**UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA**

LUCIANO PEREIRA SOARES

**UM AMBIENTE DE MULTIPROJEÇÃO TOTALMENTE
IMERSIVO BASEADO EM AGLOMERADOS DE
COMPUTADORES**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Engenharia Elétrica

São Paulo
2005

LUCIANO PEREIRA SOARES

**UM AMBIENTE DE MULTIPROJEÇÃO TOTALMENTE
IMERSIVO BASEADO EM AGLOMERADOS DE
COMPUTADORES**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Engenharia Elétrica

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Marcelo Knörich Zuffo

São Paulo
2005

FICHA CATALOGRÁFICA

Soares, Luciano Pereira

Um ambiente de multiprojeção totalmente imersivo baseado em aglomerados de computadores / L.P. Soares. -- São Paulo, 2005.

123 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Realidade virtual 2.Software livre 3.Arquiteturas paralelas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

A minha mãe e meu pai por todo o apoio e incentivo que me deram durante todos estes anos e a Andréia que tem sido uma excelente companheira.

AGRADECIMENTOS

Ao amigo e orientador Prof. Dr. Marcelo Knörich Zuffo pelas diretrizes seguras e permanente incentivo.

Aos professores João Antônio Zuffo, Márcio Lobo Netto e Roseli de Deus Lopes, que sempre me apoiaram nesta pesquisa e apresentaram sugestões excelentes.

Aos diversos colegas de laboratório, Paulo Bressan, Hilton Fernandes, Marcelo Paiva, Marcio Cabral, Glauco Todesco, Lucas Dulley, Breno Santos, Victor Henrique, Leonardo Nomura que sempre se dispuseram a ajudar quando tive dificuldades, técnicas ou pessoais.

Também aos diversos bolsistas e estagiários do laboratório, Tiago, Helena, Silvia, Rafael, Olavo, Aislan, Mário, Renato que sempre se empenharam em aprender e a desenvolver as tarefas que lhes foram requisitadas.

Toda a equipe da administração do laboratório que sempre me auxiliou nas dificuldades do nosso sistema burocrático.

O Prof. Hank Kaczmariski do Integrated Systems Laboratory da Universidade de Illinois, o Prof. Don Brutzman do Naval Postgraduate School em Monterey e o Prof. Joaquim Jorge do Instituto Superior Técnico de Lisboa que me acolheram para aperfeiçoar meus conhecimentos

Agradeço também ao Ana Teresa, Andréia, Lucas e a minha mãe Maria José pelas diversas revisões deste documento

APOIO

As seguintes empresas colaboraram direta ou indiretamente com este projeto

Absolut-Technologies

Barco

Cyclades

Furukawa

Itautec-Philco S. A.

Merfilumina

NetPro

Osram

Petrobrás

Silicon Graphics Inc.

T-Systems

FINANCIAMENTO

Este projeto contou com o financiamento dos seguintes projetos:

FINEP SISCOMPRO

Sistemas de Comunicação e Processamento Avançados e Aplicações Interativas
Nº 01.02.0288.00

FINEP HIPERSISTEMAS

Hipersistemas Integráveis
Nº 56.940260.00

INTEL do Brasil

Programa Educacional

RESUMO

Nas últimas décadas a tecnologia de realidade virtual tem sido cada vez mais utilizada em várias aplicações na sociedade. Particularmente à partir do início década de 90, observamos o advento da realidade virtual baseada em multiprojeções, com o surgimento do paradigma de construção de sistemas de realidade virtual totalmente imersivo como por exemplo o CAVE™. Esta tese tem por objetivo apresentar pesquisas e desenvolvimentos relacionadas com a implementação do primeiro sistema de multiprojeção totalmente imersivo baseado em aglomerados de computadores convencionais, sistema este denominado de CAVERNA Digital. O trabalho aqui apresentado é composto por um panorama atual do estado da arte das tecnologias de realidade virtual baseadas em multiprojeção. O conceito de aglomerados de computadores convencionais para realidade virtual, ou especificamente VRCluster é apresentado, bem como uma taxonomia de organização interna dos VRClusters. O sistema de multiprojeção da CAVERNA Digital é apresentado, descrevendo em detalhes as opções dos inúmeros subsistemas como projetores, telas, subsistema de vídeo, dispositivos de interação e outros. Outra contribuição desta tese é uma plataforma de visualização interativa voltada para ambientes de realidade virtual imersivos, plataforma esta denominada de JINX, baseada no padrão aberto X3D. Aspectos relacionados com a operação destes sistemas de realidade virtual avançados são apresentados como é o caso das interface de gerenciamento de ambientes imersivos denominado de SIRIUS e do núcleo de gerenciamento de ambientes imersivos denominado de ZEUS. Por fim vários testes de desempenho são realizados para a validação da integração dos inúmeros recursos utilizados na CAVERNA Digital.

ABSTRACT

In the last decades, virtual reality technology has been increasingly used in many applications for the society. Particularly in the beginning of 90's, we can observe the advent of virtual reality based in multi-projections, leading to the construction paradigm of CAVE™ like virtual reality systems. The goal of this thesis is to present the research and developments related to the construction of the first multi projection completely immersive system based on commodity computer clusters called CAVERNA Digital. The work here presented is composed by an updated panorama of the state-of-art multi-projection virtual reality technologies. The concept of commodity computer clusters for virtual reality, or specifically VRCluster is shown, as well as the taxonomy for internal organization of the VRClusters. The CAVERNA Digital multi-projection system is shown, including details of the options of uncountable subsystems, like projectors, screens, video, and interaction devices, among others. This thesis contributes with another innovation: the visualization platform, called JINX, based on the open standard X3D. Aspects related to the operation of these advanced virtual reality systems are presented, which is the case of the user interface for immersive environments called SIRIUS and the management core for immersive environments called ZEUS. Finally, various performance tests were conducted to validate the integration of the uncountable resources used in the CAVERNA Digital.

SUMÁRIO

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

NOTAÇÃO

1.	INTRODUÇÃO	1
1.1	Relevância.....	2
1.2	Motivação	5
1.3	Objetivo	5
1.4	Metodologia de pesquisa.....	7
1.5	Metodologia de desenvolvimento.....	8
1.5.1	Engenharia de software baseada em <i>Unified Modeling Language</i> - UML	8
1.5.2	Organização de código fonte baseada na ferramenta <i>Concurrent Versions System</i> - CVS	8
1.6	Trabalhos correlatos.....	9
1.7	Estrutura da tese.....	9
2.	O ESTADO DA ARTE EM AGLOMERADOS DE COMPUTADORES PARA REALIDADE VIRTUAL	11
2.1	Introdução	11
2.2	Agglomerados de computadores convencionais	11
2.2.1	Agglomerados de computadores para cálculo numérico.....	12

2.2.2	Aglomerados de computadores para aplicações transacionais.....	13
2.2.3	Aglomerados de computadores para aplicações gráficas	14
2.3	Sincronização em aglomerados de computadores para aplicações da realidade virtual	14
2.3.1	O Sincronismo de vídeo - <i>Genlock</i>	16
2.3.2	Sincronismo de quadros - <i>Frame-lock</i>	18
2.3.3	Sincronismo de taxa de atualizações - <i>Rate-lock</i>	20
2.3.4	Sincronismo de dados - <i>Data-lock</i>	20
2.4	Sistemas paralelos de geração de imagens	20
2.4.1	Taxonomia de paralelização em computação gráfica	22
2.5	Sistemas centralizados de realidade virtual.....	24
2.6	Bibliotecas de desenvolvimento para realidade virtual	25
2.7	Conclusões	29
3.	ALTERNATIVAS TECNOLÓGICAS PARA A IMPLEMENTAÇÃO DO VRCLUSTER.....	31
3.1	Introdução	31
3.2	O sistema operacional.....	31
3.3	Sistemas de transmissão de dados e sincronismo	32
3.4	Protocolos de memória compartilhada	33
3.4.1	Interface de programação PThreads.....	33
3.4.2	Interface de programação OpenMP	33
3.5	Protocolos de gerenciamento de dados.....	33
3.6	Bibliotecas de desenvolvimento para CG/RV	34
3.7	Formatos de descrição de cena	34
3.7.1	Formato de descrição de cena VRML.....	35
3.7.2	Formato de descrição de cena X3D	35
3.8	Sistemas de vídeo	35
3.8.1	Cartões gráficos	36
3.8.2	Comutador de rede.....	37

3.9	Taxonomia do VRCluster.....	37
3.10	Conclusões	40
4.	CAVERNA DIGITAL	41
4.1	Introdução	41
4.1.1	Sistema semi-esférico	42
4.2	Aspectos arquitetônicos e estruturais.....	43
4.2.1	A CAVERNA de baixo custo	44
4.2.2	O aglomerado de realidade virtual	44
4.3	Conclusões	51
5.	OPERAÇÃO E GERENCIAMENTO DE AMBIENTES DE REALIDADE VIRTUAL	52
5.1	Gerenciamento de instalação e configuração	52
5.2	Gerenciamento de operação.....	52
5.3	Gerenciamento de dispositivos eletrônicos	54
5.3.1	Sistema multiseria.....	54
5.3.2	Mesa de controle.....	55
5.3.3	Reconhecimento de voz	55
5.3.4	Régua de força inteligente	55
5.3.5	Comutador de teclado, vídeo e mouse	55
5.4	Disponibilidade do sistema.....	56
5.5	Integração dos recursos de gerenciamento	56
5.5.1	Estação de controle	56
5.5.2	Núcleo do sistema de gerenciamento - ZEUS.....	57
5.5.3	Interface do sistema de gerenciamento - SIRIUS	58
5.5.4	Rastreador – um exemplo de caso.....	65
5.6	Conclusões	66
6.	PROPOSTA E DESENVOLVIMENTO DE UMA PLATAFORMA DE VISUALIZAÇÃO IMERSIVA	67

6.1	Introdução	67
6.1.1	Sistema de projeções e estereoscopia.....	68
6.1.2	Gerenciamento das geometrias	71
6.1.3	Sistema de som	71
6.1.4	Sistema de controle e navegação	72
6.1.5	Sincronismos e comunicação.....	74
6.2	Desenvolvimento baseado no método proposto	76
6.2.1	Leitura de arquivos	76
6.2.2	Hierarquia de sistemas	78
6.2.3	Avanços em <i>Shading Languages</i>	78
6.3	Imagens diversas de aplicações desenvolvidas	79
6.4	Conclusões	85
7.	VALIDAÇÃO EXPERIMENTAL DOS SISTEMAS DESENVOLVIDOS	86
7.1	Introdução	86
7.1.1	Geração de imagens paralelas	86
7.2	Comparações entre plataformas.....	86
7.3	Testes de desempenho.....	90
7.3.1	Teste de escalabilidade	90
7.3.2	Teste de qualidade visual	95
7.3.3	Teste de dependência de rede.....	97
7.3.4	Teste de estabilidade	97
7.3.5	Teste do sistema com aplicações de terceiros	97
7.3.6	Teste de desempenho do sistema de gerenciamento	97
7.4	Pesquisas em aglomerados de outros centros	98
7.5	Conclusões	99
8.	CONCLUSÕES E TRABALHOS FUTUROS	100
8.1	Introdução	100
8.2	Contribuições	100
8.2.1	Trabalhos publicados	101

8.2.2	Inovações tecnológicas	102
8.2.3	A transferência de tecnologia para a Itaotec-Philco.....	104
8.3	Trabalhos futuros	104
8.4	Perspectivas futuras.....	106
ANEXOS		107
ANEXO I – Componentes da CAVERNA Digital		107
	Telas	107
	Espelhos	108
	Rastreadores e leitores de posição.....	108
	Sistema de som.....	109
	Comutador de vídeo	109
	Projetores	110
	Proteção acústica.....	111
	Subsistema de iluminação	112
	Refrigeração	112
	Óculos estereoscópicos	112
ANEXO II – Diagrama UML do sistema SIRIUS		113
ANEXO III – Detalhes de implementação da JINX		115
	IDF0	115
	Diagrama de classes	116
	Diagrama de seqüência	119
	Componentização.....	120
ANEXO IV – Arquivo de configuração da JINX		121
ANEXO V - Reconhecimento de voz		122
LISTA DE REFERÊNCIAS		123
APÊNDICES.....		130
GLOSSÁRIO		157
ÍNDICE		161

LISTA DE FIGURAS

Figura 1-1 – Gráfico da evolução dos processadores convencionais.....	2
Figura 1-2 – Simulação de grande complexidade geométrica	4
Figura 1-3 – Estrutura da CAVERNA Digital	6
Figura 2-1 – Ilustração de aglomerado de computadores baseado em lâminas	13
Figura 2-2 – Ilustração de sistema de computação em grade.....	13
Figura 2-3 – Distribuição de tarefas em um aglomerado de computadores.....	14
Figura 2-4 – Tubo de raios catódicos	17
Figura 2-5 – Sinais de vídeo do aglomerado.....	18
Figura 2-6 – Projeções sincronizadas (esquerda) e dessincronizadas (direita).....	18
Figura 2-7 – Pontos de distribuição de dados para paralelismo gráfico	22
Figura 2-8 – Técnica <i>sort-first</i> e <i>sort-last</i>	23
Figura 2-9 – Placas da Lightning2 e DVG.....	24
Figura 2-10 – Esquema de cartões gráficos de uma estação gráfica Onyx.....	25
Figura 2-11 – Exemplo do CAVELib	26
Figura 2-12 – Exemplo do CAVERNSoft	26
Figura 2-13 – Exemplo do OpenSG.....	26
Figura 2-14 – Exemplo do AVANGO	27
Figura 2-15 – Exemplo do VRJuggler	27
Figura 2-16 – Exemplo do Syzygy	27

Figura 2-17 – Exemplo da DIVERSE.....	28
Figura 2-18 – Exemplo da Glass.....	28
Figura 2-19 – Exemplo do Avalon.....	28
Figura 2-20 – Exemplo do IC:IDO	29
Figura 2-21 – Exemplo do DeltaGen	29
Figura 3-1 – Logo X3D.....	35
Figura 3-2 – Conectores VGA e DVI	36
Figura 3-3 – Comutador de Rede.....	37
Figura 3-4 – VRCluster assimétrico distribuído heterogêneo.....	38
Figura 3-5 – VRCluster simétrico centralizado homogêneo.....	39
Figura 3-6 – VRCluster assimétrico centralizado heterogêneo.....	39
Figura 4-1 – O primeiro <i>CAVE</i> TM	41
Figura 4-2 –Projeção esférica.....	42
Figura 4-3 – Leiaute da CAVERNA Digital e pilares e vigas de madeira	43
Figura 4-4 – Acomodação visual do usuário	44
Figura 4-5 – Primeiro e segundo aglomerado.....	45
Figura 4-6 – Sistemas de comunicação.....	47
Figura 4-7 – Cartões gráficos 4210 e 3000G, utilizados no aglomerado gráfico	48
Figura 4-8 –Organização desestruturada de cabos.....	50
Figura 5-1 – Servidor Multiserial.....	54
Figura 5-2 – Sistema de gerenciamento baseado no servidor multiserial.....	54

Figura 5-3 – Mesa de controle	55
Figura 5-4 – Régua de força inteligente.....	55
Figura 5-5 – KVM.....	55
Figura 5-6 – Estação de controle.....	56
Figura 5-7 – Diagrama de funcionamento do sistema	57
Figura 5-8 – Diagrama de transmissão de comandos.....	58
Figura 5-9 – Sistema de controle de iluminação.....	59
Figura 5-10 – Painel de aplicações	59
Figura 5-11 – Painel de controle de projetores	60
Figura 5-12 – Painel do comutador de vídeo	61
Figura 5-13 – Comunicação via Telnet.....	61
Figura 5-14 – Monitoramento do aglomerado	62
Figura 5-15 – Controle das câmeras robotizadas	62
Figura 5-16 – Sistema de rastreamento.....	63
Figura 5-17 – Painel de desenvolvedores	63
Figura 5-18 – Sistema de reconhecimento de voz	64
Figura 5-19 – Telas do SIRIUS Light.....	64
Figura 5-20 – Comunicação do sistema de rastreamento.....	65
Figura 6-1 – Câmeras formato dedão.....	68
Figura 6-2 – Câmeras formato correto	68
Figura 6-3 – Menu de interação	69

Figura 6-4 – Formatos de telas suportados	70
Figura 6-5 – Tipos de transformações suportados	71
Figura 6-6 – Outros dispositivos suportados pela JINX	72
Figura 6-7 – Sistema de comunicação de dispositivos.....	73
Figura 6-8 – Sistema híbrido de comunicação.....	74
Figura 6-9 – Sistema de comunicação da JINX	75
Figura 6-10 – Exemplo de transmissão de mensagens	76
Figura 6-11 – Exemplo de grafo de cena	77
Figura 6-12 – Hierarquia da JINX	78
Figura 6-13 – Exemplo de aplicação com <i>Shading Languages</i>	78
Figura 6-14 – Primeiras cenas da JINX (captura de tela)	79
Figura 6-15 – Primeiras visualizações da JINX.....	79
Figura 6-16 – Simulação de Escher	79
Figura 6-17 – Simulação arquitetônica	80
Figura 6-18 – Imagens da simulação do Rio de Janeiro (captura de tela)	80
Figura 6-19 – Sistema do Rio de Janeiro na CAVERNA Digital.....	81
Figura 6-20 – Passeio no Rio de Janeiro em <i>PowerWall</i>	81
Figura 6-21 – Sistema apresentado na COMDEX03 e SIBGRAPI.....	82
Figura 6-22 – Apresentação no SIGGRAPH 04	82
Figura 6-23 – Visualização molecular	83
Figura 6-24 – Treinamento de disjuntores elétricos.....	83

Figura 6-25 – Visualização da Universidade de São Paulo	84
Figura 6-26 – Navegação de cenário produzido pelo LEGAL	84
Figura 6-27 – Imagens diversas do METLAB, sistema desenvolvido em VTK	85
Figura 7-1 – Pipeline gráfico	86
Figura 7-2 – Estação Onyx.....	86
Figura 7-3 – Cartões gráficos comerciais (NVidia, FileGL, WildCat).....	87
Figura 7-4 – Amostras dos dados usados pelo SpecViewPerf.....	89
Figura 7-5 – Aplicativo glxgears	90
Figura 7-6 – Testes preliminares da queda de desempenho em aglomerados	91
Figura 7-7 – Resolução relativa ao número de nós	92
Figura 7-8 – Contensão de rede do modelo Rio de Janeiro em monoscopia.....	92
Figura 7-9 – Contensão de rede do modelo Rio de Janeiro em estereoscopia.....	93
Figura 7-10 – Contensão de rede do modelo Rio de Janeiro em modo esférico.....	93
Figura 7-11 – Queda de desempenho em relação a número de polígonos.....	94
Figura 7-12 – Tempo de carga de arquivos X3D.....	95
Figura 7-13 – Efeito da resolução na CAVERNA Digital 640x480 X 1280x1024...	96
Figura 7-14 – Teste de ping-pong para calcular latência da rede	98
Figura 7-15 – Exemplos do ISL, INRIA e Fraunhofer	98
Figura 8-1 – Info Cluster Gráfico	104
Figura I - 1 – Telas	107
Figura I - 2 – Espelho.....	108

Figura I - 3 – Rastreador eletro-magnético	108
Figura I - 4 – Caixa acústica	109
Figura I - 5 – Diagrama de utilização de comutador de vídeo	110
Figura I - 6 – Projetor.....	110
Figura I - 7 – Simulação da CAVERNA Digital.....	111
Figura I - 8 – Proteção acústica.....	111
Figura I - 9 – Controle de luz	112
Figura I - 10 – Ar condicionado.....	112
Figura I - 11– Óculos estereoscópico ativo.....	112
Figura II - 1 – Diagrama de responsabilidades	113
Figura II - 2 – Diagrama de classes.....	114
Figura II - 3 – Diagrama de hierarquia.....	114

LISTA DE TABELAS

Tabela I – Latência de comunicação para várias tecnologias de interconexão.....	15
Tabela II – Comparativo de sistemas imersivos	30
Tabela III – Comparação SGI Onyx IR3 x 3Dlabs Wildcat 4210	88
Tabela IV – Resultados de um nó do Perfect Cluster para o SpecViewPerf	89
Tabela V – Desempenho relativo ao aplicativo glxgears.....	91
Tabela VI – Referências das siglas de queda de desempenho	94

LISTA DE ABREVIATURAS E SIGLAS

ACK	ACKNOWLEDGMENT
AGP	ACCELERATED GRAPHICS PORT
API	APPLICATION PROGRAMMING INTERFACES
BTU	BRITISH THERMAL UNIT
CAVE	CAVE AUTOMATIC VIRTUAL ENVIRONMENT
CG	COMPUTAÇÃO GRÁFICA
CORBA	COMMON OBJECT REQUEST BROKER ARCHITECTURE
CPU	CENTRAL PROCESSING UNIT
CRT	CATHODE RAY TUBE
CVS	CONCURRENT VERSION SYSTEM
DDR	DOUBLE DATA RATE
DDWG	DIGITAL DISPLAY WORKING GROUP
DHCP	DYNAMIC HOST CONFIGURATION PROTOCOL
DLP	DIGITAL LIGHT PROCESSING
DMD	DIGITAL MICROMIRROR DEVICE
DOM	DOCUMENT OBJECT MODEL

DTD	DOCUMENT TYPE DEFINITION
DVG	DIGITAL VIDEO GRAPHICS
DVI	DIGITAL VIDEO INTERFACE
EVL	ELECTRONIC VISUALIZATION LABORATORY
GLUT	OPENGL UTILITY LIBRARY
GLX	OPENGL EXTENSION TO THE X WINDOW SYSTEM
GNU	GNU'S NOT UNIX
GPU	GRAPHICAL PROCESSOR UNIT
HA	HIGH AVAILABILITY
HMD	HEAD MOUNTED DISPLAYS
IEEE	INSTITUTE OF ELECTRICAL AND ELETRONICS ENGINEERS
IDE	INTEGRATED DISK ELECTRONICS
IP	INTERNET PROTOCOL
IPMI	INTELLIGENT PLATFORM MANAGEMENT INTERFACE
KVM	KEYBOARD VIDEO MOUSE
LCD	LIQUID CRYSTAL DISPLAY
LED	LIGHT EMITING DIODE
LSI	LABORATÓRIO DE SISTEMAS INTEGRÁVEIS

MODEM	MODULATOR-DEMOLUTADOR
MPI	MESSAGE PASSING INTERFACE
NASA	NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
OMG	OBJECT MANAGEMENT GROUP
PAL	PHASE ALTERNATION LINES
PCI	PERIPHERAL COMPONENT INTERCONNECT
POSIX	PORTABLE OPERATING SYSTEM INTERFACE
PVM	PARALLEL VIRTUAL MACHINE
PXE	PRE BOOT EXECUTION ENVIRONMENT
RGBHV	RED GREEN BLUB HORIZONTAL VERTICAL
RV	REALIDADE VIRTUAL
SAX	SIMPLE API FOR XML
SCSI	SMALL COMPUTER SYSTEM INTERFACE
SGI	SILICON GRAPHICS INC.
SMP	SYMMETRICAL MULTIPROCESSOR
SPEC	STANDARD PERFORMANCE EVALUATION CORPORATION
TCP	TRANSMISSION CONTROL PROTOCOL
UDA	UNIFIED DRIVER ARCHITECTURE

UDP	USER DATAGRAM PROTOCOL
UML	UNIFIED MODELING LANGUAGE
USB	UNIVERSAL SERIAL BUS
USP	UNIVERSIDADE DE SÃO PAULO
UTP	UNSHIELDED TWISTED PAIR
VIA	VIRTUAL INTERFACE ARCHITECTURE
VRML	VIRTUAL REALITY MODELING LANGUAGE
VTk	VISUALIZATION TOOL KIT
VSS	VIRTUAL SOUND SERVER OU VANILLA SOUND SERVER
X3D	EXTENSIBLE 3D
XML	EXTENSIBLE MARKUP LANGUAGE

NOTAÇÃO

Abaixo descrevemos a notação utilizada nesta tese:

A) Termos em inglês onde não se encontraram traduções diretas ao português, até o momento da escrita da tese, serão apresentados em itálico.

