apresentados os valores para os atributos *característica sexual* e *idade*. Essas informações, assim como as anteriores servem apenas para comparações entre o animal que foi selecionado e os valores totais de todos os animais. Na figura 5.17 abaixo temos o resultado de uma visualização baseado no mapeamento informado na figura 5.13.

Apenas para ressaltar, os animais machos estão com um mapeamento bem simples: sem divisões e o fundo do ícone representando para o peso. Assim notamos que o animal *TJ26* é mais pesado que o *TJ73* (usando azul para valores menores e vermelho para valores maiores). Os animais fêmeas só não estão utilizando a divisão 4

que na figura aparece em branco. Os animais *TJ22*, *TJ23*, *TJ25* são fêmeas subadultas. Note que temos duas incidências do animal *TJ22*. Isso significa que o animal foi recapturado. Na primeira vez o animal era jovem e quando foi recapturado já era subadulto como podemos verificar na figura. O animal *TJ24* é uma fêmea adulta. A divisão 2 está mapeada para o peso e a divisão 3 para o comprimento total. Ambas utilizam o azul para valores menores e o vermelho para valores maiores. As marcas indicam a característica sexual das fêmeas. Notamos que o animal *TJ25* foi recapturado também. Na primeira vez era uma fêmea imperfurada e na segunda já era uma fêmea perfurada recentemente

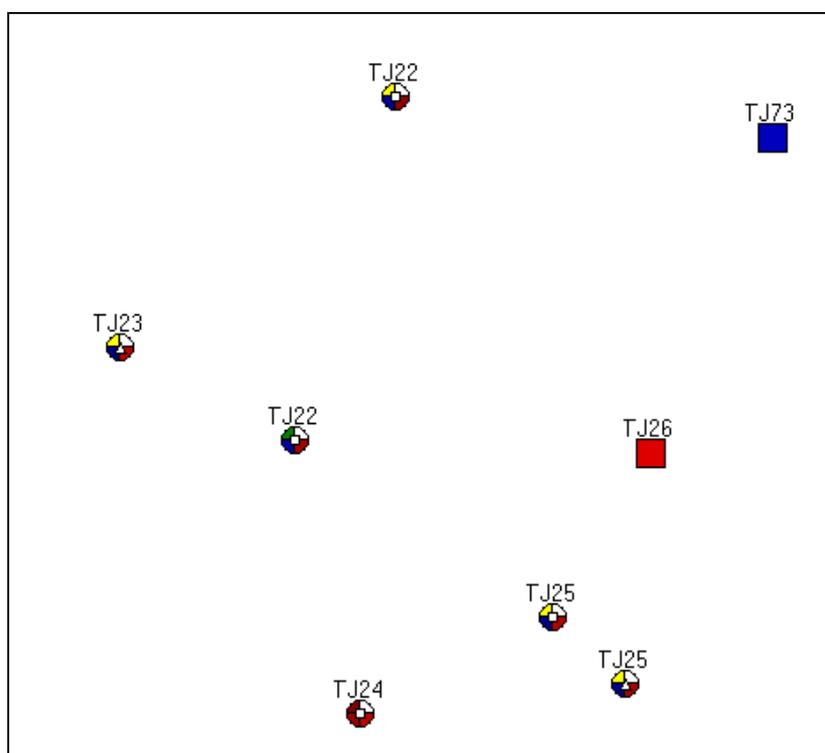


FIGURA 5.17 - Visualização após o mapeamento

6 Avaliação

Técnicas de visualização são úteis na medida em que sintetizam dados e informações, facilitando o processo de análise que, se realizado sobre um grande conjunto de dados, seria difícil e, eventualmente, pouco conclusivo. Quando são buscadas relações, baseadas em atributos, o processo de visualização pode revelar padrões auxiliando sobretudo a inferência de novas informações e a construção do conhecimento. Entretanto, a escolha da técnica de visualização a ser empregada em cada situação não é tarefa fácil, principalmente se os fatos que se quer descobrir não são conhecidos.

Este trabalho dedicou-se especificamente à visualização icônica, recomendada quando se necessita visualizar um conjunto de entidades (localizadas espacialmente ou não) e seus atributos [FRE 93]. Uma linguagem de descrição de ícones foi proposta e um subconjunto dela foi implementado num protótipo de modo que se pudesse avaliar o emprego de ícones numa situação prática.

