

Universidade Metodista de Piracicaba
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

**Ambientes Virtuais para Visualização Dinâmica
de Informação**

Carolina Valim Buk
Orientador: Prof. Dr. Claudio Kirner

PIRACICABA, SP

2006



Universidade Metodista de Piracicaba
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

AMBIENTES VIRTUAIS PARA VISUALIZAÇÃO DINÂMICA
DE INFORMAÇÃO

Carolina Valim Buk
Orientador: Prof. Dr. Claudio Kirner

Dissertação apresentada ao Mestrado em
Ciência da Computação, da Faculdade de
Ciências Exatas e da Natureza, da
Universidade Metodista de Piracicaba –
UNIMEP, como requisito para obtenção do
Título de Mestre em Ciência da Computação.

PIRACICABA, SP

2006

AMBIENTES VIRTUAIS PARA VISUALIZAÇÃO DINÂMICA DE INFORMAÇÃO

AUTORA: CAROLINA VALIM BUK

ORIENTADOR: PROF. DR. CLAUDIO KIRNER

Dissertação de Mestrado defendida e aprovada em 15 de fevereiro de 2006,
pela Banca Examinadora constituída dos Professores:

Prof. Dr. Claudio Kirner

UNIMEP

Prof^a. Tereza Gonçalves Kirner

UNIMEP

Prof. Dr. Jorge Luis e Silva

ICMC-USP

DEDICATÓRIA

*Aos meus Pais, pelo exemplo, às irmãs, pelo
companheirismo, ao Paulo, namorado
atencioso e amigo de todas as horas.*

DEDICO

A G R A D E C I M E N T O S

Ao Prof. Dr. Claudio Kirner, a segura e precisa orientação.

Ao Prof. Dr. Nivaldi Calonego Júnior a colaboração para a realização deste trabalho.

Aos Professores do Mestrado em Ciência da Computação da FACEN – UNIMEP, os ensinamentos.

Ao colega Rafael Santin pela ajuda no desenvolvimento das aplicações com realidade aumentada.

Aos colegas de curso as preciosas trocas de idéias e de conhecimentos.

AMBIENTES VIRTUAIS PARA VISUALIZAÇÃO DINÂMICA DE INFORMAÇÃO

RESUMO

Com a crescente disseminação da informação e seu aumento em quantidade e complexidade, surge a necessidade da utilização de novas técnicas para sua representação, visualização e interação. Gráficos 3D interativos, implementados com técnicas de realidade virtual, vêm sendo usados em ferramentas acadêmicas e comerciais, apresentando vantagens e desvantagens inerentes à área, como a ampliação dos limites da tela do monitor. No entanto, a realidade virtual apresenta dificuldades de navegação e interação, que resultam em custos de implantação e treinamento. No sentido de minimizar as dificuldades da realidade virtual, novas técnicas de visualização com realidade aumentada estão surgindo, trazendo os gráficos interativos para o espaço físico do usuário, facilitando sua inspeção e manipulação. Este trabalho apresenta o desenvolvimento de técnicas de visualização de informação, em ambientes de realidade virtual e de realidade aumentada, mostrando algumas versões de ferramentas de visualização e suas aplicações.

PALAVRAS-CHAVE: Gráficos 3D, Realidade Virtual, Realidade Aumentada.

VIRTUAL ENVIRONMENTS FOR DYNAMIC VISUALIZATION OF INFORMATION

ABSTRACT

With the increasing growth of the information dissemination and its increase in quantity and complexity, it is necessary to use new techniques for its representation, visualization and interaction.

Interactive 3D Graphics which are implemented with virtual reality techniques are been used in academic and commercial tools, giving us advantages and disadvantages beneath its area, like monitor screen limits growing. However, virtual reality presents us interaction and navigation difficulties witch results in training and implementation costs. Trying to minimize virtual reality difficulties new visualization techniques with augmented reality are being developed, presenting interactive graphics to the end-user physical place making easier its inspection and manipulation.

The purpose of this work is to show the information visualization technique development, virtual reality environments and augmented reality environments, presenting some versions of visualization tools and its applications.

KEYWORDS: Graphics 3D, Virtual Reality, Augmented Reality.

SUMÁRIO

LISTA DE FIGURAS	V
LISTA DE TABELAS	VII
CAPÍTULO 1	1
1. <u>Introdução</u>	1
1.1. <u>Caracterização do Problema</u>	1
1.2. <u>Motivação</u>	2
1.3. <u>Delimitação do Assunto</u>	3
1.4. <u>Objetivos da Pesquisa</u>	3
1.5. <u>Metodologia</u>	4
1.6. <u>Estrutura da Dissertação</u>	4
CAPÍTULO 2	6
2. <u>Realidade Virtual e Realidade Aumentada</u>	6
2.1. <u>Realidade Virtual</u>	6
2.1.1. <u>VRML</u>	9
2.1.2. <u>Realidade Virtual Imersiva e Não-imersiva</u>	12
2.1.3. <u>Imagem Estereoscópica</u>	13
2.2. <u>Realidade Aumentada</u>	14
2.2.1. <u>Alguns Tipos de Sistemas de Realidade Aumentada</u>	16
2.2.2. <u>Exemplos de Sistemas de Realidade Aumentada</u>	19
2.2.3. <u>Studierstube</u>	19
2.2.4. <u>ARToolKit</u>	20
2.2.5. <u>MXR Toolkit</u>	23
CAPÍTULO 3	26
3. <u>Visualização da Informação</u>	26
3.1. <u>Modelagem da Informação</u>	26
3.2. <u>Classificação da Informação</u>	27
3.3. <u>Representação de Informações</u>	28
3.4. <u>Ferramentas de Visualização de Informações</u>	29
3.4.1. <u>Seesoft</u>	29
3.4.2. <u>Information Mural</u>	30
3.4.3. <u>Tile Bars</u>	31
3.4.4. <u>Document Lens</u>	33
3.4.5. <u>Geographic Information System</u>	34
3.4.6. <u>NIRVE – NIST Information Retrieval Visualization Engine</u>	35
3.4.7. <u>Table Lens</u>	37
3.4.8. <u>Advisor Authoring Toolkit</u>	38
3.4.9. <u>Open Viz</u>	39
3.4.10. <u>Open DX</u>	40
3.5. <u>Ferramentas de Visualização de Informação avaliadas neste trabalho</u>	41
3.5.1. <u>3D Active Chart</u>	41
3.5.2. <u>Nchart Editor 2.0</u>	43
3.5.3. <u>Gannalyst Lite 3.0 Standard</u>	44
3.6. <u>Comparação das ferramentas testadas</u>	45
CAPÍTULO 4	47
4. <u>Projeto de Ambientes Virtuais para Visualização Dinâmica de Informações</u>	

4.1. Ambiente de Realidade Virtual	47
4.1.1. Modelagem dos Gráficos	47
4.1.2. Linguagens Utilizadas	49
4.1.3. Versão Com Tabela	52
4.1.4. Versão Com Banco de Dados Local	54
4.2. Ambiente de Realidade Aumentada	61
4.2.1. Versão para Visualização Estática	61
4.2.2. Versão para Visualização Dinâmica Sem Tabela	63
4.3. Aplicações	66
4.3.1. Potencialidades	67
4.3.2. Restrições	68
CAPÍTULO 5	69
5. Conclusões	69
REFERÊNCIAS BIBLIOGRÁFICAS	72
ANEXOS	81

LISTA DE FIGURAS

<u>FIGURA 1 - EXEMPLO DE UM OBJETO DESENVOLVIDO EM VRML</u>	8
<u>FIGURA 2 - EXEMPLO DE OBJETO EM FORMA DE CONE EM REALIDADE VIRTUAL</u>	10
<u>FIGURA 3 - LUVA USADA EM REALIDADE VIRTUAL (VREALITIES, 2005)</u>	13
<u>FIGURA 4 - CAPACETE USADO EM REALIDADE VIRTUAL (VREALITIES, 2005)</u>	13
<u>FIGURA 5 - ESQUEMA DO FUNCIONAMENTO DA IMAGEM ESTEREOSCÓPICA (RAPOSO, 2004)</u>	14
<u>FIGURA 6 - EXEMPLO DE REALIDADE AUMENTADA - VASO E CARRO INSERIDOS NO MUNDO REAL (KIRNER, 2004)</u>	15
<u>FIGURA 7 - - ESQUEMA DE VISUALIZAÇÃO EM REALIDADE AUMENTADA ATRAVÉS DO CAPACETE COM VISÃO ÓPTICA (AZUMA, 1997) (KIRNER, 2005)</u>	17
<u>FIGURA 8 - ESQUEMA DE VISUALIZAÇÃO EM REALIDADE AUMENTADA ATRAVÉS DE CAPACETE COM VISÃO POR VÍDEO (AZUMA, 1997) (KIRNER, 2005)</u>	18
<u>FIGURA 9 - ESQUEMA DE VISUALIZAÇÃO EM REALIDADE AUMENTADA ATRAVÉS DO MONITOR (AZUMA, 1997) (KIRNER, 2005)</u>	19
<u>FIGURA 10 - EXEMPLO DO STUDIERSTUBE (SCHMALSTIEG, 2002)</u>	20
<u>FIGURA 11 - ESTRUTURA DE ARQUIVOS DO ARTOOLKIT (ARTOOLKIT, 2004)</u>	21
<u>FIGURA 12 - EXEMPLO DE MARCADOR</u>	22
<u>FIGURA 13 - DIAGRAMA DE PROCESSAMENTO DAS IMAGENS NO ARTOOLKIT (ADAPTADA DE KATO, 2000)</u>	23
<u>FIGURA 14 - EXEMPLO DE INSERÇÃO DE PESSOAS NO AMBIENTE VIRTUAL (MXR TOOLKIT)</u>	24
<u>FIGURA 15 - FUNCIONAMENTO DO SISTEMA MXR TOOLKIT (MXR TOOLKIT)</u>	25
<u>FIGURA 16 – SEESOFT (EICK, 1992B)</u>	30
<u>FIGURA 17 - INFORMATION MURAL (JERDIN, 1998); (GEISLER, 1998)</u>	31
<u>FIGURA 18 - TILE BARS (HEAST, 1995)</u>	32
<u>FIGURA 19 - DOCUMENT LENS (ROBERTSON, 1993)</u>	33
<u>FIGURA 20 - GEOGRAPHIC INFORMATION SYSTEM (USGS – U. S. GEOLOGICAL SURVEY, 2002)</u>	35
<u>FIGURA 21 – NIRVE (SEBRECHTS, 1999)</u>	36
<u>FIGURA 22 - TABLE LENS (PIROLI, 1996)</u>	37
<u>FIGURA 23 - ADVISOR AUTHORIZING TOOLKIT (ADVIZOR SOLUTIONS INC., 2003)</u>	38
<u>FIGURA 24 - OPEN VIZ (WAYNE)</u>	39
<u>FIGURA 25 - SISTEMA DE VISUALIZAÇÃO DE INFORMAÇÕES OPEN DX (IBM, 1999)</u> ..	41
<u>FIGURA 26 - GRÁFICO DO 3D ACTIVE CHART (ARCTIVE CHART, 2004)</u>	42
<u>FIGURA 27 - GRÁFICO DO 3D ACTIVE CHART (ARCTIVE CHART, 2004)</u>	43
<u>FIGURA 28 - EXEMPLO DO NCHART EDITOR (NCHART EDITOR, 2004)</u>	44
<u>FIGURA 29 - FERRAMENTA GANNALYST E SEUS GRÁFICOS (GANNALYST, 2002)</u> ..	45
<u>FIGURA 30 - TRANSFERÊNCIA DE EVENTOS VRML/JAVA (LIMA, 2004)</u>	50
<u>FIGURA 31 - ESTRUTURA DA APLICAÇÃO</u>	51
<u>FIGURA 32 - ESQUEMA DO FUNCIONAMENTO DA VERSÃO COM TABELA</u>	52
<u>FIGURA 33 - EXEMPLO DE GRÁFICO</u>	53
<u>FIGURA 34 – EXEMPLO DA TABELA</u>	53
<u>FIGURA 35 - ESQUEMA DO FUNCIONAMENTO DA VERSÃO COM BANCO DE DADOS LOCAL</u>	54
<u>FIGURA 36 - BANCO DE DADOS ACCESS</u>	55

<u>FIGURA 37 - GRÁFICO DE VISUALIZAÇÃO DE INFORMAÇÕES</u>	56
<u>FIGURA 38 - ELEMENTO DE REFERÊNCIA - BARRA INDIVIDUAL</u>	57
<u>FIGURA 39 - AMBIENTE VIRTUAL SIMPLIFICADO</u>	57
<u>FIGURA 40 - ESQUEMA DO FUNCIONAMENTO COM AQUISIÇÃO REMOTA DE DADOS</u>	59
<u>FIGURA 41 - SISTEMA DE AQUISIÇÃO REMOTA DE DADOS</u>	59
<u>FIGURA 42 - EXEMPLO DO USO DO DATAVIS-AR</u>	61
<u>FIGURA 43 - EXEMPLOS DA FERRAMENTA DE REALIDADE AUMENTADA</u>	62
<u>FIGURA 44 - FUNCIONAMENTO DA FERRAMENTA ARTTOOLKIT COM O EDITOR</u>	64
<u>FIGURA 45 - VISUALIZAÇÃO DINÂMICA</u>	65

LISTA DE TABELAS

<u>TABELA 1 - COMPARAÇÃO DAS FERRAMENTAS TESTADAS</u>	46
---	----

CAPÍTULO 1

1. INTRODUÇÃO

1.1. CARACTERIZAÇÃO DO PROBLEMA

Atualmente existe uma grande quantidade de informações com as quais os usuários lidam todos os dias, necessitando entendê-las e explorá-las, a fim de obter resultados que gerem sua satisfação.

O uso de técnicas de visualização de informações ajuda na exploração e na interpretação dos dados, pois os convertem em imagens. Essas técnicas utilizam a capacidade computacional para realizar a conversão das informações em imagens e também fazem com que a capacidade de cognição do ser humano seja bem utilizada na interpretação das mesmas. Para o usuário, a visualização através de imagens possibilita maior entendimento das informações, ajudando na tomada de decisão. As técnicas de visualização geram as imagens a partir de informações contidas em bancos de dados, realizando seu processamento e conversão (BUK, 2004b); (RISSE, 1998).

O computador é uma ferramenta adequada para a visualização das informações, possuindo a capacidade de apresentar ao usuário um ambiente de espaço amplo para as imagens obtidas. Também possui capacidade para agregar informações já existentes, fazendo com que a aquisição e interpretação tornem-se mais completas, ajudando de forma mais consistente a tomada de decisão do usuário.

Existem várias possibilidades para a visualização das informações por meio das técnicas de multimídia, que apresentam gráficos interativos no monitor do computador. A visualização na da tela do monitor traz certas limitações à capacidade de visualização. Na visualização em realidade virtual, por exemplo, são utilizados gráficos que apresentam várias informações, que podem não estar sendo vistas pelo usuário enquanto está

navegando. Para ajudá-lo acrescenta-se som às informações críticas, pois nem sempre essas estarão no campo de visão do usuário. Para trabalhar com ambientes de realidade virtual é necessário que o usuário seja treinado, tendo assim o conhecimento dos dispositivos de navegação (luvas, capacetes, etc) para não se perder dentro do ambiente.

Uma forma mais natural de movimentar-se no ambiente tridimensional, é trazer essas informações para o espaço do usuário através de realidade aumentada. Em realidade aumentada, as informações e os gráficos do ambiente virtual são apresentados no mundo real (BUK, 2005). Com a utilização de uma *webcam* e placas de identificação, onde cada uma corresponde a uma parte de um gráfico em realidade virtual, o usuário pode ter as informações em “suas mãos”, interagindo de tal forma que lhe facilite a interpretação dos dados.

Este trabalho aborda a implementação da visualização de informações através de realidade virtual, trazendo-as para o mundo do usuário com realidade aumentada.

1.2. MOTIVAÇÃO

A motivação para a realização deste trabalho é a busca de um meio de visualizar as informações que possua interatividade, retornando respostas de maneira rápida e de fácil entendimento.

Atualmente, para visualizar as informações é necessário o uso da atividade cognitiva, a qual demanda tempo. Já com a utilização de sistemas computacionais, essa atividade torna-se mais rápida, dada a capacidade de realizar cálculos com rapidez e eficiência, característica importante para visualização de informações, pois pode-se enviar essas informações e transformá-las em gráficos rapidamente. Por outro lado, o ser humano possui grande capacidade para assimilar informações de imagens, pois está relacionada diretamente com sua atividade cognitiva, necessitando apenas visualizar a imagem gerada pelo computador para tomar suas decisões (GERSHON, 1998); (KIRNER, 2000).

Com a união do sistema computacional à capacidade humana de visualizar informações gráficas pode-se trabalhar e analisar grande quantidade de dados, pois há a complementação da visualização humana com a análise de informações do sistema computacional.

Este trabalho aborda a representação da visualização das informações em gráficos discretos de barras cilíndricas tridimensionais, sendo que para melhorar tal representação foi incorporado o uso da realidade virtual e da realidade aumentada, que possuem a habilidade de inserir o usuário em um espaço tridimensional, interagindo com os objetos do ambiente.

1.3. DELIMITAÇÃO DO ASSUNTO

Este trabalho aborda a visualização de informações em ambiente de realidade virtual e aumentada, utilizando-se banco de dados, linguagem de modelagem de realidade virtual (VRML), ARToolKit, JavaApplet e JavaScript para o desenvolvimento do sistema e aplicações de realidade aumentada.

1.4. OBJETIVOS DA PESQUISA

O objetivo deste trabalho é apresentar uma forma de visualizar as informações que propicie um retorno das mesmas de maneira mais rápida e objetiva, propiciando assim, uma ferramenta de análise e visualização de informações (SPENCE, 2001). Essa ferramenta emprega técnicas de realidade virtual, realidade aumentada, software de modelagem e visualização de dados. Usando-se a realidade virtual pode-se ter a modelagem tridimensional com interação em tempo real, tornando o usuário capaz de visualizar, analisar e interpretar as informações de maneira eficiente. A realidade aumentada permite trazer o ambiente virtual para o ambiente do usuário, facilitando sua interação com os objetos virtuais.

Foi desenvolvido um protótipo em ambiente de realidade virtual, mostrando como visualizar as informações e viabilizando a integração com o usuário por modelos de gráficos tridimensionais configuráveis e reutilizáveis.

Com esse protótipo pretende-se contribuir para a melhoria da visualização de informações, facilitando ao usuário navegar em sua interface, bem como interagir com as representações.

Como objetivos específicos do trabalho, pretende-se: 1) apresentar uma forma de visualizar as informações; 2) desenvolver uma ferramenta para visualizar as informações.

1.5. METODOLOGIA

Para realizar este trabalho foi feita revisão bibliográfica sobre realidade virtual, realidade aumentada, visualização de informações, Java, javascript, javaapplet, EAI (external authoring interface), VRML, banco de dados e HTML.

Em seguida, foi desenvolvido um protótipo simples, o DATAVIS-VR, contendo um gráfico em ambiente de realidade virtual utilizando-se VRML (linguagem de realidade virtual) e linguagem de programação Java. Esse protótipo evoluiu para a busca de dados com a inserção de um banco de dados. Para acompanhar a variação das informações foi inserido um módulo de histórico dos dados no banco de dados.

Este último protótipo evoluiu para um ambiente de realidade aumentada, resultando no DATAVIS-AR. Nesse ambiente foram realizadas duas soluções: uma com gráficos estáticos e outra com alteração no software de realidade aumentada, ARTOOLKIT, para mostrar as modificações do gráfico.

1.6. ESTRUTURA DA DISSERTAÇÃO

Este trabalho está dividido em seis partes. A primeira descreve uma introdução do trabalho.

A segunda parte apresenta a fundamentação teórica na qual o trabalho está baseado, descrevendo os ambientes de realidade virtual e realidade aumentada.

A terceira parte possui uma abordagem sobre a visualização da informação, sua modelagem, representação e exemplos de algumas ferramentas existentes.

Na quarta parte são apresentados os projetos desenvolvidos no ambiente de realidade virtual, as versões com tabela atualizada de forma dinâmica automática e com banco de dados local, e também os projetos em ambientes de realidade aumentada e suas versões de visualização estática e dinâmica sem tabela.

Por fim, apresentam-se a conclusão e as referências bibliográficas.

CAPÍTULO 2

2. REALIDADE VIRTUAL E REALIDADE AUMENTADA

2.1. REALIDADE VIRTUAL

Neste trabalho a realidade virtual será utilizada como interface, de modo que o usuário tenha possibilidade de interagir com o gráfico virtual, podendo por meio deste tomar suas decisões.

Dentre as muitas definições para realidade virtual, uma delas é a de Kirner (1999): *“É uma técnica avançada de interface, onde o usuário pode realizar imersão, navegação e interação em um mundo virtual sintético, gerado por computador, usando dispositivos multisensoriais, em tempo real”*.

Outra definição encontrada é a de que realidade virtual é um paradigma, no qual utiliza-se o computador com a finalidade de interagir com um mundo que não é real, mas pode ser considerado como tal no momento em que é utilizado (HAND, 1994).

Sherman (2003), define realidade virtual como “um meio composto de simulações computacionais interativas que dão a sensação de posição e ação ao participante, substituindo ou aumentando a realimentação de um ou mais sentidos, dando a impressão de estar mentalmente imerso ou presente em uma simulação (um mundo virtual)”.

De acordo com White (2002), a realidade virtual pode ser considerada como um protótipo ou uma realidade alternativa, onde é possível construir ambientes que não existem no mundo real.

Diante de várias definições, pode-se dizer que com a realidade virtual é possível gerar um ambiente tridimensional imitando a realidade ou um ambiente próprio, também tridimensional, no qual o usuário possa interagir e navegar facilmente, em tempo real.

Em realidade virtual são necessárias três características fundamentais: imersão, interação e envolvimento (KIRNER, 1999); (KNIGHT, 1998); (WLOKA, 1995); (KIRNER, 2000); (SHERMAN, 2003). Imersão significa atuar dentro do ambiente utilizando os sentidos: visão, audição e tato. Interação ocorre quando o usuário modificar instantaneamente o mundo virtual e suas ações sobre o mesmo. Envolvimento é o grau de engajamento do usuário em uma atividade no mundo virtual.

A realidade virtual difere de outras abordagens de conceitos computacionais tridimensionais por seus sistemas serem interativos, ou seja, os usuários movimentam-se e interagem uns com os outros e com os objetos do ambiente (DIEHL, 2001); (WLOKA, 1995). A navegação é tridimensional (3D), ocorrendo nos eixos x, y e z, da mesma forma como ocorre no mundo real. Para visualizar as imagens em três dimensões é necessário também lidar com os dados em três dimensões (x, y, z), o que afeta o modo como os dados são representados, como apresentado na Figura 1. Para desenhar o objeto desta figura foi utilizada a linguagem de realidade virtual, VRML (Virtual Reality Modeling Language), cujo código é mostrado em seguida.