Exemplo: *tracking*

B) Termos em inglês traduzidos serão apresentados à direita do termo em português em itálico e entre parênteses.

Exemplo: rastreamento (*tracking*)

C) Quando for definido um novo termo, este será apresentado em negrito pela primeira vez e utilizado ao longo da dissertação.

Exemplo: **rastreamento** é o mecanismo de determinação de posição do usuário no interior do sistema.

D) Citações à bibliografia serão apresentadas entre colchetes, a ordem de citação segue a regra: nome do primeiro autor em letras maiúsculas concatenado ao ano de publicação.

Exemplo: [SUTHERLAND97]

E) Figuras não referenciadas foram produzidas pelo autor da Tese

1. INTRODUÇÃO

Os **aglomerados de computadores** são um conjunto de computadores, capazes de compartilhar seus recursos, para um objetivo comum. Estes aglomerados, devem ser providos de mecanismo de gerenciamento e aplicações, tendo assim um ganho de desempenho proporcional ao número de estações utilizadas e podendo executar cálculos de grande complexidade numérica, transacional e gráfica.

Atualmente podemos observar uma rápida disseminação de **aglomerados de computadores convencionais** (*commodity computer clusters*) em várias aplicações desde a ciência até a educação e entretenimento. A tendência de utilizar computadores convencionais para a construção de aglomerados é cada vez maior, sendo que vários centros de pesquisas, bem como diversas empresas, estão investindo cada vez mais nesta alternativa de alto desempenho. Particularmente, o maior interesse nesta pesquisa está no desenvolvimento de aglomerados de computadores voltados para as aplicações avançadas da computação gráfica interativa e realidade virtual imersiva.

Existe um crescente uso da tecnologia de aglomerado de computadores, em especial, aqueles que utilizam produtos do mercado consumidor convencional (*commodity*). O interesse se deve ao fato de ser uma tecnologia com uma disponibilidade muito maior, o que conseqüentemente reduz seu custo.

A informática, de uma maneira geral, evolui rapidamente. Segundo a lei de Moore [MOORE65] apresentada na Figura 1-1, esta evolução é exponencial, ou seja, o ciclo de obsolescência dos sistemas computacionais é muito rápido. Atualizações em grandes servidores usualmente necessitam a troca de todo o sistema, já para aglomerados de computadores convencionais, novos nós de processamento podem ser incluídos ou substituídos facilmente, viabilizando um sistema em contínua atualização com um custo mínimo. A manutenção também fica facilitada devido a fácil aquisição dos componentes do sistema e a maior facilidade de encontrar mão de obra especializada.

Figura 1-1 – Gráfico da evolução dos processadores convencionais

Nesta pesquisa propomos o uso de aglomerados de computadores convencionais para uma nova classe de aplicações, no caso, as aplicações em Realidade Virtual. O **Aglomerado de Realidade Virtual** ou como denominamos **VRCluster** é justamente um aglomerado de computadores convencionais voltado especificamente para satisfazer as necessidades computacionais das aplicações da realidade virtual imersiva. As principais características do Aglomerado de Realidade Virtual consistem em incorporar recursos para manter a coerência visual, além do alto desempenho gráfico necessário para aplicações interativas. A **coerência visual** é fundamental em sistemas imersivos, pois faz com que as imagens tenham continuidade geométrica e temporal.

Foram desenvolvidas técnicas de projeto, instalação, configuração e utilização de um sistema de realidade virtual completo com foco em ambientes de visualização imersivos, em especial a **CAVERNA Digital** [ZUFFOJ01].

1.1 Relevância

Devido a crescente demanda por sistemas de realidade virtual e o incremento da complexidade das aplicações para estes sistemas, houve e ainda há a necessidade de se desenvolver soluções alternativas aos servidores gráficos de alto desempenho e de alto custo já disponíveis no mercado, por soluções que não utilizam eletrônica nem sistemas proprietários além de apresentar baixo custo, para assim facilitar sua utilização e disseminação na sociedade.

¹ Fonte: <http://www.intel.com/research/silicon/mooreslaw.htm>

É importante ressaltar a relevância científica e os resultados inéditos de impacto internacional deste projeto. O desenvolvimento deste projeto resultou na construção efetiva do primeiro sistema de realidade virtual de múltiplas projeções, imersivo, baseado em aglomerado de computadores convencionais, utilizando memória distribuída, capaz de oferecer o mesmo desempenho de sistemas multi-processados de memória compartilhada.

O primeiro sistema de **multiprojeção** imersivo cúbico, denominado de CAVE™ (*CAVE Automatic Virtual Environment*)™ [CRUZ93], criado e desenvolvido pelo EVL (*Electronic Visualization Laboratory*) na Universidade de Illinois em 1992, era acionado por seis estações gráficas modelo Silicon Graphics™ VGX e uma estação gráfica modelo Silicon Graphics™ Personal Iris como mestre, interconectadas por uma rede Ethernet [CRUZ92]. Logo após, este sistema foi aperfeiçoado, sendo acionado por um aglomerado de estações gráficas modelo Silicon Graphics™ Personal Iris, interconectadas por uma rede de fibra óptica Scramnet [SCRAMNET04], para o compartilhamento de memória [CRUZ93]. Na época, os custos deste aglomerado de estações gráficas eram muito elevados, e com o surgimento das estações gráficas multiprocessadas, com múltiplas saídas de vídeo, modelo Silicon Graphics™ Onyx [ONYX04], a utilização de aglomerados de estações gráficas foi abandonada pela comunidade de realidade virtual.

Com a popularização da tecnologia de realidade virtual, diversas comunidades, além daquela de informática, poderão utilizar a realidade virtual para propiciar novas pesquisas e descobertas, tendo assim uma abrangência a todas as áreas do conhecimento.

Devido à redução do custo da tecnologia de realidade virtual, acredita-se que com o decorrer dos anos esta começará a convergir com sistemas eletrônicos domésticos, como a televisão, tornando-se um meio comum para se acessar informação, cultura e entretenimento.

Podemos observar na literatura da computação de alto desempenho uma ênfase muito grande no uso de aglomerados de computadores para aplicações numericamente

intensivas. Neste projeto, pretendemos mostrar que as arquiteturas baseadas em aglomerados são também viáveis para aplicações gráficas interativas.

Além disso, podemos constatar os diversos projetos financiados para o desenvolvimento de aglomerados para realidade virtual, por diversas empresas e agências, inclusive estrangeiras, que acreditaram no desempenho do projeto e que, atualmente, se interessam em repassar esta tecnologia para a sociedade. Como exemplo, a empresa brasileira Itautec S.A. está investindo na transferência de tecnologia para viabilizar o produto aglomerados de computadores gráficos na sua linha de produção.

Esta pesquisa tem aplicação direta nas seguintes áreas: Óleo e Gás; Automobilismo; Aeronáutica; Entretenimento; Previsão de tempo; Energia; Museus; Arquitetura; Pesquisa Científica; Educação em geral; Arte e Cultura.

A utilização de realidade virtual nestas áreas tradicionalmente necessita de um grande investimento para aquisição dos equipamentos, aplicativos e em especial super computadores. Com o novo paradigma proposto nesta pesquisa, estas tecnologias ficam muito mais acessíveis e escaláveis, permitindo a sua utilização em centros que antes não dispunham de recurso para isto.

Figura 1-2 – Simulação de grande complexidade geométrica

A Figura 1-2 apresenta uma simulação para a área de óleo e gás, que permite fazer a visualização detalhada de uma plataforma petrolífera auxiliando a tomada de

decisões para planejamento e produção. Esta simulação só foi possível pela utilização de aglomerados de computadores por estes serem capazes de processar modelos de grande complexidade geométrica em tempo real.

1.2 Motivação

O LSI (Laboratório de Sistemas Integráveis) já vem desenvolvendo há muitos anos sistemas de **computação de alto desempenho** (*High Performace Computing*), baseados em aglomerados de computadores, como alternativa de baixo custo para as várias aplicações científicas e industriais. Assim, aproveitando esta experiência na tecnologia, pretendeu-se ampliar as possibilidades para ambientes de computação gráfica e realidade virtual.

Além disso, no ano de 2001, o laboratório recebeu recursos, através do projeto FINEP Hipersistemas Integráveis, para a construção do primeiro sistema de multiprojeção imersivo cúbico da América Latina, denominado de CAVERNA Digital, o que era uma chance inédita de desenvolver pesquisas e desenvolvimentos em um ambiente de computação não antes disponível para a comunidade brasileira.

Finalmente, existe o interesse do próprio pesquisador, vislumbrando a possibilidade do desenvolvimento de uma arquitetura inédita na comunidade científica internacional e aprofundamento dos seus conhecimentos na área de computação gráfica de alto desempenho e realidade virtual.

1.3 Objetivo

O objetivo desta tese é a pesquisa, desenvolvimento, implementação e análise para a viabilização de um sistema completo de realidade virtual imersiva baseado em aglomerados de computadores convencionais (VRCluster), considerando sistemas de multiprojeção, em especial a CAVERNA Digital, e todas as eventuais aplicações.

Uma das maiores barreiras em realidade virtual é a demanda computacional para a geração e multiprojeção de imagens realísticas em tempo real, demandando pesquisa e desenvolvimento em sistemas de computação e processamento gráfico de alto desempenho. Levando-se em conta estes aspectos, as aplicações correntes são resultado de compromissos entre qualidade e interatividade.

Dentre as várias possibilidades de sistemas de multiprojeção imersivos, decidiu-se implementar uma CAVERNA Digital² baseada em aglomerados de computadores convencionais. Isto é, optou-se por um sistema montado na forma de um cubo, onde imagens de alta resolução são projetadas em cada uma das faces do mesmo, permitindo que os usuários fiquem totalmente imersos em uma simulação gerada por computador, como mostra a Figura 1-3. Dentre as muitas vantagens deste tipo de sistema, podemos citar sua alta resolução, grande imersão e ampla cobertura do campo de visão (*field-of-view*).

Figura 1-3 – Estrutura da CAVERNA Digital

Sistemas de projeções têm sido utilizados há muito tempo, contudo, os mais comuns são baseados em uma única projeção, como no cinema, atrações de parques temáticos, ou mesmo, sistemas domésticos. O que torna difícil sistemas com múltiplas projeções é a coerência visual das imagens das diversas fontes de forma que se apresentem uniformes. Neste caso, existe a necessidade das fontes de vídeo iniciarem no mesmo instante e serem exibidas na mesma velocidade.

Sistemas de projeção cúbica podem ter de 3 a 6 faces e requerem sistemas computacionais capazes de gerar e sincronizar imagens em alta resolução a partir da mesma base de dados. Neste trabalho, descrevemos aspectos de projeto e implementação da CAVERNA Digital, localizado na Escola Politécnica da Universidade de São Paulo. O grande diferencial da CAVERNA Digital, em relação

² **CAVERNA Digital** é uma denominação para sistemas de multiprojeção cúbica, criada pelos pesquisadores do Laboratório de Sistemas Integráveis da Universidade de São Paulo

aos outros sistemas de projeção cúbica ao redor do mundo, é seu acionamento pelo aglomerado de computadores convencionais (VRCluster). Atualmente este é responsável pela geração, gerenciamento e multiprojeção de imagens na CAVERNA Digital.

1.4 Metodologia de pesquisa

Como primeira fase dos trabalhos, foram feitas amplas pesquisas bibliográficas, tanto da área acadêmica como comercial. Foi trabalhada uma taxonomia para os aglomerados de computadores e em seguida as implementações.

Um das metas da pesquisa foi o desenvolvimento do aglomerado de realidade virtual (VRCluster) baseado em computadores convencionais. Cada estação é responsável por diversas funcionalidades, as quais podem variar em um largo escopo, tais como, vídeo, **áudio espacializado**, entrada de dados, além de cálculos necessários para a simulação. A função essencial do aglomerado é a multiprojeção de imagens estereoscópicas com coerência visual. Este sistema incorpora uma hierarquia de recursos para sincronização de vídeo, de imagens e de dados, apresentando um ambiente virtual distribuído com coerência visual.

O sistema de gerenciamento se consolidou como uma aplicação gráfica intuitiva, que contém diversas ferramentas que são responsáveis pelo controle dos vários subsistemas do ambiente. Dentre os vários subsistemas destacamos: os projetores de imagens, o comutador de vídeo, o sistema de áudio, o sistema de iluminação, o sistema de rede. Todos podendo ser controlados de forma transparente.

Uma estação de acesso integra a eletrônica para o gerenciamento físico dos vários recursos do ambiente. Dentre eles destacamos o teclado e mouse principais, o painel de comutação de mouse e teclado, o painel de comutação de vídeo, a mesa de controle, o painel de controle de iluminação, além de 6 monitores.

O desempenho do sistema foi avaliado quanto à perspectiva de diversas aplicações, que serão transportadas para a arquitetura proposta. Algumas aplicações foram produzidas no âmbito deste projeto, em especial a plataforma de visualização JINX,

descrita no capítulo 6. Também foram feitos testes com aplicações de outras instituições.

Técnicas de documentação dos projetos implementados também foram utilizadas, como diagramas UML (*Unified Modeling Language*), bem como modelos e simulações 3D para a montagem dos sistemas imersivos, neste caso a CAVERNA Digital. Formatos atuais como XML e X3D são utilizados para gerenciamento e armazenamento de dados.

1.5 Metodologia de desenvolvimento

Esta seção busca a explicação, de forma sucinta, de algumas ferramentas que foram utilizadas durante o projeto. Os objetivos delas são variados e podem ser utilizadas para auxiliar a criação e desenvolvimento do projeto teórico, da documentação e análise de desempenho.

1.5.1 Engenharia de software baseada em *Unified Modeling Language* - UML

Devido a todos os desenvolvimentos serem orientados a objetos, os métodos e diagramas UML [UML] foram muito úteis na construção das aplicações, tornando-as robustas e de fácil compreensão, pois utiliza padrões que permitem outros desenvolvedores compreenderem o funcionamento do sistema.

Estes diagramas foram criados para o desenvolvimento e documentação das implementações realizadas e podem servir de base para futuros ajustes ou expansões. Os diagramas das aplicações produzidas se encontram nos ANEXO II – Diagrama UML do sistema SIRIUS e ANEXO III – Detalhes de implementação da JINX.

1.5.2 Organização de código fonte baseada na ferramenta *Concurrent Versions System* - CVS

Como forma de otimizar e gerenciar o desenvolvimento do código fonte, utilizou-se a ferramenta CVS [CVS], que faz o controle de armazenamento e colaboração de desenvolvimento de projetos. O controle de armazenamento é interessante para determinar os diversos estágios de desenvolvimento de um programa. Para

desenvolvimento colaborativo, diversos programadores podem compartilhar um mesmo arquivo, permitindo que possam trabalhar simultaneamente.

Basicamente o CVS possui uma árvore com alterações que foram executadas no decorrer do desenvolvimento e cada programador pode ter um arquivo em estado diferente de seus colegas, causando conflitos que devem ser corrigidos antes de atualizar os arquivos no servidor.

O CVS é uma plataforma totalmente aberta, usualmente chamado em linha de comando e acessando os dados via rede.

1.6 Trabalhos correlatos

Esta tese se insere no conjunto de projetos de pesquisa relacionados com realidade virtual, computação de alto desempenho e aplicações, que estão sendo desenvolvidos na área de Meios Eletrônicos Interativos no Laboratório de Sistemas Integráveis da Escola Politécnica da USP. Alguns desses projetos publicados como teses ou dissertações são:

- Visualização Volumétrica aplicada em Aglomerados de Computadores, por Paulo A. Bressan; [BRESSAN04]
- Um ambiente para o desenvolvimento de aplicações de Realidade Virtual baseadas em aglomerados gráficos, por Marcelo P. Guimarães; [GUIMARÃES04]
- Geração interativa de imagens por traçado de raios por Bruno B. Gnecco; [GNECCO05]
- Interação baseada em gestos de mãos e cabeça para ambientes de realidade virtual utilizando visão computacional por Marcio C. Cabral; [CABRAL05]

1.7 Estrutura da tese

Esta tese está estruturada em 8 capítulos a saber:

Capítulo 1 - Introdução: Uma breve introdução desta tese é apresentada, descrevendo a relevância, os objetivos e as metodologias adotadas durante esta pesquisa.

Capítulo 2 - O Estado da Arte em Aglomerados de Computadores para Realidade Virtual: Uma definição dos conceitos gerais utilizados nesta pesquisa é apresentada, bem como um panorama das pesquisas na área, posicionando o leitor em relação à proposta desta tese.

Capítulo 3 - Alternativas Tecnológicas para a implementação do VRCluster: Um conjunto de todas as tecnologias que foram estudadas é apresentado. As várias alternativas serão discutidas à luz de uma implementação viável de um aglomerado gráfico e um sistema de multiprojeções.

Capítulo 4 – CAVERNA Digital: Todas as decisões tomadas para a implementação da CAVERNA Digital e as dificuldades encontradas são discutidas.

Capítulo 5 – Operação e Gerenciamento de Ambientes de Realidade Virtual: São abordadas opções de organização interna, as ferramentas de gerenciamento e ambiente de programação. Apresenta também como a pesquisa foi conduzida e implementada, citando suas dificuldades de implementação e resultados obtidos.

Capítulo 6 - Proposta e Desenvolvimento de Uma Plataforma de Visualização Imersiva: A plataforma de visualização denominada JINX é apresentada. Esta é uma ferramenta que implementa os recursos oferecidos pela linguagem de descrição de cenas tridimensionais X3D (*eXtensible 3D*), viabilizando a visualização imersiva em sistemas de realidade virtual distribuídos.

Capítulo 7 - Validação Experimental dos Sistemas Desenvolvidos: Discussões são feitas sobre as opções tomadas e sua validade. Foram também realizadas comparações com outros sistemas similares.

Capítulo 8 - Conclusões e Trabalhos Futuros: Uma conclusão do trabalho executado é apresentada, verificando os pontos alcançados e o futuro da tecnologia.

Anexos : Documentos importantes para o entendimento da pesquisa desenvolvida, contendo os detalhes de implementação da CAVERNA Digital e diagramas das implementações efetuadas nesta pesquisa.

2. O ESTADO DA ARTE EM AGLOMERADOS DE COMPUTADORES PARA REALIDADE VIRTUAL

2.1 Introdução

O objetivo deste capítulo é apresentar o estado-da-arte da pesquisa e desenvolvimento em aglomerados de computadores para sistemas de realidade virtual e também definir e apresentar os conceitos utilizados ao longo desta tese.

Primeiramente serão apresentados os conceitos envolvendo a utilização atual de aglomerados de computadores, em diversos nichos como a computação numérica, transacional e gráfica. Posteriormente serão abordadas as técnicas de sincronização necessárias para aglomerados de computadores específicos para ambientes imersivos. Conclui-se o capítulo com uma avaliação das ferramentas mais atuais de desenvolvimento para sistemas de realidade virtual imersivos.

Conceitos de paralelismo serão abordados, como a taxonomia de divisão de tarefas para processamento gráfico e a taxonomia de formas de organização de aglomerados de realidade virtual.

2.2 Aglomerados de computadores convencionais

Os aglomerados de computadores convencionais [BERNAL99] vem sendo usados como alternativa para solucionar problemas numéricos de grande volume e complexidade. Em 1994 Donald Becker e Thomas Sterling iniciaram o projeto do primeiro aglomerado de computadores baseado em tecnologia de baixo custo [RESCHKE96], sendo executado no CESDIS (*Center of Excellence in Space Data and Information Sciences*), financiado indiretamente pela NASA. Este aglomerado foi chamado de Beowulf³ e foi um sucesso imediato. Atualmente muitos cientistas da área, chamam os aglomerados de computadores pessoais de Beowulf. O objetivo deste projeto era tratar problemas de grande quantidade de dados, utilizando tecnologia de baixo custo. O sistema utilizava uma rede de 10Mb/s. As primeiras aplicações do Beowulf foram sistemas de dinâmica de fluidos e análise gravitacional.

³ Beowulf é um guerreiro escandinavo do século sexto d.C., personagem de um poema [HEANEY01].

Um dos principais centros de referência para supercomputação é a lista publicada periodicamente dos 500 computadores mais rápidos do mundo, denominada de lista Top 500 [TOP500]. O objetivo da Top 500 é oferecer uma lista dos 500 mais poderosos computadores do planeta. Esta lista é disponibilizada pela Internet com diversas informações de cada sistema. Em 1998 um aglomerado GNU/Linux CPlant, desenvolvido no *Sandia National Labs*, estava na posição número 97. Já no ano de 2003, os aglomerados se apresentam em posição de destaque, como o sistema do *Lawrence Livermore National Laboratory*, que ocupa a posição de número 5. Ainda vale lembrar que os 4 primeiros sistemas, embora apresentassem desempenho superior, são sistemas que possuem custo muito superior ao dos aglomerados.

2.2.1 Aglomerados de computadores para cálculo numérico

A principal utilização dos aglomerados de computadores, atualmente, é para as aplicações de cálculo numérico, por estes realizarem cálculos matemáticos complexos rapidamente e apresentarem um custo muito inferior ao uso de super computadores comerciais.

O princípio destes sistemas é dividir equações numéricas complexas computacionalmente em diversos blocos computacionais independentes e enviar para cada nó de processamento do aglomerado, que faz o respectivo cálculo. Muitas vezes existe a necessidade de comunicação entre processos que estão executando os cálculos para troca de informação, assim, redes de dados velozes são muitas vezes fundamentais para o ganho de desempenho.

Um sistema de aglomerado de computadores que pode ser explorado para aplicações gráficas e de realidade virtual é o sistema tipo lâmina (*Blade*). Neste sistema o computador é reduzido ao tamanho de uma pequena placa e diversas destas placas são conectadas a um gabinete que compartilha recursos como fonte, entrada/saída de dados (*I/O*), levando a uma redução considerável do tamanho do sistema.

Figura 2-1 – Ilustração de aglomerado de computadores baseado em lâminas

Outra técnica que está sendo explorada é o conceito de Computação em Grade (*Grid Computing*). Neste conceito, os computadores do aglomerado estão localizados e disponíveis à grandes distâncias como apresentado na Figura 2-2, conectados por redes extremamente rápidas para a troca de informação, desta forma, cálculos de altíssima complexidade podem ser distribuídos remotamente. A realidade virtual pode explorar o conceito de grades computacionais quando uma simulação requer cálculos intensos em computadores não disponíveis localmente [COFFIN03].

Figura 2-2 – Ilustração de sistema de computação em grade

2.2.2 Aglomerados de computadores para aplicações transacionais

A demanda do mercado por aplicações transacionais é elevada, devido as necessidades que as empresas tem na organização de grandes bases de dados. Os bancos são um dos exemplos que exploram este tipo de recurso, pois necessitam realizar um grande volume de transações ao mesmo tempo, todas devendo ser confiáveis e rápidas.

Uma das formas de se preparar o ambiente é disponibilizar diversos nós para efetuarem um mesmo tipo de tarefa, como apresentado na Figura 2-3. A grande vantagem desta abordagem é que o sistema pode ser facilmente balanceado, removendo o gargalo que um computador monolítico teria. Além disso, a confiabilidade do sistema aumenta consideravelmente, pois, caso um nó apresente problema, os outros podem assumir as tarefas transacionais, que são facilmente realocáveis.