A criação de uma **linguagem simples** para atuar em conjunto com a ferramenta de visualização é o ideal. Essa linguagem é utilizada para gerar ícones automaticamente. Na maioria dos sistemas que vimos anteriormente, existe um conjunto fixo de ícones disponível ao usuário na forma de uma biblioteca. Neste caso, o sistema foi planejado para usar somente aqueles ícones ou glifos, possuindo apenas as rotinas para desenhar esses objetos. Isso faz com que a atividade do usuário fique bastante limitada, pois muitas vezes um determinado ícone ou glifo não é adequado para visualizar certos tipos de informações, mas como na biblioteca só existem esses tipos definidos, o usuário se vê obrigado a usar essa representação visual mesmo que não seja a mais adequada. A proposta da criação de uma linguagem simples, descritiva e textual faz com que o usuário possa criar seus próprios ícones.

Os ícones são gerados em um arquivo texto, obedecendo a sintaxe da linguagem proposta, e esse arquivo texto serve de entrada para o programa de visualização. Assim, dependendo da aplicação do usuário, ele cria os ícones de acordo com sua vontade usando essa linguagem e submete os novos ícones para o programa de visualização, que irá interpretá-los e desenhá-los. Ao invés de gerar esses arquivos texto manualmente, poderia ser criada uma ferramenta visual onde o usuário criasse o ícone passo a passo., tendo um *feedback* visual de como o seu ícone ficará.

Vários **tipos de dados** podem ser mapeados para ícones. No protótipo, mais especificamente no banco de dados, os tipos de dados encontrados são reais e

enumerações. Como reais podemos citar, por exemplo, o comprimento total de um animal ou o peso. Quanto às enumerações, temos como exemplo as características sexuais de um animal. Existem ícones que se adaptam melhor para um determinado tipo de dado. No *Iconvis*, as informações sobre o peso dos animais são mapeadas para uma escala de cores que varia de tons mais azuis para os animais de menor peso e tons mais avermelhados para animais mais pesados. Assim, para cada ícone (animal) que é desenhado na tela, a sua cor de fundo (supondo que o mapeamento tenha sido *peso* → *cor de fundo do ícone*) será determinada pelo peso do animal que um valor real. Esse valor real deve ser mapeado para a escala de cores definida de forma a obter a cor correspondente daquele peso.

Para os tipos enumerados, como o de característica sexual, também podem ser utilizadas cores para cada valor. Por exemplo verde para animais imperfurados, rosa para animais perfurados, vermelho para animais perfurados com secreção. Assim, para cada valor do domínio de valores da característica sexual teremos uma cor correspondente (mapeamento um para um). Também pode ser feito o uso das marcas (*Iconvis*), onde cada marca disponível pode ser mapeada para um desses valores, deixando as cores para outra informação.

O mapeamento de tipos de dados enumerados em um ícone do tipo *stick-figure* é bastante difícil de ser obtido devido a própria natureza do ícone, que é constituído apenas de segmentos de reta. O comprimento, o ângulo e a intensidade de cada segmento de reta são controlados pelos dados. Os outros ícones e glifos estão muito atrelados com a sua aplicação específica. O *infobug* por exemplo é destinado somente para a aplicação de gerenciamento de software. Para que ele possa ser utilizado com outros tipos de dados deveria ser feita recompilação da sua forma de mapeamento e das rotinas de desenho do glifo.

É importante que o usuário tenha disponível um ícone que seja bastante versátil, podendo ser utilizado para qualquer tipo de dado representando qualquer tipo de informação. Esses ícones “genéricos” podem ser construídos utilizando uma linguagem descritiva semelhante a que foi proposta neste trabalho. Assim, com essa mesma linguagem o usuário tem a liberdade de construir ícones conforme sua necessidade e de forma que o sistema de visualização não tenha que ser adaptado (e recompilado) para suportar novos tipos de ícones ou outros tipos de informações.