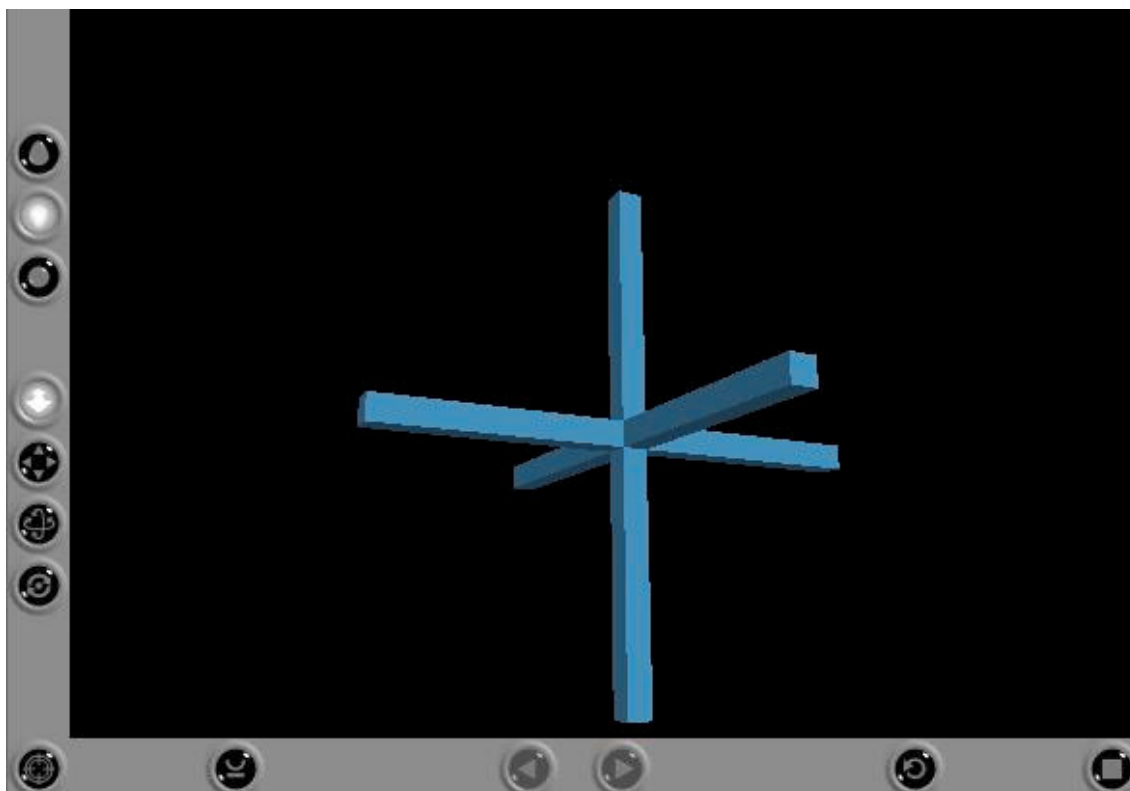


Figura 1 - Exemplo de um objeto desenvolvido em VRML

Código VRML da Figura 1:

```

Group {
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.35 0.58 0.73
          specularColor 0.23 0.39 0.49
          emissiveColor 0.06 0.09 0.12  }}
      geometry Box {size 0.1 2.0 0.1}}
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.35 0.58 0.73
          specularColor 0.23 0.39 0.49
          emissiveColor 0.06 0.09 0.12  }}
      geometry Box {size 2.0 0.1 0.1}}
    Shape {

```

```

appearance Appearance {
  material Material {
    diffuseColor 0.35 0.58 0.73
    specularColor 0.23 0.39 0.49
    emissiveColor 0.06 0.09 0.12  }}
  geometry Box {size 0.1 0.1 2.0}} }

```

O usuário precisa conhecer as técnicas de navegação do mundo virtual, como a orientação e a posição para controlar seus movimentos no ambiente. DARKEN (1993) descreve algumas características do mundo virtual que interferem em sua navegabilidade, como: a) o tamanho do ambiente, que pode interferir no discernimento dos objetos existentes, na sua estrutura e no desvio da atenção do usuário; b) a densidade que pode dificultar a navegação, pois um ambiente denso possui grande quantidade de objetos e dependendo da distribuição dos mesmos pode causar transtornos à mobilidade do usuário; c) o nível de atividade existente que pode aumentar o nível de dificuldade na navegação, conforme aumenta a movimentação de objetos (DARKEN, 1993). Uma ferramenta, bastante utilizada para sanar as dificuldades de navegabilidade, é o mapa do ambiente, contendo o local onde os objetos estão localizados e seus caminhos.

O sistema de realidade virtual faz com que o usuário seja colocado dentro de um ambiente virtual, que contém todas as informações necessárias a fim de tornar esse ambiente virtual o mais real possível.

O usuário de realidade virtual pode navegar livremente entre os objetos do ambiente, interagir e tomar decisões sobre as informações que estão sendo representadas pelos mesmos. O ambiente virtual, além de conter os objetos virtuais, também pode apresentar informações adicionais, como uma caixa de texto ou *links* para sites da Internet.

2.1.1. VRML

A linguagem VRML (Virtual Reality Modeling Language - Linguagem de Modelagem de realidade virtual) foi criada pelo Web3D

Consortium para o desenvolvimento de aplicações para a internet (INTERNATIONAL STANDARD ISO/IEC 14772-1,1997).

VRML é um formato de arquivo para descrever objetos e mundos tridimensionais, podendo ser acessada via Internet, Intranet e sistemas locais (INTERNATIONAL STANDARD ISO/IEC 14772-1, 1997); (DIEHL, 2001). A proposta dessa linguagem é tornar-se padrão para gráficos e multimídias 3D, sendo utilizada em várias áreas de aplicação, como visualização de engenharia, apresentações multimídia, entretenimento, educação, medicina, dentre outros.

Essa linguagem permite descrição de hierarquias simples com o uso de formas primitivas, como cilindro, cubo e esfera. Recursos mais complexos podem ser definidos, utilizando superfície de materiais, texturização e nível de detalhes. Os objetos também são usados como conexão para outros objetos ou endereços da Internet. Outras características incluem a transformação de objetos, mudanças de pontos de vista, iluminação, renderização. É possível também trabalhar com outros objetos de arquivos utilizados na Internet, como sons e animação.

Na Figura 2 tem-se um objeto em forma de cone no ambiente de realidade virtual.

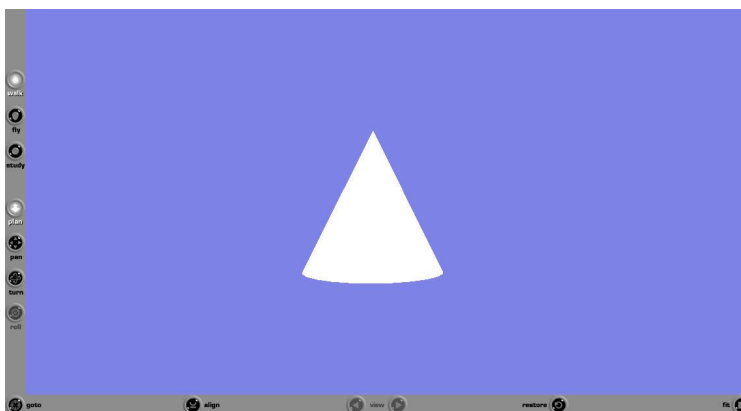


Figura 2 - Exemplo de objeto em forma de cone em realidade virtual

Por meio da linguagem VRML as cenas tridimensionais podem ser exploradas no mundo virtual, podendo suportar dinamismo em diferentes níveis de detalhes (RISSE, 1998). No código VRML é detalhado o plano de fundo onde está o objeto, através do comando “Background”, definindo-se sua cor. Logo em seguida há a descrição do objeto em si, “geometry Cone” e seus atributos que são tamanho e largura da base.

Código VRML do Cone:

```
#VRML V2.0 utf8
Background {
    skyColor 0.5 0.5 0.9
}
Shape {
    geometry Cone {
        height 3
        bottomRadius 1.5
        side TRUE
        bottom TRUE
    }
}
```

A linguagem pode ser escrita com qualquer editor de textos e visualizada em um *browser*. Existem também pacotes para desenvolvedores com programas que verificam a sintaxe e corrigem o código.

Cada arquivo VRML (INTERNATIONAL STANDARD ISO/IEC 14772-1, 1997); (DIEHL, 2001):

- Estabelece as coordenadas para cada objeto definido no arquivo;
- Define e compõe um grupo 3D e objetos multimídia;
- Pode especificar conexões para outros arquivos e aplicações;
- Pode definir o comportamento dos objetos.

O VRML também pode ser utilizado com outras linguagens de programação, como Java, para implementar a troca de informações, sendo

necessário o uso de uma interface chamada EAI (External Authoring Interface). Através do EAI é realizada a comunicação entre Java e VRML.

2.1.2. REALIDADE VIRTUAL IMERSIVA E NÃO-IMERSIVA

Existem dois tipos de realidade virtual: a imersiva e a não-imersiva. A realidade virtual não-imersiva utiliza monitores, mouses e teclados para que o usuário possa interagir com o mundo virtual. Já a realidade virtual imersiva faz uso de luvas, capacetes, salas de projeção, etc, para que o usuário seja colocado “dentro” do mundo virtual (KIRNER, 1999).

Os equipamentos utilizados em sistemas de realidade virtual imersivos estão cada vez mais sofisticados, utilizando luvas com sensores (Figura 3) para dar a sensação de toque, peso, força, fazendo com que o usuário realmente sinta as mesmas sensações do mundo real no mundo virtual.

Os capacetes (Figura 4) utilizados apresentam imagens estereoscópicas, que emitem imagens diferentes para cada olho. O cérebro humano recebe a imagem de cada olho e processa as informações, causando a impressão de profundidade.

A combinação dos sentidos de visão, audição e tato causa o efeito de percepção no mundo virtual.



Figura 3 - Luva usada em realidade virtual (VREALITIES, 2005)



Figura 4 - Capacete usado em realidade virtual (VREALITIES, 2005)

2.1.3. IMAGEM ESTEREOSCÓPICA

Os capacetes ou óculos especiais utilizados na realidade virtual possuem imagens estereoscópicas, ou seja, imagem com impressão de profundidade. O sistema visual humano é baseado em duas imagens diferentes, sendo uma do olho direito e outra do olho esquerdo, cada olho vendo a mesma imagem de pontos diferentes. Essas imagens são processadas no cérebro humano e essa diferença de ponto de visão é o que gera a impressão de profundidade (KNIGHT, 1998); (MALETIC, 2001).

Para formar imagens 3D no computador são utilizados dois pontos de visão da mesma imagem, uma para o olho direito e outra para o olho esquerdo, cada olho possuindo uma visão diferenciada da mesma imagem. A fusão dessas imagens resulta uma cena tridimensional, com a sensação de profundidade. Para visualizar essa imagem no computador há a necessidade do uso de óculos especiais, pois o monitor irá apresentar essas imagens mescladas, estampadas na forma bidimensional. Os óculos especiais alternam as imagens do monitor para cada um dos olhos, sendo que cada uma será processada no cérebro, gerando, então, a imagem tridimensional, conforme a Figura 5. A fusão das imagens leva em conta os desvios recebidos em cada

olho, gerando uma visão estereoscópica, ou seja, a impressão de profundidade da imagem apresentada (RAPOSO, 2004).

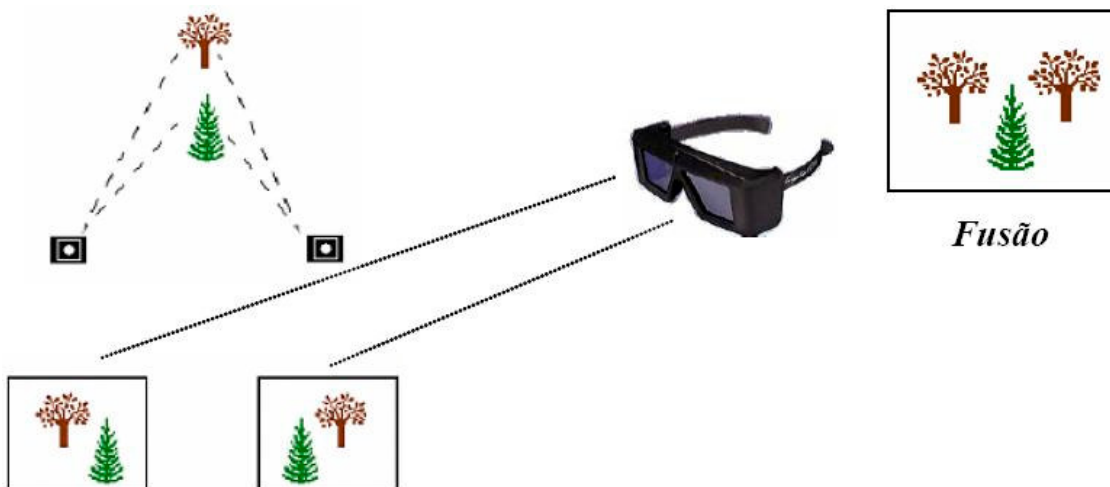


Figura 5 - Esquema do funcionamento da imagem estereoscópica (RAPOSO, 2004)

2.2. REALIDADE AUMENTADA

De acordo com Azuma (1995), a realidade aumentada teve seu início a partir da realidade virtual, onde o usuário interage com objetos virtuais imersos em um ambiente real. Por meio dessa tecnologia o usuário pode acrescentar ou remover objetos do ambiente.

A realidade aumentada adiciona ao mundo virtual informação e objetos do mundo real, podendo também, através de técnicas computacionais, remover objetos do mundo real (AHLERS, 1995); (AZUMA, 2001). As técnicas computacionais de realidade aumentada fazem com que o usuário manipule os objetos virtuais de maneira natural. O usuário recebe as imagens do mundo real por meio de câmeras colocadas no ambiente, e se vê imerso em uma representação gráfica, a qual combina imagens de vídeo do mundo real e imagens virtuais do sistema computacional, trazendo para o ambiente do usuário a visualização dos objetos do ambiente de realidade virtual.

A Figura 6 apresenta um exemplo de inserção de objetos virtuais no ambiente real, podendo-se adicionar objetos como um vaso e uma

miniatura de carro virtual em uma mesa real, enriquecendo, dessa forma, o ambiente.



Figura 6 - Exemplo de realidade aumentada - vaso e carro inseridos no mundo real (KIRNER, 2004)

Para inserir-se no ambiente aumentado, o usuário pode utilizar um capacete, recebendo as imagens diretamente nos olhos, ou visualizar o conjunto em um monitor de vídeo, da mesma forma como ocorre com a realidade virtual (BUK, 2005).

O sistema de realidade aumentada pode utilizar os mesmos dispositivos de hardware (monitor de vídeo, capacetes, óculos especiais) usados em realidade virtual (FEINER, 2002), além de câmeras para capturar as imagens do mundo real.

Ao fazer-se necessário o uso desses dispositivos, é preciso que o usuário possua treinamento para tal, pois sua posição e orientação influenciam na organização das informações contidas na imagem visualizada. Koller (1997) apresenta alguns problemas que podem ocorrer devido à colocação errada da câmera, da quantidade de luz incidente e de como as informações são inseridas e depois apresentadas ao usuário. Em seu artigo, o autor foca nos problemas relacionados à câmera e apresenta sua solução através de cálculos matemáticos. Dependendo de como for a visualização, não há a necessidade de treinamento, como mostrado neste trabalho. A visualização através de algumas ferramentas, como o ARToolKit, que é de

utilização simples faz com que o usuário realize a atividade de visualização naturalmente, pois a visualização está inserida no ambiente do próprio usuário.

Um grupo de pesquisadores utiliza um sistema chamado *Studierstube* com a finalidade de realizar pesquisas em realidade aumentada e realidade virtual de objetos tridimensionais (SCHMALSTIEG, 2002).

A realidade aumentada pode ser aplicada em diversas áreas como: medicina, manufatura, tele-robótica, manutenção e reparo de máquinas, militar, engenharia, dentre outros.

2.2.1. ALGUNS TIPOS DE SISTEMAS DE REALIDADE AUMENTADA

A maneira como é visualizada a realidade aumentada está relacionada a alguns de seus tipos (AZUMA, 2001), apresentados a seguir.

2.2.1.1. REALIDADE AUMENTADA BASEADA EM CAPACETE COM VISÃO ÓPTICA

O capacete combina a transmissão da imagem virtual com a imagem real projetando-as nos olhos do usuário (KIRNER, 2005). Um dos problemas no uso do capacete é o fato da quantidade de luz ser diferente da que está no mundo real, pois o capacete reflete parte dessa luminosidade. O capacete recebe imagens do mundo real através de um sistema óptico, verifica a posição da cabeça do usuário, gera a cena e a apresenta ao usuário, como mostrado no esquema da Figura 7 (AZUMA, 1997).

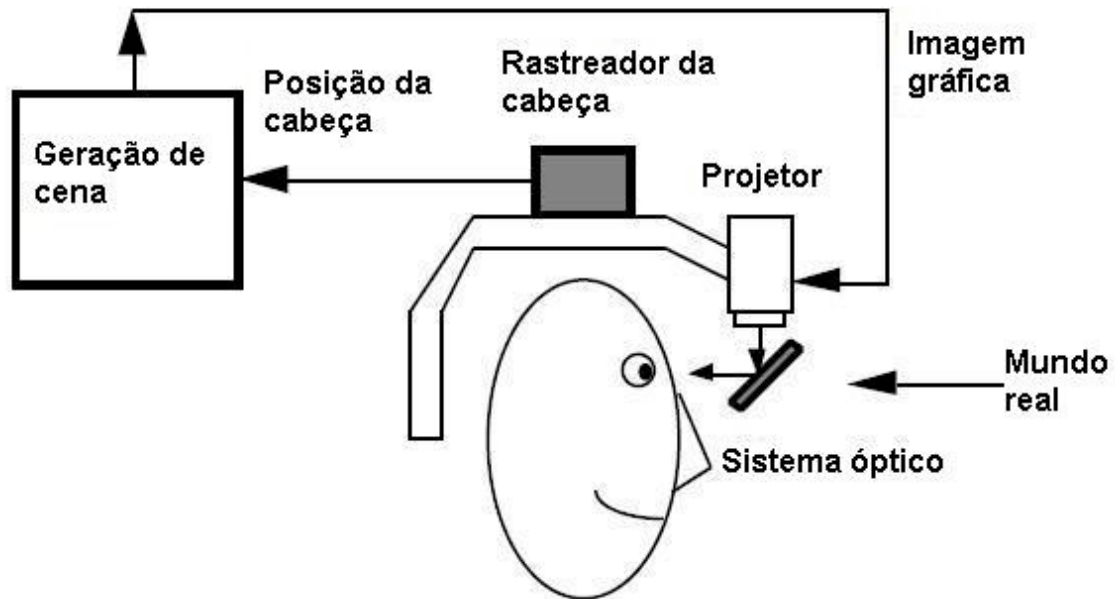


Figura 7 - - Esquema de visualização em realidade aumentada através do capacete com visão óptica (AZUMA, 1997) (KIRNER, 2005)

2.2.1.2. REALIDADE AUMENTADA BASEADA EM CAPACETE COM VISÃO POR VÍDEO

Para visualizar a imagem por meio de vídeo, são utilizadas câmeras de vídeo que capturam a imagem real, misturam essa imagem aos elementos virtuais utilizando processamento computacional e as enviam para o sistema de visualização óptica do usuário, conforme Figura 8 (AZUMA, 1997); (KIRNER, 2005).

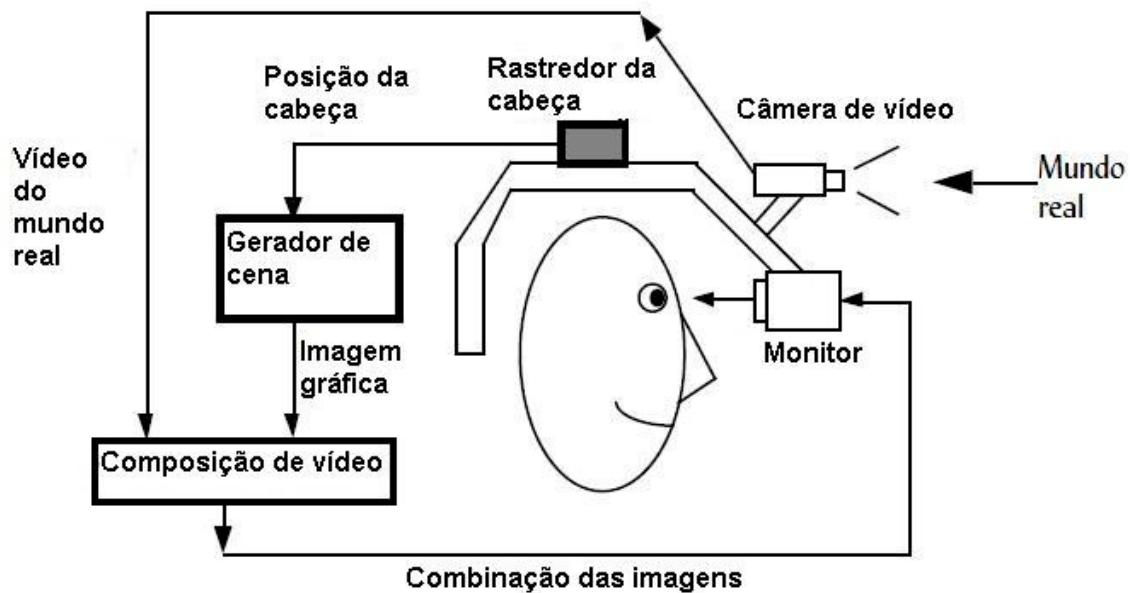


Figura 8 - Esquema de visualização em realidade aumentada através de capacete com visão por vídeo (AZUMA, 1997) (KIRNER, 2005)

2.2.1.3. REALIDADE AUMENTADA BASEADA EM MONITOR

No sistema de vídeo baseado em monitor, a imagem é gerada na estação de trabalho e apresentada em um segundo monitor de alta resolução. Um conversor transforma a imagem gráfica recebida da câmera em um formato padrão de vídeo, fazendo também a mistura entre o sinal de vídeo e o sinal da câmera, gerando a fusão da imagem do vídeo e do gráfico em tempo real. O resultado é apresentado em um monitor de vídeo padrão (Figura 9). O mouse e o teclado são dispositivos para interface do usuário (AZUMA, 1997); (KIRNER, 2005).

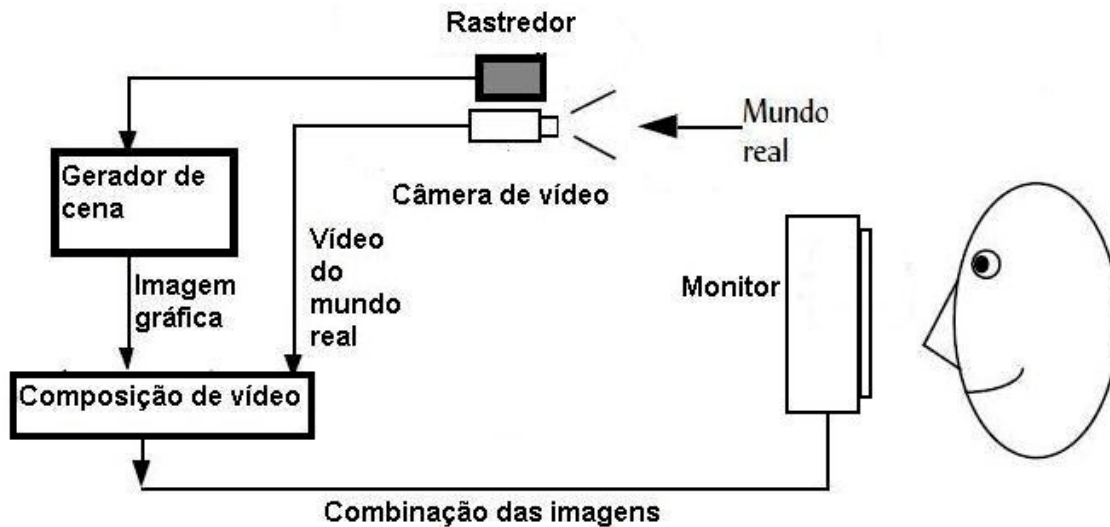


Figura 9 - Esquema de visualização em realidade aumentada através do monitor (AZUMA, 1997) (KIRNER, 2005)

Existe também um sistema de realidade aumentada utilizando projeção para a visualização do ambiente, no qual são utilizadas superfícies do ambiente real para apresentar os objetos virtuais sem equipamento auxiliar.