Figura 2-3 – Distribuição de tarefas em um aglomerado de computadores

2.2.3 Aglomerados de computadores para aplicações gráficas

Uma aplicação relevante de aglomerados de computadores para aplicações gráficas é a utilização de aglomerados numéricos para a síntese de imagens não interativas (*rendering farms*). Este tipo de utilização consiste na geração de imagens em alta resolução que consomem muita computação numérica. Assim, diversos nós de um aglomerado numérico são utilizados para gerar os quadros de uma animação.

Este sistema é muito utilizado para a produção de filmes, contudo não supre a principal necessidade de realidade virtual que é a geração das imagens em tempo real.

2.3 Sincronização em aglomerados de computadores para aplicações da realidade virtual

Uma das principais requisitos em sistemas distribuídos é a eficiência de sua sincronização dos computadores e subsistemas. É fundamental que a informação que se deseja compartilhar entre os diversos nós computacionais do sistema, seja

transferida de forma eficiente, assim mantendo a coerência global em sistemas dinâmicos.

Muitas pesquisas em sistemas de sincronização têm sido realizadas na área de realidade virtual [SINGHAL99], entretanto, elas estão focadas principalmente em sistemas colaborativos, nos quais computadores, com recursos individuais, comunicam-se uns com os outros para uma experiência virtual de um grupo de usuários, o que não é o foco desta pesquisa. Contudo, muitas dessas idéias já estudadas podem ser reaproveitadas em sistemas de aglomerados de computadores para a realidade virtual em tempo real, como por exemplo: a divisão de mundos virtuais em regiões, a preempção de dados e técnicas de organização de dados. Porém, há a necessidade de se adaptar e criar novos mecanismos de comunicação, pelo fato do sistema ser muito menos tolerante a atrasos de transmissão e falhas.

A **latência**, que é o tempo que uma mensagem demora para chegar de um ponto ao outro do sistema, pode variar muito e é um ponto importante que deve ser considerado, muitas vezes, como principal fator de decisão da tecnologia de rede a ser utilizada [CLUSTER01]. A Tabela I apresenta, de uma forma genérica, o atraso de transmissão de dados considerando várias tecnologias de interconexão de dados em supercomputadores, contudo existem outras soluções de interconexão, muitas vezes dedicadas, que também são apresentadas no decorrer deste trabalho.

Sistema	Latência
Shared Memory	3.4 μ s
Myrinet	8.3 μ s
Gigabit TCP/IP	137 μ s
Gigabit M-VIA	16 μ s
MPICH 1.1.2	142 μ s
MVICH v0.02	26 μ s
LAM v6.3	140 μ s

Tabela I – Latência de comunicação para várias tecnologias de interconexão

Das técnicas apresentadas, a de memória compartilhada é o método de transferência de dados mais eficiente em uso e também pode ser utilizada para sincronismos. Entretanto, a memória compartilhada considera a utilização de tecnologia proprietária, que normalmente tem um custo superior à soluções convencionais.

Outra forma é a sincronização por rede de computadores, que cria a necessidade de trocas de mensagens entre os computadores da rede, porém com um custo muito menor, pois podemos usar componentes convencionais nesta abordagem.

Nas próximas seções serão explicados os principais tipos de sincronismo em aglomerados gráficos de computadores.

2.3.1 O Sincronismo de vídeo - *Genlock*

Quando necessário a utilização de diversos dispositivos de vídeo, o ideal é que as imagens geradas sejam coerentes, ou seja, apresentem uma amostragem e iluminação (*chroma phase*) compatível, para isto é utilizado o recurso de *genlock* [GENLOCK03]. Mais importante ainda, este recurso será fundamental para sistemas estereoscópicos ativos, controlando o redesenho da tela. Caso não se deseje utilizar um sistema estereoscópico ativo, o recurso de *genlock* não é fundamental e pode vir a ser dispensado. Existem diversos estilos de estereoscopia, em especial nesta pesquisa, o interesse maior está nas tecnologias ativas e passivas. Para mais informações sobre estereoscopia consulte apêndice E. O sinal de *genlock* é usualmente tratado direto pelo cartão gráfico, contudo apenas poucos cartões gráficos oferecem tal recurso. Neste caso é necessária uma solução alternativa. Nesta pesquisa foi estudada e aplicada o projeto SoftGenlock [ALLARD02] que permite gerenciar parcialmente sinais de *genlock* entre cartões gráficos, mediante o uso da interface paralela, consultar apêndice F.

O *genlock* (*generator locking*) é um sinal compartilhado entre os sistemas de geração de vídeo, que é utilizado como referência para a taxa e momento das atualizações das imagens bem como cor e brilho. Um sistema é o mestre, que é usado como referência, porém podem ser utilizados dispositivos específicos para gerar este tipo de sinal. Este sistema permite então combinar imagens geradas por computador, com imagens de uma câmera. Caso o *genlock* não seja usado, efeitos de pulsação e erros de cores ocorrerão. Alguns sistemas ainda possuem um sinal único adicional para definir o nível de preto global (*true black burst*). Em sistema de vídeo composto como televisão, a correção de cor é baseada na fase da crominância.

Os monitores de tubo de raios catódicos utilizam um sistema de varredura de um feixe eletrônico, que faz com que as imagens sejam desenhadas em uma matriz na face frontal do tubo, ou seja a tela. A velocidade da varredura é controlada por dois sinais que definem os sincronismos vertical e horizontal. No caso de televisores, este sinal é gerado internamente, pois seguem padrões bem precisos de frequência, contudo para sistemas de computação há a necessidade de várias resoluções e níveis de cores, portanto estes sinais de varreduras, devem ser informados em conjunto com os sinais da imagem.

Os sistemas de estereoscopia ativa fazem a seleção das imagens, exibindo alternadamente as imagens de cada olho e filtrando-as pelos óculos. Esta comutação de imagens é feita no instante do redesenho no sistema de exibição, o que é um inconveniente, caso se esteja usando mais de um monitor ou projetor.

Figura 2-4 – Tubo de raios catódicos

Nesta situação a varredura dos diversos dispositivos do sistema devem ser sincronizada, para que o redesenho seja executado no mesmo instante em todos os dispositivos, permitindo que os óculos funcionem corretamente, independentemente de qual tela esteja se olhando. Esta sincronização é feita compartilhando o sinal de varredura, que é inserido no dispositivo gerador de imagens.

Na Figura 2-5 podemos verificar que os sinais de vídeo do aglomerado estão sincronizados, pois a foto capturada mostrou todos os monitores realizando o redesenho da imagens ao mesmo tempo. O monitor, marcado com um círculo, é de outro computador não conectado ao aglomerado de computadores, ou seja não sincronizado, mostrando que a varredura é realizada em instantes diferentes (não sincronizada) neste monitor.

Figura 2-5 – Sinais de vídeo do aglomerado

2.3.2 Sincronismo de quadros - *Frame-lock*

Para se obter um bom sistema de múltiplas projeções, é necessária que todas as imagens geradas apresentem uma coerência visual precisa, o suficiente para não ser percebida por um usuário comum. Esta coerência deve englobar as cores das diversas projeções, a calibragem geométrica das projeções e a continuidade das imagens apresentadas. É desejado que alterações na imagem, que ocorrem em uma projeção, ocorra ao mesmo tempo em todas as outras projeções. Por exemplo, na Figura 2-6 se quer que não haja descontinuidade no personagem virtual em movimento, que atravessa de uma imagem projetada para outra, para isto usamos o recurso de sincronismo de quadros também denominado de *frame-lock*. A ausência do recurso de *frame-lock* pode apresentar problemas de descontinuidade facilmente percebidos pelo usuário.

Figura 2-6 – Projeções sincronizadas (esquerda) e dessincronizadas (direita)

Em algumas aplicações comerciais utilizadas no setor de entretenimento, a mídia de armazenamento de imagens é feita em fita de vídeo ou filme de película, e quando se deseja apresentá-lo utiliza-se dois sistemas de projeção. Na exibição de um filme em estéreo passivo, existem dispositivos que permitem com que os dois filmes iniciem exatamente ao mesmo tempo, para haver coerência das imagens. Este mecanismo de sincronização é uma alternativa satisfatória, contudo somente apresenta resultados bons para curtas apresentações, pois, mesmo que os aparelhos de exibição sejam da mesma marca e modelo, um sistema mecânico vai apresentar sensíveis variações de tempo na apresentação de cada quadro, que conforme esta variação vai se acumulando, pode gerar uma falha de sincronização visível, assim prejudicando a qualidade da projeção. Neste caso, seria necessário algum mecanismo para, de tempos em tempos, verificar se os filmes se apresentam no mesmo ponto de exibição.

O problema de sincronização de quadros apresentado ocorre também em imagens sintetizadas por computador. Para evitar este problema, primeiramente é necessário recorrer à técnica chamada de **armazenamento duplo** (*double-buffering*), que permite que, enquanto um quadro esteja sendo processado em uma porção da memória de vídeo (*frame-buffer*), outro esteja sendo exibido utilizando outra porção da memória de vídeo. Em certo momento, um comando é executado para se trocar a posição da memória que será exibida e liberada a outra porção para se poder desenhar a próxima imagem. Enquanto o comando de troca de quadros não for executado, a imagem permanecerá a mesma.

Com a técnica de armazenamento duplo (*double-buffering*), enquanto uma região da memória de vídeo é exibida outra vai sendo gravada com a nova imagem, para uma posterior troca de papéis. Pode-se controlar o momento exato da troca de quadros de um dispositivo, que esteja gerando as imagens e desta forma, se as comutações de quadro dos diversos computadores que compõe o aglomerado, fizerem a troca de memória ao mesmo tempo, haverá uma consistência nas imagens.

Para se conseguir a sincronização de quadros, mensagens devem ser trocadas entre os diversos computadores. Um nó do aglomerado fará a troca de memória, quando receber uma mensagem, informando que todos estão prontos também.

O recurso de *frame-lock* pode ser confundido com o recurso de sincronismo dos dados que compõem a cena virtual, também denominado na literatura de *data-lock*, pois para o *frame-lock* funcionar corretamente, é necessário que as informações do cenário virtual, como posição geométrica, cores, dentre outras, sejam as mesmas em todos os nós do aglomerado gráfico.

2.3.3 Sincronismo de taxa de atualizações - *Rate-lock*

A frequência de atualização dos quadros pode ser variável, ou seja, o número de quadros por segundo (*frames per second*) não é constante devido a variações na complexidade da imagem gerada. Dentro de certos limites aceitáveis, não há necessidade de maiores controles, contudo, conforme as variações vão se tornando maiores, o sistema pode se tornar lento e inoperável, do ponto de vista do usuário, e trazer uma sensação desagradável para quem estiver operando. Assim, o recurso de *rate-lock* permite que as atualizações ocorram sempre em um período pré-estabelecido, criando um sistema de exibição de quadros contínuo.

2.3.4 Sincronismo de dados - *Data-lock*

O *data-lock* ou sincronismo de dados é responsável pela coerência dos elementos dinâmicos da cena, e para isto é importante que as informações em cada nó do aglomerado sejam idênticas em cada instante. Estas informações, normalmente, contêm a posição da câmera virtual, a direção que esta tem, movimentos da cena, ou qualquer outro atributo dinâmico da cena. Estes sincronismos são importantes não somente para a imagem, mas para qualquer outro tipo de retorno para o usuário, gerado pela simulação, como som ou mesmo algum tipo de retorno de força.

Desta forma, as informações devem ser transmitidas para cada nó, e para que isto ocorra, é necessário que estas informações trafeguem sobre algum meio de comunicação, usualmente uma rede de computadores.

2.4 Sistemas paralelos de geração de imagens

Um sistema de geração de imagens (*rendering*) paralela é necessário quando se deseja imagens com tal complexidade que somente um computador não é capaz gerar. Este problema pode ocorrer por limites de processamento, por exemplo, uma

complexidade poligonal da cena maior do que a capacidade do computador de gerar imagens em tempo real, ou limites físicos, como a necessidade de mais saídas de vídeo, que o cartão gráfico do computador disponibiliza. Assim, estes dados gráficos podem ser distribuídos e processados separadamente, posteriormente, compostos para sua visualização.

Dentre as vantagens do uso do paralelismo, temos a potencial escalabilidade de desempenho e números de componentes gráficas que compõem a cena virtual. Utilizando-se aglomerados gráficos para acelerar o processamento gráfico 3D, podemos simplesmente adicionar processadores ou estações, que teremos resultados mais complexos em menos tempo. Com este sistema é possível se obter uma saída visual com maior resolução, podendo-se chegar a ponto de se obter uma imagem foto-realista com cartões gráficos relativamente simples.

O processamento gráfico pode ser executado de várias formas, como por exemplo, no processador principal do computador, contudo, devido à grande complexidade e volume de cálculos para sistemas gráficos, estes processadores multiuso não conseguem tratar bem os problemas de computação gráfica, assim é necessário a utilização de processadores gráficos dedicados.

Desta forma, para uma melhor desempenho do sistema, o paralelismo gráfico deve ocorrer nestes processadores dedicados a cálculos de computação gráfica, que estão disponíveis nos cartões gráficos com recursos 3D. Porém, há uma série de inconvenientes nisso. A tecnologia atual inserção de placas gráficas para computadores pessoais utiliza o barramento AGP, que embora tenha uma excelente capacidade de transferência de dados, está limitada a apenas uma conexão por placa mãe. Para se efetuar o paralelismo deste sistema é necessário replicar todo o conjunto necessário para seu funcionamento, neste caso, a replicação de todos os elementos de um computador, o que leva aos aglomerados de computadores.

Existem algumas vantagens e desvantagens nesta proposta. Primeiramente o balanceamento de recursos fica melhor, pois teremos completa disponibilidade do computador. A desvantagem é o não aproveitamento completo dos recursos computacionais, ou seja muitos elementos são duplicados sem necessidade.

Durante a pesquisa foram efetuados alguns testes, visando avaliar a possibilidade de utilizar diversos cartões gráficos em um só computador, porém para isto foi necessário utilizar a tecnologia de conexão de placas PCI, ao invés da tecnologia de conexão de placas AGP. Um dos inconvenientes da PCI é possuir uma baixa taxa de transmissão de dados, quando comparada com a AGP, prejudicando o desempenho do sistema como um todo. Além disso, este sistema apresentou dificuldades de programação, pois, o computador não está preparado para possuir mais de um cartão gráfico ao mesmo tempo.

2.4.1 Taxonomia de paralelização em computação gráfica

Existem diversas formas de se paralelizar um sistema gráfico [CHALMERS02]. Os tipos de sistemas gráficos paralelos são tradicionalmente definidos pelo ponto que há uma transição do paralelismo na fila de processamento gráfico (*pipeline*). As três principais técnicas de paralelismo em computação gráfica são *sort-first*, *sort-middle*, *sort-last* [ELDRIDGE01]. A Figura 2-7 apresenta as fases de um *pipeline* gráfica padrão e a correlação do possível ponto de paralelismo e seu nome. Conforme a técnica explorada, um ou outro arranjo, deve ser feito para a maior facilidade de distribuição e seu balanceamento.

Figura 2-7 – Pontos de distribuição de dados para paralelismo gráfico

A técnica *sort-first* realiza a distribuição dos dados antes do processamento geométrico na fila de processamento gráfica. Neste caso, cada nó do sistema é responsável por processar uma parte da imagem final, para posteriormente, serem visualizadas uma ao lado da outra (*tiling*). As imagens geradas podem ser compostas em uma nova imagem completa, ou senão diretamente exibidas de cada nó do sistema, como é o caso da CAVERNA Digital. Conforme necessário pode haver uma sobreposição das imagens geradas.

A técnica *sort-middle* realiza o paralelismo entre o processamento geométrico e a rasterização (*rasterization*) da fila de processamento gráfico, para isso são distribuídos para os nós de processamento, os vértices das malha ou as primitivas geométricas inteiras. A técnica de *sort-middle* não pode ser utilizada em placas gráficas convencionais.

A técnica *sort-last* realiza a distribuição de pixels depois da rasterização (*rasterization*), mas antes da visualização. Este método permite uma subdivisão das tarefas muito refinada, o que assegura um melhor balanceamento de carga entre os nós. Na técnica de *sort-last* é necessária alguma forma de composição das imagens no final do processo da fila de processamento gráfico.

Figura 2-8 – Técnica *sort-first* e *sort-last*

Na Figura 2-8 podemos observar como as imagens são geradas para os casos citados. Para se utilizar estas técnicas de paralelismo podem ser utilizadas várias abordagens de implementação, em geral é necessário um conjunto de operações de *software* e *hardware*. Um dos problemas normalmente enfrentados é a composição final das imagens para a respectiva exibição. Devido a restrições da capacidade de transmissão de dados das redes de comunicação, são necessárias algumas técnicas para se transmitir as imagens pela rede considerando o atraso mínimo. Métodos baseados em compressão de dados e métodos que levam em conta a oclusão de objetos são normalmente eficientes [BUES03].

Um sistema proposto para implementar o método de *sort-last* direto por *hardware* é a Lightning-2 [STOLL01], este sistema possui um equipamento que recebe sinais DVI para a composição, e são informados tanto os canais de cores como os canais de profundidade das imagens. Já a o sistema proposto pela empresa ORAD™ [ORAD] possui uma placa dedicada em cada estação, chamada de DVG, que combina as imagens.

Figura 2-9 – Placas da Lightning2 e DVG

2.5 Sistemas centralizados de realidade virtual

O equipamento, até então, mais utilizados para ambiente de realidade virtual foram as estações Silicon Graphics™, isto se deve as facilidades de programação e processamento gráfico de alto desempenho que elas oferecem para o desenvolvimento de aplicações de realidade virtual avançadas, e a escalabilidade de desempenho, tanto no sistema de processamento numérico, quando no sistema de processamento gráfico.

O modelo de estação gráfica Onyx da Silicon Graphics™ consiste de um sistema internamente baseado em aglomerado, utilizando-se de memória compartilhada. Assim, todos os processadores acessam uma mesma memória virtual, o que facilita muito o desenvolvimento de programas, além da transferência de dados ser muito eficiente entre os nós de processamento.

O esquema de organização de processadores dos servidores Silicon Graphics™ é o Hyper-cubes [PACHECO97], o que em algumas situações poder ser muito vantajoso para a escalabilidade de desempenho numérico, e, por consequência, pode ser aproveitado no sistema gráfico, pois conseguem uma eficiente comunicação entre os processadores do sistemas, mesmo considerando uma grande quantidade de nós de processamento

⁴ <http://graphics.stanford.edu/papers/lightning2/>

⁵ <http://www.orad.tv/>

As estações gráficas Onyx, possuem um conjunto de placas especializadas, que juntas compõem a fila de processamento gráfico (*graphic pipeline*), sendo cada placa responsável por uma das etapas da síntese das imagens, aumentando assim o desempenho de geração de conteúdo para computação gráfica. Estas placas são a GE (*Graphics Engine*) responsável pelo cálculo das geometrias, a RM (*Raster Manager*) responsável pela aplicação das texturas e a DG (*Display Generator*) responsável pela geração do sinal de vídeo.

Figura 2-10 – Esquema de cartões gráficos de uma estação gráfica Onyx

2.6 Bibliotecas de desenvolvimento para realidade virtual

O ambiente de desenvolvimento de programas para sistemas de multiprojeção estereoscópicas é complexo, pois leva em conta diversos aspectos, como a representação de grandes estruturas de dados, o gerenciamento de vários periféricos e o mapeamento das imagens em vários projetores e sincronizações. No caso da CAVERNA Digital é necessário alimentar cinco faces de projeção. Além disso, devido ao tamanho da tela, deve-se utilizar resoluções suficientemente altas para que não se note a discretização das imagens.

A seguir serão apresentadas as principais bibliotecas para o desenvolvimento de aplicações para ambientes imersivos, identificadas na literatura, sendo que a maioria delas já estão caminhando para a plataforma de aglomerados de computadores convencionais.

⁶ Imagens obtidas em <http://techpubs.sgi.com>

O CAVELib [VRCO03] é uma biblioteca que permite ao usuário desenvolver aplicativos para realidade virtual com um amplo grau de abstração. Desenvolvido originalmente na Universidade de Illinois no EVL, é atualmente comercializado pela VRCO. Um dos sistemas mais utilizado no mundo. A memória compartilhada é a principal forma de troca de dados entre os processos. (Figura 2-11)

Figura 2-11 – Exemplo do CAVELib

O CAVERNsoft [PARK00] é um sistema de desenvolvimento em C++, para a produção de aplicações para sistemas cúbicos de projeção, sendo suportado em SGI, Solaris, Linux, FreeBSD e Windows. (Figura 2-12)

Figura 2-12 – Exemplo do CAVERNSoft

O OpenSG é um ambiente de programação para geração de imagens em tempo real, baseado em um grafo de cena (*scene graph*). Dentre de suas características está o suporte a paralelos, controle de múltiplas telas com estereoscopia.. A Figura 2-13 mostra uma parede de projetores (*projector wall*) composto de 20 imagens. Um dos principais recursos do OpenSG é fazer a distribuição da hierarquia por processos computacionais separados (*threads*), que já são automaticamente sincronizadas. (Figura 2-13)

Figura 2-13 – Exemplo do OpenSG

⁷ Imagem obtida em <http://www.vrco.com>

⁸ Imagens obtidas em http://www.openchannelsoftware.org/projects/CAVERNsoft_G2/

⁹ Imagem obtida em <http://www.opensg.org/>

O AVANGO [AVANGO03], tem sua origem em 1996, no GMD (German National Research Center for Information Technology). Ele é um sistema de desenvolvimento orientado a objetos, para aplicações de realidade virtual distribuídas. A distribuição dos dados é feita através de redundância e replicação, para os vários nós da aplicação. Sua programação é feita sobre a biblioteca Iris Performer [DAI97]. Atualmente o AVANGO suporta recursos de aglomerados de computadores [HINKENJAN02]. A Figura 2-14 mostra uma aplicação exemplo desenvolvida com o Avango. (Figura 2-14)

10

Figura 2-14 – Exemplo do AVANGO

O VRJuggler [VRJUGGLER03] é uma poderosa biblioteca de desenvolvimento para realidade virtual, sendo totalmente gratuita. Devido às suas características, é possível utilizar um mesmo aplicativo em diversos tipos de sistema, como CAVEs, Doms, Panoramas. Dentre as possíveis linguagens para funções de desenho temos o OpenGL, o Performer e até mesmo o OpenSG. (Figura 2-15)

11

Figura 2-15 – Exemplo do VRJuggler

A biblioteca gráfica Syzygy [SCHAEFFER03] desenvolvida no ISL na Universidade de Illinois em Urbana/Champagne é utilizada para acionar seu sistema cúbico de projeção de 6 lados da instituição, contudo é disponibilizada livremente pela Internet. O princípio desta biblioteca é inovador e eficiente, contudo o desenvolvimento de aplicações não é trivial. (Figura 2-16)

Figura 2-16 – Exemplo do Syzygy

¹⁰ Imagem obtida em <http://www.avango.org/>

¹¹ Imagem obtida em <http://www.vrjuggler.org/>

A Biblioteca DIVERSE (Device Independent Virtual Environments – Reconfigurable, Scalable, Extensible) [DIVERSE03] possui diversas ferramentas para desenvolvimento de aplicações de realidade virtual. Ele é encontrado gratuitamente na Internet e sua biblioteca base é o OpenGL Performer, o que limita este sistema às arquiteturas suportadas. A biblioteca permite que novas funcionalidades possam ser estendidas [KELSO02]. (Figura 2-17)

12

Figura 2-17 – Exemplo da DIVERSE

A Glass [GNECCO01] é uma biblioteca de sincronização para aplicações gráficas dirigidas a aglomerados de computadores. Sua principal utilização é em aplicações que necessitam múltiplas projeções, em especial a Caverna Digital, onde há a necessidade de coerência de diversos pontos de vista. A utilização da Glass é relativamente simples, o que facilita a adaptação de programas já existentes para o ambiente de aglomerado gráfico. (Figura 2-18)

Figura 2-18 – Exemplo da Glass

O Avalon [BEHR04] é um sistema para realidade virtual e realidade aumentada iniciado em 1997. Ele usa um grafo de cena internamente. O seu principal formato de operação atual é o X3D, onde foram implementadas diversas extensões para suportar os diferentes tipos de dispositivos de realidade virtual que não são padrões no formato X3D. (Figura 2-19)

13

Figura 2-19 – Exemplo do Avalon

¹² Imagem obtida em <http://www.diverse.vt.edu>

¹³ <http://www.zgdv.de/avalon/>

<p>O IC:IDO é um sistema comercial que apresenta uma estrutura muito robusta para a visualização e operação de conteúdos. Seu destaque esta para a forma de interação, que é muito rápida e eficiente. Contudo apresenta algumas dificuldade de instalação e operação. (Figura 2-20)</p>	<p>Figura 2-20 – Exemplo do IC:IDO</p>
<p>O DeltaGen explora largamente os últimos recursos das placas gráficas, como os <i>programmable shaders</i>, que permitem gerar imagens com um realismo muito elevado. Além disso este sistema possui rotinas que conseguem otimizar arquivos diversos para o formato interno com um ganho de desempenho. (Figura 2-21)</p>	<p>Figura 2-21 – Exemplo do DeltaGen</p>

2.7 Conclusões

Este capítulo apresentou diversas tecnologias estado-da-arte para aplicação direta e indiretamente na realidade virtual imersiva. Verificamos que o uso crescente de aglomerados de computadores em realidade virtual é irreversível.