Um primeiro critério a ser analisado no protótipo, diz respeito à **quantidade de informações** que está sendo visualizada num dado momento. Através da utilização dos ícones propostos na dissertação, os biólogos (futuros usuários do sistema) podem visualizar as informações sobre seus animais que são consideradas mais importantes

para eles. Pela forma do ícone (quadrada ou circular) pode-se distinguir o sexo dos animais. Através da utilização dos ícones existentes na literatura como em [MEI 98] (*infobug glyph*) e em [ERB 95] (sistema *Exvis*, ícones *stick-figures*) seria difícil essa representação pois a forma desses ícones não varia, tornando a percepção da visualização um pouco mais demorada, requerendo um pouco mais de atenção do usuário para poder distinguir quais animais são fêmeas e quais são machos, por exemplo.

Ao utilizar o protótipo *Iconvis*, basta aos biólogos anotar a posição onde os animais foram capturados e carregar as informações diretamente no banco de dados. Atualmente, toda a fase de pré-processamento descrita neste texto é realizada manualmente pelos biólogos, que não contam com nenhuma ferramenta de visualização. Utilizando o *IconVis*, na tela principal serão vistos os animais que estão selecionados em uma porção do universo de pesquisa dos biólogos. Se eles desejarem visualizar os animais que estão nos arredores da posição atual ou que estão em uma região distante basta mover a janela de seleção para a posição desejada e a tela principal será automaticamente atualizada mostrando os animais que estão contidos na nova região selecionada.

A ferramenta *IconVis* foi desenvolvida levando em consideração as linhas gerais e os requisitos pesquisados para construção de uma boa ferramenta de visualização (citados na página 19). Cabe ressaltar que nem todos os requisitos citados se aplicam para o *IconVis*. Por exemplo, o protótipo *IconVis* atualmente não trabalha com quantidades vetoriais, sendo assim, o requisito que afirma que deve-se usar forma para representar quantidades vetoriais não se aplica nesse caso.

Antes do desenvolvimento do *IconVis*, foi feito um levantamento junto aos biólogos para determinar quais eram as informações mais importantes para serem visualizadas. Assim reduziu-se o conjunto de informações a apenas um subconjunto de interesse. Isso tornou mais fácil o planejamento dos ícones e os seus respectivos parâmetros para posterior mapeamento desse subconjunto de informações selecionadas.

Algumas informações desse subconjunto, como o peso e o comprimento de um animal, podem ser mapeadas para cores. Um dos requisitos recomenda usar cores para valores escalares. Isso foi feito para o peso e o tamanho do animal. As cores escolhidas, vermelho para valores mais altos e azul para valores mais baixos, foram escolhidas segundo um estudo feito sobre cores e a percepção do homem em relação a elas, onde valores altos são melhor percebidos quando em tons vermelhos e os tons azuis são associados a valores baixos).

O requisito que recomenda usar animação para evoluções no tempo não foi implementado no estágio atual do protótipo *IconVis*, mas poderia ser perfeitamente utilizado. Uma idéia seria a utilização da animação para representar a movimentação dos animais ao longo do tempo. Como no banco de dados temos a informação de captura e recaptura isso poderia ser feito. Mas como são apenas duas informações (captura e recaptura) essa característica do protótipo não ficaria muito interessante, o ideal seria que tivéssemos várias informações de recaptura, assim poderíamos ver o suposto caminho percorrido pelos animais entre as capturas.

O *IconVis* procura integrar consulta e navegação na mesma ferramenta. Assim é possível também fazer uma manipulação dos dados disponíveis para a visualização. Para dar suporte a esse requisito, foi implementado o construtor de consultas, que permite que o usuário possa selecionar apenas os dados que tem interesse. Existe a possibilidade do usuário ir realizando refinamentos sucessivos nas consultas até obter um conjunto de dados que esteja satisfatório para ele. Assim, com um menor número de informações, uma visualização mais “clara” e “limpa” é gerada.

Os itens relativos a usuários experientes e visualizações simultâneas não foram implementados no protótipo. O primeiro não o foi devido ao fato de que os usuários do *IconVis* possuem o mesmo conhecimento, não havendo a necessidade da criação de níveis de usuário. A segunda não foi implementada, porém até poderia ter sido feita. O local de estudo dos biólogos está dividido em várias regiões. No estágio atual do protótipo *IconVis*, o usuário pode somente ver uma determinada região por vez, não sendo possível, por exemplo, a visualização de duas ou mais regiões simultaneamente. É também sempre utilizada a mesma representação visual baseada em os ícones.