2.2.2. EXEMPLOS DE SISTEMAS DE REALIDADE AUMENTADA

2.2.3. STUDIERSTUBE

O sistema Studierstube, criado por Schmalstieg, é utilizado para estudar modelos tridimensionais em realidade aumentada, proporcionando uma interface de maior abrangência de visualização (BILLINGHURST, 2002); (SCHMALSTIEG, 2002). O sistema apresenta cinco características: trabalha com objetos virtuais, utiliza realidade aumentada, promove a cooperação, permite independência do usuário (cada usuário pode controlar seu próprio ponto-de-visão) e o usuário mantém sua individualidade (os dados são apresentados separadamente para cada usuário). A realidade aumentada é utilizada para gerar as imagens virtuais no mundo real através desse sistema.

Sua arquitetura é composta por realidade aumentada colaborativa, dessa forma, vários usuários podem usá-lo ao mesmo tempo. O

Studierstube também pode ser utilizado, através de redes onde cada usuário trabalha na mesma visualização separadamente (SCHMALSTIEG, 2000).

Na Figura 10 são apresentados dois usuários examinando uma imagem tridimensional.



Figura 10 - Exemplo do Studierstube (SCHMALSTIEG, 2002)

2.2.4. ARToolKIT

O ARToolKit é um sistema de desenvolvimento de aplicações de realidade aumentada, onde objetos virtuais são projetados no mundo real. Elementos de calibração reais e virtuais devem coincidir para que os ambientes real e virtual possam estar alinhados. Esse sistema possui algoritmos que calculam a posição da câmera de acordo com marcações colocadas no seu campo de visão (ARTOOLKIT, 2004); (KATO, 2000). Será dada maior ênfase a esse sistema, já que foi escolhido para o estudo.

O sistema de desenvolvimento ARToolKit é livre para uso não-comercial, sendo distribuído “open-source” sob licença GPL, ou seja, licença

pública geral a qual garante que o software seja gratuito e livre, podendo inclusive alterar seu código-fonte. A versão utilizada neste trabalho é a 2.65 com VRML, e está disponível no endereço:

http://www.hitl.washington.edu/research/shared_space/download/.

A Figura 11 mostra a estrutura dos arquivos do ARToolKit.

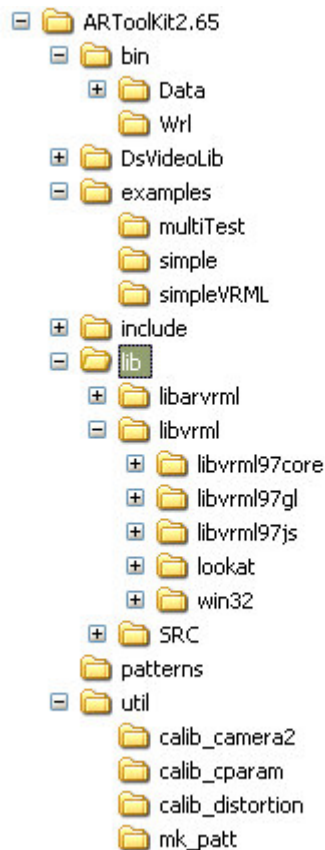


Figura 11 - Estrutura de arquivos do ARToolKit (ARTOOLKIT, 2004)

Para o ARToolKit funcionar corretamente são necessárias uma câmera de vídeo e um dispositivo que mostre as imagens misturadas ao usuário, podendo ser um monitor ou óculos de realidade aumentada, por exemplo. Conforme é executado o programa de visualização do ARToolKit, aparecem telas mostrando as saídas das imagens em formato de texto e logo em seguida a imagem propriamente dita.

Para capturar as imagens, o ARToolKit utiliza marcadores de referência, em formato de quadrado, onde cada marcador é relacionado a uma

imagem, conforme pode-se ver na Figura 12. Pode-se utilizar vários marcadores simultaneamente, pois cada marcador é relacionado a uma imagem separadamente.

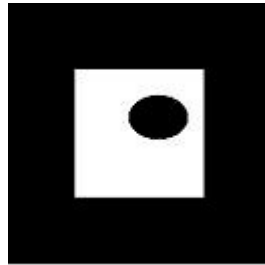


Figura 12 - Exemplo de marcador

Como pode-se ver na Figura 13, primeiro a câmera identifica os marcadores convertendo a imagem em binário, calculando em seguida sua posição e orientação em relação à câmera. Com a orientação e a posição calculadas, os marcadores são comparados a imagens existentes na memória, alinhando as imagens dos marcadores aos objetos virtuais. No vídeo de saída os objetos são renderizados, ou seja, processados, para, então, serem apresentados aos usuários através de seus dispositivos de visualização. Como na Figura 13, o marcador utilizado é reconhecido e sua posição em relação à câmera é calculada. Em seguida a imagem do cubo colorido é comparada àquela que está na memória do computador. Com a imagem alinhada ao marcador, a imagem é processada e apresentada no dispositivo de visualização (monitor de vídeo, óculos especial, por exemplo).

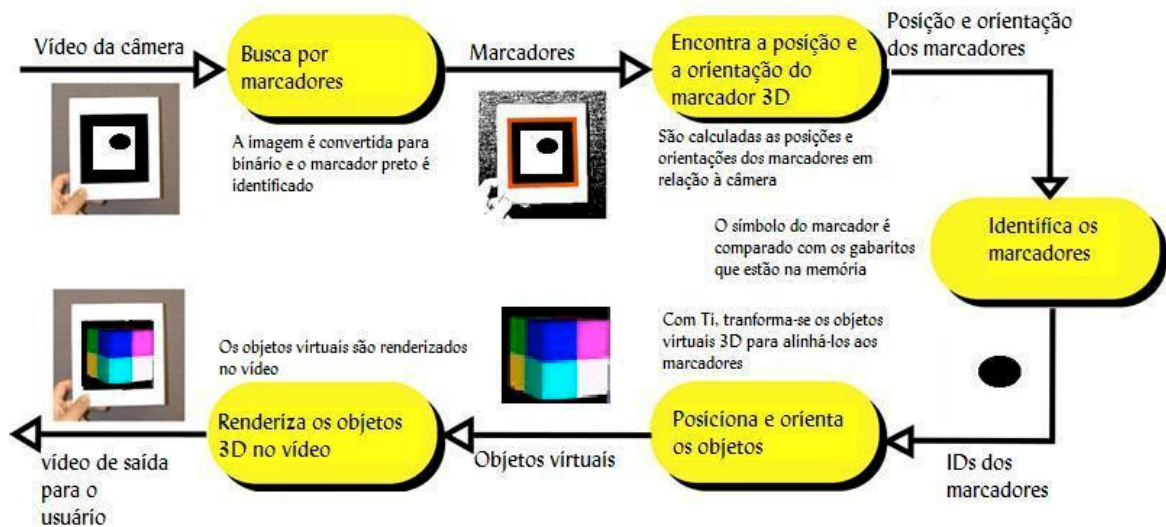


Figura 13 - Diagrama de processamento das imagens no ARToolKit (adaptada de KATO, 2000)

2.2.5. MXR TOOLKIT

O sistema MXR Toolkit não é vendido comercialmente. Assim como o ARToolKit, é um sistema aberto para fins não-comerciais. Consiste em uma biblioteca para ajudar a construir ambientes de realidade aumentada, possuindo várias rotinas para trabalhar imagens recebidas através de câmeras. Existem também soluções para técnicas de visualização as quais possuem rotinas matemáticas e geométricas, as quais o desenvolvedor do ambiente pode acrescentar à biblioteca (MXR Toolkit).

Nesse sistema é possível inserir pessoas no ambiente virtual, conforme mostrado no exemplo da Figura 14, onde as pessoas posicionam-se em frente à câmera e são inseridas no ambiente de realidade virtual, podendo interagir com os objetos virtuais. Neste exemplo pode-se observar que a pessoa está inserida em um museu virtual.



Figura 14 - Exemplo de inserção de pessoas no ambiente virtual (MXR Toolkit)

Os marcadores utilizados nesse sistema são os mesmos do ARToolKit. No entanto, sua utilização é diferente pois as pessoas são inseridas em um ambiente não-real.

Na Figura 15 pode-se ver como é feita a captura das imagens, onde inicialmente existem algumas câmeras que capturam a imagem da pessoa. Essas imagens são salvas e uma outra câmera captura a imagem do marcador. Então essas imagens são processadas no computador e finalmente misturadas, realizando então a inserção da pessoa no novo ambiente.

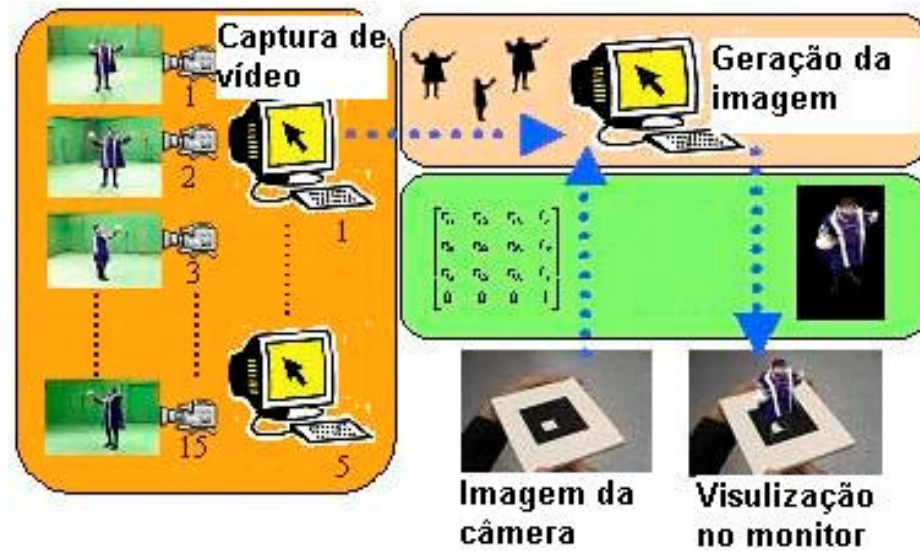


Figura 15 - Funcionamento do sistema MXR Toolkit (MXR Toolkit)

Também podem ser utilizados alguns artifícios para transportar a pessoa para outro ambiente virtual através do uso de uma pá, onde o usuário coloca a pá no ambiente atual onde está a pessoa e a retira, inserindo-a então em outro ambiente totalmente diferente.

CAPÍTULO 3

3. VISUALIZAÇÃO DA INFORMAÇÃO

Atualmente as informações são apresentadas em forma de documentos, normalmente extensos, exigindo do usuário sua leitura e entendimento. A visualização da informação por meio de imagens possibilita ao usuário melhor interpretação das mesmas (GERSHON, 1998).

Segundo pesquisas (NAKAKOJI, 2001); (HAWKINS, 1999), os usuários possuem maior facilidade em visualizar informações em gráficos animados, pois chamam mais atenção e facilitam a visualização da mudança dos dados no decorrer do tempo.

Com a visualização da informação é possível unir a capacidade do ser humano em interpretar gráficos e a qualidade das informações geradas por computador, otimizando o processo de interpretação dos dados.

3.1. MODELAGEM DA INFORMAÇÃO

De acordo com Crapo (2000), a visão humana possui mecanismos para enviar informações multidimensionais ao cérebro, podendo ser processadas várias informações sobre o objeto visualizado, como: movimentação, cor, profundidade, intensidade, tamanho, direção, distância. Dessa forma é realizado o modelo cognitivo do objeto apresentado.

A modelagem de informação baseia-se no processo de formação de um modelo cognitivo a partir de dados. A visualização é uma atividade realizada por sujeitos humanos, usando uma imagem ou lembrança que fica guardada na memória (SPENCE, 2001); (BUK, 2004c). O computador pode facilitar o processo de visualização, através de ferramentas que transformam os dados para que o usuário possa vê-los, por meio de gráficos.

Para realizar a modelagem da informação, é necessário obter os dados, organizá-los e representá-los. A modelagem de informações é uma atividade antiga e deve ser realizada através do estudo e das relações entre as informações. Atualmente, a modelagem é feita por meio de ferramentas específicas de visualização de informação, as quais devem ser transparentes ao usuário (CRAPO, 2000).

3.2. CLASSIFICAÇÃO DA INFORMAÇÃO

De acordo com Shneiderman (1996), a visualização da informação pode ser classificada em sete tipos: unidimensional, bidimensional, tridimensional, multidimensional, sendo estes três do tipo espacial e, temporal, hierárquica e rede, que são do tipo estrutural (CARR, 1999); (BUK, 2004a). As classificações do tipo escritural não serão abordadas neste trabalho. Esta classificação refere-se à quantidade de características representadas na visualização das informações, tais como: cor, tonalidade, altura, largura, altitude, longitude, dentre outros, sendo de grande interesse para a visualização de dados.

Os tipos de visualização podem ser utilizados tanto para classificar o ambiente no qual os dados estão inseridos, como também para classificar as informações. Neste trabalho, a classificação refere-se às informações utilizadas, pois o ambiente no qual estão inseridas é tridimensional. A classificação, portanto, refere-se às informações apresentadas.

A visualização unidimensional é baseada em um único elemento, organizado de maneira seqüencial, localizado em uma posição, focando algum elemento no monitor, como um conjunto de arquivos, um código fonte de um programa, documentos contendo textos, lista de nomes. A interface desse tipo de visualização pode conter elementos com características de tipos de fontes de letras, cores ou tamanhos.

A visualização bidimensional trabalha dois aspectos: a posição que o elemento está e um atributo. O atributo pode ser a altura, cor, largura.

Esse tipo de visualização é muito utilizado em gráficos (por exemplo: gráfico de barras ou pizza) e plantas de prédios.

Na visualização tridimensional, é necessário trabalhar três elementos, os dois aspectos da visualização bidimensional e o elemento que representa o volume do objeto em estudo. É muito utilizado para estudos do corpo humano, moléculas, átomos e projetos de arquitetura.

Para algumas representações, as visualizações unidimensionais e bidimensionais não são adequadas. Por exemplo, para visualização de objetos do mundo real, como o corpo humano, então utiliza-se a visualização tridimensional, que além dos elementos da visualização bidimensional, possui mais um elemento, normalmente trazendo a noção de volume. Nesse tipo de visualização tem-se a aplicação da realidade virtual, sendo a mesma propícia a representar, por exemplo, informações bidimensionais em gráficos de barras dispostos no mundo virtual.

Através da adição de atributos aos dados em gráficos bidimensionais tem-se a visualização multi-dimensional. Esses atributos podem ser: tonalidade, forma, cor, como ocorre com a representação de alguns bancos de dados e estatísticos.

3.3. REPRESENTAÇÃO DE INFORMAÇÕES

A representação de informações faz com o usuário as entenda de forma mais simplificada. Essa representação é normalmente realizada através de gráficos discretos bidimensionais ou tridimensionais, nos quais se pode acrescentar outros atributos além de altura, largura e profundidade. Esses outros atributos podem ser cor, tonalidade, transparência, pulsação de cor, pulsação do objeto do gráfico (pulsação das barras como elemento de animação), dentre outros elementos passíveis de ser incorporados.

Os elementos visuais representam as informações atualizadas ou até mesmo seu histórico, mostrando tendências. É possível também

incorporar elementos sonoros para sinalizar alguma irregularidade ocorrida (NAKAKOJI, 2001).

3.4. FERRAMENTAS DE VISUALIZAÇÃO DE INFORMAÇÕES

3.4.1. SEESOFT

O sistema SeeSoft foi desenvolvido com a finalidade de ajudar na visualização das informações através do uso de arranjos coloridos com linhas e colunas.

O SeeSoft é uma ferramenta que analisa linhas de códigos, relacionando os documentos, mapeando-os e colorindo cada linha de acordo com a modificação que foi realizada no código. Também reduz o arquivo em uma representação compacta sendo utilizado para fazer controle de versões de código e sua análise, verificando as chamadas das funções (GERSHON, 1998); (KNIGHT, 2001).

Por meio das características e análises feitas nos documentos por essa ferramenta, junto com interações gráficas e técnicas de manipulação, o usuário é capaz de tomar decisões (EICK, 1992a).

Com o SeeSoft o programador fica sabendo quais são os arquivos do programa e quando foram modificados, facilitando o gerenciamento do projeto.

Esta ferramenta é um exemplo de visualização de informação tridimensional, por trabalhar três aspectos: cor, linha e coluna para a representação.

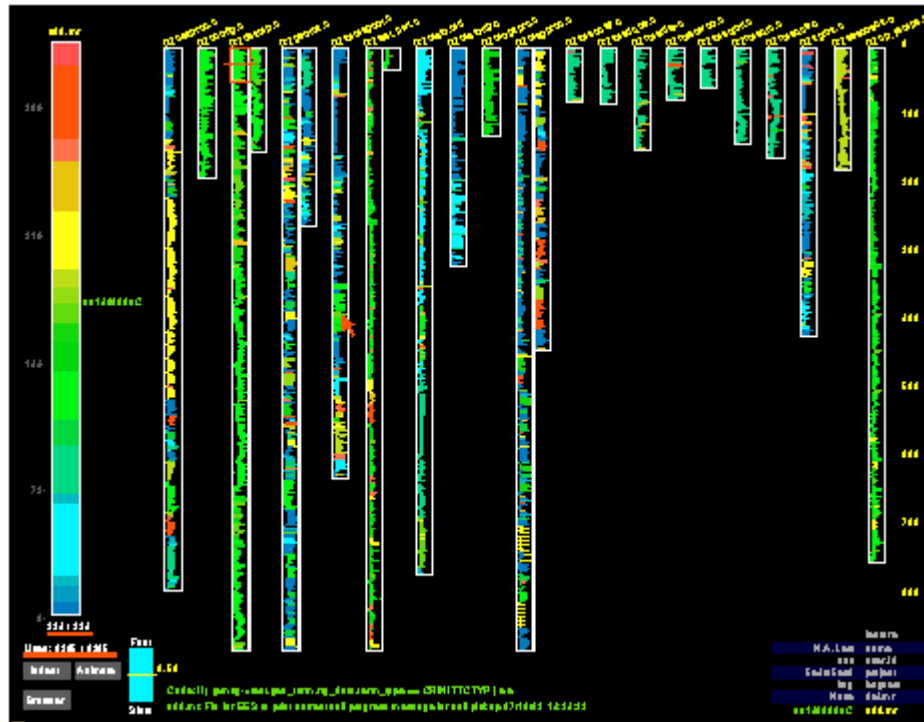


Figura 16 – SeeSoft (EICK, 1992b)

A Figura 16 mostra uma tela do sistema SeeSoft, onde cada barra colorida representa um arquivo de código de programação, sendo cada linha do programa representado pelas linhas de cada barra. As cores das barras mostram as modificações realizadas nos códigos.

3.4.2. INFORMATION MURAL

Este sistema de visualização de informações faz a representação das informações por meio de atributos de escala, intensidade, cor, textura, tamanho do pixel, podendo ser utilizado para visualizações isoladas ou para navegação global da visualização das informações. Essa técnica mostra a densidade real da informação, analisando a intensidade do pixel, sendo capaz de lidar com grande quantidade de informações on-line (JERDIN, 1998); (GEISLER, 1998).

Alguns dos tipos de informações com as quais o Information Mural trabalha são: arquivos de textos, gráficos, outros programas de visualização de informações e imagens.

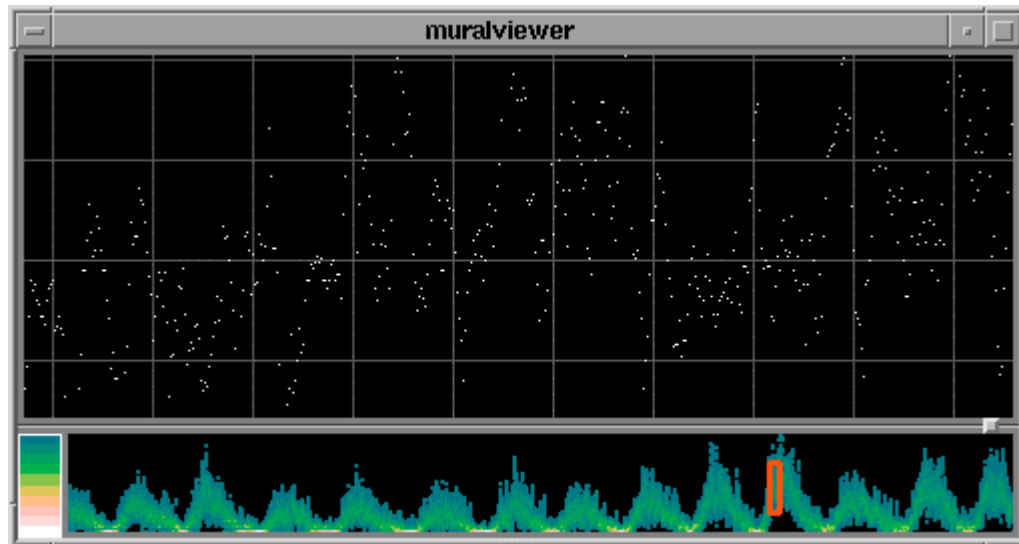


Figura 17 - Information Mural (JERDIN, 1998); (GEISLER, 1998)

Como exemplificado pela Figura 17, esse sistema pode representar um conjunto de informações de forma ampla e ao mesmo tempo, através de um gráfico, cada área específica.

O Information Mural é um exemplo de um sistema de visualização multidimensional por ser utilizado para visualizar gráficos, conferindo-lhes vários atributos, como: escala, intensidade, cor, textura, tamanho do pixel.

3.4.3. TILE BARS

A informação desse tipo de software é baseada em textos, os quais devem conter os resumos para facilitar a busca (HEAST, 1995).

O Tile Bars é utilizado em pesquisas de grandes coleções de documentos, sendo realizadas buscas através de comparações e similaridades com os textos procurados. Essas informações utilizam termos de conjunções, disjunções e negações em textos e mostram apenas os que possuem as

palavras procuradas, fazendo a relação entre essas palavras e os documentos (HEAST, 1995).

Este sistema mostra o tamanho do documento, a frequência com que o termo aparece no documento e sua distribuição, facilitando a visualização das informações procuradas num texto ou mídia.