O atual estágio das tecnologias de sincronismo e paralelismo gráfico, não apresentam nenhum limitante que inviabilizaria seu uso em aglomerados de computadores.

Baseado nas informações apresentadas, foi possível fazer as decisões de projeto para viabilizar técnicas e ferramentas da implementação desta pesquisa. O que foi apresentado pode ser aproveitado na pesquisa diretamente, ajudando a estabelecer um embasamento em todas as áreas pertinentes desta tese. Em especial as sincronizações e paralelismos serão reutilizados no capítulo sobre a ferramenta de visualização desenvolvida.

Como foi apresentado, existem diversos pontos a serem trabalhados em um aglomerado gráfico, além de diversas ferramentas que já existem e podem ser

¹⁴ <http://www.icido.de/>

¹⁵ <http://www.realtime-technology.com/>

utilizadas em plataformas monolíticas e em alguns casos em aglomerados. Um dos principais problemas é a preparação dos aglomerados para que os mesmos funcionem corretamente, além do seu gerenciamento. Isto será discutido nos próximos capítulos. Um ponto importante a ser mencionado é que as ferramentas de visualização aqui apresentadas basicamente não existiam no início deste projeto, ou então não suportavam aglomerados de computadores.

A Tabela II apresenta um comparativo dos sistemas imersivos de realidade virtual. A partir desta análise, podemos constatar um incremento de oferta de ambientes de programação para realidade virtual com suporte à aglomerados de realidade virtual, o que não existia a alguns anos atrás.

	Aglomerados	Gratuita	Linux/Windows/SGI
CaveLib	Não	Não	Linux/Windows/SGI
CavernSoft	Não	Sim	Linux/Windows/SGI
OpenSG	Sim	Sim	Linux/Windows/SGI
Avango	Sim	Não	Linux/Windows/SGI
VRJuggler	Não*	Sim	Linux/Windows/SGI
Syzygy	Sim	Sim	Linux/Windows/SGI
Diverse	Sim	Sim	Linux/Windows/SGI
Glass	Sim	Sim	Linux/Windows/SGI
Avalon	Sim	Não	Linux/Windows/SGI
IC:IDO	Sim	Não	Linux
DeltaGen	Sim	Não	Windows

Tabela II – Comparativo de sistemas imersivos

* embora o VRJuggler não suporte aglomerados suas variações NetJuggler e ClusterJuggler suportam

3. ALTERNATIVAS TECNOLÓGICAS PARA A IMPLEMENTAÇÃO DO VRCLUSTER

3.1 Introdução

Este capítulo apresenta o elenco de tecnologias para a construção de aglomerados computadores de realidade virtual (VRCluster). Existem várias formas de integração e alternativas tecnológicas para a construção dos VRClusters, as opções tecnológicas e de integração, devem levar em conta o resultado que se pretende em termos do desempenho da aplicação final e os recursos disponíveis. O capítulo apresenta uma contribuição que é uma proposta de taxonomia de organização de VRClusters.

3.2 O sistema operacional

Para o desenvolvimento do projeto estão sendo utilizados os sistemas operacionais Linux, Windows e IRIX. O que terá maior foco de utilização nesta tese será a plataforma Linux, por a ser um sistema operacional baseado no modelo de software livre, mais adequada para a implementação de sistemas de baixo custo.

- **Linux:** Este foi o sistema operacional mais utilizada ao longo do desenvolvimento do projeto. Isto se deve ao fato do Linux apresentar um excelente controle dos seus recursos, o que facilita muito o desenvolvimento de aplicações e também por ser um sistema robusto. Além de não ter custo direto, existem diversos aplicativos e bibliotecas de desenvolvimento para esta plataforma que também são gratuitos. Também existe uma maior praticidade, pois não é necessário lidar com peculiaridades e restrições do sistema operacional no programa a ser desenvolvido. Um exemplo é a possibilidade de configurar resoluções de vídeo com total liberdade.
- **Microsoft Windows:** Esta família de sistemas operacionais proprietários apresentam dentre suas principais vantagens, a qualidade dos controladores de dispositivos (*drivers*), para os periféricos que compõem o aglomerado gráfico. Por exemplo, os cartões gráficos possuem normalmente um melhor desempenho no sistema operacional Microsoft Windows, do que no Linux. Contudo normalmente os sistemas proprietários não possuem a flexibilidade suficiente

para a configuração das necessidades especiais das aplicações requeridas pelo VRCluster, muitas vezes inviabilizando a aplicação final. Além disso as rotinas e o próprio código dos sistemas proprietários são totalmente fechados, não permitindo a total compreensão do sistema nem permitindo sua alteração.

- IRIX: Devido ao grande parque de equipamentos da empresa Silicon Graphics™ instalados no mundo para o uso de realidade virtual, é natural a realização de estudos e desenvolvimentos utilizando-se esta tecnologia de sistema operacional proprietário. Assim, os sistemas desenvolvidos para aglomerados de computadores também poderão operar neste sistema operacional. Devido aos recursos de realidade virtual oferecidos nas estações gráficas Silicon Graphics™, que permitem uma integração mais rápida de aplicações finais, os aplicativos podem tirar proveito de vários recursos oferecidos pelo sistema operacional IRIX como por exemplo o sistema de memória compartilhada.

3.3 Sistemas de transmissão de dados e sincronismo

Para o correto funcionamento do aglomerado de computadores, um sistema de comunicação eficiente é de fundamental importância para o desempenho do conjunto. Devido aos quesitos de desempenho gráfico, a latência de comunicação aceitável é normalmente inferior a 100ms para que o usuário não perceba visualmente este atraso. Em especial sistemas que utilizem alguma técnica de rastreamento de cabeça, a latência tolerável cai para 10ms [AKELEY01]. Além disso é ideal a utilização de protocolos de comunicação baseados em padrões abertos, pois assim se pode tirar proveito da eletrônica disponível reduzindo os custos de desenvolvimento. Finalmente, caso os equipamentos padrões não sejam adequados, pode haver a necessidade de desenvolver alguma eletrônica de comunicação, como foi o caso do sistema de *genlock* (atualmente disponível em algumas placas gráficas).

Para uma discussão sobre os diversos protocolos e ferramentas de transferência de dados, muitos deles usados no desenvolvimento da pesquisa direta ou indiretamente ou trabalhando em conjunto, explorando suas vantagens e desvantagens, para verificar suas capacidade e adequação ao sistema em desenvolvimento, consultar apêndice D e apêndice C respectivamente.

3.4 Protocolos de memória compartilhada

Os tradicionais supercomputadores utilizam sistemas de memória compartilhada, permitindo que diversos processadores acessem uma memória única, resultando em um melhor desempenho para as aplicações. Contudo este recurso já é encontrado também em computadores tradicionais, que dispõem da tecnologia SMP (*Symmetrical MultiProcessor*). A forma mais utilizada para explorar este recurso é por meio da programação utilizando-se *threads* [LEWIS95], que basicamente são uma sequência de instruções independentes que podem ser processadas paralelamente.

3.4.1 Interface de programação PThreads

O sistema de programação baseado em *threads* mais utilizado é o Pthreads, que é definido pelo padrão IEEE POSIX 1003.1c. Existem várias distribuições do Pthreads, muitas vezes acompanhando o sistema de cada computador. No esquema de programação utilizando-se *threads* uma aplicação pode disparar diferentes sequências de código em paralelo e todas acessarem a mesma base de dados, reduzindo assim a replicação de dados. Para poder utilizar as *threads*, a biblioteca de desenvolvimento disponibiliza ferramentas para gerenciamento, como as de exclusão mútua e variáveis condicionais. A biblioteca Pthreads foi desenvolvida originalmente para a linguagem C, assim ela necessita ajustes para orientação a objetos.

3.4.2 Interface de programação OpenMP

O sistema OpenMP [OPENMP03] foi concebido para tirar proveito dos sistemas que possuem diversos processadores. Algumas rotinas, como laços (*loops*), podem tirar grande proveito disso. Este sistema pode aumentar consideravelmente a velocidade de algumas aplicações, contudo embora o OpenMP, tenha Open no seu nome, atualmente somente compiladores comerciais suportam o OpenMP. Este foi um dos motivos de ter menos enfoque na pesquisa, pois eventualmente teríamos uma dependência comercial nesta opção tecnológica.

3.5 Protocolos de gerenciamento de dados

Protocolos para organizar e gerenciar os dados são fundamentais para sistemas distribuídos. Nesta pesquisa decidiu-se utilizar o protocolo XML (*eXtensible Markup*

Language), que embora novo já é largamente adotado e bem aceito. Um dos motivos de seu sucesso é superar as dificuldades da comunicação em rede é a transferência de dados entre plataformas diferentes. O protocolo XML foi criado com o propósito de fornecer um protocolo simples e eficiente, permitindo que qualquer tipo de computador seja capaz de interpretá-lo [MCLAUGHLIN01], pois basicamente se apresenta no formato texto, permitindo uma rápida compreensão. Além disso, o XML pode ser estendido, ou seja, novas formas de armazenar as informações podem ser criadas. Também existem mecanismos de validação e verificação de conteúdos XML, que pode ser executado pelo sistema do DTD (*Document Type Definition*) ou *Schema*, que é mais atual.

Por ser um padrão aberto, diversas organizações estão desenvolvendo e disponibilizando leitores (*parsers*) de XML, algumas gratuitamente. Os primeiros leitores a serem desenvolvidos, e ainda os mais utilizados, são o SAX (*Simple API for XML*) e o DOM (*Document Object Model*), porém hoje existem diversos leitores e durante o desenvolvimento deste trabalho um leitor foi desenvolvido, porém não utilizado devido a necessidades de melhor suporte aos recursos do XML.

3.6 Bibliotecas de desenvolvimento para CG/RV

Para a produção de imagens em computação gráfica são necessárias diversas instruções. Nos primórdios da computação gráfica, estas instruções eram somente vetoriais, devido ao tipo de monitor disponível, que também era vetorial. Com o advento do monitor de rastreamento (*raster*), no qual um feixe eletrônico faz uma varredura constante e uniforme em toda a área da tela, novas possibilidades surgiram, como a visualização de sólidos e a utilização de texturas. Para maiores detalhes sobre as ferramentas e bibliotecas para computação gráfica consulte apêndice A.

3.7 Formatos de descrição de cena

Existem diversos formatos para armazenamento de arquivos de dados 3D. Contudo usualmente são proprietários ou fechados. O VRML (*Virtual Reality Modeling Language*) e seu sucessor, o X3D, são abertos, e disponibilizam diversos recursos poderosos e compreensíveis.

3.7.1 Formato de descrição de cena VRML

O VRML [MATSUBA96] foi concebido com o propósito de descrever uma cena 3D de forma simples e eficiente para que pudesse ser disponibilizado na Internet, através de uma linguagem independente de plataforma. O VRML se baseou originalmente no OpenInventor, e sua primeira apresentação foi em 1994, no *First International Conference on the World Wide Web*. A forma mais tradicional de se acessar um conteúdo VRML é através de um navegador de páginas da Internet, que suporte este formato. Normalmente, basta instalar um aplicativo complementar (*plug-in*), para viabilizar esta visualização e navegação.

3.7.2 Formato de descrição de cena X3D

O X3D [X3D03] [X3DFAQ04] foi um padrão proposto em Agosto de 2001 pelo Web3D Consortium, como um padrão aberto para suceder e expandir as capacidades do VRML97, adicionando novos recursos e utilizando o XML (*Extensible Markup Language*) como principal forma de codificação.

16

Figura 3-1 – Logo X3D

O X3D tem o objetivo de suportar diversos tipos de sistemas de realidade virtual como CAVEs, luvas, panoramas, *mouses* 3D, HMDs (*Head Mounted Displays*), bem como computadores domésticos. O X3D tenta cobrir as áreas mais importantes para realidade virtual, em especial na Internet, como gráficos 3D e 2D, animações diversas, espacialização de som, interatividade e programabilidade. Por a ser uma modelagem textual e programável, pode ser facilmente criada e alterada, criando bons efeitos gráficos, com um grande grau de refinamento.

3.8 Sistemas de vídeo

Os sinais de vídeo podem ser codificados e transferidos de várias formas, o formato de transmissão de sinal de vídeo mais utilizado para informática é o RGB. Embora apresente problemas, como o de não cobrir completamente o espectro visível, possui uma especificação muito simples, tornando seu uso muito interessante. Porém este

¹⁶ Imagem obtida em: <http://www.web3d.org/>

sistema, como os demais analógicos, é influenciado pela qualidade do cabeamento utilizado, sendo susceptível a ruído e a diferenças de comprimento de cabo, que podem interferir no posicionamento da imagem. Já com o novo padrão DVI (*Digital Video Interface*) é possível enviar sinais de vídeo em modo analógico e digital. Contudo um dos inconvenientes com o DVI é o comprimento máximo do cabo, que não deve ultrapassar 2 metros. Maiores informações sobre o padrão DVI encontram-se no apêndice A.

Figura 3-2 – Conectores VGA e DVI

Já os sistemas de multiprojeção apresentam outros pontos relevantes em relação aos sistemas com uma só projeção. Primeiramente é necessário que todas as projeções estejam devidamente sincronizadas e calibradas (geometria, cor, etc..). Algumas vezes as distâncias dos cabos começam a se tornar muito elevadas, neste caso as fibras ópticas são uma forma viável de transferência de dados, que podem cobrir distâncias de centenas de quilômetros, sem a necessidade de nenhum outro equipamento.

3.8.1 Cartões gráficos

A oferta de cartões gráficos cresceu rapidamente nos últimos tempos, principalmente devido a indústria de jogos, que baseou este tipo de produto nas plataformas de computadores pessoais. Isto resultou em um rápido crescimento deste nicho, e permitiu que novas tecnologias surgissem rapidamente, mais velozes, eficientes e baratas.

O padrão de barramento AGP que se estabeleceu por muito tempo como principal padrão de barramento para os cartões gráficos, chegou em sua versão 8X e agora está sendo substituído pelo padrão de barramento PCI-Express. A largura de banda é maior, que o padrão AGP, além de menor latência. Para mais informações sobre o padrão AGP, consultar o apêndice H. Sobre o padrão PCI-Express consultar o apêndice M.

3.8.2 Comutador de rede

Os comutadores de rede (*switches*) são equipamentos que aceitam uma série de conexões com diversos computadores, e criam rotas inteligentes para a comunicação entre eles. O número de portas e velocidades é bem variado, possibilitando inúmeras configurações flexíveis de aglomerados.

Figura 3-3 – Comutador de Rede

3.9 Taxonomia do VRCluster

Existem diversas formas de se organizar um aglomerado de computadores [RAFFIN03], em especial para o VRCluster foi desenvolvida uma taxonomia que serve de referência para a implementação de um sistema deste tipo. Esta taxonomia leva em conta os tipos de nós computacionais a serem utilizados, que podem ser nós para operações puramente numéricas, nós para síntese de imagens e nós para entrada e saída de dados. Outras variáveis que podem ser utilizadas para a definição do VRCluster são seu custo, desempenho, recursos de gerenciamento e facilidade de atualização.

O VRCluster pode ser simétrico ou assimétrico. No caso dele ser simétrico todos os nós do mesmo tipo no aglomerado mapeiam seus dados da mesma forma, ou seja, possuem as mesmas conexões, já aglomerados assimétricos não possuem uma conexão uniforme para a transferência de seus dados. Estes dados usualmente são redes, dispositivos de realidade virtual bem como teclado e mouse, e os sinais de vídeo.

Uma outra taxonomia é em relação ao aglomerado possuir seus computador centralizados em um ponto ou distribuídos em um ambiente. O aglomerado que se encontra confinado em um bastidor (*rack*) é considerado centralizado, já um aglomerado que fique fisicamente espalhado é considerado distribuído.

Outra das formas de se especificar que tipo de aglomerado se deseja implementar é se este vai ser heterogêneo ou homogêneo. Os aglomerados homogêneos são compostos de computadores idênticos. Neste caso, a instalação e manutenção fica

mais fácil, porém pode apresentar um custo superior. Já os aglomerado heterogêneos são usualmente montados para reaproveitar computadores antigos, desta forma a economia é ainda maior. Em termos de operação os dois sistemas apresentam basicamente a mesma estrutura, contudo no sistema heterogêneo poderá haver problemas de balanceamento, comprometendo o desempenho do sistema.

A Figura 3-4 apresenta um esquema exemplo de organização, onde os computadores estão distribuídos em um ambiente além deles não apresentarem a mesma configuração. Este esquema é muito interessante para situações onde os computadores já existam e se deseja reaproveitar o sistema para ambientes complexos de realidade virtual. Contudo existem as dificuldades de manter este sistema, pois os computadores são diferentes criando a necessidade de uma manutenção específica para cada um, além do acesso físico que é mais trabalhoso.

Figura 3-4 – VRCluster assimétrico distribuído heterogêneo

Já o exemplo da Figura 3-5 mostra uma configuração de aglomerado mais robusta, pois os computadores estão no mesmo espaço além de apresentarem uma mesma configuração, o que facilita muito seu gerenciamento e manutenção, embora apresente um custo inicial maior.

3.10 Conclusões

Neste capítulo foram descritas diversas tecnologias usadas durante a pesquisa e desenvolvimento do aglomerado de computadores para aplicações de realidade virtual, servindo de referência para os capítulos seguintes. As tecnologias apresentadas são os pilares fundamentais para o desenvolvimento do VRCluster.

Existem várias formas de integração de um VRCluster, e estas formas variam em função do resultado final que se deseja, resultado normalmente limitado pelo desempenho da aplicação final, recursos disponíveis e características específicas do ambiente de realidade virtual como número de telas, periféricos utilizados e natureza dos aplicativos.

A adoção de ambientes de desenvolvimento abertos como por exemplo a escolha do sistema operacional Linux, se enquadra muito bem com a utilização de ferramentas de paralelismo, neste caso as Pthreads e o OpenMP. O ambiente de programação OpenGL que se apresenta muito flexível consegue um mapeamento direto de suas funções com o padrão X3D, que por consequência é uma implementação das extensões 3D em XML.

Uma contribuição deste capítulo é taxonomia de organização de VRClusters que se mostra muito útil para definir as várias possibilidades de integração de VRClusters, esta taxonomia foi apresentada em vários cursos apresentados em congressos internacionais e nacionais da área [ZUFFO02][RAFFIN03][SOARES02b] e já se encontra em uso pela comunidade.

4. CAVERNA DIGITAL

Este capítulo descreve em detalhes o projeto e implementação de um ambiente de realidade virtual baseado em multiprojeções totalmente imersivo denominado de CAVERNA Digital.

4.1 Introdução

O sistema de projeção da CAVERNA Digital[ZUFFO01] é um sistema de realidade virtual, que consiste de uma sala, na qual cada parede é uma tela de projeção. O observador se encontra dentro do ambiente tendo um grande grau de imersão, não sendo intrusivo, ou seja, não existem muitos equipamentos para serem carregados.

O primeiro sistema cúbico de projeção (CAVE™) no mundo foi desenvolvido pelo *Electronic Visualization Laboratory* (EVL) da Universidade de Illinois em Chicago pela pesquisadora Carolina Cruz Neira [CRUZ93]. Sua primeira proposta foi a colocação de monitores de vídeo ao redor da cabeça do usuário, mas logo se percebeu que telas e projetores seriam bem mais práticos e eficientes.

O sistema original [CRUZ92] usava resoluções de 1280x512 a 120HZ , e os tubos dos projetores não usavam fósforo P43 (verde) o que criava uma sombra na imagem devido ao longo tempo de comutação entre elas. As dimensões da CAVE™ do EVL eram de 7'x7'x7' com sistema sonoro. Constatou-se que poucas pessoas passavam mal nesta equipamento.

17

Figura 4-1 – O primeiro CAVE™

Os sistemas cúbicos apresentam uma ótima qualidade de imersão em relação a outras formas de sistema de realidade virtual como, por exemplo, os capacetes de realidade virtual HMD (*Head Mounted Display*) [SUTHERLAND68], que são óculos pesados e apresentam sérios problemas de rastreamento, navegabilidade e campo de visão

¹⁷ Imagem obtida em: <http://resumbrae.com/ub/dms424/03/05.html>

podendo induzir ao mal estar no usuário. Na CAVERNA Digital os recursos incorporados ao usuário para a imersão são geralmente mínimos, baseando-se apenas em óculos para suportar a estereoscopia e dispositivos de rastreamento. Além disso, em sistemas HMD, o corpo do usuário deve ser reproduzido, já na CAVERNA Digital não, pois ele existe fisicamente.

Um outro exemplo é o monitor convencional que cobre menos que 45% da capacidade dos gânglios celulares da retina [TAYLOR03], já uma CAVERNA Digital tem uma cobertura visual muito maior.

Uma grande vantagem da CAVERNA Digital é sua flexibilidade, pois possibilita o compartilhamento do mundo virtual por vários usuários simultaneamente. A quantidade de pessoas está limitada diretamente às dimensões físicas da CAVERNA Digital. É possível também o recálculo do ponto de vista do usuário, baseado em um sistema de rastreamento do ponto de visão de um usuário, e assim produzir o efeito perfeito de imersão e navegabilidade, porém para somente um usuário por vez.

Dentre as possíveis aplicações da CAVERNA Digital, podemos destacar: a visualização de lençóis petrolíferos para facilitar as perfurações; as maquetes digitais na indústria automobilística para solucionar problemas de design e aerodinâmica; a análise meteorológica, como a visualização de massas de ar; simulações cirúrgicas na medicina; planetário virtual; dentre muitas outras.

4.1.1 Sistema semi-esférico

Existem variações dos sistemas imersivos, como os **sistemas de projeção semi-esféricos** [ELUMENS03][GREENE86], que ao invés de possuírem uma tela plana, possuem uma tela esférica e a projeção é feita por um projetor disposto entre o usuário e a tela. Este sistema apresenta as vantagens de não possuir descontinuidades na imagem, porém impede que o usuário se aproxime da tela.