O *IconVis* procura prover detalhe com o contexto através do uso do conceito do mapeamento *window-viewport*. O espaço de informação (região de estudo) que deve ser visualizado é bastante grande e sua redução para visualização numa única imagem tornaria os detalhes muito pequenos para serem vistos, além de causar o problema da sobreposição de ícones pela proximidade com que diferentes animais são capturados. Assim, através do uso de uma janela, o usuário tem a possibilidade de ver todo o espaço de informação bem como a seleção da porção que está sendo visualizada num dado momento.

O protótipo procura ser bastante fácil e intuitivo no seu uso, requerendo um tempo mínimo de treinamento. O programa segue os padrões de outros aplicativos da linha *Windows*, não requerendo portanto uma adaptação maior a sua interface. A ‘navegação’ do usuário pelas várias telas do sistema é bastante fácil. Para fazer uma visualização de uma nova região, por exemplo, basta ao usuário deslocar a janela de seleção com o mouse para a nova posição no universo.

O *IconVis* provê a funcionalidade de salvar uma imagem de uma visualização realizada. Desse modo, o usuário tem uma 'fotografia' da última operação realizada. Juntamente com essa fotografia é criado um arquivo que contém todos os dados da visualização salva. O usuário pode, então, salvar o estado corrente do sistema, podendo voltar a trabalhar mais tarde do ponto onde parou.

No desenvolvimento do protótipo, não se levou muito em consideração a extensibilidade do mesmo nem a generalidade com respeito ao tipo de informação. Entretanto, apesar disto, com pequenas modificações no código fonte do programa, é possível estender o protótipo para outros tipos de informação ou mesmo para outras formas de visualização. Com relação à generalidade do tipo de informação, o construtor de consulta desenvolvido, como descrito anteriormente se adapta a qualquer tipo de informação, não requerendo praticamente nenhuma mudança no código fonte. O mapeador de ícones não é totalmente genérico, mas pode ser também facilmente adaptado para outros tipos de informação.

Cabe ressaltar que apesar da linguagem desenvolvida para a criação e geração automática de ícones ser bastante simples e limitada, é possível gerar diversas variações de ícones, permitindo que várias informações possam ser mapeadas para os mesmos. Os ícones gerados se adaptam perfeitamente ao tipo de informação do estudo de caso do capítulo 5 e as representações visuais escolhidas refletem exatamente as que os biólogos utilizam para representar seus animais em seus mapas, não requerendo assim, uma adaptação a novas formas de representação.

7 Conclusões e extensões

Este trabalho teve por objetivo o estudo de técnicas de visualização de informações, especificamente as que utilizam ícones como forma de representação visual. Um protótipo foi implementado para auxiliar os biólogos da Universidade Federal do Rio Grande do Sul a estudarem informações sobre pequenos roedores que são capturados na localidade de Itapuã, no Rio Grande do Sul.

Através da visualização usando ícones, as informações sobre os animais que estão num banco de dados são visualizadas graficamente e pode-se obter informações importantes que seriam difíceis de serem descobertas pela simples análise dos dados brutos.

Tomando-se o estudo de caso realizado, por exemplo, se um usuário deseja verificar no banco de dados os animais que estão acima de um determinado peso, a procura por esses animais diretamente no banco de dados é mais difícil. Para essa tarefa ele teria que usar um programa que lhe permitisse fazer consultas sobre o banco de dados. Na maioria das vezes esses programas requerem o conhecimento de *SQL*, linguagem que os usuários leigos dificilmente dominam. Assim, uma ferramenta de consulta com uma interface visual mais fácil e intuitiva foi implementada apoiando a ferramenta de visualização icônica.

O usuário pode usar a ferramenta visual de consulta para gerar a sua consulta, e o resultado dessa consulta pode ser visualizado graficamente através do uso de ícones. Dessa forma, no exemplo anteriormente citado, os animais mais pesados poderiam ser representados por ícones maiores ou ainda poderiam ser utilizadas cores para representação dessa informação, fazendo com que, por exemplo, os animais mais pesados tenham tons mais avermelhados e como consequência os animais de menor peso seriam representados por ícones menores e/ou através de tons mais azulados por exemplo.