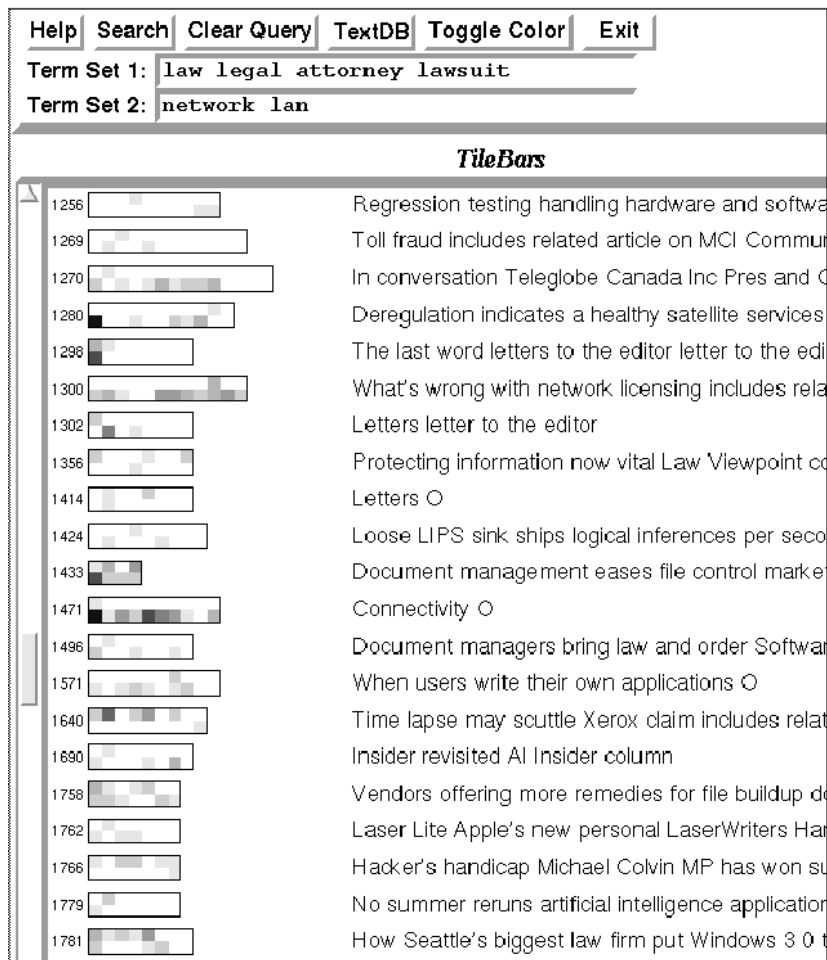


Figura 18 - Tile Bars (HEAST, 1995)

O exemplo da Figura 18 mostra um sistema de Tile Bars onde cada retângulo corresponde a um documento e os quadrados representam cada segmento de texto. Os quadrados podem variar de tom de cor de acordo com a frequência da palavra no texto. Em frente ao retângulo aparecem as palavras iniciais do texto que o documento contém.

O sistema de visualização Tile Bars pode ser classificado como bidimensional por utilizar duas características para sua visualização, o tamanho do texto e a freqüência com que os termos aparecem no mesmo.

3.4.4. DOCUMENT LENS

Esse sistema representa os documentos por meio de vários mapas como se fossem páginas de um texto em uma figura tridimensional, mostrando-as em uma estrutura global que identifica cada uma delas (GEISLER, 1998). O Document Lens é utilizado apenas para visualizar informações de texto, pois sua técnica é específica para esse tipo de documento.

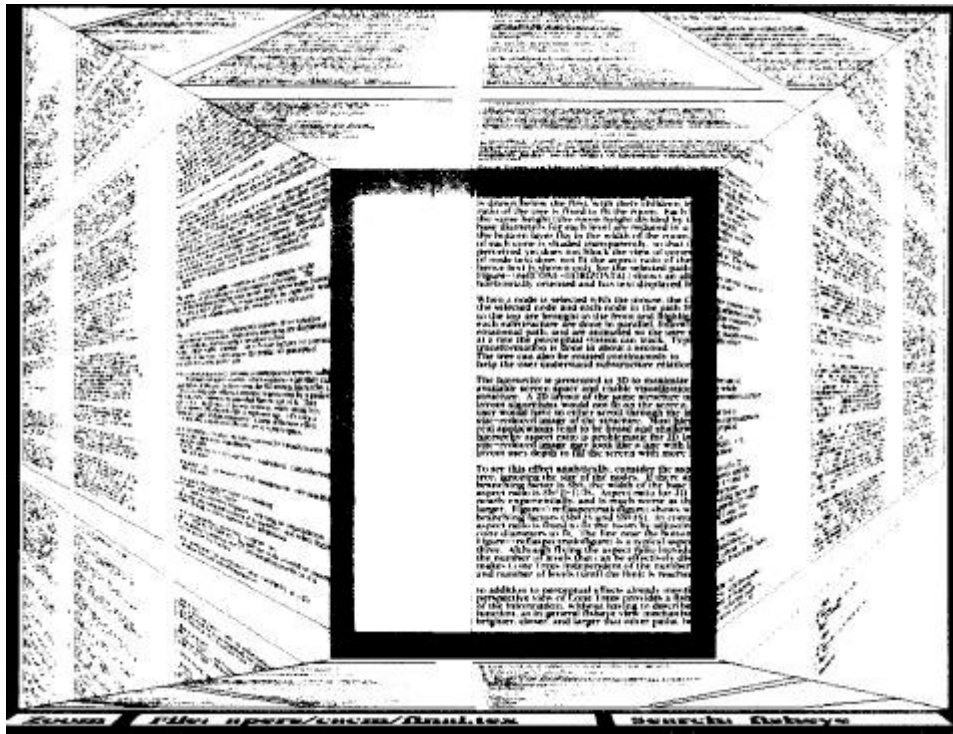


Figura 19 - Document Lens (ROBERTSON, 1993)

O Document Lens, como pode ser visto na Figura 19, funciona como uma “lente de aumento” que focaliza uma parte específica do documento que o usuário deseja visualizar. No entanto, o usuário continua tendo uma ampla visão do documento sem perder o contexto global. Essa técnica passa a

impressão de informações estampadas em um muro. A parte que o usuário deseja ver aparece mais próxima, como se estivesse com uma lente de aumento, e as outras partes do documento aparecem mais distantes, continuando visíveis (ROBERTSON, 1993).

O Document Lens é um sistema de visualização unidimensional, utilizado para representar informações tipo textos com “lentes de aumento”.

3.4.5. GEOGRAPHIC INFORMATION SYSTEM

Sistema que representa e analisa as informações referentes aos mapas geográficos (GEISLER, 1998); (OLIVEIRA, 1997). É capaz de armazenar, manipular e disponibilizar uma grande quantidade de informações referentes à geografia de uma determinada região obtendo informações de mapas existentes, fotos de satélites e censo demográfico. Com essa combinação de informações, o Geographic Information System constrói os mapas e analisa as novas variáveis, sendo possível determinar, por exemplo, a utilização de recursos hidrográficos (USGS – U. S. GEOLOGICAL SURVEY, 2002). Para inserir manualmente dados sobre um determinado local é necessário informar sua latitude, longitude e altitude e com isso o sistema analisa e monta o gráfico dessa região.

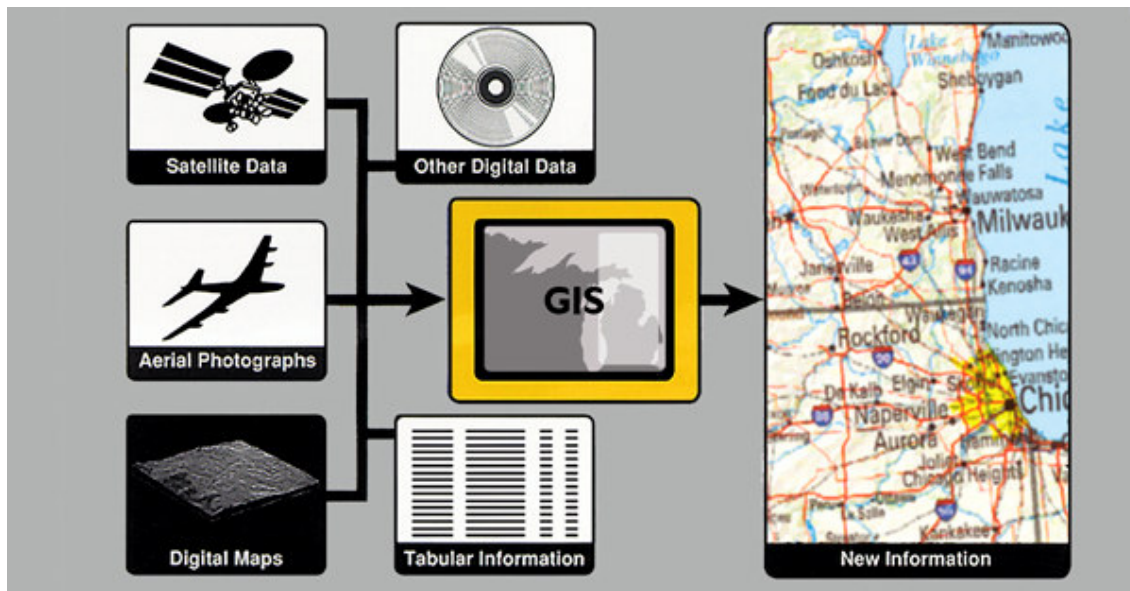


Figura 20 - Geographic Information System (USGS – U. S. GEOLOGICAL SURVEY, 2002)

Nesse sistema pode-se notar que os dados são coletados de diversas formas, conforme mostrado na Figura 20, sendo enviados para o sistema realizar o mapeamento do local.

O Geographic Information System pode ser classificado como um sistema Tridimensional ou Multidimensional, pois representa e analisa informações geográficas de mapas através de cores, latitude, longitude e até tonalidades de cores.

3.4.6. NIRVE – NIST INFORMATION RETRIEVAL VISUALIZATION ENGINE

Esse sistema foi desenvolvido para pesquisas em motores, através do uso de palavras-chave. Os documentos são organizados em grupos para facilitar a busca (NIRVE, 2003).

O NIRVE possui um menu para controlar as informações, pesquisar e visualizar documentos. Possui uma janela que mostra as informações como uma representação abstrata. O usuário é capaz de movimentar essa janela da forma que achar melhor, fazendo com que fique mais confortável a sua visualização.

As informações são apresentadas ao usuário através de ícones, os quais contêm os dados referentes a posições, como latitude e longitude de um objeto.

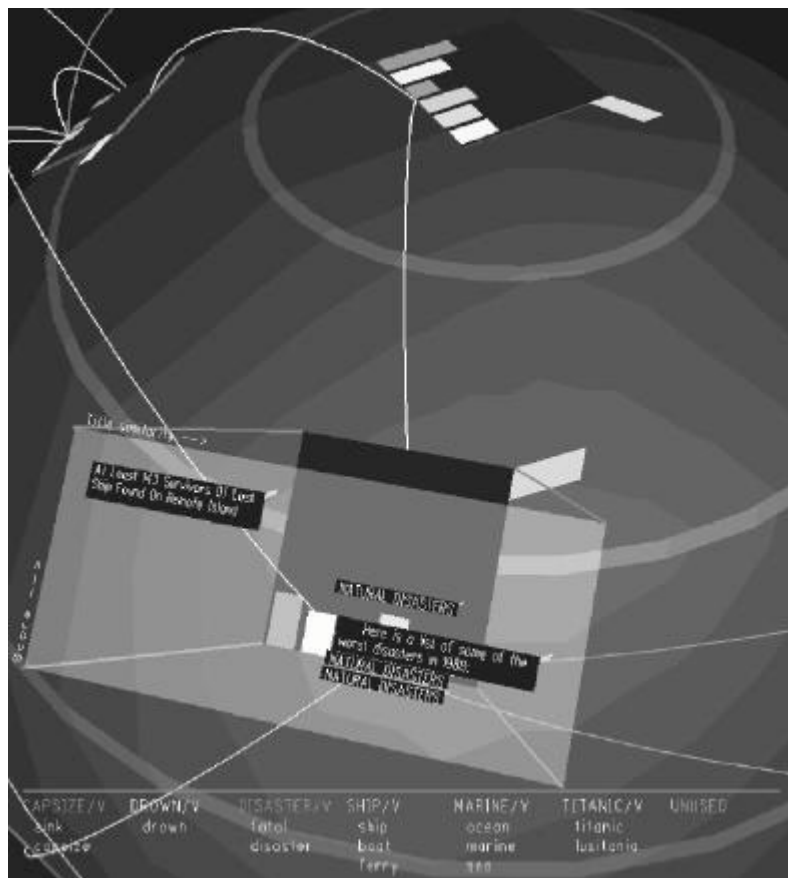


Figura 21 – Nirve (SEBRECHTS, 1999)

Conforme apresentado na Figura 21, um retângulo tridimensional é projetado para fora da superfície da esfera, mostrando o título do documento.

O NIRVE é um exemplo de sistema tridimensional. Serve para visualizar informações de objetos do mundo real, e é utilizado para visualização de informações de motores e posição de navios, por exemplo.

3.4.7. TABLE LENS

Desenvolvido pela empresa Xerox PARC, o Table Lens trabalha com informações de texto, bancos de dados e representações gráficas (GEISLER, 1998). É utilizado para visualizar informações contidas em grandes tabelas e baseia-se na interação entre o foco e o contexto das informações, integrando símbolos e representações gráficas em uma mesma forma de visualizar as informações (RAO, 1994). O Table Lens pode trabalhar com diferentes tipos de representações gráficas: texto, cor, diferentes tonalidades, tamanhos e posições.

Esta ferramenta faz a interação entre grandes estruturas de informação em um formato dinâmico, no qual o usuário pode colocar seus próprios padrões e características gráficas. A combinação do foco, contexto e esquema gráfico fornecem ao usuário uma alta interatividade com a informação disponível.

Table Lens: Baseball Player Statistics

Calculate:

Players	At	Hi	Ho	Ru	Rb	Wa	Ye	Career At	Career Hit	Co	Ca	Ca	Ca	Le	Di	Te	Po	Fu	As	Bt	Sa	Le	Te
Pete Rose								14053	4256														
Charlie Moore								3926	1029														
Bob Dernie								1931	491														
Mike Brown								853	228														
Tony Azmas								28	4														

Row: 277: Steve Lombardozzi:
 Column: 10: Career Home Runs:
 Value: 8

74 --- ????

Figura 22 - Table Lens (PIROLI, 1996)

A Figura 22 mostra uma Table Lens, onde cada coluna representa o desempenho de um jogador de baseball.

O sistema Table Lens, por utilizar informações de tabelas relacionais, interagindo e representando essas informações, pode ser caracterizado como um sistema multidimensional.

3.4.8. ADVIZOR AUTHORIZING TOOLKIT

O Advizor Authoring Toolkit possui interação com visualização de dados, sendo utilizado em Business Intelligence (BI), Customer Relationship Management (CRM) e outros sistemas de análise de informações (ADVIZOR SOLUTIONS INC, 2003). Representam-se as informações de várias formas: barras, histogramas, constelações, linhas, gráficos diversos. Com essas combinações o usuário é capaz de visualizar e avaliar as informações de diversas maneiras, o que lhe permite tomar decisões de forma rápida e eficiente.

Esse software é capaz de fazer um protótipo do ambiente e mostrá-lo com uma série de recursos visuais.



Figura 23 - Advisor Authoring Toolkit (ADVIZOR SOLUTIONS INC., 2003)

A Figura 23 mostra as formas de visualização da informação do sistema Advisor Authoring Toolkit, onde a análise dos dados é feita por meio de vários tipos de gráficos (barras, tridimensional, torta, histograma, etc.).

Mesmo possuindo várias formas de gráficos para representar as informações (solução que pode ajudar a tomada de decisão), existe a deficiência em interagir com os mesmos, mudando seu ponto de vista, navegando entre as informações.

O sistema de visualização Advizor Authoring Toolkit pode ser classificado como uma visualização desde unidimensional, bidimensional, tridimensional e até multidimensional dependendo do tipo de gráfico utilizado.

3.4.9. OPEN VIZ

O OpenViz é uma ferramenta desenvolvida para ajudar na tomada de decisões, utilizando gráficos com várias opções de visualização de informações (WAYNE, 2003). Assim como o Advisor Authoring Toolkit, também é um sistema utilizado em Business Intelligence (BI), Customer Relationship Management (CRM) e outros sistemas de análise de decisão.

Essa ferramenta é capaz de criar apresentações e vários tipos de gráficos, como de barras, pizza, linha, explorando as informações, apresentando-as em cenas interativas, onde o usuário pode ter a liberdade de explorar, questionar e pesquisar sobre as informações que necessita (WAYNE,2003).

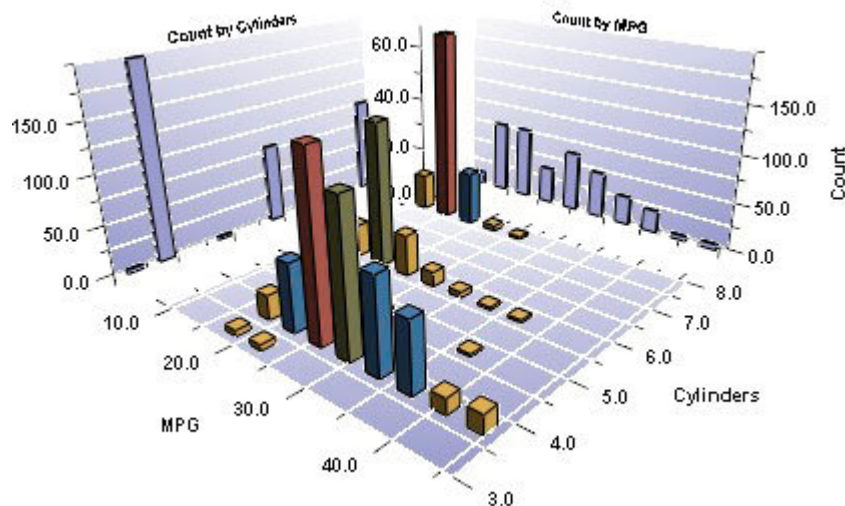


Figura 24 - Open Viz (WAYNE)

O sistema Open Viz possui a capacidade de lidar com vários tipos de informações, como o exemplo mostrado na Figura 24, no qual o sistema trabalha com parâmetros específicos, sendo esses: barras, cores, legendas.

Assim como o Advizor Authoring Tollkit, o Open Viz também pode ser um sistema tridimensional, por trabalhar com as variáveis cores, barras e legendas.

3.4.10. OPEN DX

Software desenvolvido pela IBM para visualização e análise de informações (IBM, 1999). Disponibiliza ferramentas para manipulação, transformação, processamento, animação de documentos, além de vários métodos de exploração e análise de informações.

Esse sistema dispõe de técnicas de geração de apresentação de informações gráficas, histogramas, gráficos 2D, 3D e figuras geométricas. Na Figura 25 pode-se verificar a visualização do estudo da camada de ozônio.

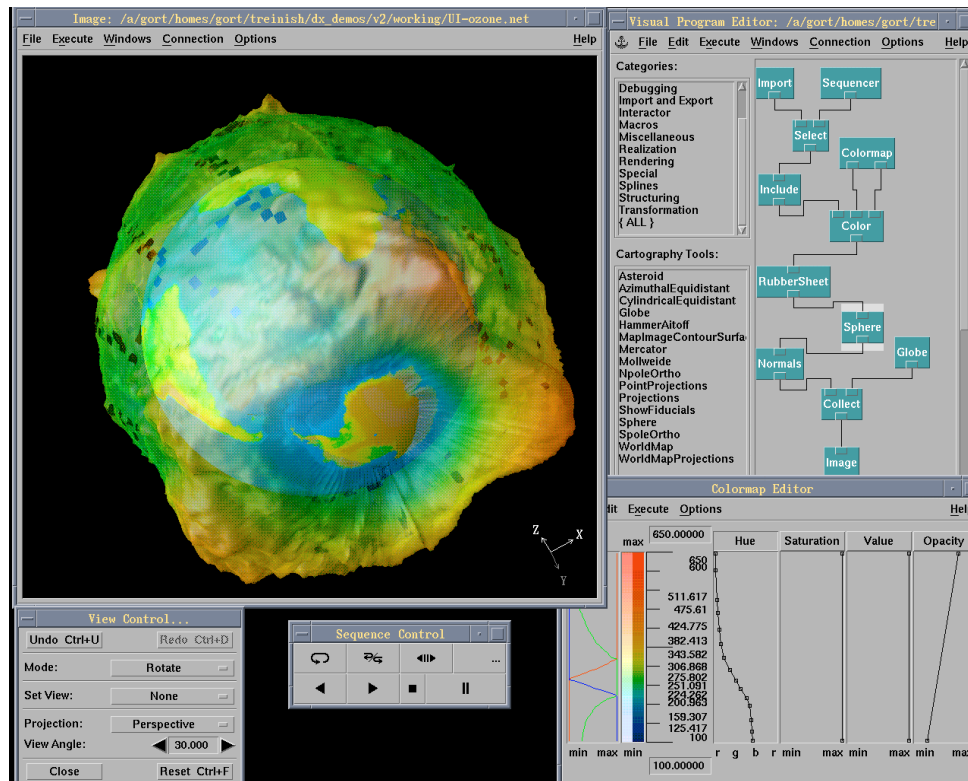


Figura 25 - Sistema de visualização de informações Open DX (IBM, 1999)

A ferramenta Open DX é outro exemplo de um sistema multidimensional por representar informações do mundo real com volume (altura, largura, profundidade) e também com a utilização de cores e suas tonalidades.

3.5. FERRAMENTAS DE VISUALIZAÇÃO DE INFORMAÇÃO AVALIDADAS NESTE TRABALHO

3.5.1. 3D ACTIVE CHART

A ferramenta 3D Active Chart é um visualizador tridimensional de informações, no qual as informações são obtidas através de banco de dados, cuja extensão é dbf, ou seja, utiliza-se o banco de dados em Dbase (ACTIVE CHART, 2004).

A migração dos dados para o visualizador é realizada através da conexão ODBC e comandos SQL (Structured Query Language – Linguagem Estruturada de Manipulação de Dados).

A atualização das informações é feita no próprio banco de dados e para mostrá-las no 3D Active Chart, é necessário salvar as alterações no banco de dados e em seguida atualizar os dados do visualizador.

O gráfico do 3d Active Chart é feito em WRML, linguagem de realidade virtual, mostrando as informações através de gráficos tridimensionais contendo barras coloridas e com diferenciados formatos, como por exemplo: cones, cubos e cilindros, como mostrado nas Figuras 26 e 27 a seguir. É possível ao usuário modificar o gráfico, alterando-se as informações que serão mostradas, a ordem de apresentação das mesmas, se serão utilizadas outras formas de barras (cubos, cones, cilindros), ou somente será apresentada a barra em um formato, se o gráfico terá várias cores diferenciadas, ou será de uma mesma cor.