Figura 4-2 –Projeção esférica

Algumas adaptações, que foram executadas neste sistema, permitiram criar um sistema de colaboração portátil muito eficiente. Uma câmera foi implantada na parte superior da semi-esfera. Embora apresente uma grande inclinação, esta funciona perfeitamente para capturar imagens e reenviá-las para outro ambiente.

4.2 Aspectos arquitetônicos e estruturais

A CAVERNA Digital proposta é composta por um cubo de dimensão 3m x 3m x 3m com projeção nas 4 paredes laterais mais o piso. O sistema pode comportar até 6 usuários no seu interior, compartilhando a mesma experiência virtual [DISZ97].

No leiaute arquitetônico da CAVERNA Digital, Figura 4-3, um dos aspectos levados em conta foi embutir totalmente os projetores e outros dispositivos, escondendo do usuário final as particularidades técnicas do sistema.

Figura 4-3 – Leiaute da CAVERNA Digital e pilares e vigas de madeira

Do ponto de vista estrutural deve-se levar em conta também aspectos como compatibilidade eletromagnética, iluminação, isolamento acústico, controle de temperatura e umidade, e, finalmente, a possibilidade de montagem de periféricos adicionais no interior da CAVERNA Digital.

A estrutura do cubo é toda feita em madeira e plástico, envolvendo o uso de metal apenas em alguns parafusos estruturais e projetores, buscando minimizar a distorção dos campos magnéticos gerados pelos rastreadores eletromagnéticos de posição. Na Figura 4-3 os pilares e vigas do prédio próximos as telas de projeção foram construídos de madeira, pois estes, diferentes dos outros de metal, não criam interferência no sistema eletromagnético.

A preparação do usuário para uma experiência virtual imersiva completa começa antes mesmo do usuário entrar no interior do cubo. Um dos cuidados é a adaptação do sistema visual humano à intensidade de luz projetada nas faces, conforme Figura 4-4. Dentre os vários cuidados necessários, temos a eliminação do ofuscamento, através do uso de difusores de luz e a utilização de lâmpadas com alta resposta de cor. A flexibilidade de iluminação é também importante, demandando a utilização de sistemas de iluminação heterogêneos, baseados em lâmpadas alógenas e lâmpadas fluorescentes, ambos com controle automatizado de intensidade.

Figura 4-4 – Acomodação visual do usuário

A escolha das faces da CAVERNA Digital, onde há projeção, foi baseada no potencial de uso de aplicações relevantes para ciências gerais, engenharias e indústria, nas quais há a necessidade de se visualizar objetos, e o maior interesse é a visualização destes por cima, e ambientes para passeios virtuais, onde a referência do piso é importante para a sensação de imersão do usuário.

Para detalhes dos diversos componentes da CAVERNA Digital verificar ANEXO I – Componentes da CAVERNA Digital

4.2.1 A CAVERNA de baixo custo

A CAVERNA Digital, atualmente implementada, utilizou tecnologias topo de linha, porém, infelizmente, estas apresentam um custo ainda muito elevado. Foi desenvolvida uma variante: o mesmo sistema com custo reduzido. Para isto, foram utilizados projetores mais simples e de menor dimensões, apresentando custos inferiores. O sistema de controle também utiliza *joysticks* de jogos de computadores, que são práticos, rápidos o suficiente e baratos.

4.2.2 O aglomerado de realidade virtual

Os tradicionais sistemas de realidade virtual são baseados em supercomputadores gráficos. A alternativa proposta baseia-se em aglomerados de computadores

convencionais. Uma das vantagens dos aglomerados gráficos é sua escalabilidade. Os supercomputadores gráficos, muitas vezes apresentam dificuldades de expansão, sendo necessária uma troca do equipamento por um mais moderno. Este tipo de escalabilidade é conhecido como **vertical**. Já para ambientes de aglomerados de computadores, os antigos sistemas podem ser totalmente mantidos, apenas sendo necessária a instalação de mais equipamentos, novos ou não, que trarão maior desempenho para as aplicações. Este tipo de escalabilidade é conhecido como **horizontal**. Um dos grandes inconvenientes da escalabilidade vertical é a interrupção causada quando é necessário uma atualização do sistema. Como o sistema deve ser trocado, é necessário desligar-se um para instalar o outro, tornando o sistema indisponível durante esta troca. Contudo sistemas de escalabilidade horizontal apresentam maiores dificuldades de manutenção devido ao sistema não ser unificado.

4.2.2.1 Sistemas desenvolvidos

O primeiro aglomerado desenvolvido e utilizado é composto por 6 computadores Pentium III. A escolha de 6 nós é devido à complexidade do sistema utilizado, que é uma CAVERNA Digital de 5 lados. Desta forma, há necessidade de pelo menos 5 nós para sinais de vídeo, e um nó para agir como servidor, executando tarefas como leitura de dispositivos e gerenciamento da base de dados, além de um KVM PS2 e um comutador de rede [SOARES02a].

Figura 4-5 – Primeiro e segundo aglomerado

O segundo aglomerado é mais bem definido e balanceado. Este aglomerado é baseado em 6 computadores Dual-Xeon que são principalmente responsáveis por imagem e áudio, uma estação Pentium IV que serve como nó de acesso, uma estação Itanium de 64 bits responsável pelas simulações numéricas, além de um KVM USB, uma multiseriál, amplificador de som e comutadores de rede e vídeo.

Em muitos momentos foi utilizada uma estação Silicon Graphics™ Onyx 3000 para testes de compatibilidade e desempenho.

4.2.2.2 Sistema de interconexão

O sistema de interconexão físico é baseado no padrão de rede Gigabit-ethernet, mediante portas UTP e ópticas. A escolha do sistema de interconexão deve-se ao fato da excelente largura de banda disponibilizado pelo sistema GigaBit, compatibilidade com o sistema ethernet e sua fácil aquisição, visto que o sistema GigaBit já é facilmente encontrado no mercado. Um outro fato que corrobora a escolha do GigaBit é sua possível utilização com o protocolo VIA que permite uma maior eficiência da transmissão de dados na rede, sem corromper com a compatibilidade. O padrão de barramento escolhido foi o PCI 64 bits a 66MHz, que é fundamental para as placas de rede mais evoluídas. O PCI-X não foi escolhido por apresentar uma latência muito elevada, prejudicando o balanceamento do sistema.

Como forma de disponibilizar outra meio de acesso, o sistema de conexão por porta serial padrão mostra-se muito eficiente. Este pode ser utilizado para uma comunicação que consuma menos largura de banda, como gerenciamento ou sincronismos. Contudo sua maior importância é ser um acesso alternativo, caso o sistema de conexão principal esteja indisponível ou com problemas. A nomenclatura destes sistemas é: ***In-Band Networking Management*** para o gerenciamento por rede tradicional e ***Out-of-Band Networking Management*** para sistemas alternativos, como o serial.

A utilização do sistema serial é feita por intermédio da comunicação de dois computadores, ou seja, ponto a ponto. Como esquema de conexão, pode haver várias formas de organização: a primeira, de mais baixo custo, seria conectar cada nó do aglomerado em outro computador diretamente, podendo-se concentrar todos os computadores em um nó principal. Muitas vezes é necessária a aquisição de placas adicionais para esta tarefa. Também podem ser feitos sistemas de corrente (*chain*), onde cada computador se comunica com o próximo. Esta forma é mais simples, pois muitos computadores possuem duas conexões seriais. Uma terceira opção para as conexões seriais é a utilização de um sistema dedicado, capaz de suportar diversas conexões seriais. Para o protótipo foi utilizado o sistema *Terminal Server* da Cyclades™ [CYCLADES01]. Este sistema é um computador com dimensões e recursos reduzidos com até 42 portas, podendo, desta forma, acessar diversos

dispositivos ao mesmo tempo. Esta opção tem vantagens de sua alta densidade de conexões não necessitando a instalação de placas adicionais. A Figura 4-6 apresenta um diagrama para os sistemas apresentados.

Figura 4-6 – Sistemas de comunicação

Uma alternativa para a comunicação de nós em um aglomerado de computadores, é a utilização da tecnologia sem fio. Contudo o padrão atual de mercado, o IEEE 802.11, apresenta velocidades muito baixas, inviabilizando seu uso para tarefas que necessitem grandes transferências de dados, porém suficiente para situações de gerenciamento. O padrão IEEE 802.11e, permite taxas de transmissão melhores, usando uma frequência de transmissão de 5GHz, maior que o padrão antigo, que é de 2.4GHz, permitindo transferências a taxas de até 54 Mbps e controle de Qualidade de Serviço, que já permite transmissão de informações como vídeo e áudio em baixa resolução.

4.2.2.3 Sistema gráfico

O sistema de exibição gráfica de cada nó do sistema implementado é baseado originalmente em cartões gráficos 3Dlabs WildCat 4210 para o primeiro protótipo e cartões gráficos Nvidia Quadro FX 3000G para o último. Outros conjuntos de estações e cartões gráficos foram utilizados proporcionando melhores validações.

A escolha destes cartões gráficos deve-se ao fato de serem os únicos a oferecer acesso ao sinal de *genlock*, fundamental para estereoscopia ativa. Os cartões gráficos apresentam um sistema de interconexão entre as placas de cada nó, onde diversos sinais são transmitidos. Além do *genlock*, também são sincronizados *frame-lock* e *rate-lock*. Contudo os cartões gráficos 3DLabs não forneciam *drivers* para o ambiente Linux, o que dificultou seu uso. Já os cartões da Nvidia apresentam um sistema similar e *drivers* Linux eficientes. Um esquema físico dos cartões gráficos pode ser visto na Figura 4-7.

Figura 4-7 – Cartões gráficos 4210 e 3000G, utilizados no aglomerado gráfico

Estas placas possuem um sistema digital de vídeo, o DVI, este padrão pode ser puramente digital ou misturado com analógico. Foi necessário utilizamos o sistema analógico devido a tecnologia dos projetores disponíveis na época da construção da CAVERNA Digital. Algumas placas tem a opção de duas saídas de vídeo por cartão gráfico, que foi utilizado para estereoscopia passiva ou dois projetores ativos. Para maiores informações consultar apêndice A.

Placas Volume PRO VP500-2X Double Memory foram utilizadas para a aceleração do processamento em aplicações de visualização volumétrica [ZUFFO98].

¹⁸ Imagem obtida em <http://www.3dlabs.com/>

¹⁹ Imagem obtida em <http://www.nvidia.com/>

4.2.2.4 Sistema de armazenamento

O sistema de armazenamento de dados em aglomerados pode ser local ou remoto. As vantagens do sistema remoto é a possibilidade de um grande armazenamento de dados, contudo este tipo de armazenamento possui uma certa latência, que aumenta o tempo de carregar uma aplicação. Assim muitas vezes é necessária a disponibilidade de um disco local, para um rápido acesso. Um disco local mostra-se muito conveniente para armazenamento temporário de dados (*cache*). Neste caso, pode ser feita uma cópia do disco remoto para o local, e este utilizar a cópia.

A escolha inicial de discos SCSI em relação aos IDE se deve ao fato que embora os discos IDE sejam mais baratos e facilmente encontrados no mercado, e baixo custo é um dos objetivos da pesquisa, apresentam menores taxas de transmissão e robustez que os discos SCSI. Além disso o sistema SCSI permite a utilização de diversos discos ao mesmo tempo, facilitando a criação de volumes lógicos (*stripes*) para acessos muito mais velozes. Contudo o controlador SCSI só é encontrado em placas-mãe mais evoluídas e caras, muitas vezes sendo necessária a aquisição de um chip adicional ou uma placa de expansão. Assim, uma configuração mais balanceada para um aglomerado seria a utilização de um conjunto de discos SCSI em **volume lógico**, no servidor de dados, ou seja como se os diversos discos fossem um só para o sistema, permitindo o acesso de dados a maiores taxas, e um disco local IDE para reduzir custos e permitir acessos locais rápidos.

Outro ponto importante é o tipo de memória RAM. Existem diversos tipos, porém os utilizados foram as tecnologias RAMBUS e DDR. A memória RAMBUS embora apresente uma melhor taxa de transferência, o que é muito importante em aplicações gráficas, é uma tecnologia patenteada comprometendo seu custo. Em contrapartida, as memórias DDR apresentam uma taxa atual de transferência menor. Porém, com a utilização de diversos canais em paralelo é possível ter velocidades equivalentes.

Devido ao padrão XML ser adotado por diversos bancos de dados, ele se torna uma alternativa interessante para um sistema de armazenamento. Além disso, é a base do padrão X3D de descrição de dados 3D. Desta forma o banco de dados pode gerenciar o conteúdo 3D das aplicações.

4.2.2.5 Sistema de conexões e cabos

Um cuidado a ser tomado com a montagem de um aglomerado é a organização dos seus cabos, que usualmente é muito densa, evitando assim acidentes e interferência entre os sinais. Sistemas sem fio baseados no padrão *Wi-Fi 802.11* são muito práticos em algumas situações, porém tem pouca banda e necessitam cuidados com as antenas.

Figura 4-8 –Organização desestruturada de cabos

Para a melhor organização não só dos cabos, como dos computadores, é muito conveniente a montagem em um *rack*, que permite um fácil acesso, além de compactar todos os sistemas em um único módulo. A grande densidade de nós em um *rack* é interessante, pois ocupa menos espaço e auxilia o gerenciamento. Sistemas tipo lâmina (*blade*) Figura 2-1, conseguem excelente compactação em bastidores (*racks*).

Os primeiros testes com fibra óptica, em curtas distâncias, mostraram dificultar a montagem do sistema devido à sua fragilidade, sem trazer grandes vantagens. Contudo para distâncias maiores pode ser fundamental, conforme limitações do sistema por cabo.

Cuidados com energia também são importantes. Os aglomerados alteram muito seu consumo quando estão com diferentes níveis de processamento, oscilando o consumo de energia elétrica. Assim, é extremamente recomendável a utilização de um sistema de cabos robustos, para suportarem grandes potências.

4.3 Conclusões

Neste capítulo foi apresentada a CAVERNA Digital e suas variações. A CAVERNA Digital necessitou diversos cuidados para o seu correto funcionamento, criando a necessidade de diversos estudos na literatura. Embora, diferente dos outros tópicos desta pesquisa, pois o sistema de multiprojeção fosse algo já existente, tentou-se inovar criando uma infra-estrutura de maior qualidade.

Os sistemas alternativos como a projeção esférica e a CAVERNA de baixo custo, foram ensaiados para serem substitutos da CAVERNA Digital em situações necessárias, bem como validar o resto desta pesquisa em ambientes diferentes.

5. OPERAÇÃO E GERENCIAMENTO DE AMBIENTES DE REALIDADE VIRTUAL

Neste capítulo serão apresentadas as técnicas desenvolvidas para o gerenciamento completo do VRCluster, desde sua instalação e configuração até a sua operação. Será verificada a questão dos diversos dispositivos utilizados e sua integração. Finalizando o capítulo com um exemplo prático.

5.1 Gerenciamento de instalação e configuração

Um dos grandes dilemas em computação baseada em aglomerados é a instalação e atualização dos diversos nós. A instalação manual em todos os nós do aglomerado se tornaria muito lenta conforme o número de nós aumenta. Assim, foi selecionado um instalador automático para esta tarefa.

O mecanismo selecionado é o KickStart da RedHat™ [KICKSTART03], pois além de possuir uma rotina de instalação, também permite programar-se nela por intermédio de um interpretador de comandos (*scripts*), garantindo uma adaptação para necessidades de aglomerados gráficos, já que estes não existem na distribuição da RedHat. No Kickstart se informa todos os parâmetros de uma instalação normal.

O sistema de instalação automática Kickstart, através de um servidor, disponibiliza pela rede todas as informações necessárias para se instalar os computadores individualmente. O processo inicia-se no momento em que o computador é ligado. Neste instante, é executada uma busca na rede pelo sistema dinâmico de endereçamento DHCP (*Dynamic Host Configuration Protocol*) e o sistema de inicialização (*boot*) pela rede PXE (*Pre boot eXecution Environment*). Após o sistema entrar no ar, a rotina de instalação é automaticamente iniciada. Esta rotina, que foi previamente definida, instala o sistema definitivo, conforme foi programada. Assim, o computador fica preparado para suas operações de aglomerado.

5.2 Gerenciamento de operação

Para a operação de aglomerados é essencial a disponibilidade de um conjunto de ferramentas para o seu gerenciamento. Embora esta afirmação pareça óbvia, ainda há

uma grande deficiência de ferramentas para estes fins. Além disso, as soluções de gerenciamento existentes têm o foco exclusivo em sistemas numéricos, muitas vezes com interfaces não intuitivas e de baixa qualidade visual.

Dentre as ferramentas de gerenciamento que mais se destacam temos:

- O Oscar [OSCAR03] é um sistema disponível gratuitamente, que apresenta uma série de ferramentas muito úteis para aglomerados de computadores. Com ele é possível a instalação de forma fácil e rápida, utilizando arquivos RPM. Existem ferramentas para distribuir comandos, terminar processos e desligar remotamente todos os nós do aglomerado. Além de apresentar uma otimização para ferramentas de aglomerados como o LANMPI, PBS e PVM;
- O SPADE Tools é outro conjunto de ferramentas muito poderosas, contudo este só é disponível para os aglomerados de computadores da Itautec-Philco. Com ele é possível gerenciar um aglomerado da mesma forma que o Oscar, além de outras ferramentas como o gerenciador de console.

Para um melhor uso de um aglomerado de computadores, é necessário um balanceamento de seus recursos. Este balanceamento deve ser feito de tal forma que, cada nó do aglomerado, tenha uma carga de processamento correspondente a seu desempenho. No caso de aglomerados homogêneos, a carga deve ser aproximadamente a mesma em cada nó. Para isto, é fundamental que o sistema que esteja controlando os processos, não sobrecarregue um nó em relação aos outros.

Uma forma de acessar as informações dos nós do aglomerado é através de serviços de diagnósticos do computador. Um dos padrões utilizado por computadores convencionais é o sistema IPMI [IPMI02]. Este sistema foi desenvolvido pela Intel™, e faz o gerenciamento dos componentes conectados à placa-mãe, como processadores, sensores, sistemas de controle e registros de operações, tudo isso por intermédio de um protocolo bem simples. Este protocolo é aberto, assim outros desenvolvedores poderão usar o sistema livremente, padronizando e reduzindo custos. Estas informações podem ser colhidas por um sistema de rede, ou mesmo, por uma porta serial.

5.3 Gerenciamento de dispositivos eletrônicos

Diversos sistemas eletrônicos apóiam o gerenciamento do sistema como um todo. Apresentaremos aqui os principais sistemas e suas utilidades.

5.3.1 Sistema multiseria

O sistema multiseria de controle de dispositivos, foi implementado em um servidor de terminais dedicado [CYCLADES01] apresentado na Figura

Figura 5-1 – Servidor Multiseria

5-1. Este é um computador compacto, (1 posição de bastidor (*rack*)), que possui diversas portas seriais (até 48). Internamente o sistema utilizado é um computador com sistema operacional Linux, Power PC 50 MHZ, 128MB memória e mais 8MB flash. Ele também possui uma porta console dedicada, possibilitando conectar este equipamento a computadores ou modems, em situações críticas, além de possuir protocolos de segurança para as comunicações.

Desta forma, o gerenciamento de dispositivos usualmente é executado via uma comunicação serial RS232. Um sistema multiseria permite comunicar com diversas portas seriais, culminando com um sistema Ethernet. Esta tecnologia pode ser utilizada para gerenciar servidores ou nós de um aglomerado, bem como dispositivos como projetores, gerenciadores de energia (*no-breaks*), comutadores (Figura 5-2).

Figura 5-2 – Sistema de gerenciamento baseado no servidor multiseria

5.3.2 Mesa de controle

A mesa de controle apresenta uma série de controles, como botões, discos e alavancas, que podem gerenciar diversos recursos da CAVERNA Digital. Ela pode ser programada, e seus comandos transmitidos por portas seriais. Um programa servidor interpreta os comandos, baseado em um arquivo de configuração, executando as funções desejadas (Figura 5-3).

20

Figura 5-3 – Mesa de controle

5.3.3 Reconhecimento de voz

Através de um sistema de reconhecimento de voz, os diversos recursos da CAVERNA Digital podem ser acionados. Os comandos podem ser fornecidos por um microfone ligado ao sistema de gerenciamento desenvolvido. Para maiores detalhes consultar ANEXO V - Reconhecimento de voz.

5.3.4 Régua de força inteligente

Algo que pode ocorrer em aglomerados de computadores é o travamento total de um nó, neste caso, somente reiniciando o faria operacional. Contudo, muitas vezes, o acesso físico ao computador é de grande dificuldade. Neste caso, régua de força inteligentes permitem desligar remotamente a força, por um instante, de uma tomada específica, para forçar o computador a reiniciar. (Figura 5-4).

21

Figura 5-4 – Régua de força inteligente

5.3.5 Computador de teclado, vídeo e mouse

Uma outra forma de gerenciamento físico de aglomerados é a utilização de KVMs (*Keyboard Video Mouse*). Sendo prático para a instalação, porém, não acessa outros dispositivos, como projetores ou comutadores (Figura 5-5).

Figura 5-5 – KVM

²⁰ Imagem obtida em <http://www.jlcooper.com/>

²¹ Imagem obtida em <http://www.cyclades.com/>

5.4 Disponibilidade do sistema

Um ponto fundamental é preparar um sistema de alta disponibilidade (*High Availability*). Um dos argumentos contra aglomerados de computadores de baixo custo é sua instabilidade, muitas vezes devido aos componentes eletrônicos, que não são muito robustos para operações exaustivas. Além disso, não possuem um suporte de boa qualidade das empresas que os vendem [SOARES02b]. Para ambientes corporativos existem distribuições do GNU/Linux já preparadas para ambientes de aglomerados de computadores, fornecendo ferramentas e um suporte completo, para atingir clientes que necessitem de alta disponibilidade.

Usualmente, os aglomerados executam operações muito demoradas e uma falha no sistema pode causar a perda de todo o trabalho efetuado. Assim a disponibilidade de energia elétrica é fundamental. Um aglomerado que está ligado a um sistema de *no-break* é capaz de detectar a ausência de força e iniciar o processo de desligamento automático, salvando toda informação essencial.

5.5 Integração dos recursos de gerenciamento

5.5.1 Estação de controle

Devido a grande diversidade de recursos da CAVERNA Digital o seu gerenciamento torna-se complexo, por este motivo foi concebido uma estação de controle para ela, apresentada na Figura 5-6. Esta consiste de um mesa com uma série de dispositivos integrados, destacando-se a série de monitores, teclado, mouse e uma mesa de controle que é conjunto de pequenas alavancas para controle de áudio, iluminação, comutação de teclado, mouse e vídeo. O sistema de visualização deve suportar altas frequências, devido às estereoscopias, por isso a utilização de monitores analógicos.

Figura 5-6 – Estação de controle

5.5.2 Núcleo do sistema de gerenciamento - ZEUS

O sistema de gerenciamento proposto é bem amplo e tende a resolver os problemas operacionais de uma infra-estrutura de realidade virtual multiprojativa baseada em aglomerados de computadores. Este sistema é parte básica da pesquisa e desenvolvimento, pois permite a usabilidade dos recursos disponíveis. Sem ele poderia não ser conveniente usar um aglomerado, pois seria difícil seu gerenciamento.