Poderíamos ter um conjunto maior de ícones, ou ícones mais representativos dos animais, ao invés de usar somente círculos e quadrados, que permitisse um mapeamento maior de características. Uma idéia seria a adição de um módulo que permitisse ao usuário criar seus próprios ícones através de um editor de ícones. Através desse editor também seria possível determinar os parâmetros dos ícones que podem ser utilizados no mapeamento dos dados e informações.

O módulo construtor de consultas é genérico, podendo ser portado para qualquer outra aplicação, não importando o tipo de dados que essa aplicação irá tratar. O protótipo em si, entretanto, está muito “amarrado” às definições fornecidas pelos usuários atuais. Assim, tem-se uma série de cálculos (pré-processamento) que devem ser realizados para que seja possível obter a visualização dos dados. A generalização do protótipo é uma extensão natural deste trabalho, para que sua adequação a novas aplicações.

Poderia também ser feita a generalização dos pontos de referência utilizados para cálculo da posição dos animais. No estágio atual do protótipo, esses pontos de referência estão fixos (posições exatas foram fornecidas pelos biólogos). Para que essa generalização seja possível, esses pontos de referência devem ser informados junto com o banco de dados de informações sobre os animais. Atualmente o banco de dados contém apenas informações sobre os animais propriamente dito, não contendo nenhuma informação sobre a posição dos mesmos, ou mesmo dos pontos de referência. Com esses pontos de referência informados no banco de dados, essas informações seriam utilizadas na fase de pré-processamento para fazer o cálculo da posição de captura dos animais.

Uma outra extensão que poderia ser feita seria a determinação da distância entre os animais que foram recapturados e um mapa de deslocamento dos animais entre o ponto da primeira captura e o ponto de recaptura. Para que essa informação possa ser calculada, o ideal é que o banco de dados contenha informações sobre os pontos de referência, como mencionado acima.

A linguagem de descrição de ícones também poderia ser tornada mais poderosa, com mais recursos do que no estágio atual. No entanto, a utilização de ícones revelou-se bastante útil na identificação de informações associadas à distribuição espacial dos animais na região de estudo. Os biólogos do Departamento de Genética, numa primeira avaliação, julgaram o protótipo adequado às suas necessidades. Entretanto, seu uso regular com uma base de dados maior deverá fornecer uma avaliação mais rigorosa que poderá, então, ser levada em conta numa segunda versão do *IconVis*.

Bibliografia

- [AND 95] ANDREWS, K.; KAPPE, F.; MAURER, H. Serving Information to Web with Hyper-G. **Computer Networks and ISDN Systems**, Amsterdam, v.27, n.6, p.919-926, Apr.1995. Trabalho apresentado na International World-Wide Web Conference, 3.,1995.
- [AHL 95] AHLBERG, C.; WISTRAND, E. IVEE: An information visualization & exploration environment. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1995, Atlanta, GA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1995. p. 66-73
- [BED 99] BEDERSON, B.B.; BOLTMAN, A. Does Animation HelpUsers Build Mental Maps of Spatial Information? In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1999, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1999. p. 28-35.
- [BJO 99] BJORK, S.; HOLMQUIST, L.E.; REDSTORM, J. A Framework for Focus+Context Visualization. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1999, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1999. p. 53-56.
- [CHE 73] CHERNOFF, H. The Use of Faces to Represent Points in k-Dimensional Space Graphically. **Journal of the American Statistical Association**, [S.l.], v. 68, n. 342, p. 361-367, 1973.
- [CHI 98] CHI, E.H.; RIEDL, J.T. An Operator Framework for Visualization Systems. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1999, Research Triangle Park, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1998. p. 63-70.
- [DER 97] DERTHICK,M.; ROTH,S.F.; KOLOJEJCHICK, J.A. Coordinating Declarative Queries with a Direct Manipulation Data Exploration Environment, In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1997, Phoenix, AZ. **Proceedings...** Los Alamitos: IEEE Computer Society, 1997. p. 65-72.
- [DON 88] COX, D.; ELLSON, R. Visualization of injection molding. **Simulation**, [S.l.], p.184-188, 1988.
- [ERB 95] ERBACHER, R. et al. Exploratory Visualization Research at the University of Massachusetts at Lowell. **Computer & Graphics**, [S.l.], v.19, n.1, p.131-139,1995. Disponível em: <http://www.cs.uml.edu/~rerbache/ps/Visualization_Research_Summary.ps.gz>. Acesso em 18jul.2000.
- [FAY 96] FAYYAD, U.M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From Data Mining to Knowledge Discovery: An Overview. In: FAYYAD, U.M. et al.