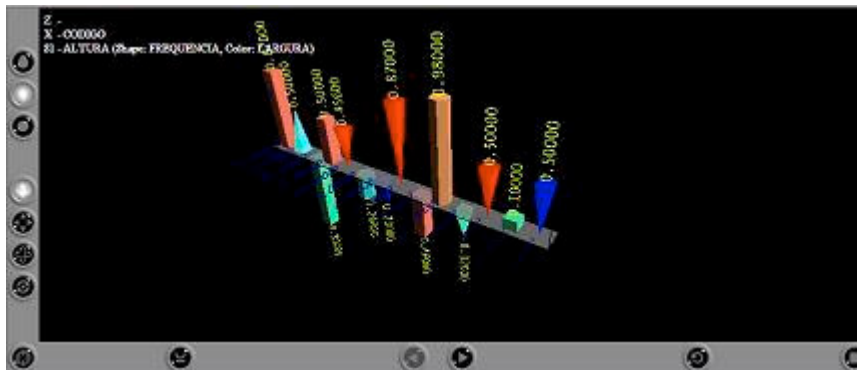


Figura 26 - Gráfico do 3D Active Chart (ARCTIVE CHART, 2004)

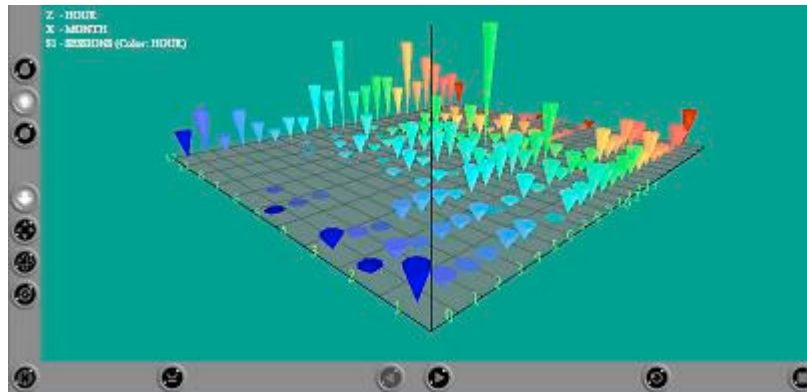


Figura 27 - Gráfico do 3D Active Chart (ARCTIVE CHART, 2004)

3.5.2. NCHART EDITOR 2.0

O visualizador NChart Editor mostra as informações através de um gráfico pré-selecionado pelo usuário. Ao criar um novo arquivo, é necessário antes escolher o tipo de gráfico a ser usado. Os tipos são variados, como barra, colunas, linha, gráfico de torta, pontilhado, radar, dentre outros (NCHART EDITOR, 2004).

As informações são inseridas em tabelas existentes no próprio programa, onde são editados as categorias, séries e níveis dos componentes. Dessa forma são criadas as colunas e seus respectivos valores. Existe a possibilidade de inserir mais de um valor na mesma barra, sendo este mostrado através de diferentes cores. Os dados ficam armazenados no próprio programa.

Para criar as imagens apresentadas, o NChart Editor utiliza linguagens de programação que podem ser escolhidas pelo usuário, como: Visual Basic, VBScript, JScript, C++ e C#.

Na Figura 28 é demonstrado um exemplo no qual o gráfico possui cinco barras, onde cada barra possui três valores que podem ser alterados através da tabela ao lado do gráfico. O gráfico é atualizado ao mesmo tempo em que seus dados são alterados.

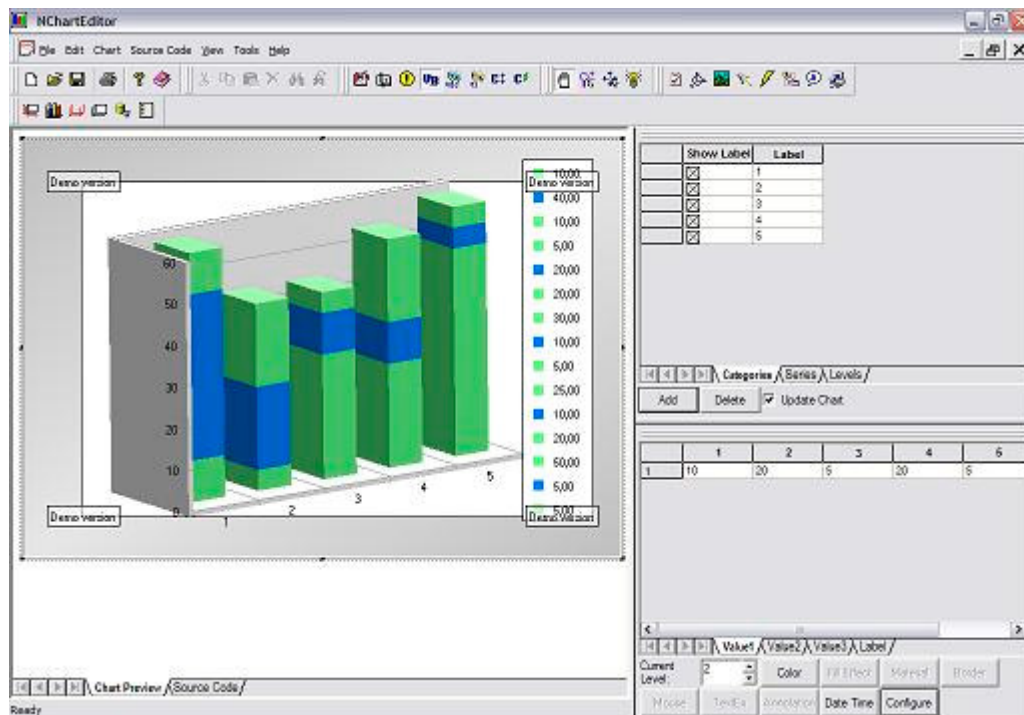


Figura 28 - Exemplo do NChart Editor (NCHART EDITOR, 2004)

3.5.3. GANNALYST LITE 3.0 STANDARD

É uma ferramenta utilizada para analisar informações técnicas do mercado financeiro. Trabalha com gráficos de linhas, barras, vela e “gann swing”. Por meio das propriedades do gráfico, podem-se alterar os modos de visualização, exportar o gráfico para arquivo de texto, mudar o estilo do gráfico (GANNALYST, 2002).

O Gannalyst é uma ferramenta que trabalha com a análise dos dados. Contém alguns objetos que ajudam nessa análise, como: inserção de linhas, indicadores (símbolos) e ferramenta de texto. Conforme mostrado na Figura 29, possui várias formas de mostrar as informações. Os dados são armazenados em uma tabela própria, onde o usuário os insere.

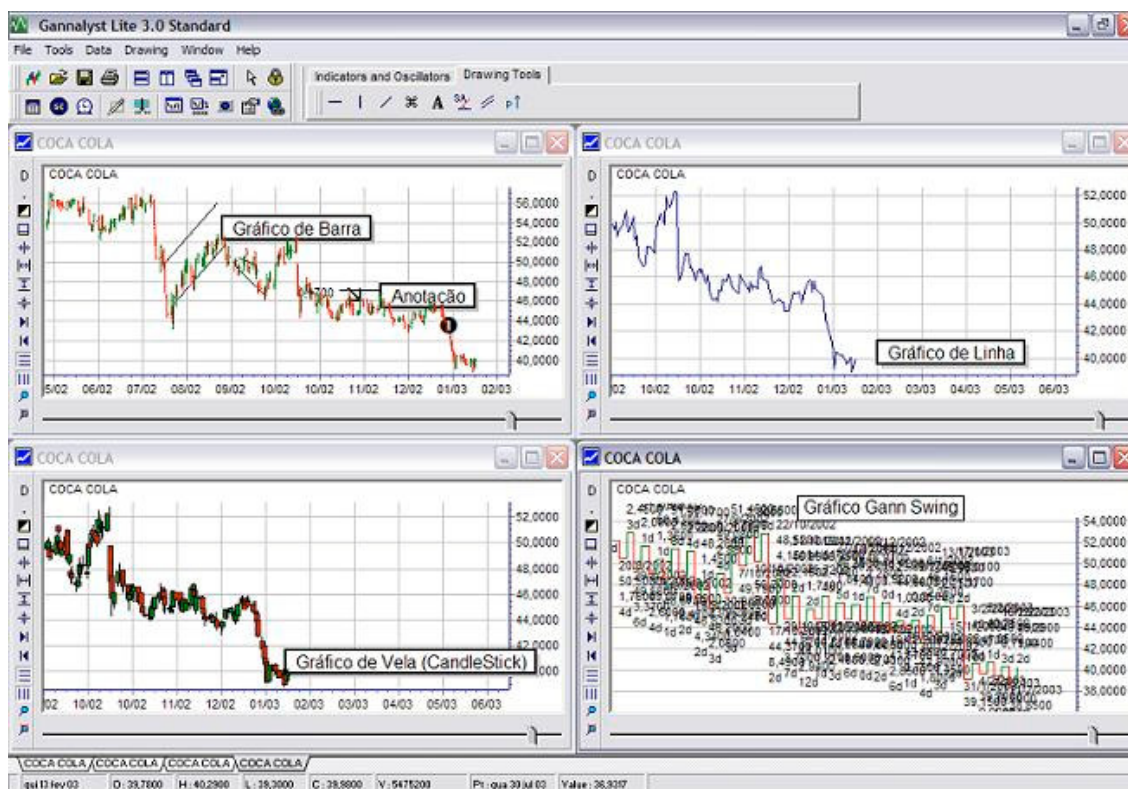


Figura 29 - Ferramenta Gannalyst e seus gráficos (GANNALYST, 2002)

3.6. COMPARAÇÃO DAS FERRAMENTAS TESTADAS

Como pode-se observar na Tabela 1, as três ferramentas testadas (3D Active Chart, Nchart Editor 2.0 e Gannalyst Lite 3.0 Standard) são visualizadores tridimensionais, trabalham com gráficos 3D, mas no entanto cada uma possui sua particularidade.

A inserção das informações no 3D Active Chart é realizada através de uma tabela, cujo histórico é armazenado em um banco de dados da ferramenta. O Nchart Editor 2.0 e o Gannalyst Lite não possuem banco de dados, uma vez que as informações são inseridas e mantidas nas tabelas existentes.

Cada sistema possui seu tipo de gráfico. O 3D Active Chart trabalha com gráficos VRML, o Nchart Editor 2.0 utiliza gráfico de barras e o Gannalyst Lite possui quatro tipos de gráficos: linhas, barras, vela e “gann swing”. Em todas essas ferramentas, o usuário pode modificar o gráfico de

alguma forma, seja inserindo mais elementos gráficos, como no Gannalyst Lite, seja através de mudanças no formato das barras no 3D Active Chart, ou até pré selecionar o gráfico a ser utilizado, como no Nchart Editor 2.0.

Tabela 1 - Comparação das Ferramentas Testadas

3D Active Chart	NChart Editor 2.0	Gannalyst Lite
<ul style="list-style-type: none"> • Visualizador tridimensional • Atualização de dados feita pelo Banco de Dados • Gráfico VRML com barras coloridas e em vários formatos • Usuário modifica o gráfico 	<ul style="list-style-type: none"> • Visualizador tridimensional • Inserção de dados através de tabelas • Gráfico com cores diferentes na mesma barra • Usuário faz uma pré seleção do tipo de gráfico 	<ul style="list-style-type: none"> • Visualizador tridimensional • Inserção de dados através de tabelas • 4 tipos de gráficos • Usuário pode inserir elementos gráficos para ajudar na análise dos dados

A comparação das ferramentas, apresentada na Tabela 1, mostra que elas apresentam diferentes graus de interatividade com o usuário e capacidade de alteração dos gráficos. No NChart Editor 2.0 e no Gannalyst Lite, o usuário fica atrelado a gráficos pré-definidos, enquanto o 3D Active Chart permite alteração na estrutura dos gráficos.

CAPÍTULO 4

4. PROJETO DE AMBIENTES VIRTUAIS PARA VISUALIZAÇÃO DINÂMICA DE INFORMAÇÕES

4.1. AMBIENTE DE REALIDADE VIRTUAL

4.1.1. MODELAGEM DOS GRÁFICOS

Os gráficos foram modelados na linguagem VRML. O gráfico é multidimensional, ou seja, utiliza cor, altura e largura, dispostos em ambiente tridimensional, para propiciar a representação de grande quantidade de informações.

Nesse sentido, como existe grande percepção tridimensional, utilizaram-se planos para facilitar a visualização. Esses são de planos de referência, pois podem ser movimentados através do mouse ou da tabela existente logo abaixo do gráfico, inserindo o valor no campo apropriado.

Para a implementação foi usada a linguagem de realidade virtual VRML na modelagem do ambiente e dos gráficos.

Cada gráfico possui barras cilíndricas e dois planos de referência. Cada barra contém um plano de referência individual. Em uma tabela, logo em seguida ao gráfico é apresentado o valor correspondente à posição do plano sensor do mesmo, tanto superior quanto inferior.

O gráfico utilizado no projeto está inserido em um ambiente virtual tridimensional. Nesse ambiente, o gráfico ainda possui suas características tridimensionais, de acordo com a classificação de Shneiderman (1996), sendo apenas representadas suas três características, altura, largura e cor. No entanto, para representar as informações também são utilizados os

planos de referência das barras individuais e cores das barras, sendo possível inserir mais elementos, como som.

A interface de manipulação foi desenvolvida em linguagem HTML (Hyper Text Markup Language – Linguagem de Marcação de Hiper Texto), JavaApplet e JavaScript.

A linguagem HTML proporciona a possibilidade do sistema estar funcionando em ambiente de internet ou intranet.

O JavaApplet realiza a troca de informações e divide-se em três partes: a conexão JDBC-ODBC, a biblioteca AWT e a interface de comunicação EAI. A parte JDBC-ODBC faz a conexão entre o applet e o banco de dados, realizando as pesquisas e enviando-as para o banco de dados, de onde recebe os resultados e faz seu processamento. A parte AWT é responsável pela interface com o usuário. O EAI (External Authoring Interface) faz a comunicação entre o applet e o VRML, processando os dados para sua visualização. Com o JavaScript a linguagem VRML controla as cenas.

Exemplo da conexão JDBC-ODBC:

```
// String de conexão usada para acessar os dados por JDBC - Windows
String url = "jdbc:odbc:Historicos";
// Objetos de conexão com o banco de dados.
Statement st;
ResultSet rs;
try {
    // Setando as classes de conexão do JDBC
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con = DriverManager.getConnection(url);
    String strCmd;
    int nIndex = 0;
    st = con.createStatement(); ...}
rs = st.executeQuery(sql);
//fechar conexão com banco
con.close();
```

4.1.2. LINGUAGENS UTILIZADAS

4.1.2.1. EAI

O EAI (External Authoring Interface) é a interface externa que define as funções da linguagem VRML, fazendo a comunicação entre Java e VRML e o processamento das informações.

O VRML e o JavaApplet devem estar na mesma janela do *browser* para que funcione.

Utiliza-se o EAI para a comunicação entre as linguagens Java e VRML, fazendo a customização da navegação em VRML, bem como o controle dos eventos gerados.

O EAI recebe as informações do JavaApplet e as envia para o gráfico VRML.

A interface EAI faz o acesso aos nós do VRML, enviando as informações processadas através do JavaApplet. O External Authoring Interface permite controlar a linguagem VRML utilizando JavaApplet na mesma janela do *browser*, adaptando sua interface a fim de fornecer acesso à linguagem VRML (MARRIN, 1997); (HANUSKA, 1999). A interface imita o acesso do Script do nó através da referência do nome deste.

A aplicação comunica-se com a linguagem VRML obtendo a referência do objeto no *browser*, podendo identificar uma cena particular no ambiente VRML quando múltiplas cenas estão disponíveis (MARRIN, 1997); (DIEHL, 2001). A aplicação que criou o objeto deve fornecer toda informação necessária para introduzir o objeto no ambiente, realizando essas ações através da associação da função *callback* e do evento a ser aplicado.

A linguagem Java é utilizada para realizar o processamento dos dados e enviar o resultado para o VRML, através de EAI. Essa linguagem é utilizada por suas características serem aplicadas à internet, proporcionando também acesso via rede. Também é possível criar dinamismo entre as

aplicações, assim o programa pode ser atualizado de maneira rápida com acesso dinâmico aos dados, como pode-se ver na Figura 30.

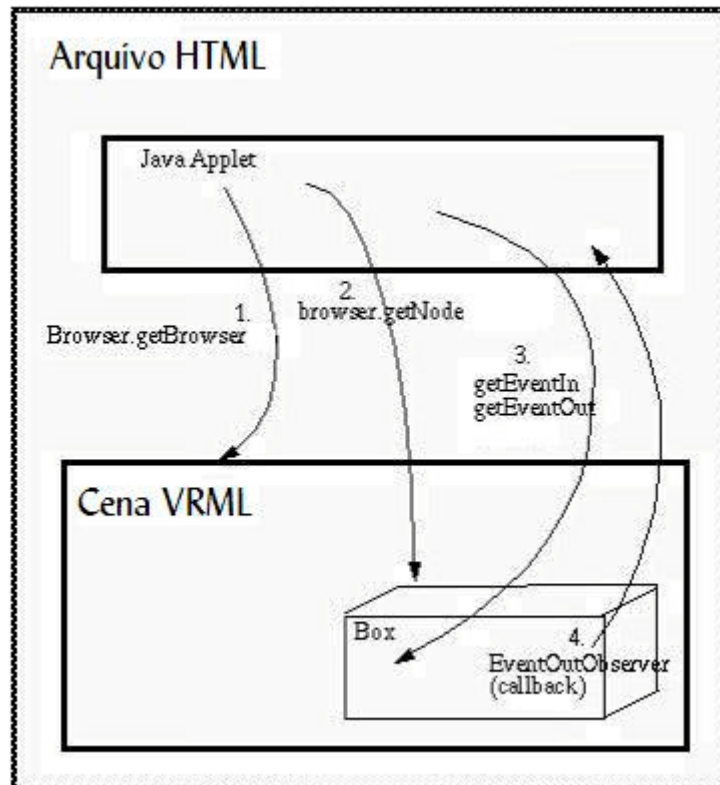


Figura 30 - Transferência de eventos VRML/JAVA (LIMA, 2004)

Na página HTML ocorre a visualização da cena em vrml e do applet em Java. Este envia requisições à cena VRML e ao script da cena. O script, em Java também, realiza as operações requisitadas e envia os resultados ao JavaApplet. O JavaApplet, irá, então, apresentar os resultados.

A visualização dinâmica é gerada a partir de uma página em HTML (Hyper Text Markup Language – Linguagem de Marcação de Hiper Texto) contendo um JavaScript específico. O JavaScript realiza a troca de informações com a página HTML e também gera algumas funções que não podem ser feitas pelos nodes do VRML.

4.1.2.2. JAVA

A integração das linguagens Java e VRML foi desenvolvida de maneira que as mesmas estivessem na mesma janela do *browser*.

Para mostrar a interação entre o Java e o VRML, foi incluído no projeto um banco de dados em Access. Nesse banco de dados as informações são inseridas, sendo depois enviadas ao gráfico do VRML através do JavaApplet.

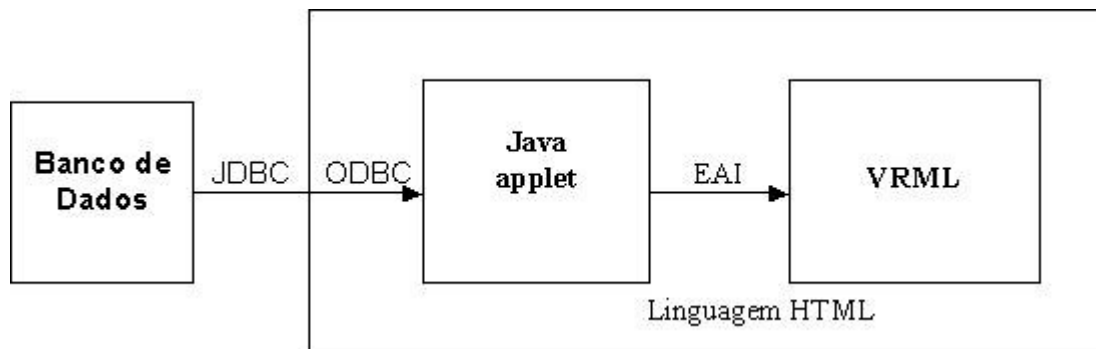


Figura 31 - Estrutura da aplicação

No JavaApplet as informações são processadas e então enviadas ao VRML através do EAI, como mostrado na Figura 31.

4.1.2.3. FUNCIONAMENTO

As ferramentas citadas anteriormente (VRML, EAI, JAVA, HTML, BD), em conjunto fazem a visualização dos dados dinamicamente, uma vez que o visualizador de dados possui realidade virtual para a visualização tridimensional do gráfico de barras em um ambiente interativo que consiste também em um banco de dados atualizado. O gráfico de barras é atualizado através de consultas ao banco de dados, sendo a imagem renderizada e com isso atualizada dinamicamente.

O banco de dados armazena tanto as informações atuais quanto os dados históricos, ambos atualizados sistematicamente através de uma aplicação Java. Essa aplicação, além de atualizar o banco de dados,

envia essas informações ao gráfico VRML pelo EAI. No HTML essas informações são apresentadas ao usuário por meio de uma tabela e de do gráfico VRML.

4.1.3. VERSÃO COM TABELA

No esquema da Figura 32, pode-se ver o funcionamento dessa versão, na qual o usuário insere manualmente os dados em uma tabela, e a mesma atualiza o gráfico através de renderização da imagem apresentada. Assim, cada vez que a tabela for atualizada com novas informações, estas são apresentadas por cada barra cilíndrica, cada uma correspondendo a cada célula da tabela.

Essa versão (Figura 33) do projeto mostra um gráfico em linguagem de realidade virtual (VRML) e logo em seguida uma tabela em Java (Figura 34), onde podem ser inseridos números diretamente na tabela que irão modificar o tamanho das barras do gráfico. Cada célula da tabela corresponde a uma barra do gráfico em realidade virtual. A cada modificação de dados na tabela, automaticamente esta é apresentada na barra correspondente.

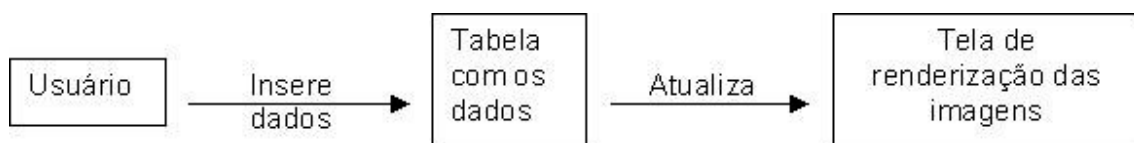


Figura 32 - Esquema do funcionamento da versão com tabela



Figura 33 - Exemplo de gráfico

O gráfico foi modelado na linguagem de realidade virtual VRML, e possui cilindros representando cada barra, e também cada eixo (x, y e z). Foi inserido um plano semi-transparente, que pode ser movimentado através do mouse, para facilitar na visualização das informações. Cada célula da tabela corresponde a uma barra cilíndrica. Toda vez que ocorre mudança no valor de cada célula da tabela, a barra correspondente apresenta seu valor alterando sua altura.

	X1	X2	X3	X4	X5
	0.1	-0.2	0.6	0.4	0.2
	0.6	0.1	1.0	0.3	0.6
	1.2	1.0	1.5	1.2	0.8
Posição do Cursor	0.0				

Figura 34 – Exemplo da tabela

Nessa versão a inserção dos dados na tabela é completamente manual e o próprio usuário encarrega-se de realizar essa atividade.