A aplicação de gerenciamento é toda dividida em módulos independentes capazes de executar ou mandar tarefas e se comunicar por mensagens em um protocolo comum, baseado em XML. O **ZEUS** é o aplicativo coração do sistema de gerenciamento proposto. Embora muito importante, seu funcionamento é muito simples. Ele é encarregado de transportar pela rede as mensagens de gerenciamento, ou seja, os pedidos e respostas de cada ponto do sistema. Assim quando alguém faz uma requisição ao sistema, o ZEUS se encarrega de enviar a mensagem a quem é capaz de executar e retornar possíveis mensagens de resposta ao requerente. Na Figura 5-7 temos uma simulação do sistema, onde o hexágono maior representa a multiseri

Figura 5-7 – Diagrama de funcionamento do sistema

5.5.3 Interface do sistema de gerenciamento - SIRIUS

Para o controle do ambiente da CAVERNA Digital, além dos elementos físicos, como a mesa de controle, é possível utilizar um aplicativo totalmente automático, gerenciado por uma rede de computadores e optou-se por este ser totalmente independentemente de arquitetura, já que podemos utilizar Linux, Windows e IRIX. Assim mediante o **SIRIUS**, um aplicativo em java, é possível controlar remotamente todos os recursos de realidade virtual do ambiente.

A Figura 5-8 apresenta uma visualização do sistema do ponto de vista do SIRIUS. Este envia comandos para o ZEUS que se responsabiliza pelo envio para os respectivos *daemons*.

Figura 5-8 – Diagrama de transmissão de comandos

Os *daemons* controlam diversos recursos como a iluminação, vídeo, demonstrações, rastreamento, áudio e câmeras robotizadas. Todas as operações e problemas são registrados (*log*), para possíveis depurações e controle de uso. O SIRIUS e ZEUS apresentam aproximadamente 6.000 linhas de código (7 abr. 03). Para detalhes da implementação consultar ANEXO II – Diagrama UML do sistema SIRIUS.

O controle de iluminação na Figura 5-9 é usado para ligar e desligar as luzes do ambiente, assim como controlar o brilho, pois aplicações imersivas, necessitam de um tipo de iluminação específica. O ambiente de infra-estrutura deve ser escondido do usuário, para que este não se distraia e preste atenção somente na simulação. Caso existam diversos ambientes é possível dividir os controles em várias telas, auxiliando a operação.

Figura 5-9 – Sistema de controle de iluminação

O painel de aplicações, apresentado na Figura 5-10, é utilizado para executar, de forma simples, as demonstrações. Neste painel existe um ícone associado com cada aplicação, o que torna fácil seu reconhecimento. Existem também pastas permitindo que as aplicações sejam divididas em grupos, facilitando sua localização. Na parte inferior do painel, há também uma tela que exibe informações adicionais de cada aplicação.

Figura 5-10 – Painel de aplicações

Quando um clique é feito sobre o ícone, a aplicação é disparada nos nós do aglomerado de computadores previamente definidos. Com um novo clique, a aplicação será finalizada. Contudo, se uma aplicação estiver rodando e um ícone de outra aplicação do mesmo grupo for selecionado, a aplicação anterior será desligada e a nova entrará em operação. A existência de grupos se dá a fim de separar sistemas que não compartilham o mesmo recurso. Esta lógica foi testada e apresentou-se bem funcional, evitando erros de operações.

Outro painel de grande importância para o sistema é o responsável pelo controle dos projetores apresentado na Figura 5-11. Caso o ambiente tenha projetores gerenciáveis remotamente, estes poderão ser conectados no sistema para utilizar suas funções. Dentre elas as mais importantes seriam ligar e desligar o projetores, bem como alterar o sistema de

calibragem, quando um aplicativo é disparado este automaticamente pode alterar a seleção de calibragem dos projetores em uso. Existem ícones representando cada projetor do sistema, e neste é apresentado por intermédio de cores, a situação atual deles, no caso as cores são:

- Vermelho: O projetor está desligado;
- Verde: O projetor está ligado;
- Amarelo: O projetor está em estado de dormência (*stand-by*).

Embora os comutadores de vídeo possuam painéis físicos de comando, estes não são eficientes para constantes mudanças. Com o painel do comutador de vídeo, na Figura 5-12, é possível montar uma configuração de vídeo facilmente, visto que existem imagens representativas que identificam os elementos de entrada e saída de vídeo, facilitando a operação do

Figura 5-12 – Painel do comutador de vídeo

painel. Além disso, existem configurações pré-definidas que servem para um ajuste instantâneo das várias entradas e saídas de vídeo.

O acesso aos computadores, tanto pelo sistema normal de rede, como pelo sistema de multiseriial, pode ser realizado pelo protocolo **telnet**, informando o IP e porta conforme a Figura 5-13. Este protocolo abre uma ponta de comunicação diretamente com o computador, ou indiretamente pela multiseriial, permitindo a execução de comandos, ou mesmo um gerenciamento específico de cada nó do aglomerado de computadores.

Figura 5-13 – Comunicação via Telnet

O monitoramento do aglomerado é algo essencial. Com ele é possível obter-se diversas informações de cada nó computacional, como temperatura, velocidade das ventoinhas, alimentação. Esta ferramenta é especialmente útil para a detecção de problemas e, principalmente, para prevenir algum acidente.

Figura 5-14 – Monitoramento do aglomerado

Muitas vezes a visualização dos usuários no interior da CAVERNA Digital pode ser útil, desde simplesmente saber como os usuários estão se comportando, até executar algum rastreamento, tudo isso baseado em câmeras.

O painel de visualização e controle das câmeras robotizadas, apresentado na Figura 5-15, permite que as imagens capturadas pelas câmeras, sejam enviada por rede e visualizadas, bem como selecionar e controlar a movimentação das câmeras, baseados em rotação.

Figura 5-15 – Controle das câmeras robotizadas

O sistema de rastreamento é uma peça fundamental da CAVERNA Digital. Existe o interesse de exibir o estado deste sistema e permitir uma configuração dele bem facilitada. O recurso de rastreamento pode ser mapeado diretamente em uma aplicação reduzindo possíveis latências causadas pelas diversas trocas de mensagens geradas pelo ZEUS. Além disso é possível visualizar os dados de rastreamento em diversos formatos, inclusive em quaternions, que após vários testes foi o formato utilizado mais adequado.

Figura 5-16 – Sistema de rastreamento

O sistema de comunicação do SIRIUS é uma variação do XML. Para futuras versões, o padrão deverá ser totalmente seguido, visto que a implementação atual ainda não está seguindo fielmente os requisitos. Uma característica que ainda não foi totalmente explorada no SIRIUS é a sua capacidade de aceitar programações externas, como por exemplo módulos adicionais (*plug-ins*).

Figura 5-17 – Painel de desenvolvedores

Figura 5-18 – Sistema de reconhecimento de voz

Algo que se apresenta em todos os painéis é o sistema de reconhecimento de voz, apresentado na Figura 5-18. Para utilizar, este sistema basta um clique no botão da caixinha acústica. Falar os comandos que são redirecionados para o ZEUS, executando assim as tarefas. Para maiores detalhes sobre o reconhecimento de voz consultar anexo V.

5.5.3.1 Interface SIRIUS simplificada

Devido ao interesse de utilizar-se o SIRIUS em computadores compactos, como por exemplo os computadores de mão, criou-se uma versão reduzida conhecida como SIRIUS Light. Suas telas são apresentadas na Figura 5-19.

Figura 5-19 – Telas do SIRIUS Light

Esta versão do SIRIUS não apresenta componentes gráficos mais detalhados, pois o espaço em tela pode ser muito reduzido, além disso o código é mais simples para pode ser executado em computadores com menos recursos de programação. As funcionalidades são as mesmas apresentadas na versão completa do SIRIUS.

5.5.3.2 Comunicação via rede

Os comandos que são trocados entre o ZEUS, SIRIUS e *daemons* são XML em modo texto, permitindo que uma simples comunicação com telnet, possa ser feita e, assim, operar o sistema. Um ponto negativo aqui é a ausência de segurança, contudo o sistema interno, ou seja, a rede do aglomerado é considerada segura e livre de intrusos.

Um estudo que foi encaminhado é a utilização de sistemas sem fio, porém ainda não implementado. Consultar apêndice K

5.5.4 Rastreador – um exemplo de caso

Um exemplo prático do sistema de gerenciamento é apresentado na Figura 5-20. Neste exemplo o SIRIUS pede para o rastreador enviar comandos para o sistema de visualização no aglomerado. Os comandos enviados pelo SIRIUS chegam ao ZEUS que retransmite os comandos para o rastreador e para o aglomerado informando as ações. O rastreador abre um canal de comunicação com o aglomerado de computadores e retorna comandos informando que o processo funcionou.

Figura 5-20 – Comunicação do sistema de rastreamento

5.6 Conclusões

Neste capítulo foi apresentado um conjunto de dispositivos para o gerenciamento de ambientes de realidade virtual, sua integração e o conjunto de ferramentas desenvolvidas para permitir toda a integração de forma simples para o usuário.

Também foi apresentado o cuidado com a disponibilidade do sistema apresentado, que deve ser robusto o suficiente para as aplicações a serem utilizadas.

O sistema ZEUS que é o coração do sistema de gerenciamento e o SIRIUS que é a interface gráfica que permite todo o gerenciamento.

Esta pesquisa apresenta uma contribuição muito importante visto que existem diversas ferramentas para o gerenciamento de aglomerados numéricos mas não para aglomerados gráficos.

6. PROPOSTA E DESENVOLVIMENTO DE UMA PLATAFORMA DE VISUALIZAÇÃO IMERSIVA

Este capítulo apresenta a JINX que é uma plataforma de *software* para visualização imersiva voltada para ambientes avançados de realidade virtual, como a CAVERNA Digital e o VRCluster. Dentre as principais características da JINX descritas neste capítulo, destacamos a compatibilidade desta com padrões abertos, no caso o X3D e XML, além do suporte de computação distribuída.

6.1 Introdução

Como foi apresentado, a opção tecnológica oferecida pela Silicon Graphics™ é muito utilizada, principalmente pela facilidade de desenvolvimento e tradicional uso desta plataforma que levou a grande quantidade de programas disponíveis para ela. Já, os aglomerados de computadores, ainda não possuem grande quantidade de programas desenvolvidos, especialmente para área industrial, que necessitam tecnologias consolidadas e robusta.

A principal idéia da plataforma aqui apresentada é fornecer uma ferramenta que permita o desenvolvimento fácil e rápido de novas soluções e, ao mesmo tempo, seja possível reaproveitar códigos para sistemas de memória compartilhada neste novo paradigma. O programa chamado de JINX apresenta 34.445 linhas de código (em 27 de maio de 2005). Maiores detalhes da sua estrutura interna podem ser encontrado no ANEXO III – Detalhes de implementação da JINX, que apresenta o diagrama IDEF0, o diagrama de classes, o diagrama de sequência e componentização.

A JINX foi proposta para funcionar de duas formas, a primeira é a simples leitura de arquivos X3D para navegação imersiva. Outra forma é portar aplicações, isto é, utilizando a idéia de orientação a objetos, onde existe uma classe, que funcionaria como motor principal, bastaria conectar a ele um código de uma aplicação gráfica já existente. A JINX então é capaz de gerenciar o sistema para esta aplicação funcionar em ambientes imersivos distribuídos. Após sua implementação foi constatado que a maioria dos usuários preferem usar as facilidades do X3D a portar aplicações.

6.1.1 Sistema de projeções e estereoscopia

Os sistemas de realidade virtual podem ser apresentados de diversas formas, desde simples telas, como um conjunto de telas paralelas (*power-wall*), como até a CAVERNA Digital, em comum todas utilizam estereoscopia. Para conseguir o efeito estereoscópico é necessária a criação de duas câmeras que devem estar posicionadas de forma similar aos olhos humanos. Existem duas técnicas principais para o posicionamento das câmeras [BOURKE99] [BOURKE02]:

- **Dedão (*Toe-in*):** Neste método as duas câmeras são separadas e rotacionadas para o centro. Onde os raios de visão se colimarem, será definido o ponto zero, ou seja, onde os objetos aparecerão na mesma posição da tela. Este método não é totalmente correto, criando problemas em sistemas de multiprojeção, principalmente nas bordas (Figura 6-1).

Figura 6-1 – Câmeras formato dedão

- **Método correto (*Correct method*):** Este método, também conhecido como projeção perspectiva, é baseado em cones de visão assimétrica de eixos paralelos. Os eixos das câmeras mantêm-se sempre paralelos, e o efeito de estereoscopia é feito por intermédio do cone de visão, que é ligeiramente deformado e deslocado (Figura 6-2).

Figura 6-2 – Câmeras formato correto

A paralaxe, que é o ângulo que faz com que os olhos separados tenham um foco é muito importante para uma perfeita noção de profundidade. Este pode ser controlado conforme cada pessoa, tela e simulação. A Equação 1 calcula a paralaxe.

$$paralax = 2 * \arctan\left(\frac{DX}{2 * d}\right)$$

Equação 1 – Cálculo da paralaxe

Nesta fórmula, DX é a separação horizontal de dois pontos projetados e d é a distância entre o usuário e a tela.

Pode-se verificar que um objeto pode ser projetado na tela da CAVERNA Digital pela matriz homogênea [CRUZ92] da Equação 2.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{e_x}{e_z - PP} & -\frac{e_y}{e_z - PP} & 1 & -\frac{1}{e_z - PP} \\ \frac{e_x PP}{e_z - PP} & \frac{e_y PP}{e_z - PP} & 0 & \frac{e_z}{e_z - PP} \end{bmatrix}$$

Equação 2 – Matriz homogênea de projeção em sistemas cúbicos

Sendo e_x, e_y, e_z o ponto de vista do usuário no interior da CAVERNA Digital e PP a distância da origem até a tela. Esta equação é para a tela frontal.

O cálculo de visão em situações de rastreamento de cabeça mostra algumas complexidades a mais devido à irregularidade do cone de visão. Ou seja, a origem do cone é exatamente o ponto de vista onde o usuário está observando e este dado é obtido através de algum sistema de rastreamento.

Figura 6-3 – Menu de interação

O ambiente gráfico é responsável tanto pela exibição como pela leitura de dados, desta forma existem menus para a interação com o ambiente. Estes menus afetam o

sistema como um todo, e podem ser utilizados para o gerenciamento do sistema. A Figura 6-3 apresenta o menu de interação, que cobre todas as opções desejadas desta implementação, tanto para navegação como comandos para um sistema de calibração de telas, que é fundamental para a instalação e manutenção dos projetores.

Curiosamente existem poucos aplicativos que disponibilizem ferramentas para a calibração de sistema de realidade virtual. Esta pesquisa apresenta estudos sobre as principais técnicas de calibração, sendo estas selecionadas e adaptadas, chegando a uma série de padrões apresentados no apêndice I.

6.1.1.1 Método de visualização

Devido a grande variedade de dispositivos de visualização, a JINX foi concebida para suportar os mais tradicionais, apresentados na Figura 6-4. As visualizações podem ser com ponto de vista rastreado ou não.

Figura 6-4 – Formatos de telas suportados

Da mesma forma, os sistemas de projeção podem estar de tal forma que seja conveniente uma movimentação da imagem a fim de posicioná-la de forma correta. Os tipos de transformações suportados pela JINX são rotação e espelhamento, como apresentados na Figura 6-5.

Figura 6-5 – Tipos de transformações suportados

6.1.2 Gerenciamento das geometrias

O OpenGL, embora uma ferramenta muito poderosa, não implementa os conceitos de grafo de cena, como por exemplo o X3D propõe. Desta forma, se torna necessária a organização dos dados geométricos para que este se torne o mais eficiente possível. Assim, a JINX implementa este recurso otimizando o uso da memória do computador. Já, devido ao OpenGL ser uma máquina de estados é ideal que os dados estejam organizados de forma a minimizar o número das mudanças de estados. As técnicas mais modernas usam o conceito de árvore de visita (*visiting tree*) que não utilizam a ordem do grafo de cena, mas sim a mais conveniente para ganho de desempenho. Contudo, um cuidado a mais, que foi tomado aqui, é a ordenação destes dados para posterior compartilhamento dentro do aglomerado de computadores, possibilitando futuramente explorar recursos como *sort-last* ao máximo.

Um exemplo de vantagem, que vai além da árvore de visita, é ordenar a cena de forma que seja fácil balancear a carga de textura entre os nós, assim se poderia usar ao máximo a memória de textura de cada nó e ter imagens com melhor qualidade.

6.1.3 Sistema de som

A utilização de som na simulação é muito importante, melhorando radicalmente a imersão. Existem diversas formas de sonorizar um ambiente:

- Definindo pontos emissores no ambiente virtual. Estes pontos agem como fontes, que sugerem ao usuário que o som vem daquele ponto no espaço.
- Som ambiente, neste caso o som não possui direção e pode ser ouvido de qualquer ponto da cena. Este tipo de som é muito útil para uma música no ambiente, ou passar informações independente de onde o usuário esteja.

- Sons baseados em eventos, ocorre devido a uma colisão ou interação de outro tipo, emitindo um som característico.

Para a correta espacialização do som, o usuário pode receber o som de diversas caixas acústicas espalhadas no ambiente ou utilizar-se de algum sistema de fone de ouvido, e este deve ser rastreado, para o conhecimento da posição e orientação de quem está ouvindo. Para detalhes dos sistemas consultar apêndice B.

6.1.4 Sistema de controle e navegação

Para o controle e navegação no ambiente virtual, é necessário enviar comandos para o sistema e estes comandos devem ser interpretados e distribuídos para os nós do aglomerado. Para aplicações imersivas é necessária alguma forma de navegar por este mundo nas três dimensões e, muitas vezes, interagir com algum objeto ou estrutura presente no ambiente.

Para esta entrada de dados podem ser utilizados diversos dispositivos, desde os mais tradicionais, como teclado e mouse, até sistema mais evoluídos, como rastreadores. O grande inconveniente do teclado e mouse é o fato de não serem eficientes para ambientes com grande liberdade de movimento, como no caso da CAVERNA Digital. Desta forma, o uso de teclado e mouse é recomendado apenas para sistema de simulação. Mas, para o usuário final, existem dispositivos mais recomendáveis, como o rastreador eletromagnético. Com o rastreador existe um grande grau de liberdade, mesmo que estes tenham fios. Outros dispositivos nos quais foram desenvolvidos suportes são as luvas de realidade virtual, joysticks alternativos e rastreadores de rotação, conforme apresentados na Figura 6-6.

Figura 6-6 – Outros dispositivos suportados pela JINX

²² Imagem obtida em <http://www.5dt.com/>

²³ Imagem obtida em <http://www.intersense.com/>

Usualmente a comunicação destes dispositivos é feita através de uma porta de comunicação serial, seja ela RS232 ou USB. Por exemplo, o rastreador ou mouses espaciais (*space ball*) utilizam o sistema RS232. Já para os *joyticks* atuais, é utilizada a conexão USB que é mais interessante, pois permite a utilização de repetidores (*hub*), garantindo mais conexões por portas, e melhor reconhecimento de dispositivo.

Os comandos dos dispositivos devem ser interpretados. Assim, por intermédio de um código específico para cada dispositivo, os comandos são lidos e, então, traduzidos para um formato padrão que consegue abstrair todas as principais informações necessárias para a navegação e interação. Este código pode estar junto do aplicativo principal, contudo nem sempre isto é possível ou desejável. Assim existe a possibilidade de um aplicativo enviar uma seqüência de dados no formato XML e a aplicação converter isto para o formato padrão. Posteriormente são enviadas somente as informações necessárias de cada dispositivo para o sistema central da JINX. Um esquema de como as informações dos dispositivos trafegam está apresentada na Figura 6-7.

Figura 6-7 – Sistema de comunicação de dispositivos

6.1.5 Sincronismos e comunicação

Para o sincronismo e comunicação de dados foram estudadas diversas abordagens, contudo a que se mostrou mais prática e eficiente em relação a proposta foi o MPI(Message Passing Interface), no caso da implementação desenvolvida usa-se o MPICH, que possui a vantagem de ser gratuito e operar tanto em plataformas Unix, como no ambiente Windows. Os tipos de sincronismos, com exceção do *genlock*, são totalmente implementados nele. Temos o *frame-lock* que é feito bloqueando todos os processos antes da execução da troca de memórias de vídeo (*frame buffers*), e somente quando todos estiverem neste ponto, os programas serão desbloqueados, permitindo que todos os processos executem a troca de buffer, que ocorrerá, basicamente, no mesmo período, e o *data-lock*, que será feito trocando-se mensagens das mudanças de estado que o sistema vai sofrendo.

Contudo para a comunicação dentro do próprio nó, o MPI não se apresentou interessante. Assim a comunicação interna nos nós é baseada em thread em especial o Pthread. Este sistema é muito eficiente e opera em diversas plataformas. A troca de mensagem deve ser cuidadosa para a correta implementação da exclusão mútua.

Na Figura 6-8 é esquematizado um sistema de aglomerado de computadores multi-processados. Cada nó do aglomerado possui dois processadores. Estes podem utilizar a mesma área de memória, não sendo necessário o envio de mensagem, daí a vantagem das threads. Além

Figura 6-8 – Sistema híbrido de comunicação

disso, a velocidade de transferência é a mais alta do sistema. Já para processadores de computadores diferentes, é necessário o envio de mensagens pela rede, que por consequência é um processo mais lento. Por este motivo, é fundamental que o sistema estudado implemente os dois conceitos de transferência de dados.

A comunicação principal envolverá diversos movimentos, ou seja, conforme o usuário se desloca no ambiente virtual, cada nó estará a par deste novo posicionamento. A informação de movimentação será assíncrona, ou seja, a leitura dos dispositivos é contínua. Já a parte gráfica e sonora é síncrona e estará na forma bloqueada, ou seja, somente quando todos os nós estiverem prontos para a atualização gráfica e sonora que ocorrerá o desbloqueio e os nós usarão novos dados. Assim, se não houver atualização da posição, o servidor simplesmente ignora, não bloqueando o sistema. Um diagrama de como os módulos se comunicam é apresentado na Figura 6-9.

Figura 6-9 – Sistema de comunicação da JINX

Os comandos dos sistemas de controle, usualmente *joysticks*, são enviados para uma rotina respectiva, que executa a sua leitura. Então, são feitos cálculos para tratar o sistema de coordenadas lido, para o sistema do aplicativo. Posteriormente, todas estas rotinas enviam suas informações de movimentação para uma rotina principal de controladores, fazendo uma média dos movimentos. Isto é conveniente quando se está usando dois dispositivos ao mesmo tempo para controlar o ambiente. Estes comandos são então enviados para a rotina principal, que empacota tudo e reenvia de forma organizada para todos os nós do aglomerado. Uma representação prática é apresentada na Figura 6-10.

Figura 6-10 – Exemplo de transmissão de mensagens

Um tipo de sincronismo que foi proposto nesta arquitetura é o que nós denominamos *TimeLock*, este sincronismo é especialmente válido para os arquivos X3D, pois comumente as animações nestes formatos são baseadas em percursos pré definidos animados durante o tempo. Assim se o tempo universal dos nós do aglomerado foi sincronizado a cada frame a animação será correta e isso não acarretará nenhum grande consumo de banda da rede.

6.2 Desenvolvimento baseado no método proposto

Devido ao princípio utilizado são possíveis duas formas de desenvolver aplicações. A primeira consistiria de somente a leitura de um arquivo em X3D e a outra baseada em uma aplicação gráfica, gerenciada pela biblioteca. Devido a menor relevância do sistema baseado em aplicações gráficas, aqui haverá um enfoque na leitura de arquivos.

6.2.1 Leitura de arquivos

A leitura e conversão de arquivos na maioria dos programas de realidade virtual são de grande dificuldade. Normalmente, estas aplicações possuem um formato nativo e não suportam todas as características de formatos não nativos. Nesta pesquisa escolhemos o X3D, que mostra ser um formato robusto e livre de qualquer tipo de restrição, como patente, e possui muitos recursos interessantes para o desenvolvimento proposto.