- (Eds.). **Advances in Knowledge Discovery and Data Mining**. Cambridge: AAAE Press:MIT Press, 1996. p. 1-34.
- [FOR 97] FORTNER, B.; MEYER, T.E. **Number By Color** : A Guide To Using Color To Understand Technical Data. [S.l.]: Springer Telos, 1997.
- [FRE 93] FREITAS, C.M.S. A Methodology for Selecting Visual Representations in Scientific and Simulation Applications. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE IMAGENS, SIBGRAPI, 6., 1993, Recife. **Anais...** Recife: SBC/UFPE, 1993. p.89-97.
- [FUR 86] FURNAS, G.W. Generalized fisheye views. In: CHI CONFERENCE, 1986, Boston. **Proceedings...** [S.l.:s.n.], 1986. p.16-23.
- [GER 95] GERSHON, N.; ELICK, S.G. Visualization's New Track: Making Sense of Information. **IEEE Spectrum**, New York, p.38-56, 1995.
- [GER 97] GERSHON, Nahum; ELICK, S. G. Information Visualization. **IEEE Computer Graphics and Applications**, New York, p. 29-31, 1997.
- [GRE 95] GREENBERG, S.; GUTWIN C.; COCKBURN, A. Sharing fisheye views in relaxed – WYSIWIS group-ware applications. In: GRAPHICS INTERFACE, 1995, Toronto, Canada. **Proceedings...** [S.l.:s.n.], 1995. P.28-38
Disponível em:
<http://www.cpsc.ucalgary.ca/redirect/grouplab/papers/Gi96/gi96_fisheye.html>. Acesso em:20nov.2000.
- [HAB 90] HABER, R.B.; MCNABB, D.A. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In: NIELSON, G.M.; SHRIVER, B.D.; ROSENBLUM, L. (Eds.). **Visualization in Scientific Computing**. Los Alamitos: IEEE Computer Society Press, 1990. p.74-92.
- [HAN 98] HANSEN, M.; MEADS, D.; PANG, A. Comparative Visualization of Protein Structure-Sequence Alignments. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1998, Research Triangle Park. **Proceedings...** Los Alamitos: IEEE Computer Society, 1998. p. 106-110.
- [HIG 99] HIGGINS, M.; LUCAS, P.; SENN, J. VisageWeb: Visualizing WWW data in Visage. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1999, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1999. p. 100-111.
- [JOS 94] JOSLYN, C. M. **PV-WAVE:Glyph Multidimensional Visualization Toolkit User's Guide**. [S.l.], 1994. Disponível em:
<<http://www.vni.com/products/wave/>>. Acesso em 27set.2000.
- [KEA 96] KEAHEY, T., ROBERTSON, E. Techniques for Non-Linear Magnification Transformations, In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1996. **Proceedings...** Los Alamitos: IEEE Computer Society, 1996. p.38-45.
- [KER 98] KERPEDJIEV, S. et al. Saying it in Graphics: from Intentions to Visualizations. In: IEEE INFORMATION VISUALIZATION