4.1.4. VERSÃO COM BANCO DE DADOS LOCAL

A segunda versão do projeto consistiu na utilização da linguagem de realidade virtual VRML, da linguagem JavaApplet e JavaScript, do banco de dados access e também da linguagem HTML.

Desenvolvido em realidade virtual, nessa versão os valores das barras dos gráficos são alimentados por meio do banco de dados, por valores aleatórios, inseridos automaticamente pelo mesmo, como pode-se ver pelo esquema da Figura 35. O banco de dados (Figura 36) é atualizado automaticamente e por meio de programação específica atualiza a tabela de dados. Esta, por fim, passa os dados para o gráfico. No banco de dados Access foi criada uma tabela onde o *código* corresponde a cada barra do gráfico (código 1 corresponde à barra C01, código 2, corresponde à barra C02, por exemplo); o *número* é sua altura correspondente; *corr*, *corg*, *corb* é a composição das cores de cada barra; e, por fim, *largura* é a largura de cada barra.



Figura 35 - Esquema do funcionamento da versão com banco de dados local

The screenshot shows the Microsoft Access interface with a table named 'tbNumero : Tabela'. The table has six columns: 'codigo', 'numero', 'corr', 'corg', 'corb', and 'largura'. The data is as follows:

codigo	numero	corr	corg	corb	largura
1	-0,262	96	0	188	0,34
2	-0,199	96	0	204	0,286
3	0,43	0	145	255	0,359
4	-0,168	96	0	212	0,875
5	-0,175	96	0	210	0,646
6	0,345	255	167	0	0,163
7	-0,456	192	192	139	0,028
8	-0,385	192	192	157	0,04
9	0,902	255	25	0	0,466
10	0,856	255	37	0	0,117
11	0,267	128	255	187	0,527
12	0,816	128	255	47	0,055
13	-0,737	64	67	64	0,343
14	-0,26	64	188	64	0,409
15	0,404	128	255	152	0,686
16	0,324	0	172	255	0,674
17	-0,224	192	192	198	0,109
18	0,254	128	255	190	0,027

The interface also shows a status bar at the bottom indicating 'Modo folha de dados' and 'NUM'.

Figura 36 - Banco de dados Access

O gráfico da Figura 37 mostra as barras cilíndricas, cada uma contendo seu próprio valor, tanto de largura, quanto de altura, espessura e cor e vários elementos para representar as muitas formas de apresentar as informações ao usuário. Além de mostrar os dados na própria barra também podem-se utilizar outros elementos gráficos, como os planos do gráfico. Essa versão foi nomeada DATAVIS-VR.

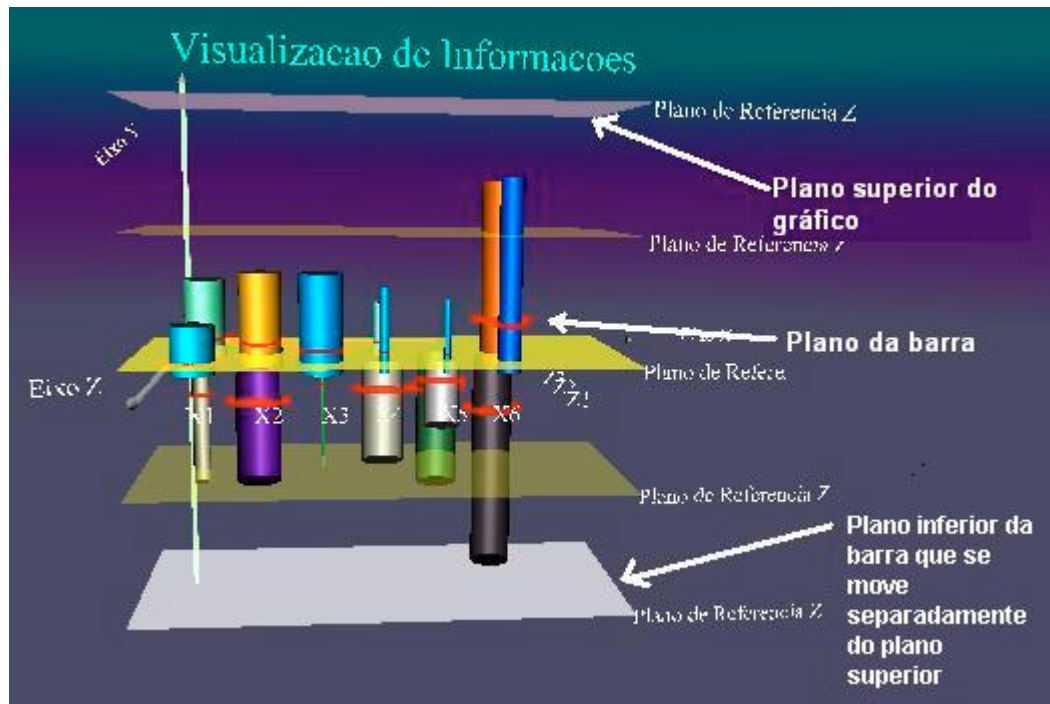


Figura 37 - Gráfico de visualização de informações

O código do programa VRML pode ser visto no Anexo 1.

Os planos sensores do gráfico podem ser movidos tanto através do uso do mouse, quanto através da alimentação manual feita pelo usuário.

Através do JavaScript é realizada a pesquisa ao banco de dados que contém as informações, sendo o resultado enviado ao JavaApplet e em seguida mostrado no gráfico VRML.

Parte do programa JavaScript que realiza as pesquisas no banco de dados para atualização dos valores pode ser visto no Anexo 2.

Trecho do programa JavaApplet que mostra o resultado da pesquisa sendo enviado para o gráfico em realidade aumentada encontra-se no Anexo 3.

Nesse ambiente de realidade virtual, o usuário pode interagir com o gráfico, navegando e alterando seus dados, através da interação com o banco de dados.

4.1.4.1. FUNCIONAMENTO DA FERRAMENTA DE VISUALIZAÇÃO DE DADOS

Ao monitorar o mundo real, podem-se visualizar as informações através de gráficos. Um exemplo de aplicação seria o uso do sistema de obtenção de dados, onde os dados são enviados ao sistema e mostrados através de um gráfico em realidade virtual.

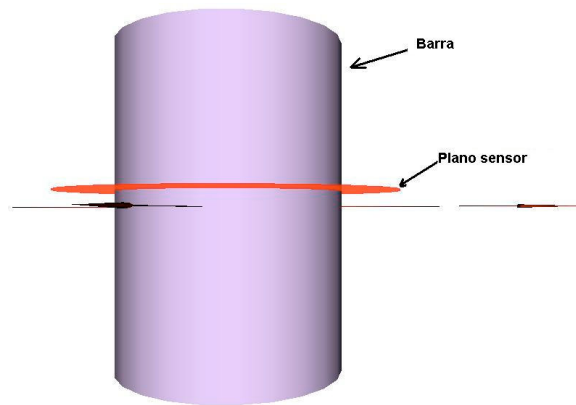


Figura 38 - Elemento de referência - barra individual

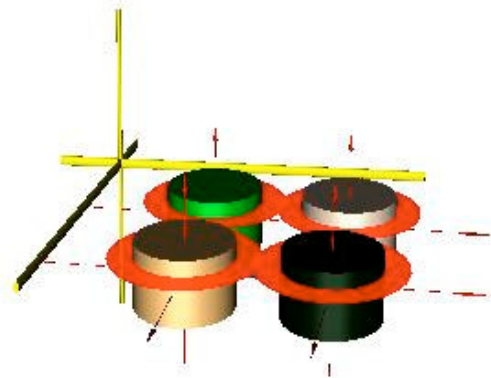


Figura 39 - Ambiente Virtual Simplificado

A barra da Figura 38 é o elemento de referência, pois pode-se acrescentar à mesma várias características para representar as informações, como cores variadas, animação, tamanho etc. Os planos sensores da barra podem ser configurados separadamente, indicando os valores mínimos e máximos. Replicando-se o elemento da Figura 39, forma-se um ambiente virtual simplificado. Com a replicação do elemento de referência pode-se ter gráficos com quantas barras o usuário desejar.

4.1.4.2. CONSIDERAÇÃO DOS DADOS HISTÓRICOS

O Sistema de Aquisição do DATAVIS-VR envia as informações diretamente para o banco de dados, o qual armazena as informações para análise. Além de armazenar as informações atuais, o banco de dados também contém informações anteriores, ou seja, possui um histórico.

Cada dado isolado possui a sua importância. No entanto, um conjunto desses mesmos dados pode trazer outro significado no contexto de um sistema de informações. Como exemplo pode-se citar o estudo do nível de poluição de um lago. Em uma certa data é verificada a taxa de poluição e sabe-se, então, que se essa taxa estiver alta naquela data, o lago estará poluído. Mas se essa taxa for considerada em um certo período de tempo, outras informações poderão ser verificadas sobre a poluição do lago, como por exemplo, se a poluição está diminuindo ou aumentando, se essa taxa está em crescimento ou queda acelerada, qual seu desvio padrão, e outras estatísticas que podem ser estudadas.

No sistema DATAVIS-VR foi utilizado o banco de dados Access da Microsoft. Foram utilizadas duas tabelas, contendo a primeira os dados atuais das barras, e a segunda armazenando os dados antigos, ou seja, o histórico. A pesquisa do histórico e a limitação da quantidade de informações que serão mostradas no gráfico de barras são determinadas por meio de comandos no programa servlet, feito em Java. O resultado dessa pesquisa é apresentado no gráfico através do plano sensor individual das barras, sendo apresentada a média dos valores.

Parte do código do JavaServlet que atualiza o banco de dados do Access pode ser vista no Anexo 4.

À versão atual pode ser adicionado um sistema de aquisição de dados remoto, onde o usuário insere os dados em uma ferramenta de aquisição de dados, e esses são enviados para o banco de dados, como pode ser visto na Figura 40. Após a inserção de dados feita pelo usuário, o sistema de aquisição os envia para o banco de dados para serem armazenados e então promoverem a atualização da tabela do gráfico, assim esses dados da tabela serão mostrados no gráfico.

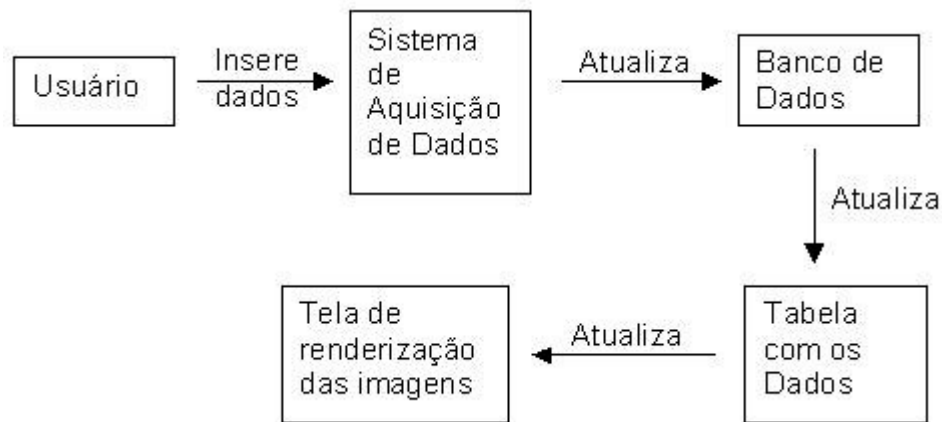


Figura 40 - Esquema do Funcionamento com aquisição remota de dados

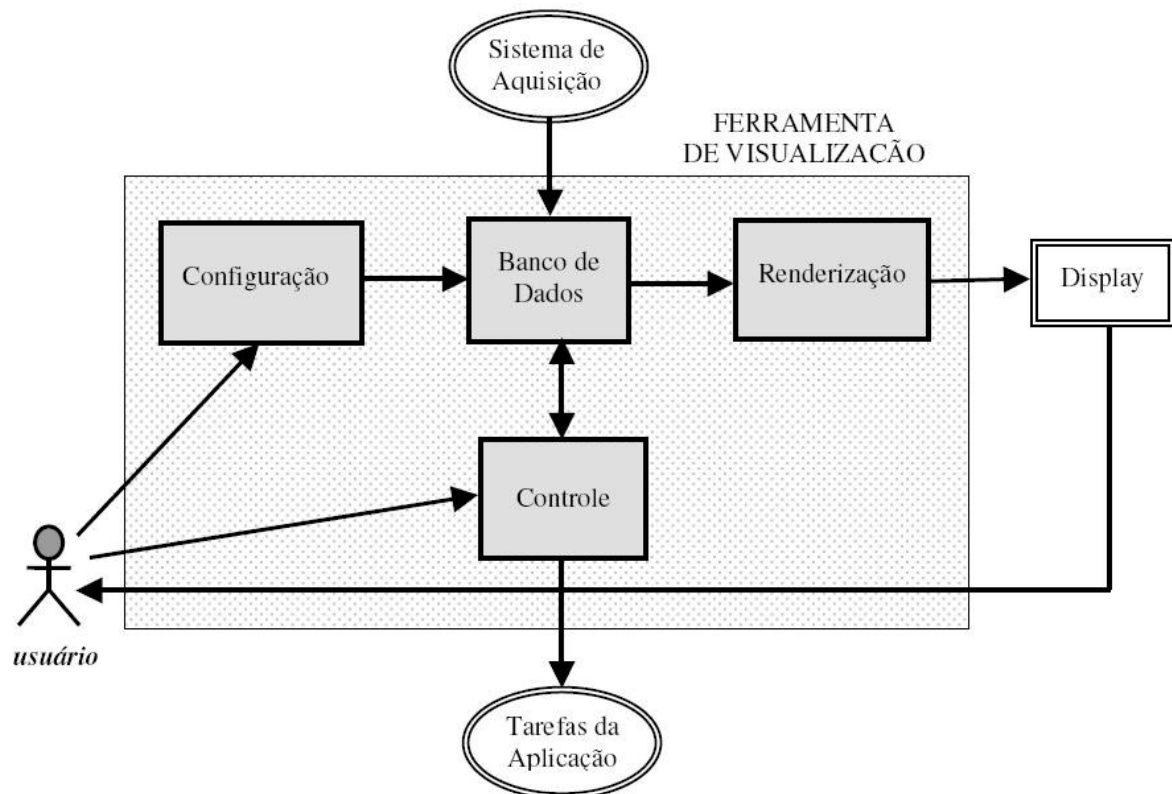


Figura 41 - Sistema de aquisição remota de dados

No esquema da Figura 41 pode-se observar o funcionamento da Ferramenta de Visualização com um sistema de aquisição de informações.

O Sistema de Aquisição envia dados continuamente ao banco de dados. Este, por sua vez, armazena n quantidades de dados em seu

histórico, apresentando sempre os mais recentes. A quantidade de dados históricos, definida como n , pode ser configurada através do Java.

O módulo de controle analisa as informações do banco de dados de acordo com os fatores de decisão apontados e restrições configuradas, realizando ações de acordo com o que for configurado. É também através desse módulo que o usuário pode interagir com o gráfico, clicando sobre o mesmo, realizando a configuração nos planos sensores, tanto do gráfico, como das barras individualmente, ou navegando através das barras.

O módulo Renderização permite ao usuário visualizar as informações do Banco de Dados através de imagens. No caso deste trabalho, isso se dá através de gráficos de barras tridimensionais.

O módulo de Configuração permite que o usuário ajuste o sistema conforme sua necessidade, estabelecendo as restrições e os ajustes de fatores de decisão, como, por exemplo, ajustar os valores mínimos e máximos dos planos sensores através do *mouse* para uma melhor orientação do gráfico, facilitando sua visualização e tomada de decisão.

O usuário pode alterar a configuração (pelo módulo de Configuração) e o controle (através do módulo de Controle) alterando seus parâmetros da maneira que mais lhe convier para que gere dados de seu interesse. Dessa forma, o usuário pode alterar os valores mínimos e máximos dos planos sensores, a taxa de variação dos dados do histórico, a quantidade de dados históricos utilizados para calcular a média dos dados em um determinado período de tempo.

O resultado das configurações e controles pode ser visto por meio das características representadas no gráfico, como cores das barras, tonalidades, tamanhos das barras e planos sensores, tanto do gráfico, como das barras individuais.

O Display é como o usuário irá visualizar as informações, ou seja, através do monitor do computador.

4.2. AMBIENTE DE REALIDADE AUMENTADA

4.2.1. VERSÃO PARA VISUALIZAÇÃO ESTÁTICA

Com o DATAVIS-AR, descrito anteriormente, e a ferramenta ARToolkit, foram realizadas várias experiências de inserção dos gráficos de realidade virtual em realidade aumentada.

Na Figura 42 pode-se ver um exemplo de como pode ser utilizado o DATAVIS-AR com o ARToolkit. Na placa do sistema é feita a chamada para o gráfico em VRML, que dessa forma aparece na mão do usuário. Com o gráfico “em suas mãos”, o usuário pode então girá-lo para qualquer posição a fim de estudá-lo melhor.



Figura 42 - Exemplo do uso do DATAVIS-AR

Também com a ferramenta ARToolkit foram inseridas várias placas contendo partes do gráfico geral. O usuário pode trocar de lugar as partes, substituí-las, eliminá-las ou até mesmo incluir novas partes, assim interagindo com o gráfico, conforme pode-se ver na Figura 43. Dessa maneira o usuário tem a liberdade de visualizar o gráfico como um todo ou apenas em parte, escolhendo um único pedaço do gráfico para um estudo mais

aprimorado. A base de visualização pode ser em qualquer formato, pois são colocadas placas sobre a mesma.

As placas dos gráficos foram previamente cadastradas no ARToolKit, cada uma contendo seus marcadores de referência, e em seguida foram cadastrados os gráficos correspondentes. Ao colocar a placa na frente de uma *webcam*, a ferramenta reconhece a placa e insere automaticamente seu gráfico correspondente para ser estudado pelo usuário. Dessa forma o usuário pode interagir com o objeto em suas mãos.



Figura 43 - Exemplos da ferramenta de realidade aumentada

Como pode-se ver na Figura 43, é possível construir um tabuleiro com pedaços de gráficos para maior interação, podendo-se trocar de lugar as partes, retirar gráficos que não interessam, ou até mesmo inserir novas partes, com a finalidade de facilitar a tomada de decisão do usuário.

Pode-se notar também, nessa mesma Figura, que a base onde ficam as placas dos gráficos não precisa ter um formato único como quadrado ou retangular, como mostrado, a base pode ser inclusive circular, ou seja, não importa seu formato.

4.2.1.1. USO DE MARCADORES E CONTROLE

Inicialmente a visualização em realidade aumentada é estática, pois o ARToolKit não está preparado para atualizar automaticamente as imagens conforme as informações vão sendo alteradas. Então é necessário realizar uma modificação no código fonte e juntamente utilizar uma placa de controle para fazer com que o gráfico apresente as informações atualizadas (SANTIN, 2005).

A alteração do código pode ser vista no artigo de Santin (2005), no qual encontra-se a explicação sobre quais os arquivos e suas modificações para que o sistema funcione. Com a placa de controle, cada vez que o usuário mostra essa placa na frente da placa do gráfico, ocorre uma mudança programada de evento que altera o objeto virtual. Nesse caso altera a maneira o gráfico está apresentado na placa.

4.2.2. VERSÃO PARA VISUALIZAÇÃO DINÂMICA SEM TABELA

Nessa versão os dados são enviados para o gráfico através de um editor do VRML, o qual irá alterar os dados do gráfico (SANTIN, 2005). Para realizar essa versão, além de alterar o código fonte do ARToolKit, foi criada uma função para iniciar a leitura dos arquivos dat e alocação das cenas VRML logo na inicialização da ferramenta. Essa alteração do código permite que seja feita uma re-leitura do código VRML, e, se houver alguma modificação, o sistema irá mostrar.

Esse editor permite ao usuário alterar remotamente os valores do gráfico, bastando para isso inserir no código VRML marcações com as

respectivas linhas a serem alteradas, de acordo com o esquema apresentado na Figura 44:

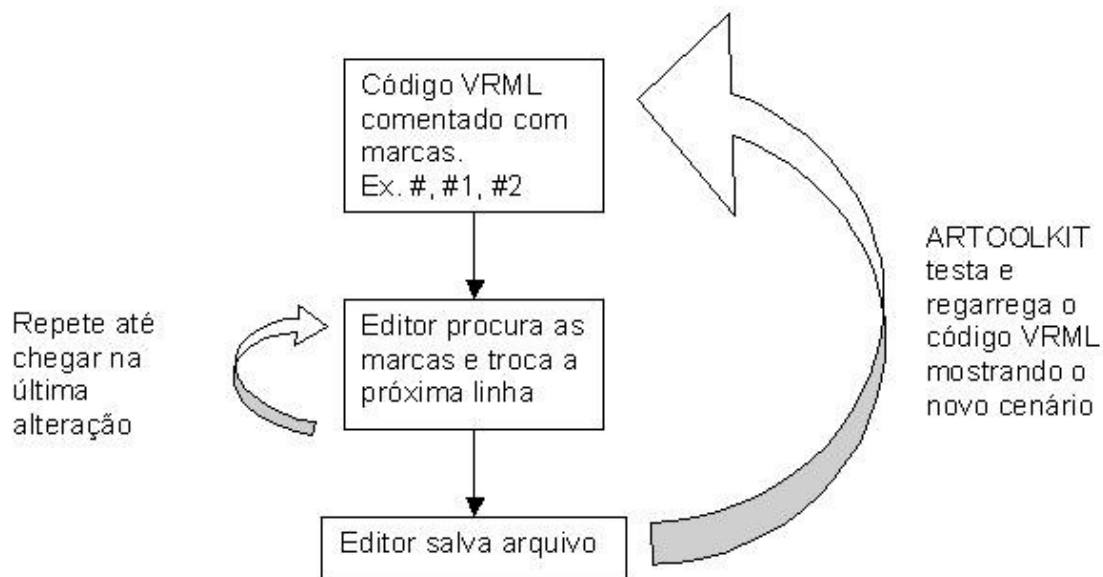


Figura 44 - Funcionamento da ferramenta ARToolKit com o Editor

Trecho do código VRML mostrando a linha que contém as marcas para o Editor realizar a atualização dos dados:

```

DEF Bar Transform{
children[
DEF C01 Transform {
#modifica
translation -2 1 0
#translation -2 0 0
#translation -2 -1 0
#translation -2 0 0
  
```

Cada vez que o editor passa na linha onde está o código comentado “#modifica”, realiza a alteração especificada. Nesse exemplo a

alteração realiza a modificação da translação do objeto (altura da barra C01), em seguida o arquivo VRML é salvo. O ARToolKit recarrega o arquivo e mostra a nova cena, sendo esta a modificação ocorrida da translação do objeto.

A Figura 45 mostra a visualização do gráfico, onde o exemplo de código anteriormente citado refere-se à primeira barra e suas possíveis posições.