Além disso existem aplicativos e *plug-ins* que convertem outros formatos geométricos para X3D, permitindo assim que a JINX opere indiretamente com diversos programas de modelagem tridimensional.

```
#include "engine.h"

int main(int argc, char* argv[]) {

    Engine* engine = new Engine(&argc,argv);

    engine->Configuration("http://foo/render.xml");

    engine->X3D("http://foo/exemplo.x3d");
    //engine->OpenGL(&myDrawGLScene);

    engine->run();

    delete engine;
    return(0);

}
```

Neste código vê-se que o processo de criação de um programa é muito simples. O arquivo de configuração é baseado em XML e foram propostas, mediante uma análise das tecnologias, outras soluções disponíveis. Mais detalhes consultar ANEXO IV – Arquivo de configuração da JINX.

Quando a JINX abre um arquivo X3D ela otimiza os dados para serem alocados na memória. Os dados são organizados em forma de árvore, como na Figura 6-11, ou seja, em um grafo de cena. Devido as vantagens da orientação a objetos, é criada uma classe para cada nó do X3D, que são posteriormente organizados na memória por ponteiros. Estes ponteiros

Figura 6-11 – Exemplo de grafo de cena

podem acessar os nós da árvore irregularmente. Esta técnica, embora muito fácil e prática, não é a mais eficiente, devido às placas gráficas usarem máquinas de estado, a mudança constante de estados diminui consideravelmente o desempenho. Neste caso, é recomendável fazer uma árvore de visitas, que tenta reduzir a mudança de

estados. Um fato importante é os modernos cartões gráficos poderem carregar as informações geométricas em sua própria memória, desta forma, todos os vértices e polígonos ficam na placa gráfica, aumentando ainda mais o desempenho.

6.2.2 Hierarquia de sistemas

A Figura 6-12 apresenta um resumo da hierarquia dos sistemas utilizados na JINX. Para a geração de áudio é utilizado o Fmod. Para a comunicação, principalmente o MPI, contudo para algumas tarefas onde o MPI não seria possível ou interessante, é utilizado o sistema de *Sockets*. Outras alternativas como CORBA foram estudadas porém não foram

Figura 6-12 – Hierarquia da JINX

utilizadas, para maiores informações sobre padrões de comunicação verifique o apêndice C. Para a parte gráfica é utilizado o OpenGL, que possui uma comunicação direta com os cartões gráficos, acelerando a síntese de imagens.

6.2.3 Avanços em *Shading Languages*

A nova geração de placas gráficas oferece inúmeros novos recursos, em especial, as *Shading Languages*, que oferecem recursos muito poderosos de síntese de imagens. Estes recursos já começaram a ser especificados inclusive pela comunidade X3D.

Estas técnicas já foram adaptadas e implementadas na JINX, como forma de melhorar a qualidade dos conteúdos. Na Figura 6-13 é possível verificar duas esferas que foram produzidas com tal técnica. A qualidade é bem superior a tradicional, porém, é grande queda de desempenho, conforme a proporção visualizada de esfera.

Figura 6-13 – Exemplo de aplicação com *Shading Languages*

6.3 Imagens diversas de aplicações desenvolvidas

Esta seção apresenta uma série de projetos que utilizaram a JINX, sendo que a maioria também utilizou o VRCluster e a CAVERNA Digital.

A Figura 6-14 e Figura 6-15 apresentam as primeiras cenas produzidas com a JINX. Elas são muito simples e foram criados conforme os nós X3D eram implementados na JINX. A partir disto ficou comprovado que este deveria ser um foco da pesquisa

Figura 6-14 – Primeiras cenas da JINX (captura de tela)

Figura 6-15 – Primeiras visualizações da JINX

As imagens da Figura 6-16 são baseadas nos desenhos de Escher. Este modelo foi produzido por Leonardo Nomura, no seu trabalho de conclusão de curso.

Figura 6-16 – Simulação de Escher

A casa, apresentada na Figura 6-17, foi produzida por Richard Ibara utilizando como ferramenta de modelagem o 3D Studio, e foi baseada em plantas reais fornecidas pelo arquiteto.

Figura 6-17 – Simulação arquitetônica

O modelo do Rio de Janeiro visto na Figura 6-18, que possui cerca de 150.000 polígonos, foi primeiramente utilizado em um evento de turismo e, devido ao sucesso, reutilizado em diversos eventos. Este modelo foi produzido por Leonardo Nomura e Mario Nagamura.

Figura 6-18 – Imagens da simulação do Rio de Janeiro (captura de tela)

Um extensão para sistemas de projeção esférica foi desenvolvida pelo engenheiro Lucas Dulley, permitindo a visualização dos modelos em sistemas tipo domos, como mostra a Figura 6-19.

Figura 6-19 – Sistema do Rio de Janeiro na CAVERNA Digital

As imagens da Figura 6-20 foram apresentadas em um sistema tipo *PowerWall* com estereoscopia passiva e acionada por um cluster de dois nós no congresso da SBPC 2004 em Cuiabá. O controle era mediante um joystick.

Figura 6-20 – Passeio no Rio de Janeiro em *PowerWall*

A Figura 6-21 apresenta, uma simulação em quatro monitores dispostos em paralelo, que foi montado na COMDEX 2003, apresentando um passeio na casa com um caixa eletrônico da Itaútec. Este mesmo sistema foi montado no SIBGRAPI2003 em São Carlos.

Figura 6-21 – Sistema apresentado na COMDEX03 e SIBGRAPI

A JINX e o modelo do Rio de Janeiro foram submetidos para o SIGGRAPH 04 e foram apresentados, na seção “3DX Demo or Die”, conforme Figura 6-22, em um tela de grandes proporções, para um público aproximado de 2.500 pessoas. Este mesmo vai ser apresentado novamente no SIGGRAPH 2005, porém desta vez na estrutura completa da Asa Delta.

Figura 6-22 – Apresentação no SIGGRAPH 04

Um dos projetos de físico-química é o de visualização molecular, apresentado na Figura 6-23. Neste sistema é possível visualizar moléculas e estudar o comportamento delas. Este projeto foi assessorado pela Silvia Esparrachiari Ghirotti.

Figura 6-23 – Visualização molecular

O projeto de treinamento em disjuntores de grandes estações de energia, apresentado na Figura 6-24, foi o primeiro a utilizar o sistema de roteiro (*scripts*), desenvolvido em colaboração com o Rodrigo de Godoy Domingues. Neste sistema, os funcionários podem testar suas habilidades de trocar peças e efetuar manutenções diversas.

Figura 6-24 – Treinamento de disjuntores elétricos

Outro projeto, que ainda está em andamento, é da visualização da Universidade de São Paulo, apresentado na Figura 6-25. Nele é possível visualizar diversos pontos da Universidade. Este modelo foi desenvolvido por Mario Nagamura.

Figura 6-25 – Visualização da Universidade de São Paulo

A ferramenta de autoria LEGAL desenvolvida pela Andréia Regina Pereira é utilizada para educação infantil, sua última etapa consiste da geração de um mundo X3D, que as crianças produzem. A Figura 6-26 apresenta alguns mundos que foram gerados e visualizados na CAVERNA Digital através da JINX.

Figura 6-26 – Navegação de cenário produzido pelo LEGAL

6.3.1.1 Gerenciamento de aplicações diversas

Com as técnicas apresentadas é possível executar rotinas, utilizando aplicações específicas, como as de Visualização Científica [BATTAIOLA98]. Ou seja, uma aplicação desenvolvida com outra ferramenta pode ser portada para o sistema de desenvolvimento proposto.

Na Figura 6-27, são apresentadas imagens de um software desenvolvido em VTK, que se pode embutir na JINX para então ser visualizado na CAVERNA Digital.

Figura 6-27 – Imagens diversas do METLAB, sistema desenvolvido em VTK

6.4 Conclusões

Neste capítulo foi apresentada a ferramenta JINX e todas as técnicas desenvolvidas para viabilizar seu funcionamento. O principal objetivo inicial dela, que era ser uma ferramenta que permitiria de forma fácil a imersão em ambientes de realidade virtual desenvolvidos com ferramentas comuns ou mesmo textuais, foi alcançado. Isto é comprovado pelos inúmeros projetos de pesquisa paralelos que utilizaram a JINX e que alavancaram a evolução dela.

Pelos diagramas apresentado neste capítulo se tentou mostrar abstratamente que os princípios dela são claros, porém houve diversas dificuldades de implementação, que conforme foram solucionadas foram documentadas no próprio código.

Apresentamos diversos estudos realizados, como estereoscopia, gerenciamento de geometrias e som. Pôde ser visto que esta área está muito aberta, permitindo o desenvolvimento de muitas técnicas de otimização de simulações em realidade virtual usando o VRCluster a CAVERNA Digital. Alguns avanços com as *Shading Languages* estão sendo estudas e trarão avanços muito poderosos para o sistema de visualização.

7. VALIDAÇÃO EXPERIMENTAL DOS SISTEMAS DESENVOLVIDOS

7.1 Introdução

Neste capítulo serão apresentadas diversas análises e testes sobre esta pesquisa, verificando-se sua validade experimental. O objetivo da validação é provar de forma experimental as inúmeras decisões técnicas tomadas ao longo da implementação.

7.1.1 Geração de imagens paralelas

Um dos inconvenientes da síntese de imagens de um mesmo modelo em diversas visões é dela ser processada na íntegra a cada quadro, criando uma grande redundância. Para a visualização de uma imagem 3D [CUNNINGHAN03] são necessários alguns passos, conforme a Figura 7-1. Os modelos nas coordenadas do mundo são os mesmos para todas as cenas, o que poderia ser reaproveitado. Contudo, devido as técnicas atuais, isto não é possível.

Figura 7-1 – Pipeline gráfico

Este tópico de pesquisa ainda precisa ser avaliado, pois não foi resolvido nas pesquisas de paralelização gráfica. Em especial, em sistemas esféricos, um melhor aproveitamento do *pipeline* gráfico aumentaria consideravelmente o desempenho.

7.2 Comparações entre plataformas

Até recentemente a plataforma de computadores para ambientes imersivos, que inclui a CAVERNA Digital, mais utilizada é a da Silicon Graphics™, baseada em estações Onyx, apresentada na Figura 7-2. Esses computadores podem ter um número

24

Figura 7-2 – Estação Onyx

²⁴ Imagem obtida em <http://www.sgi.com/>

variável de **filas de processamento gráfico** (*graphic pipelines*), que são escolhidas conforme a complexidade do sistema de projeção. A Silicon Graphics™ também oferece uma outra possibilidade baseada em cartões gráficos mais simples para a plataforma Onyx. Este é o InfinitePerformance podendo ter até 8 placas. Para uma completa utilização deste computador é necessário um sistema de composição, que permite, dentre outras possibilidades, a junção das imagens. Mais detalhes consultar seção 2.5.

Este tipo de estação oferece recursos como paginação de textura, visualização volumétrica, múltiplos canais de vídeo. Devido ao tradicional desempenho gráfico desta plataforma e seu legado, a maioria dos programas de realidade virtual imersiva é dirigido para a família de servidores Onyx.

Uma alternativa de baixo custo são os cartões gráficos de mercado, para computadores pessoais (*workstations*), dirigidos para a plataforma PC, apresentando aceleração OpenGL, suporte para textura 3D, chip gráfico reprogramável, e em alguns casos, a capacidade de *frame lock*, *genlock* e *rate lock*. Este último, de particular interesse para este trabalho.

Figura 7-3 – Cartões gráficos comerciais (NVidia, FileGL, WildCat)

Da mesma forma que a Onyx, estes cartões gráficos apresentam a capacidade de diversas filas de processamento gráfico, que em alguns casos podem ser gerenciados para um melhor ganho de desempenho.

²⁵ Imagem obtida em <http://www.nvidia.com/>

²⁶ Imagem obtida em <http://www.ati.com/>

²⁷ Imagem obtida em <http://www.3dlabs.com/>

Indicador	InfiniteReality3 Desempenho por Fila de processamento Gráfico ¹	Wildcat 4210 (2 Filas de Processamento Gráfico)	Wildcat 7210 (2 Filas de Processamento Gráfico)	Nvidia Quadro FX 3000G
Polígonos/segundo	13,1 Milhões	11 Milhões	37,9 Milhões	100 Milhões
Voxels/segundo	400 Milhões	Não informado	Não informado	1,8 GB/s
Trilinear textured	200 a 800 Milhões	243 Milhões	400 Milhões	3.2 Bilhões
Interpolation/segundo				
Vetores/segundo	8,6 Milhões (anti-aliased)	17,6 Milhões (solid)	33,9 Milhões (solid)	Não informado
Cor	48-bit (RGBA)	32-bit (RGBA)	32-bit (RGBA)	128-bit
Luzes suportadas por <i>hardware</i>	Não informado	24	32	Não informado
Overlay planes	16	8	8	Suportado/Não informado
Anti-aliasing multisampling	8x8 = 64	SuperScene = 16	SuperScene = 16	16x (FSAA)
Max. Bits/pixel	2048	256	128	Não informado
Memória de textura	256 MB	128 MB	256 MB	256 MB (total)
Frame buffer	80 MB a 320MB	128 MB	128 MB	256 MB (total)
Resolução padrão máxima	1920x1200	1280 x 1024 ²	1280 x 1024 ²	3840x2400

Tabela III – Comparação SGI Onyx IR3 x 3Dlabs Wildcat 4210

¹ dados referentes a estações Onyx 2, embora não encontrada variação para Onyx 3000.

² cartão oferece resoluções maiores, mas não com suporte à estereoscopia.

A **Tabela III** apresenta uma comparação de desempenho gráfico, para vários indicadores de desempenho, entre as plataformas Onyx e o cartão gráfico WildCat e Nvidia em um computador convencional. Estes dados foram obtidos no início do projeto. Os dados expostos, foram adquiridos de catálogos dos fabricantes, sendo que no caso da Wildcat e Nvidia as informações são em condições de pico, já para a Onyx não foram informadas as condições de medida.

Uma outra fonte de informações, para medidas de desempenho gráfico, é o órgão SPEC (*Standard Performance Evaluation Corporation*), uma organização sem fins lucrativos que mantém e padroniza testes de desempenho, que são aplicados nas novas gerações de computadores, bem como mantém e publica os resultados. Ela apresenta o SPECViewPerf, que é uma ferramenta de medida de desempenho para OpenGL, porém as informações na página são superficiais, desatualizadas e restritas somente a alguns modelos de sistemas, utilizando Windows, inviabilizando uma comparação prática com sistemas Unix.

3dsmax-03 = 21.65	catia-01 = 14.46	ensight-01 = 8.691	light-07 = 6.320
maya-01 = 29.61	proe-03 = 23.79	sw-01 = 10.71	ugs-04 = 19.63

Tabela IV – Resultados de um nó do Perfect Cluster para o SpecViewPerf

Figura 7-4 – Amostras dos dados usados pelo SpecViewPerf

Dos dados informados, na Tabela IV, é visível que o aglomerado desenvolvido apresenta um desempenho muito bom, em relação aos sistemas SGI, que evoluíram em um passo mais lento que os componentes convencionais para computador. Contudo, no quesito memória gráfica, a Onyx apresenta uma melhora considerável, que é essencial para os sistemas imersivos, que necessitam de grandes taxas de pontos (*pixels*), nos outros recursos a evolução foi muito lenta ou inexistente.

Devido à concorrência do mercado de cartões gráficos para computadores pessoais, muitos produtos foram lançados recentemente, cada vez apresentando melhores velocidades e recursos. Desta forma, a utilização de aglomerados gráficos torna-se mais interessante, pois é um mercado altamente competitivo, que só traz benefício aos usuários, com soluções cada vez melhores.

²⁸ Imagens obtidas através do SpecViewPerf

Duas outras empresas, que apresentam soluções gráficas de alto desempenho, são a HP™ e a SUN™, contudo, estão bem distantes das necessidades exploradas, assim não serão abordadas em maiores detalhes.

7.3 Testes de desempenho

Diversos testes de desempenho foram realizados a fim de detectar a validade do sistema desenvolvido em um ambiente prático. Eles foram planejados para detectar os pontos críticos de uso do sistema de realidade virtual imersivo.

7.3.1 Teste de escalabilidade

Os testes de escalabilidade pretendem mostrar como o aumento linear de nós afeta o desempenho. Devido ao sincronismo de dados, *data-lock*, a velocidade do sistema será relativa à estação que apresentar menor desempenho, ou seja, a velocidade de toda a simulação será a do computador mais lento.

A velocidade da simulação depende de diversos fatores, sendo os pontos mais críticos a complexidade da imagem gerada e o cartão gráfico utilizado. Nem todas as técnicas de otimização de síntese de quadros foram exploradas, como, por exemplo, a baseada na geração de imagem somente para os objetos no campo de visão (*view frustum culling*), assim o cartão gráfico tem grande influência no desempenho.

7.3.1.1 Testes preliminares

As estações utilizadas para estes testes são bem distintas, de forma a verificar se o sistema se adapta a sistemas heterogêneos, sem perder o desempenho. O cálculo é baseado em quadros por segundo (*frames/second*). As estações utilizadas para estes cálculos preliminares estão na Tabela V.

O aplicativo *glxgears* é uma demonstração originária dos sistemas GLUT e portada para GLX. Ele desenha três engrenagens coloridas rotacionando entre si e exibe a taxa de quadro por segundo periodicamente. Originalmente, acompanhava os utilitários do MESA, porém

Figura 7-5 – Aplicativo *glxgears*

agora é facilmente encontrado em vários ambientes Linux e Windows.

A Tabela V apresenta o desempenho de cada sistema para o aplicativo glxgears. Este aplicativo foi escolhido, por ser comumente utilizado e fornecer uma idéia da velocidade do sistema, que se reflete nos testes com o aplicativo desenvolvido.

Estação	Desempenho (quadros/seg)	Sistema
PC1	$5.011,2 \pm 0,2$	Processador: Intel Pentium 4 2,2 GHz Memória: 768 MB Cartão Gráfico: Nvidia Quadro 4 XGL 550
PC2	$839,4 \pm 0,2$	Processador: Intel Pentium 4 2,2 GHz Memória: 512 MB Cartão Gráfico: Nvidia Gforce 256
SGI	$1.292,2 \pm 0,2$	Processador: MIPS R12000 400 MHz Memória: 1024 MB Cartão Gráfico: Infinite Reality 3
Perfect Cluster (por nó)	$16.950,0 \pm 70,0$	Processador: Intel Dual Xeon 3,1 GHz Memória: 2048 MB Cartão Gráfico: Nvidia Quadro FX 3000G

Tabela V – Desempenho relativo ao aplicativo glxgears

Para os testes preliminares de escalabilidade foi utilizado a JINX. Os valores apresentados estão em quadros por segundo. A cena está em 640x480 pontos em monoscopia, o que é facilmente suportado por todos os sistemas. Contudo, não deve ser considerado como um resultado refinado. Para os teste finais foram utilizados sistemas mais precisos.

Figura 7-6 – Testes preliminares da queda de desempenho em aglomerados

7.3.1.2 Testes finais

Devido a arquitetura Silicon Graphics™ não suportar os modelos de alta complexidade, esta não será mais usada. Os testes a seguir foram realizados no Perfect cluster, com resolução em 1280x1024, e os dados utilizados foram os do Rio de Janeiro disponíveis em: <http://www.lsi.usp.br/interativos/nrv/hangGlidingRJ.html>. O teste verificou o desempenho conforme o número de nós. O modelo foi carregado,

percorrido para estabilizar as texturas, e posicionado em certo ponto para a leitura dos dados.

A Figura 7-7 mostra o ganho, em pixels, conforme se aumenta o número de nós.

Figura 7-7 – Resolução relativa ao número de nós

A Figura 7-8 mostra a contensão da rede conforme se aumenta o número de nós do aglomerado de computadores.

Figura 7-8 – Contensão de rede do modelo Rio de Janeiro em monoscopia

O teste apresentado na Figura 7-9 é o mesmo, porém com modo de visualização estereoscópico.

Figura 7-9 – Contensão de rede do modelo Rio de Janeiro em estereoscopia

É possível verificar, mediante Figura 7-8 e Figura 7-9, que o desempenho continua caindo conforme o número de nós aumenta, em especial a mudança de 1 para 2 nós encontramos a maior redução em velocidade. Isto se deve porque em um só nó toda a comunicação é local, e assim não há necessidade de transmitir dados pela rede. Outro ponto a ser comentado é que nós em números pares apresentam melhor desempenho que seu predecessor ímpar, contudo testes com números maiores de nós devem ser realizados para confirmar esta afirmação.

Figura 7-10 – Contensão de rede do modelo Rio de Janeiro em modo esférico

A Figura 7-10 apresenta os resultados do mesmo modelo do Rio de Janeiro, utilizando um sistema de visualização esférica. Este sistema internamente necessita 5 sínteses de imagens para gerar a projeção esférica, motivo deste ser mais lento.

Outro teste realizado tratou da verificação de como o número de polígonos e sua organização alteram o desempenho da aplicação. A Figura 7-11 apresenta uma série de testes realizados em uma só estação para verificar o comportamento da JINX. Todos os testes usaram 100 coletas de tempo, somente uma luz pontual e resolução de 1280x1024 pixels. O primeiro teste, T1, consistiu em desenhar consecutivamente esferas de 1024 faces e medir a quantidade de quadros por segundo, conforme o número de esferas aumentava a velocidade diminuía. O teste T2 utilizou a mesma geometria, contudo, com uma textura de 4096 x 4096 pixels aplicada nela. Posteriormente, no teste T3 foi aplicado consecutivamente grupos de 10 esferas com texturas diferentes menores de 2048x2048 pixels, mostrando que a quantidade de texturas influencia na velocidade. O teste T4 aplicou um sistema que agrupava 1024 estruturas, em especial nós Transform, com 1 face cada. Pode-se perceber que a queda de desempenho é muito acentuada, deixando claro que, se os polígonos não estiverem organizados em conjunto, o desempenho cairá consideravelmente. Para validar esta teoria, foi realizado um último teste, que exibiu esferas de 100 faces agrupadas em 10 nós Transform e pôde-se verificar o comportamento intermediário. Algo constatado é que a queda de desempenho é sempre logarítmica.

Figura 7-11 – Queda de desempenho em relação a número de polígonos

- T1** n esferas de 1024 faces sem textura
- T2** n esferas de 1024 faces com textura de 4096x4096 pixels
- T3** n grupos de 10 esferas de 1024 faces e texturas de 2048x2048 pixels
- T4** n grupos de 1polígono dentro de 1024 nós Transform
- T5** n esferas de 100 faces dentro de 10 nós Transform

Tabela VI – Referências das siglas de queda de desempenho

Uma outra medida importante é saber o tempo carga, desde o acionamento da aplicação até sua visualização. Este tempo tem relação direta com o arquivo a ser utilizado que deve ser recuperado do sistema de armazenamento distribuído. A Figura 7-12 apresenta o tempo de carga de arquivos X3D que apresentam estruturas complexas de dados e múltiplos de 512K. Os tempos apresentados são os menores durante 10 medidas. Durante as medidas houve uma grande instabilidade do sistema, podendo a mesma levar 10 vezes o tempo mínimo. Conforme os arquivos ficavam maiores esta instabilidade aumentava. Uma provável causa deste comportamento é devido ao sistema ser baseado em um servidor de http, que pode não apresentar as condições ideais para esta aplicação.