- SYMPOSIUM, 1999, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1999. p. 97-101.
- [LEE 94] LEE, J.P. **Database Exploration**. Berlin: Springer-Verlag, 1994. p.118-138. (Lecture Notes in Computer Science, v.871).
- [LEV 91] LEVKOWITZ, H. Color Icons: Merging color and texture perception for integrated visualization of multiple parameters. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1991, San Diego, CA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1991.
- [MAC 91] MACKINLAY, J. D.; ROBERTSON, G. G.; CARD, S. K. Perspective wall : detail and context smoothly integrated. In: SIGCHI, 1991. **Proceedings...** [S.l.:s.n.], 1991. p.173-179.
- [MEI 98] CHUAH, M. C.; EICK, S. G. Information Rich Glyphs for Software Management Data. **IEEE Computer Graphics and Applications**, New York, p.24-29, 1998.
- [MUK 94] MUKHERJEA, S.; STASKO, J. Towards Visual Debugging: Integrating Algorithm Animation Capabilities within a Source Level Debugger. **ACM Transactions on Computer-Human Interaction**, New York, v.1 n.3, p.215-244, Sept. 1994.
- [MUK 95] MUKHERJEA, S.; FOLEY, J. D. **Requirements and Architecture of Information Visualization Tool**. Berlin: Springer-Verlag, 1995. (Lecture Notes in Computer Science, v. 1183).
- [NEU 87] NEUWIRTH, C.; KAUFFER, D.; CHIMERA, R.; TERILYN, G. The Notes Program: A Hypertext Application for Writing from Source Texts. In: HYPERTEXT CONFERENCE, 1987, Chapel Hill, NC. **Proceedings...** [S.l.:s.n.], 1987. p. 121-135.
- [PIK 70] PICKETT, R. Visual analyses of texture in the detection and recognition of objects. In: LIPKIN, B.S.; ROSENFELD, A. (Eds.). **Picture Processing and Psycho-Pictorics**. New York:Academic Press, 1970.
- [PIN 99] PINKNEY, D. **Intelligent Iconic Visualization**. Disponível em: <<http://glug.cs.uml.edu/~dpinkney/thesis/thesis.shtml>>. Acesso em: 16out.1999.
- [PIN 99] PINKNEY, D. **Review of Some Visualization Techniques**. Disponível em: <http://glug.cs.uml.edu/~dpinkney/vis/vis_review.html>. Acesso em: 20ago.1999.
- [PIR 2000] PIROLI P.; CARD, S. K; Wege, M. M. V. D. The Effect of Information Scent on Searching Information Visualizations of Large Tree Structures. In: **AVI**, 2000, Palermo, Italy. **Proceedings...** [S.l.:s.n.], 2000.
- [RAO 94] RAO, R.; CARD, S. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In: **ACM SIGCHI CONFERENCE ON HUMAN FACTORS**

- IN COMPUTING SYSTEMS, 1994, Boston, MA. **Proceedings...** [S.l.:s.n.], 1994. p. 318-322.
- [REK 93] REKIMOTO, J. ; GREEN, M. The Information Cube: Using Transparency in 3D Information Visualization. In: THIRD ANNUAL WORKSHOP ON INFORMATION TECHNOLOGIES & SYSTEMS, 1993. **Proceedings...** [S.l.:s.n.], 1993. p. 125-132. Disponível em: <<http://www.csl.sony.co.jp/person/rekimoto/cube.html>>. Acesso em: 29out.2000.
- [RIB 94] RIBARSKY, W. et al. Glyphmaker: Creating Customized Visualizations of Complex Data. **IEEE Computer**, New York, p. 57-64, 1994.
- [ROB 93] ROBERTSON, G.G.; CARD, S.K.; MACKINLAY, J.D. Information Visualization Using 3D Interactive Animation. **Communications of ACM**, New York, v.36, n.4, p.57-71, 1993.
- [ROB 98] ROBERTSON, G. Leveraging Human Capabilities in Information Perceptualization. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1998, Research Triangle Park. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1998. p. xi.
- [ROT 94] ROTH, S.F. et al. Interactive graphic design using automatic presentation knowledge. In: ACM CHI HUMAN FACTORS IN COMPUTER SYSTEMS, 1994, Boston. **Proceedings...** [S.l.:s.n.], 1994. p.112-117.
- [ROT 96] ROTH, S.F. et al. Visage: A User Interface Environment for Exploring Information. In: IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1996, San Francisco, CA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1996. p.3-12.
- [SAR 94] SARKAR M.; BROWN M. Graphical Fisheye Views. **Communications of the ACM**, New York, v.37, n.12, p.73-84, 1994.
- [SIL 95] SILVER, D. Object-Oriented Visualization. **IEEE Computer Graphics and Applications**, [S.l.], v. 15, n. 3, p. 54-63, May 1995.
- [UPS 89] UPSON, C. et al. The Application Visualization System: A Computational Environment for Scientific Visualization. **IEEE Computer Graphics and Applications**, [S.l.], v.9, n.4, p.30-42, 1989.
- [WAL 96] WALSUM, T. V. et al. Feature Extraction and Iconic Visualization. **IEEE Transactions on Visualization and Computer Graphics**, [S.l.], v.2, n.2, 1996.
- [WAT 91] WATERORTH, J.; CHIGNELL, M. A Model for Information Exploration. **Hypermedia**, [S.l.], v.3, n.1, p.33-58, 1991.