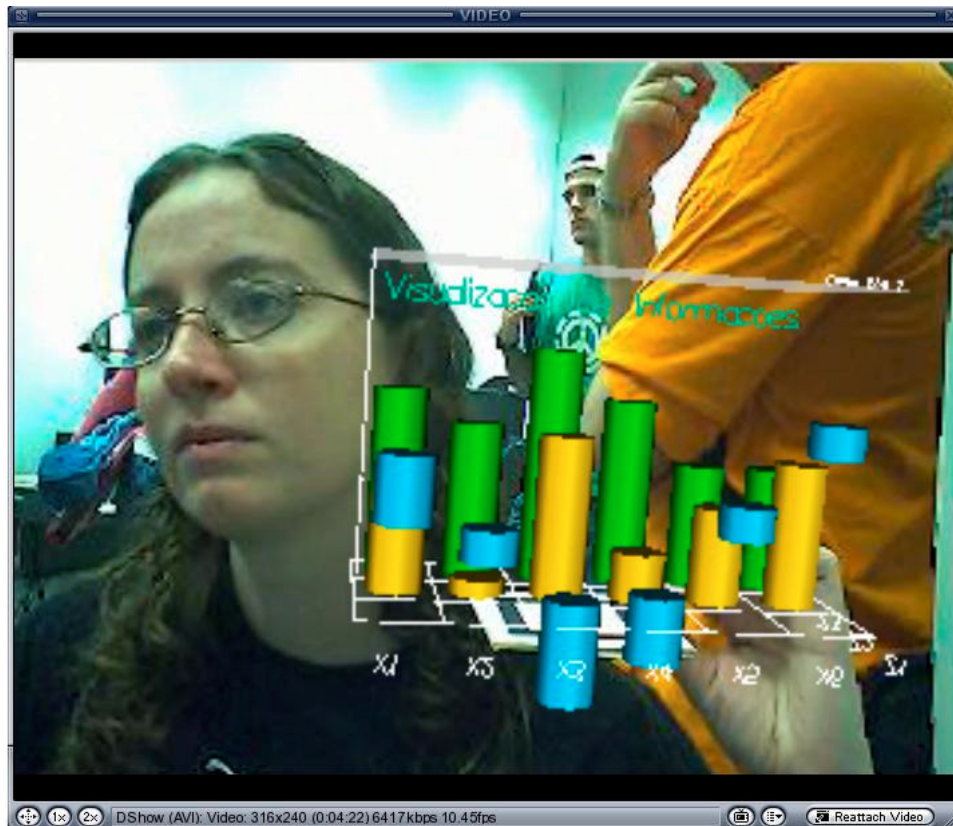


Figura 45 - Visualização dinâmica

Nessa versão pode ser incluída uma tabela para que o usuário insira os dados no momento em que é apresentado o gráfico. Por meio dessas modificações no código fonte, cada vez que o usuário alterar os dados na tabela, serão apresentadas as informações referentes no gráfico, dinamicamente.

A tabela a ser inserida funciona como sendo um sistema de aquisição de informações na qual pode ser acrescentado um sistema de histórico para arquivar os dados relevantes.

4.3. APLICAÇÕES

As aplicações desse projeto podem ser implementadas em sistemas que necessitem de tomada de decisão baseada em análise de dados, como mercado financeiro, administrativo, militar, medicina, e outros que foram citados no decorrer deste trabalho. O sistema de visualização de informações facilita o entendimento do usuário através da conversão dos dados em informações gráficas no espaço bidimensional ou tridimensional.

As aplicações implementadas com realidade virtual permitem visualizar imagens em ambiente tridimensionais, às quais podem-se acrescentar elementos para facilitar a tomada de decisão, incluindo informações adicionais como: endereços de internet e textos explicativos a cada elemento de um gráfico.

Para trabalhar com realidade virtual, é necessário o treinamento do usuário que precisa saber utilizar os dispositivos de visualização para interagir com o ambiente.

Com a utilização de realidade aumentada, o usuário estará lidando com um ambiente familiar, sem a necessidade de treinamento para interação com o ambiente. As aplicações podem ser semelhantes àquelas que utilizam realidade virtual.

Por exemplo, pode ser feito um programa para medir a poluição de um lago, de forma que, em cada parte do lago, seja colocado um medidor de poluição que envia os dados ao programa. Esses dados serão representados através de um gráfico, onde cada barra equivale a uma parte do lago. Quando a barra apresenta uma cor vermelha (dado crítico), significa que naquela parte do lago o nível de poluição está bastante alto. Outra utilização de um sistema de visualização de informações pode ser uma aplicação de análise

de solo. Uma câmera de vídeo captura a imagem do solo, montando um mapa da área no computador. Podem ser colocados aparelhos que fazem a leitura das condições do solo para enviar os dados o sistema, sendo que cada aparelho representa uma barra no gráfico. Assim como no exemplo do lago, se a barra ficar com a cor vermelha, as condições daquela parte de solo estariam bastante ruins e assim por diante.

Uma aplicação financeira poderia mostrar o comportamento de ações da bolsa de valores, indicando em cada barra a situação das ações, através de cores e altura. Com isto o usuário tem condições de avaliar a situação de mercado e tomar decisão sobre a compra e venda de ações.

Tanto a realidade virtual quanto a realidade aumentada pode beneficiar em muito a visualização de dados, dando ao usuário uma visão espacial e facilidades de interação, dependendo dos equipamentos e aplicativos disponíveis.

4.3.1. POTENCIALIDADES

No sistema projetado neste trabalho pode ser inserido um sistema de realidade aumentada, colocando-se os gráficos virtuais no mundo real.

Foi realizada uma experiência com gráficos tridimensionais usando o visualizador de dados com realidade virtual e o software ARTOOLKIT. Com uma *webcam* ligada ao computador e o software ARTOOLKIT foi possível associar o objeto gráfico a uma marca previamente cadastrada, e, colocando-se a placa com a marca no campo de visão da *webcam* o sistema insere o gráfico sobre a placa, podendo ser estudado pelo usuário, como foi mostrado na Figura 43.

O software ARTOOLKIT reconhece diferentes marcas em diferentes placas ao mesmo tempo, podendo trabalhar com vários objetos no gráfico.

4.3.2. RESTRIÇÕES

Uma das restrições do sistema é a necessidade de configuração manual, tanto do gráfico, como do banco de dados.

Em realidade virtual, os usuários necessitam de treinamento, pois devem ter conhecimento dos dispositivos de visualização como capacetes e luvas.

Em realidade aumentada deve-se estar atento ao tamanho das placas utilizadas, pois se forem muito grandes prejudicam a visualização, restringindo seu uso.

CAPÍTULO 5

5. CONCLUSÕES

A partir dos resultados obtidos com os experimentos em realidade virtual e realidade aumentada, pôde-se verificar que essas duas tecnologias possuem grande capacidade para a visualização de informações, facilitando ao usuário seu entendimento por meio de imagens que podem ser manipuladas pela interação existente.

A visualização das informações é um importante passo para o estudo e compreensão de dados relevantes para a tomada de decisão que, em muitos casos, deve ser realizada de forma acertada e rápida, o que pode ser proporcionado pela visualização.

Um ponto importante a ser considerado é a familiarização com o ambiente proposto em realidade virtual, onde os usuários necessitarão, muitas vezes, de treinamento na ferramenta, pois o sistema lida com movimentos em um espaço tridimensional, que pode causar dificuldade na movimentação do usuário. O usuário tende a se “perder” no mundo virtual, por essa razão, um treinamento antes de utilizar a ferramenta é recomendado, não apenas pela movimentação, mas também pela utilização dos objetos existentes, já que há a necessidade de interação com os mesmos. Essa forma de visualizar a informação potencializa o aspecto cognitivo e o entendimento de dados, facilitando sua análise.

A realidade aumentada é utilizada para tornar a interface mais amigável ao usuário, contribuindo para a interação do usuário com o sistema de visualização, permitindo sobreposição dos dois ambientes, o de realidade virtual e a realidade real do usuário. Por meio desse sistema o usuário já não necessita de treinamento, pois realiza os mesmos movimentos naturais do ambiente real.

Neste trabalho, foi apresentada a ferramenta DATAVIS-VR que foi desenvolvida em um ambiente de realidade virtual. Esse sistema possui um gráfico tridimensional que apresenta as informações contidas em um banco de dados alimentado automaticamente. Para trabalhos futuros pretende-se implementar um sistema de aquisição de informações para que o próprio usuário insira os dados no sistema.

Também foi desenvolvida uma ferramenta, o DATAVIS-AR, que incorporou recursos de realidade virtual e realidade aumentada. A realidade aumentada foi utilizada como uma ferramenta para otimizar a interação do usuário com os gráficos contendo as informações de seu interesse. Dessa forma, o usuário tem a liberdade de “ter em suas mãos” o gráfico, podendo visualizá-lo da maneira que lhe convier, inserindo, excluindo, alterando o gráfico.

Essa ferramenta apresenta versão com atualização dinâmica dos dados por meio de modificações no código fonte do ARToolKit, onde um editor altera os dados do gráfico em VRML, apresentando suas modificações ao usuário. Para futuros trabalhos pode ser implementado um sistema para aquisição de informações onde o usuário insere o valor das informações e automaticamente as apresenta no gráfico.

Este trabalho abordou a visualização de informações em ambiente de realidade virtual e aumentada, com interação, e a sua representação por meio de gráficos em ambientes tridimensionais.

Para os próximos trabalhos, a ferramenta pode ser testada com outros tipos de banco de dados, como o SQL, por exemplo, ou outros que são mais comuns em empresas de grande porte.

Atualmente o sistema funciona no ambiente operacional Windows XP. Sugere-se fazer uma versão para ser utilizada em outros sistemas operacionais, como o Linux, que é um sistema operacional que está em constante crescimento e utilização por ser um software livre.

Os gráficos utilizados na ferramenta podem ser alterados, podendo-se fazer um estudo para inserir vários tipos de gráficos, tornando a visualização das informações mais completa.

Encontra-se em andamento estudos sobre realidade aumentada relacionados à utilização da mão do usuário para controlar os objetos no ambiente, pois atualmente é utilizado o *mouse*. Dessa forma a interação ambiente e usuário pode ocorrer de forma natural, já que este irá “pegar”, por exemplo, uma parte do gráfico que lhe interessa e realizar seus estudos mais detalhadamente.

REFERÊNCIAS BIBLIOGRÁFICAS

ACTIVE CHART. Disponível em: <<http://www.rinvoice.com/activechart.htm>>
Acesso em: 28/02/2004.

ADVIZOR SOLUTIONS INC. **Advizor - Product Overview**, Fev/2003.
Disponível em:
<<http://www.advizorsolutions.com/whitepapers/requestprocess.asp>> Acesso
em: 06/05/2003.

AHLERS, Klaus H. et al. Distributed Augmented Reality for Collaborative Design Applications. In: EUROGRAPHIC'S, 1995. Proceedings. Maastricht, NL, 1995. p. 3-14.

ARTOOLKIT. Disponível em:
<http://www.hitl.washington.edu/research/shared_space/>. Acesso em:
12/06/2004.

AVS ADVANCED VISUAL SYSTEM. **Decisions at the Speed of Sight – OpenViz**. Disponível em: < <http://www.avv.com/pdf/index.html>> Acesso em:
19/05/2003.

AZUMA, Ronald T. et al. Recent Advances in Augmented Reality. **IEEE Computer Graphics and Applications**, v .21, n.6, p. 34-47, 2001.

AZUMA, Ronald T. A Survey of Augmented Reality. In: COMPUTER GRAPHICS - SIGGRAPH '95, Los Angeles, 1995. Proceedings, Course Notes #9: Developing Advanced Virtual Reality Applications. Los Angeles, 1995. p. 1-38.

BILLINGHURST, Mark; KATO, Hirokazu. Collaborative Augmented Reality. **Communications of the ACM**, p. 64-70, 2002.

BUK, Carolina V. et al.(2004a). Uso da Realidade Aumentada em Ambientes Virtuais de Visualização de Dados. In: Symposium on Virtual Reality –

SRV2004, VII, São Paulo, 2004. **Anais**. Editora SENAC, São Paulo, 2004. V.1, p. 149-160.

BUK, Carolina V.et. al. (2004b). Visualização Interativa de Informações em Ambiente de Realidade Aumentada. In: Congresso de Pós-Graduação da Universidade Metodista de Piracicaba 2º, Piracicaba, 2004. **Anais**. Piracicaba: Universidade Metodista de Piracicaba, 2004. p. 175-177.

BUK, Carolina V. et. al. (2004c). Visualização de Dados em Ambientes com Realidade Aumentada. In: Workshop de Realidade Aumentada – WRA2004, 1º, Piracicaba, 2004. **Anais**. Piracicaba: Universidade Metodista de Piracicaba, 2004. p. 45-48.

BUK, Carolina V. et. At. (2005). Visualização de Informações em Ambientes de Realidade Aumentada. In: Workshop de Realidade Aumentada, II, Piracicaba, 2005. **Anais**. Piracicaba: Universidade Metodista de Piracicaba, São Paulo. 2005. p. 68-71.

CARR, David A., Guidelines for Designing Information Visualization Applications. In: ECUE'99, Stockholm, 1999. Proceedings. Stockholm, 1999.

Como ver VRML. Disponível em: <<http://www.geocities.com/MotorCity/4630/ver/cp-download.html>> Acessado em: 06/05/2003

CRAPO, Andrew W. et al., Visualization an the Process of Modeling: A Cognitive-theoretic View. Boston: KDD, MA USA, ACM, 2000. p. 218-226.

CRISPEN, Bob. **Comp.Lang.Vrml FAQ**. 2000. Disponível em: <<http://vrmlworks.crispen.org/>> Acesso em: 06/01/2003.

DARKEN, Rudy P., SIBERT, John L. A Toolset for Navigation. In: VIRTUAL ENVIRONMENS. ACM User Interface Software & Technology, Atlanta,GA, 1993. **Proceedings**. Atlanta,GA, 1993 p. 157-165.

DIEHL, Stephan. **Distributed Virtual Worlds: Foundations and Implementation Techniques Using VRML, Java, and CORBA**. Berlin: Springer, 2001, 166 páginas.

EICK, Stephen G. et al. (1992a) Seesoft - A Tool for Visualizing Line Oriented Software Statistics. **IEEE Transactions on Software Engineering**, p. 957-968, 1992.

EICK, Stephen G. et al.(1992b) Seesoft - A Tool for Visualizing Line Oriented Software Statistics. Imagem disponível em <<http://pitecan.com/presentations/DEWS98/slide/page63.html>> Acesso em 20/05/2004.

FEINER, Steven K. Augmented Reality: A New Way of Seeing. **Scientific American**, 24/04/2002. Disponível em: <http://www.sciam.com/print_version.cfm?articleID=0006378C-CDE1-1CC6-B4A8809EC588EEDF> Acesso em: 20/05/2004.

GANNALYST. Copyright Gannalyst 2000 - 2002. All Rights Reserved. Disponível em: <<http://www.gannalyst.com/main.asp>>. Acesso em: 28/02/2004.

GEISLER, Gary. **Making Information More Accessible: A Sourvey of Information Visualization Applications and Techniques**, 1998. <Disponível em: <http://www.ils.unc.edu/~geisg/info/infovis/paper.html>> Acesso em: 06/05/2003.

GERSHON, Nahum, et al. Information Visualization. In: interactions, Jg. 1998, ACM Press, New York, NY, vol. 5, ISSN:1072-5520, p. 9-15.

HAND, Chris. Other Faces of Virtual Reality. In: East-West International Conference on Multimedia, Hypermedia and Virtual Reality, Moscow, 1994. **Proceedings**. Moscow, 1994. p. 69-74.

HANUSKA, Dusan. Information System and VRML Worlds. Department of Computer Science and Engineering, Faculty of Electrical Engineering and

Information Technology, Slovak University of Technology
Bratislava / Slovakia, 1999.

HAWKINS, Donald T. Information Visualization – Don't Tell Me, Show Me. Online ABI/INFORM Global, 1999, p. 88-90.

HEAST, Marti A. TileBars: Visualization of Term Distribution Information in Full Text Information Access, ACM CHI'95 , Denver, Colorado, 1995. **Proceedings**. Denver, Colorado, 1995. p. 59-66.

HERMAN, I. et al. Graph Visualization and Navigation in Information Visualization: a Survey. IEEE Transactions on Visualization and Computer Graphics, 2000, p. 24-43.

IBM. **Open Visualization Data Explorer**. Disponível em:
<<http://researchweb.watson.ibm.com/dx/>> Acesso em: 21/05/2003.

INTERNATIONAL STANDARD ISO/IEC 14772-1:1997 . **The Virtual Reality Modeling Language**, 1997. Disponível em:
<http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/> Acesso em: 06/10/2004.

JERDING, Dean F.; STASKO, John T. Information Mural: A Technique of Displaying and Navigating Large Information Spaces. **IEEE Transactions on Visualization and Computer Graphics**, 1998, p 257-271.

KATO, Hirokazu, et al. **ARToolKit 2.33 Documentation**. Seattle, 2000.

KIRNER, Claudio. Realidade Virtual: Dispositivos e Aplicações. In: Escola Regional de informática da SBC Regional Sul - VII ERI, SBC, Londrina, Chapecó, Novo Hamburgo, 1999. **Anais**. Londrina, Chapecó, Novo Hamburgo, 1999, p. 135-158.

KIRNER, C ; TORI, R. Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade. In: Claudio Kirner; Romero Tori. (Ed.). **Realidade Virtual: Conceitos, Tecnologia e Tendências**. 1ed. São Paulo: Editora SBC, 2004, capítulo 1, v. 1, p. 3-20.

KIRNER, Cláudio; ZORZAL, Ezequiel R. Aplicações Educacionais em Ambientes Colaborativos com Realidade Aumentada. **XVI Simpósio Brasileiro de Informática na Educação**, Juiz de Fora – MG, 2005.

KIRNER, Tereza G. et al. Development Of An Information Visualization Tool Using Virtual Reality. In: ACM Symposium on Applied Computing – SAC'2000, 15th, Como, 2000. **Proceedings**. 2000, p. 604-607.

KNIGHT, Claire. Virtual Reality for Visualization. **Computer Science Technical Report**, 1998. Disponível em: <http://vrg.dur.ac.uk/papers/papersearch.php3?year=ALL> Acesso em: 12/04/2003.

KNIGHT, Claire; MUNRO, Malcolm. Comprehension with[in] Virtual Environment Visualizations. In: **IEEE International Workshop on Program Comprehension**, Pittsburgh - PA, 1999. **Proceedings** Pittsburgh - PA, 1999. p. 4-11.

KNIGHT, Claire; MUNRO, Malcolm. Mediating Diverse Visualizations for Comprehension. In: International Workshop on Program Comprehension (IWPC' 01), IEEE, 9th, Washington - DC, 2001. **Proceedings**. Washington - DC, 2001, p. 18-25.

KOLLER, Dieter et al. Real-time {Vision-Based} Camera Tracking for Augmented Reality Applications. In: Symposium on Virtual Reality Software and Technology, ACM, Lausanne – Switzerland, 1997. **Proceedings**. Lausanne – Switzerland, 1997. p. 87-94.

LIMA, Johnatan Daniel F. **Interagindo VRML e Java, uma visão Geral**. Disponível em: <http://www.mundooo.com.br/php/mooartigos.php?pa=showpage&pid=16>. Acesso em: 28/06/2003.

MALETIC, Jonathan I.; LEIGH, Jason; MARCUS, Adrian, Visualizing Software in an Immersive Virtual Reality Environment, In: ICSE' 01 Workshop on

Software Visualization, Toronto, Ontario, 2001. **Proceedings.** Toronto, Ontario, 2001. p. 49-54.

MARRIN, Chris. **External Authoring Interface Reference.** Silicon Graphics Inc., 1997. Disponível em: <http://www.web3d.org/WorkingGroups/vrml-eai/history/eai_draft.html> Acesso em: 10/01/2003.

MARSHALL, M. S., et al. An object-oriented design for graph visualization. **Software: Practice and Experience**, 2001, p. 739-756.

MXR Toolkit. Disponível em <http://155.69.54.110/SOFTWARE/FILES/SOFTWARE/MXR.Tool.Kit/All_document/HTM_format/MXRToolkit.htm>. Acesso em 03/12/2005.

NAKAKOJI, Kumio et al. Cognitive Effects of Animated Visualization in Exploratory Visual Data Analysis. In: International Conference of Information Visualization, IEEE, 5th, London, 2001. **Proceedings.** London, 2001, p. 77-84.

NCHART EDITOR. Disponível em: < <http://www.programurl.com/nevron-chart-editor.htm>>. Acesso em 01/03/2004.

NIRVE. User Guide for the NIST Information Retrieval Visualization Engine (NIRVE). Disponível em: <<http://www.itl.nist.gov/iaui/vvrg/cugini/uicd/nirve-home.html>> Acesso em: 21/05/2003.

OLIVEIRA, Juliano Lopes de.; MEDEIROS, Claudia Bauzer; CILIA, Mariano Ariel, Active Customization of GIS User Interfaces. In: IEEE International Conference on Data Engineering, Washington - DC, 1997. **Proceedings.** Washington - DC, 1997, p. 487-496.

PACIFIC NORTHWEST NATIONAL LABORATORY. **About Visualization at PNNL.** Disponível em: <<http://www.pnl.gov/infoviz/about.html>> Acesso em: 06/01/2003.

PIROLI, Peter; RAO, R. Table Lens as a Tool for Making Sense of Data. In: Workshop on Advanced Visual Interfaces, Gubbio - Umbria, 1996. **Proceedings.** Gubbio - Umbria, 1996. p. 67-80.

RAO, Ramana; CARD, Stuart K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In: ACM Conference on Human Factors in Computing Systems (CHI'94), Boston, Massachusetts. **Proceedings**. Boston, Massachusetts 1994. p. 318--322.

RAPOSO, Alberto B.; SZEMBERG, Flavio; GATTASS, Marcelo; CELES, Waldemar. Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração. In: Congresso da Sociedade Brasileira de Computação, XXIV, V.2, XXIII JAI – Livro Texto, SBC, Brasil, 2004, cap. 7. **Anais**. Brasil, 2004. p. 289-331.

RISSE, Thomas et al. Supporting Dynamic Information Visualization With VRML and Databases. In: Workshop on New Paradigms in Information Visualization and Manipulation, Washington D. C., 1998. **Proceedings**. Washington D.C., 1998. p. 69-72.

ROBERTSON, George G.; MACKINLAY, Jock D. The Document Lens, **ACM**, 1993, pp. 101-108.

SANTIN, Rafael; KIRNER, Claudio. Alteração Dinâmica de Objetos Virtuais em Realidade Aumentada. In: Workshop de Realidade Aumentada – WRA2005, II, Piracicaba: Universidade Metodista de Piracicaba, 2005. **Anais**. Piracicaba, 2005. p. 37-40.

SANTOS, C. Russo dos. Et al. **Mapping Information Onto 3D Virtual Worlds**, 2000. Disponível em: <<http://citeseer.nj.nec.com/418679.html>> Acesso em: 15/05/2003.

SCHMALSTIERG, Dieter et al. Bridging multiple user interface dimensions with augmented reality. In: IEEE and ACM International Symposium on Augmented Reality (ISAR'00). Munich, Germany, 2000. **Proceedings**. Munich, Germany. 2000. p. 20-29.

SCHMALSTIERG, Dieter et al. The Studierstube Augmented Reality Project. Teleoperators and Virtual Environments, Vol. 11, No. 1, MIT Press, 2002. pp. 33-54.

SEBRECHTS, M. et al. Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces, 2nd Annual Int'l ACM-SIGIR Conf. Research and Development in Information Retrieval, ACM Press, NewYork, 1999. p 3-10.

SHERMAN, Willian R.; CRAIG, Alan B. **Understanding Virtual Reality**. 2^a Ed. San Francisco - CA: Morgan e Morgan, 2003.

SHNEIDERMAN, Ben, Inventing Discovery Tools: Combining Information Visualization with Data Mining, **Information Visualization – Palgrave Macmilian Ltd**, 2002, pp. 5-12.

SHNEIDERMAN, BEN, The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization, In: IEEE Conference on Visual Languages, Boulder, CO, September, 1996. **Proceedings**. Boulder, CO, 1996. p. 336-343.

SPENCE, Robert. **Information Visualization**, Addison-Wesley, 2001.

STASKO, John T., Tree-Dimensional Computation Visualization, In: IEEE Symposium Visual Languages, New York - NY, 1993. **Proceedings**. New York - NY, 1993. p. 100-107.

USGS – U. S. GEOLOGICAL SURVEY. Geographic Information System, 2002. Disponível em: <<http://erg.usgs.gov/lsb/pubs/gisposter/>> Acesso em: 07/05/2003.

VREALITIES. Disponível em: <<http://www.vrealities.com>> Acesso em: 28/02/2005.

WAYNE, Rick. **The Folks at AVS Get It**, Software Development Magazine. Disponível em: <http://www.avs.com/software/soft_b/opencviz/index.html> Acesso em: 19/05/2003.

WHITE, Jennifer. **Virtual Reality and the Built Environment**. Great Britain: Architectural Press, 2002. 160 páginas.

WILHELM, Sherie A. **Virtual Reality: Now You Live it, Now You Don't**, 1994. Disponível em: < <http://www.cs.virginia.edu/misc/news-vr6t.html>> Acesso em: 21/04/2003.

WLOKA, M. M.. Interacting with Virtual Reality. **In Virtual Environments and Product Development Processes**. Chapman and Hall, 1995.

ANEXOS

ANEXO 1 – PARTE DO CÓDIGO DO PROGRAMA VRML

```

#VRML V2.0 utf8
# Definição do início do programa
DEF Entry Viewpoint {
    position 0 2 7.0
    orientation 1.2 0.8 0 -0.3
    fieldOfView 0.8
    description "Entry View"}
# Definição do tipo de navegação
NavigationInfo {
    type [ "EXAMINE", "ANY" ]
    headlight TRUE}
# Definição do plano de fundo
Background {
    skyColor [
        0.3 0.0 0.4,
        0.0 0.4 0.4,
        0.3 0.0 0.4,
        0.3 0.3 0.4,]
    skyAngle [
        1.47,
        1.57,
        1.67,]}
# Agrupamento dos objetos que formam o gráfico
Group {
    children [
# Definição do gráfico
DEF graphType Switch {
    whichChoice 0
    choice [
#0-----
# definição das barras do gráfico
DEF Bar Transform{
    scale 0.5 1.2 0.5
    children[

```

```

DEF C01 Transform {
translation -2 0 0
scale 0.35 0.1 0.35
rotation 1 0 0 0
children [
# Definição do plano de referência individual da barra
DEF SensorPlanoC01 Group {
children [
Transform {
translation 0 0.5 0 #posicao inicial da plano crítico
rotation 0 0 1 1.57
children [
DEF SensorC01 PlaneSensor {
maxPosition 50 0 # Pos max do sensor = tamanho do aramado
minPosition -50 0}
DEF PlanoCriticoC01 Transform {
rotation 0 0 1 -1.57
children[
Shape {
appearance DEF Cor_criticoC01 Appearance {
material Material {
diffuseColor 1 0 0
emissiveColor 1 0 0
transparency 0.2}}
geometry Cylinder {
radius 1.5
height 0.05
}}}}}}
DEF Shp0 Switch {
whichChoice 0
choice [
Transform{
translation 0 1 0
children [
Shape {
appearance Appearance {
material DEF ShpCol01 Material {diffuseColor 0.9 0.8 1.0
specularColor 0.5 0.5 0.5}}
geometry Cylinder {}}}}
Transform{

```

```

translation 0 1 0
children [
Shape {
appearance Appearance {
material USE ShpCol01}
#-----
geometry Box { }}}
Transform{
translation 0 1 0
children [
Shape {
appearance Appearance {
material USE ShpCol01}
geometry Cone { }}}]
#----- continua para as barras de C02 a C26
#=====Line
# Utilizando o nó extrusion são definidas as faces do gráfico
DEF Line Transform{
    scale 0.5 0.5 0.5
children[
DirectionalLight{
direction 0 -0.5 -1}
Transform{
translation 0 0 0
scale 1 2 1
children[
Shape {
appearance Appearance {
material DEF LC1 Material {
diffuseColor 0.0 0.8 1.0
specularColor 0.5 0.5 0.5}}
geometry DEF SP1 Extrusion {
creaseAngle 1
crossSection [
1.00 0.00, 0.92 -0.38,
0.71 -0.71, 0.38 -0.92,
0.00 -1.00, -0.38 -0.92,
-0.71 -0.71, -0.92 -0.38,
-1.00 -0.00, -0.92 0.38,
-0.71 0.71, -0.38 0.92,

```

```

0.00 1.00, 0.38 0.92,
0.71 0.71, 0.92 0.38,
1.00 0.00]
spine [
-2 0.4 0,
-1 -0.2 0,
0 0.6 0,
1 0.4 0,
2 0.2 0,]
scale [
0.02 0.4,
0.02 0.4,
0.02 0.4,
0.02 0.4,
0.02 0.4,]]]]}
#----- mesmo para as linhas 1 e 2
#=====
# Definição do plano de referência da base do gráfico Grade, Eixo X, Eixo Y e Eixo Z
# Título -----
Transform{
scale 0.4 0.4 0.4
translation -1.3 2.5 -1.5
children[
Shape {
appearance Appearance {material Material {diffuseColor 0.0 0.8 0.6
specularColor 0.5 0.5 0.5}}
geometry Text {string "Visualizacao de Informacoes" }}}}
# Plano de Referência com movimento-----
DEF cursorScript Script {
eventOut SFFloat cursorValue
eventOut SFVec3f cursorPosition
eventIn SFVec3f mousePosition
field SFFloat data 0.001
url "javascript:function mousePosition(value, time)
{
data = value[1];
if (data < -2.0) { data = -2.0; }
if (data > 2.0) { data = 2.0; }
cursorPosition[1] = data;
cursorValue = data;}"}
Transform {

```



```

children [
DEF CursorPSMov PlaneSensor {}
DEF CursorMov Transform{
translation 0 2.001 0
children[
Shape {
appearance Appearance {
material Material {
emissiveColor 0.5 0.5 0.5
transparency 0.3}}
geometry IndexedFaceSet {
coord Coordinate {
point [
-1.5 0.1 0.5
2.5 0.1 0.5
2.5 0.1 -1.5
-1.5 0.1 -1.5]}
coordIndex [0,1,2,3,-1]
solid FALSE
convex FALSE}}
Transform {
translation 2.5 0 0.4
scale 0.2 0.2 0.2
children[
Shape {
appearance Appearance {}
geometry Text {string "Plano de Referencia Z" }}}}]
#-----

# Roteamento dos caminhos de entrada e saída de informações das barras C01 a C26
ROUTE SensorC01.translation_changed TO PlanoCriticoC01.translation
ROUTE CursorPSMov.trackPoint_changed TO cursorScript.mousePosition
ROUTE cursorScript.cursorPosition TO CursorMov.set_translation
}}}

```

ANEXO 2 – PARTE DO PROGRAMA JAVASCRIPT QUE REALIZA PESQUISAS NO BANCO DE DADOS PARA ATUALIZAÇÃO DE VALORES

```

public void mudaDados (String sUrl) throws ServletException, IOException
{
    // Faz uma mudanca aleatoria na tabela.
    Connection con;
    Statement st;
    ResultSet rs;
    int i = 0;
    float fNumero = 0.0F;
    int nCodigo = 1;
    float fLargura = 0.0F;
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection(sUrl);
        st = con.createStatement();
        for (i = 0; i < 18; i++)
        {
            // Criando a altura e a largura randomicos de cada barra do grafico
            fNumero = (float)(Math.random() - Math.random());
            fLargura = (float) Math.abs((Math.random() - Math.random()));
            defineCores(fNumero,i+1);
            st.executeUpdate("update tbNumero set" +
                " numero = " + String.valueOf(fNumero) +
                "; corr = " + String.valueOf(nCorR) +
                "; corg = " + String.valueOf(nCorG) +
                "; corb = " + String.valueOf(nCorB) +
                "; largura = " + String.valueOf(fLargura) +
                " where codigo = " + Integer.toString(nCodigo++) );
        }
    }
    catch (SQLException eSQL)
    {
        throw new ServletException(eSQL.getMessage());
    }
}

```

```
        catch (Exception e)
        {
            throw new ServletException(e.getMessage());
        }
    }
    st.close();
    con.close();
}
catch (SQLException eSQL)
{
    throw new ServletException(eSQL.getMessage());
}
catch (Exception e)
{
    throw new ServletException(e.getMessage());
}
}
```

ANEXO 3 – TRECHO DO PROGRAMA JAVAAPPLET MOSTRANDO O RESULTADO DA PESQUISA A SER ENVIADO PARA O GRÁFICO

```

// Rotina que atualiza as barras dos graficos exibidos.
    float af1[ ][ ][ ] = new float[3][6][3];
    float af2[ ][ ][ ] = new float[3][6][4];
    // laço para pegar todas as linhas de graficos - 3 no total
    for(int i = 0; i < 3; i++)
    {
        // laço para pegar cada barra de grafico em cada linha
        for(int j = 0; j < 6; j++)
        {
            // as coordenadas de altura, largura e circunferencia seo fornecidas
            // por uma array 3D.
            af1[i][j][0] = 0.35F;
            af1[i][j][1] = 0.35F;
            af1[i][j][2] = 0.35F;
            af2[i][j][0] = 1.0F;
            af2[i][j][1] = 0.0F;
            af2[i][j][2] = 0.0F;
            af2[i][j][3] = 0.0F;
            if(af[i][j][0] == 0.0F)
            af[i][j][0] = 0.0001F;
            if (af[i][j][1] > 0.35F)
            {
                af1[i][j][0] = 0.35F;
                af1[i][j][2] = 0.35F;
            }
            else
            {
                af1[i][j][0] = af[i][j][1];
                af1[i][j][2] = af[i][j][2];
            }
            af1[i][j][1] = af[i][j][0];
        }
    }
    // Para cada barra, é chamado cada objeto/evento que ira alterar os graficos no VRML no
    browser
    rotation01.setValue(af2[0][0]);

```

```
scale01.setValue(af1[0][0]);
discoMed01.setValue(mediasPositivas[0]);
rotation02.setValue(af2[0][1]);
scale02.setValue(af1[0][1]);
discoMed01.setValue(mediasPositivas[1]);
rotation03.setValue(af2[0][2]);
scale03.setValue(af1[0][2]);
discoMed01.setValue(mediasPositivas[2]);
rotation04.setValue(af2[0][3]);
scale04.setValue(af1[0][3]);
discoMed01.setValue(mediasPositivas[3]);
rotation05.setValue(af2[0][4]);
scale05.setValue(af1[0][4]);
discoMed01.setValue(mediasPositivas[4]);
rotation06.setValue(af2[0][5]);
scale06.setValue(af1[0][5]);
discoMed01.setValue(mediasPositivas[15]);
rotation11.setValue(af2[1][0]);
scale11.setValue(af1[1][0]);
discoMed01.setValue(mediasPositivas[5]);
rotation12.setValue(af2[1][1]);
scale12.setValue(af1[1][1]);
discoMed01.setValue(mediasPositivas[6]);
rotation13.setValue(af2[1][2]);
scale13.setValue(af1[1][2]);
discoMed01.setValue(mediasPositivas[7]);
rotation14.setValue(af2[1][3]);
scale14.setValue(af1[1][3]);
discoMed01.setValue(mediasPositivas[8]);
rotation15.setValue(af2[1][4]);
scale15.setValue(af1[1][4]);
discoMed01.setValue(mediasPositivas[9]);
rotation16.setValue(af2[1][5]);
scale16.setValue(af1[1][5]);
discoMed01.setValue(mediasPositivas[16]);
rotation21.setValue(af2[2][0]);
scale21.setValue(af1[2][0]);
discoMed01.setValue(mediasPositivas[10]);
rotation22.setValue(af2[2][1]);
scale22.setValue(af1[2][1]);
```

```

discoMed01.setValue(mediasPositivas[11]);
rotation23.setValue(af2[2][2]);
scale23.setValue(af1[2][2]);
discoMed01.setValue(mediasPositivas[12]);
rotation24.setValue(af2[2][3]);
scale24.setValue(af1[2][3]);
discoMed01.setValue(mediasPositivas[13]);
rotation25.setValue(af2[2][4]);
scale25.setValue(af1[2][4]);
discoMed01.setValue(mediasPositivas[14]);
rotation26.setValue(af2[2][5]);
scale26.setValue(af1[2][5]);
discoMed01.setValue(mediasPositivas[17]);
}
public void setaValores (int nBarra, float fValor, int corr, int corg, int corb, float FvalorLargura,
float fMediaPos, float fMediaNeg)
{
    /* Funcao executada pelo script na pagina. Esta função que recebe os valores do
banco de dados e os atualiza nos graficos.
Valores recebidos:
nBarra      - Indice da Barra que estara recebendo os valores = {1..18}
fValor      - Valor float que deve ser repassado à barra - {-inf .. +inf}
corr        - Cor R do RGB - vermelho {- 0..255}
corg        - Cor G do RGB - Verde - {0.255}
corb        - Cor B do RGB - azul - {0.255}
FvalorLargura - Largura que a barra nBarra devera ter - {0..1}
fMediaPos   - Media dos valores positivos registrados - {0..inf}
fMediaNeg   - Media dos valores negativos registrados - {0.. -inf}
<inf> - infinito */

float af[ ] = new float[3];
float cor[ ] = new float[3];

af[0] = FvalorLargura;
af[1] = fValor;
af[2] = FvalorLargura;
cor[0] = (float) corr/255;
cor[1] = (float) corg/255;
cor[2] = (float) corb/255;

```

/ Cada CASE:*

seta-se o textfield com o valor do banco. o evento action e acionado automaticamente

largura = 0

altera-se a rotacao do objeto

altera-se a cor do objeto

seta a altura do disco de medias

seta a lagura do disco de medias

seta a profundidade do disco de medias

*executa o metodo que ira transferir os dados para o objeto */*

switch (nBarra)

{

case 1:

textField1.setText(String.valueOf(af[1]));

arrLarg[0] = af[0];

rotation01.setValue(af);

scale01.setValue(af);

diffuseColor01.setValue(cor);

mediasPositivas[0][0] = fMediaPos;

mediasPositivas[0][1] = 0.0F;

mediasPositivas[0][2] = 0.0F;

discoMed01.setValue(mediasPositivas[0]);

break;

case 2:

textField2.setText(String.valueOf(af[1]));

arrLarg[1] = af[0];

rotation02.setValue(af);

scale02.setValue(af);

diffuseColor02.setValue(cor);

mediasPositivas[1][0] = fMediaPos;

mediasPositivas[1][1] = 0.0F;

mediasPositivas[1][2] = 0.0F;

discoMed02.setValue(mediasPositivas[1]);

break;

case 3:

textField3.setText(String.valueOf(af[1]));

arrLarg[2] = af[0];

rotation03.setValue(af);

scale03.setValue(af);

diffuseColor03.setValue(cor);

mediasPositivas[2][0] = fMediaPos;

```
mediasPositivas[2][1] = 0.0F;  
mediasPositivas[2][2] = 0.0F;  
discoMed03.setValue(mediasPositivas[2]);  
break;
```

case 4:

```
textField4.setText(String.valueOf(af[1]));  
arrLarg[3] = af[0];  
rotation04.setValue(af);  
scale04.setValue(af);  
diffuseColor04.setValue(cor);  
mediasPositivas[3][0] = fMediaPos;  
mediasPositivas[3][1] = 0.0F;  
mediasPositivas[3][2] = 0.0F;  
discoMed04.setValue(mediasPositivas[3]);  
break;
```

case 5:

```
textField5.setText(String.valueOf(af[1]));  
arrLarg[4] = af[0];  
rotation05.setValue(af);  
scale05.setValue(af);  
diffuseColor05.setValue(cor);  
mediasPositivas[4][0] = fMediaPos;  
mediasPositivas[4][1] = 0.0F;  
mediasPositivas[4][2] = 0.0F;  
discoMed05.setValue(mediasPositivas[4]);  
break;
```

case 6:

```
textField6.setText(String.valueOf(af[1]));  
arrLarg[6] = af[0];  
rotation11.setValue(af);  
scale11.setValue(af);  
diffuseColor11.setValue(cor);  
mediasPositivas[5][0] = fMediaPos;  
mediasPositivas[5][1] = 0.0F;  
mediasPositivas[5][2] = 0.0F;  
discoMed06.setValue(mediasPositivas[5]);  
break;
```

case 7:

```
textField7.setText(String.valueOf(af[1]));  
arrLarg[7] = af[0];
```



```
rotation12.setValue(af);  
scale12.setValue(af);  
diffuseColor12.setValue(cor);  
mediasPositivas[6][0] = fMediaPos;  
mediasPositivas[6][1] = 0.0F;  
mediasPositivas[6][2] = 0.0F;  
discoMed12.setValue(mediasPositivas[6]);  
break;
```

case 8:

```
textField8.setText(String.valueOf(af[1]));  
arrLarg[8] = af[0];  
rotation13.setValue(af);  
scale13.setValue(af);  
diffuseColor13.setValue(cor);  
mediasPositivas[7][0] = fMediaPos;  
mediasPositivas[7][1] = 0.0F;  
mediasPositivas[7][2] = 0.0F;  
discoMed13.setValue(mediasPositivas[7]);  
break;
```

case 9:

```
textField9.setText(String.valueOf(af[1]));  
arrLarg[9] = af[0];  
rotation14.setValue(af);  
scale14.setValue(af);  
diffuseColor14.setValue(cor);  
mediasPositivas[8][0] = fMediaPos;  
mediasPositivas[8][1] = 0.0F;  
mediasPositivas[8][2] = 0.0F;  
discoMed14.setValue(mediasPositivas[8]);  
break;
```

case 10:

```
textField10.setText(String.valueOf(af[1]));  
arrLarg[10] = af[0];  
rotation15.setValue(af);  
scale15.setValue(af);  
diffuseColor15.setValue(cor);  
mediasPositivas[9][0] = fMediaPos;  
mediasPositivas[9][1] = 0.0F;  
mediasPositivas[9][2] = 0.0F;  
discoMed15.setValue(mediasPositivas[9]);
```

```
break;
```

```
case 11:
```

```
textField11.setText(String.valueOf(af[1]));  
arrLarg[12] = af[0];  
rotation21.setValue(af);  
scale21.setValue(af);  
diffuseColor21.setValue(cor);  
mediasPositivas[10][0] = fMediaPos;  
mediasPositivas[10][1] = 0.0F;  
mediasPositivas[10][2] = 0.0F;  
discoMed21.setValue(mediasPositivas[10]);  
break;
```

```
case 12:
```

```
textField12.setText(String.valueOf(af[1]));  
arrLarg[13] = af[0];  
rotation22.setValue(af);  
scale22.setValue(af);  
diffuseColor22.setValue(cor);  
mediasPositivas[11][0] = fMediaPos;  
mediasPositivas[11][1] = 0.0F;  
mediasPositivas[11][2] = 0.0F;  
discoMed22.setValue(mediasPositivas[11]);  
break;
```

```
case 13:
```

```
textField13.setText(String.valueOf(af[1]));  
arrLarg[14] = af[0];  
rotation23.setValue(af);  
scale23.setValue(af);  
diffuseColor23.setValue(cor);  
mediasPositivas[12][0] = fMediaPos;  
mediasPositivas[12][1] = 0.0F;  
mediasPositivas[12][2] = 0.0F;  
discoMed23.setValue(mediasPositivas[12]);  
break;
```

```
case 14:
```

```
textField14.setText(String.valueOf(af[1]));  
arrLarg[15] = af[0];  
rotation24.setValue(af);  
scale24.setValue(af);  
diffuseColor24.setValue(cor);
```

```
mediasPositivas[13][0] = fMediaPos;  
mediasPositivas[13][1] = 0.0F;  
mediasPositivas[13][2] = 0.0F;  
discoMed24.setValue(mediasPositivas[13]);  
break;
```

case 15:

```
textField15.setText(String.valueOf(af[1]));  
arrLarg[16] = af[0];  
rotation25.setValue(af);  
scale25.setValue(af);  
diffuseColor25.setValue(cor);  
mediasPositivas[14][0] = fMediaPos;  
mediasPositivas[14][1] = 0.0F;  
mediasPositivas[14][2] = 0.0F;  
discoMed25.setValue(mediasPositivas[14]);  
break;
```

case 16:

```
textField06.setText(String.valueOf(af[1]));  
arrLarg[5] = af[0];  
rotation06.setValue(af);  
scale06.setValue(af);  
diffuseColor06.setValue(cor);  
mediasPositivas[15][0] = fMediaPos;  
mediasPositivas[15][1] = 0.0F;  
mediasPositivas[15][2] = 0.0F;  
discoMed06.setValue(mediasPositivas[15]);  
break;
```

case 17:

```
textField07.setText(String.valueOf(af[1]));  
arrLarg[11] = af[0];  
rotation16.setValue(af);  
scale16.setValue(af);  
diffuseColor16.setValue(cor);  
mediasPositivas[16][0] = fMediaPos;  
mediasPositivas[16][1] = 0.0F;  
mediasPositivas[16][2] = 0.0F;  
discoMed16.setValue(mediasPositivas[16]);  
break;
```

case 18:

```
textField08.setText(String.valueOf(af[1]));
```

```
    arrLarg[17] = af[0];  
    rotation26.setValue(af);  
    scale26.setValue(af);  
    diffuseColor26.setValue(cor);  
    mediasPositivas[17][0] = fMediaPos;  
    mediasPositivas[17][1] = 0.0F;  
    mediasPositivas[17][2] = 0.0F;  
    discoMed26.setValue(mediasPositivas[17]);  
    break;  
}  
display();  
}
```

ANEXO 4 – PARTE DO CÓDIGO DO JAVASERVLET QUE ATUALIZA O BANCO DE DADOS ACCESS

```

public void alimentaHistorico (int nCodigo, float fNumero, Statement st1) throws
ServletException, Exception
{
    /* Esta rotina simula uma constante atualizacao na base de dados.*/
    String cmdSQL;
    String lsDataHora;
    java.util.Date dta = new java.util.Date();
    lsDataHora = String.valueOf(dta.getMonth() + 1) + "/" +
        String.valueOf(dta.getDate()) + "/" +
        String.valueOf(1900 + dta.getYear()) + " " +
        String.valueOf(dta.getHours()) + ":" +
        String.valueOf(dta.getMinutes()) + ":" +
        String.valueOf(dta.getSeconds());
    cmdSQL = "insert into tbHistorico (item, datahora, valor) " +
        "values (" + String.valueOf(nCodigo) + ", " + lsDataHora + ", " +
String.valueOf(fNumero) + ")";
    try
    {
        st1.executeUpdate(cmdSQL);
    }
    catch (SQLException eSQL)
    {
        throw new ServletException (eSQL.getMessage());
    }
}

```