Figura 7-12 – Tempo de carga de arquivos X3D

7.3.2 Teste de qualidade visual

A qualidade visual depende de diversos fatores e, muitas vezes, é difícil medir esta característica. Um dos principais é a resolução das imagens. No sistema mais complexo disponível para experimentos, a CAVERNA Digital, a quantidade de pontos (*pixels*) por área, que pode ser chamado de acuidade visual (*visual acuity*), é fundamental [CRUZ92]. O tamanho útil da tela é de 3x3 metros, desta forma, se utilizarmos uma resolução padrão de vídeo (640x480) cada ponto na tela terá dimensões de 4,6875 x 6,25 milímetros. Este tipo de resolução comprometeria a qualidade da visualização. Assim são utilizadas resoluções maiores como 1280 x

1024, passando o tamanho do ponto a ser 2,34375 x 2,9296875, o que representa uma grande melhora na imagem. As imagens apresentadas na Figura 7-13 mostram a diferença entre as resoluções se estivéssemos vendo as imagens em tamanho real. Esta mesma experiência foi realizada na CAVERNA Digital, mostrando a influência da resolução.

Figura 7-13 – Efeito da resolução na CAVERNA Digital 640x480 X 1280x1024

Porém, deve ser lembrado que, a geração das imagens depende da taxa de atualização, resultando na taxa de pontos por segundo (*fill-rate*). Esta tem um fator limitante de ~2 Gigabits/segundo, que é o limite da transmissão do cobre. No caso de resoluções maiores, devem ser utilizadas diversas vias, como por exemplo, o padrão DVI, que possui dois canais, dobrando a capacidade de transmissão. Para o cálculo da taxa de quadros por segundo, deve ser utilizada a Equação 3, informando a quantidade de pontos na vertical e horizontal e a atualização deles.

$$\frac{\#PontosVertical * \#PontosHorizontal}{\Delta t} = taxa$$

Equação 3– Cálculo da taxa de pontos por segundo

²⁹ Imagem obtida em <http://www.lenna.org/>

7.3.3 Teste de dependência de rede

A rede pode ser um gargalo para o sistema, afetando o desempenho deste. Devido ao sincronismo de dados ser feito pela rede, este vai definir um ritmo. Em testes preliminares foi encontrado um teto médio de $46,7 \pm 0,1$ quadros por segundo.

Isto indica que não importa quão leve o modelo ou poderoso o cartão gráfico, a taxa de atualização de quadros não passará, na média, de 46,7 quadros por segundo. Este valor é totalmente aceitável, visto que o desempenho do sistemas de computação gráfica gira em torno de 30 quadros por segundo para aplicações tradicionais de computação gráfica.

7.3.4 Teste de estabilidade

Os sistemas de realidade virtual necessitam ser robustos o suficiente para se manterem estáveis durante suas operações. Os sistemas desenvolvidos se mostraram relativamente estáveis, contudo, para uma utilização no mercado, seria necessário um aperfeiçoamento do sistema para um melhor tratamento de erros, em especial para, arquivos X3D mal formatados.

A segurança também é importante. Pelos dados estarem circulando em uma rede comum, podem acontecer problemas devido a interferências externas. Uma alternativa é a utilização de uma rede independente para a comunicação interna do aglomerado, que conseqüentemente evitaria problemas de congestionamento da rede.

7.3.5 Teste do sistema com aplicações de terceiros

Outros sistemas de realidade virtual de outras instituições foram testados, contudo devido a não serem sistemas estáveis, ainda não foram feitos testes minuciosos de desempenho.

7.3.6 Teste de desempenho do sistema de gerenciamento

Uma das preocupações no sistema de gerenciamento é este ser rápido o suficiente para o usuário conseguir controlar o ambiente sem grandes atrasos.

Figura 7-14 – Teste de ping-pong para calcular latência da rede

A Figura 7-14 apresenta a latência das mensagens do sistema de gerenciamento composto pelo SIRIUS, ZEUS e *daemons*. Por exemplo a linha W-M-D2 significa um sistema de rede sem fio rodando o SIRIUS, a multiserial rodando o ZEUS e o computador número 2 rodando um *daemon*. É possível visualizar que a latência é linear no intervalo apresentado, onde as mensagens são pequenas o suficiente para executar o comando, permitindo um atraso tolerável para o usuário. Também é possível visualizar neste gráfico que o dispositivo sem fio é mais lento, contudo rápido o suficiente para o transportar as mensagens de gerenciamento.

7.4 Pesquisas em aglomerados de outros centros

Diversos centros de pesquisa já estão trabalhando com aglomerados gráficos para aplicações imersivas, alguns ainda em fase inicial, com poucos trabalhos publicados. Dentre os centros mais conhecidos temos O ISL da universidade de Illinois nos EUA, INRIA Rhône-alpes na França, Fraunhofer IAO na Alemanha, SKLof CAD&CG da China.

Figura 7-15 – Exemplos do ISL, INRIA e Fraunhofer

³⁰ <http://www.inrialpes.fr/sed/grimage/contacts.html>

³¹ <http://vr.iao.fhg.de/imodeling/index.en.php>

7.5 Conclusões

Neste capítulo tentou-se estabelecer uma análise comparativa entre as tecnologias disponíveis e os desenvolvimentos relacionados com esta pesquisa mostrando os pontos fortes e fracos do sistema proposto. Foi apresentada uma breve explicação da geração de imagens, e como poderemos aproveitar este sistema para criar imagens utilizando os possíveis paralelismos.

Os diversos trabalhos realizados, mostraram que ainda existem possibilidades de evoluir os sistemas desenvolvidos como um todo. Contudo já apresentam qualidade muito superior aos supercomputadores gráficos monolíticos, baseado nas comparações entre eles.

Finalizamos verificando que o uso de aglomerados em outros centros de pesquisa no mundo é um fato e deve ser observado com atenção.

8. CONCLUSÕES E TRABALHOS FUTUROS

8.1 Introdução

Sistemas de realidade virtual baseados em multiprojeção estereoscópica, utilizando o paradigma de CAVERNA Digital, são ferramentas importantes utilizadas em várias aplicações da engenharia e ciências. Este trabalho apresentou as várias decisões de projeto e inovações relacionadas com a implementação da primeira CAVERNA Digital da América Latina, tanto no aspecto das montagens físicas, eletrônicas e de aplicações de gerenciamento e desenvolvimento [ZUFFO02].

A utilização de aglomerados de computadores convencionais mostrou ser uma alternativa interessante do ponto de vista de potencial de investigação científica e tecnológica, bem como uma alternativa de baixo custo.

Atualmente várias aplicações imersivas da realidade virtual, relevantes a sociedade brasileira, encontram-se em desenvolvimento na CAVERNA Digital, desde sistemas industriais, como visualização de sistema de petróleo, até áreas de entretenimento.

8.2 Contribuições

De uma forma geral as expectativas do projeto foram alcançadas, provando sua viabilidade. Algumas mudanças do projeto original foram realizadas, contudo, os fundamentos científicos foram corretos, não sendo necessária sua alteração.

As propostas para os sistemas de comunicação e sincronização foram validadas, mostrando que a tecnologia de mercado (*commodity*) pode ser utilizada para estes sistemas como uma alternativa.

Diversos artigos e cursos em congressos nacionais e internacionais foram publicados, conseguindo cumprir o papel acadêmico do projeto, mostrando que a comunidade tem interesse e a pesquisa é relevante. Nos últimos anos pôde ser constatado, que diversas universidades estão apostando na tecnologia de aglomerado de computadores, muitas vezes recorrendo ao nosso laboratório para auxiliá-los.

Como forma de expor o conhecimento para a comunidade, documentos em diversas línguas estão publicados na página do autor www.lsi.usp.br/~lsoares, bem como na página do laboratório www.lsi.usp.br/interativos/nrv.

8.2.1 Trabalhos publicados

As pesquisas e implementações deste trabalho foram devidamente exploradas nos congressos da área, por meio de diversos cursos e publicações decorrentes das atividades de pesquisa, dentre eles podemos citar:

▪ Artigos

Soares, L. P. e Zuffo, M. K. 2004. JINX: an X3D browser for VR immersive simulation based on clusters of commodity computers. Em *Proceedings of the ninth international conference on 3D Web technology*, Monterey, California, EUA, 79 – 86

Soares, L. P., Nomura, L., Cabral, M. C., Dulley, L. P., Guimarães, M. P., Lopes, R. D. e Zuffo, M. K. 2004. Virtual Hang-gliding over Rio de Janeiro. Em *IEEE VR workshop VR for Public Consumption*, Chicago, Illinois, EUA

Soares, L. P., Cabral, M., Bressan, P., Fernandes, H., Lopes, R. e Zuffo, M. 2002. Managing commodity cluster oriented for virtual reality application, Em *Proceedings of the V symposium of Virtual Reality*, Fortaleza, Ceara, Brasil

Soares, L. P., Cabral, M. C., Bressan, P. A., Lopes, R. D. e Zuffo, M. K. 2002. Powering Multiprojection Immersive Environments, Em *Proceeding of SIACG 2002*, Guimarães, Portugal

Zuffo, J., Soares, L., Zuffo, M. e Lopes R. 2001. CAVERNA Digital - Sistema de Multiprojeção Estereoscópico Baseado em Aglomerados de PCs para Aplicações Imersivas em Realidade Virtual, Em *Proceedings of the IV symposium of Virtual Reality*, Florianópolis, Santa Catarina, Brasil

▪ Cursos

Soares, L. P. Zuffo, M. K. e Cardoso, A. 2004. Desenvolvendo AV com uso de X3D e VRML, Tutorial em VII Simpósio de Realidade Virtual, São Paulo, São Paulo, Brasil

Soares, L. P., Bressan, P. A., Guimarães, M. P., Cabral, M. C., Kurashima, C., Todesco, G. e Zuffo, M. K. 2003. Visualização de Alto Desempenho: utilizando componentes convencionais, Tutorial em VI Simpósio de Realidade Virtual, Ribeirão Preto, São Paulo, Brasil

Soares, L. P., Goudeseune, C., Kaczmarski, H., Schaeffer, B., Allard, J., Raffin, B., Cabral, M. C. e Zuffo, M. K. 2003. Commodity Clusters for Immersive Projection Environments, Tutorial em SIGGRAPH 03, San Diego, California, USA

Kaczmarski, H., Zuffo, M. K., Schaeffer, B., Goudeseune, C. e Soares, L. P. 2003. Introduction to Commodity Clusters for Multiprojector Displays, Tutorial em Workshop on Commodity Clusters for Virtual Reality, IEEE VR, San Diego, California, EUA

Zuffo, M., Soares, L., Bressan, P., e Paiva, M. 2002. Commodity Clusters for Immersive Environments, Tutorial em V Symposium of Virtual Reality, Fortaleza, Ceara, Brasil

Soares, L. P., Kaczmarski, H., Zuffo, M. K., Goudeseune, C., Schaeffer, B., Augerat, P., Raffin, B. e Bressan, P. A. 2002. Commodity Clusters for Immersive Projection Environments, Tutorial em SIGGRAPH 02, Sant Antonio, Texas, EUA

Soares, L. P., Stein, A., Lopes, R. D. e Zuffo, M. K. 2001. Projeto e Construção de Mundos Virtuais em CAVERNAs Digitais. Tutorial em IV Simpósio de Realidade Virtual, Florianópolis, Santa Catarina, Brasil

▪ **Mesa de Discussão**

Behr, J., Chittaro, L., Polys, N. and Soares, L. P. 2004. X3D User Interface. Em *Proceedings of the ninth international conference on 3D Web technology*, Monterey, California, EUA

Também foram produzidos capítulos de 4 livros. Além de diversos trabalhos práticos apresentados.

8.2.2 Inovações tecnológicas

A seguir serão apresentados as principais inovações tecnológicas desenvolvidas no âmbito desta tese:

▪ **A Família de VRClusters**

O desenvolvimento do conceito de VRCluster, bem como as suas implementações na forma de uma família foram pesquisas pioneiras no Brasil e em muitos pontos no mundo. Ao longo dos desenvolvimentos desta tese três VRClusters foram implementados pelo autor, nas configurações descritas abaixo:

VR Cluster I, integrado a partir de setembro de 2000

Processadores Pentium III Xeon, 1.2 GHz
1 GB de memória
Cartão gráfico 3DLabs Wildcat 4210

VR Cluster II, integrado a partir de Janeiro 2002.

Processadores Xeon, 2.1 GHz
1 GB de memória
Cartão gráfico 3DLabs Wildcat 5110

VR Cluster III (Perfect Cluster) integrado a partir de fevereiro de 2004

Processadores Pentium III Xeon, 3.2 GHz
2 GB de memória
Cartão gráfico Nvidia Quadro FX 3000G

- **CAVERNA Digital**

O desenvolvimento da CAVERNA Digital, primeiro sistema de multiprojeção cúbica da América Latina que viabilizou o desenvolvimento de outras diversas linhas de pesquisa, muitas fora do âmbito desta tese. A proposta de construção de uma CAVERNA de baixo custo também foi explorada, mostrando a possibilidade de utilizar-se tecnologias simples para a produção de ambientes imersivos. Este é um passo fundamental, que anda casado com o aglomerado, pois o objetivo é disponibilizar uma tecnologia facilmente acessível, tanto nos programas, como computadores e sistemas físicos de projeção.

- **O Sistema de Gerenciamento**

O sistema de gerenciamento também foi e ainda é algo inovador. Muitos centros de realidade virtual no mundo ainda usam técnicas rudimentares ou ferramentas impróprias para o gerenciamento do complexo de realidade virtual. Com o sistema de gerenciamento baseado na SIRIUS a utilização da CAVERNA Digital ficou dinâmica, permitindo uma fácil operação da mesma. O sistema supriu as necessidades fundamentais, e acabou sendo adotado por todos os usuários do laboratório

- **JINX**

O desenvolvimento de aplicações utilizando a JINX, também é algo que foi reformulado, apresentando paradigmas simples, baseados em XML e X3D. Desta forma, a produção de conteúdo ficou muito facilitada, permitindo mesmo que até usuários leigos, produzam conteúdos para ambientes imersivos. Porém, apesar das simplicidades, foi mantida toda a flexibilidade das ferramentas existentes, como, por exemplo, a opção de utilizar um sistema OpenGL diretamente, o que pode permitir ao pesquisador uma maior gama de opções.

8.2.3 A transferência de tecnologia para a Itaotec-Philco

A Itaotec, empresa nacional pioneira no desenvolvimento de tecnologia computacional, está desenvolvendo um projeto em conjunto com o Laboratório de Sistemas Integráveis, com o objetivo de disponibilizar um sistema completo e operacional de aglomerados numéricos e gráficos e alguns já foram comercializados para centros de pesquisa privados e públicos no Brasil. O protótipo do VRCluster da Itaotec apresentado na Condex 2003 está na Figura 8-1.

Figura 8-1 – Info Cluster Gráfico

Se pretende manter o VRCluster sempre atualizada com o que há de mais novo nos recursos de computação gráfica, mediante contrato de parceria entre o Laboratório de Sistemas Integráveis e a Itaotec. Documentação de instalação e operação está sendo desenvolvida, permitindo os centros de pesquisa criarem ou adaptarem suas aplicações, bem como utilizar as diversas ferramentas do sistema.

8.3 Trabalhos futuros

O desenvolvimento de um aglomerado de computadores para controlar sistemas de realidade virtual é um trabalho constante, pois novas tecnologias em ambas as áreas estão sempre surgindo.

O sistema SIRIUS de gerenciamento poderia embutir outras funcionalidades como, por exemplo, um leitor de estado dos nós do aglomerado de computadores, ou seja, descobrir se não estão trabalhando com problemas de tensão, se uma ventoinha quebrou, dentre outros.

Algo de fundamental interesse são as aplicações colaborativas entre sistemas de aglomerados para realidade virtual, ou seja um aglomerado, no Brasil, se comunicar com um aglomerado em outro país e estes compartilharem uma experiência virtual.

O sistema de desenvolvimento de aplicações para ambientes imersivos pode tirar proveito de paralelização por *sort-last*, desta forma o desempenho das aplicações seria maior ainda.

Uma das alternativas estudadas foi a construção de um aglomerado de baixo custo, ou seja, em vez de haver redundância nos nós, poderiam ser criadas estações mais robustas. Atualmente com a tecnologia PCI-Express isto pode ser uma realidade.

A CAVERNA Digital atual utiliza projetores analógicos, com o advento do padrão DVI, toda a comunicação poderá ser digital, reduzindo, assim, ruídos nos sinais e problemas de calibração.

O SIRIUS, como explicado, é uma aplicação Java, que pode ser portada para sistemas portáteis de mão (*palm-top*). Desta forma, com a utilização de algum sistema sem fio (*wireless*), o gerenciamento terá uma maior liberdade, podendo ser feito dentro da CAVERNA Digital.

A escolha do padrão X3D deve-se ao fato dele ter o propósito de ser um padrão universal, desta forma, diversos programas comerciais começariam a suportá-lo. Enquanto isso, a possibilidade de criar conversores é totalmente aceitável, visto que um simples programa que faça a análise léxica, já é capaz de converter um arquivo VRML para X3D. Para outros formatos poderão ser necessários sistemas mais evoluídos, contudo, fundamentais para a ampla utilização do sistema. Além disso, com este sistema poderá ser solucionado um dos graves problemas de aplicativos para realidade virtual: os dados após serem enviados para o sistema de realidade virtual não podem ser editados, de forma que suas alterações voltem ao programa de

origem. O objetivo é que estas alterações voltem da mesma forma que vieram para o sistema de realidade virtual.

O suporte a dispositivos é algo importante para os ambientes de realidade virtual, muitas vezes, queremos fazer utilizações específicas. Devido às limitações de equipamentos, não foi possível oferecer suporte a uma grande gama de sistemas de interatividade, contudo, a medida da disponibilidade do equipamento, eles estão sendo implementados.

8.4 Perspectivas futuras

A utilização do VRCluster e da CAVERNA Digital tem a possibilidade de se tornarem tecnologias muito exploradas comercialmente. Devido ao custo de aglomerados não ser tão elevado, este pode se tornar um produto doméstico, da mesma forma que os computadores estão nas residências de uma grande parcela da população.

Se o custo para os sistemas de projeção se tornarem mais acessíveis, toda a tecnologia apresentada tem um grande potencial de se tornar um eletrodoméstico, como está acontecendo no caso da Itautec.

A formalização de padrões é algo que começará a se concretizar, durante o congresso do Web3D 2004 e 2005, foram sugeridas expansões no padrão X3D para ambientes imersivos, mediante uma mesa de discussão.

ANEXOS

ANEXO I – Componentes da CAVERNA Digital

Existem várias listas de discussão sobre sistemas cúbicos de projeção na Internet. Um dos grupos mais importantes é o CAVERNUS (*CAVE Research Network Users Society*). O CAVERNUS é uma aliança entre a indústria e instituições de pesquisa, que fica sediada no NCSA (*National Centre for Supercomputing Applications*), que além de promover um mecanismo de interação entre diversos pesquisadores, também organiza várias atividades como seminários, bibliotecas de programação, exemplos de aplicações e demonstrações.

Uma ramificação do NCSA propôs um sistema distribuído de realidade virtual chamado Caterpillar's, que pretende comunicar vários pontos de pesquisa ao redor do mundo, todos interconectados em redes ATM [NCSA03].

Telas

As imagens das quatro paredes laterais são projetadas pela parte traseira, ou seja, oposta ao usuário, em películas de material plástico polímero branco, e o piso recebe uma projeção direta pela parte superior, como é apresentado na Figura I - 1.

Figura I - 1 – Telas

Devido a projeção ser feita pela parte de trás da tela, esta deve apresentar um material de qualidade óptica elevada, evitando-se que a luz atravesse diretamente e crie o efeito indesejável de cintilação ou estouro de luz (*hot spot*). Esta tela não deve modificar as propriedades da projeção, como cor ou contraste, nem refletir a luz. Caso se utilize um sistema de estereoscopia passiva, a tela não deve despolarizar a luz, no caso da CAVERNA Digital implementada a tela não possui esta característica, pois somente se deseja utilizar um sistema de projeção baseado em estereoscopia ativa e o custo de telas para projeção ativa são bem superiores.

Espelhos

Para as projeções das imagens das três paredes fixas são utilizados espelhos de cristal com alto índice de reflexão, como na Figura I - 2, onde esta reflexão é feita pela superfície e não a parte interna, o que reduz, significativamente, a reflexão dupla encontrada em espelhos domésticos. A projeção da porta é realizada

diretamente, sem a necessidade de espelho, pois existe uma maior disponibilidade de espaço, a projeção do piso é feita com o auxílio de um espelho de película plástica. Este material foi escolhido devido ao risco do espelho ser muito pesado e cair da parte superior no usuário [SCREEN00]. Estes espelhos também possuem uma proteção anti-estática que inibi a poeira de aderir no espelho, o que com o tempo comprometeria a qualidade visual.

Rastreadores e leitores de posição

Rastreadores são dispositivos que permitem ao usuário interagir com o ambiente, de uma forma simples e eficiente, determinando a posição e orientação de determinados sensores (normalmente fixos ao usuário), para tarefas, como manipulação do ambiente, cálculo de projeção e som. Um dos sistemas mais utilizados é o eletromagnético, apresentado na Figura I - 3. Este dispositivo capta o sinal de um emissor de ondas eletromagnéticas e calcula sua posição relativa.

Este equipamento necessita de um ambiente livre de grandes volumes de metal, que eventualmente poderiam causar interferência e conseqüentemente induzir imprecisões nas medidas de posição e orientação. Estes sensores também podem possuir botões, que são programados para incorporar uma determinada função, que geralmente é utilizado para selecionar e mover objetos presentes na cena virtual.

Figura I - 3 – Rastreador eletro-magnético

Sistema de som

O áudio é um fator determinante para a indução no usuário da sensação de imersão. Como sistema de saída existem duas soluções: a primeira é gerar o som em caixas acústicas amplificadas, como apresentado na Figura I - 4, espalhadas estrategicamente na CAVERNA Digital. Outra

Figura I - 4 – Caixa acústica

possibilidade é a utilização de fones de ouvido sem fio, gerando o som baseado na posição do observador, que é conhecido através do sistema de rastreamento.

O sistema de áudio apresenta muitas similaridades ao sistema de vídeo. Também existe a necessidade de se comutar o sinal dentre os vários sistemas contudo somente se necessita de uma mesa de som para se escolher os diversos canais de entrada e as respectivas saídas. Existem sistemas evoluídos para uso de áudio imersivo, para maiores detalhes, consultar apêndice B.

Comutador de vídeo

Este é um dispositivo muito comum em emissoras de TV. O comutador de vídeo (*video matrix switcher*) tem a finalidade de permitir um mapeamento de entradas de vídeo em diversas saídas, que é visto na Figura I - 5. Existe um circuito interno que redireciona os sinais das entradas e o disponibiliza em outras saídas, amplificando o sinal. Estes comutadores de vídeo existem para formatos digitais ou analógicos, no sistema apresentado foi utilizado o sistema analógico. Estes sistemas têm de apresentar um grande banda de transmissão, pois os sinais podem ser complexos.

Devido a disponibilidade de inúmeras fontes de sinal de vídeo do aglomerado gráfico, existe a necessidade de se comutar as entradas dos projetores para o servidor gráfico utilizado. Esta necessidade impôs o uso de um comutador de vídeo. Rapidamente pode-se alterar uma configuração com ou sem intervenção física (controlada remotamente por uma porta serial comum). A CAVERNA Digital utiliza um sistema de projeção analógico, baseado no padrão de sinal RGBHV (três canais de cor “R” vermelho, “G” verde e “B” azul e dois canais de sincronismo “H” horizontal e “V” vertical).

Figura I - 5 – Diagrama de utilização de comutador de vídeo

Projetores

Os projetores são uma peça muito importante para a qualidade de imagem, pois conforme o modelo de projetor é possível obter diferentes resoluções. Atualmente já é disponível projetores lasers da ordem de 5000x4000 pixels, chamados de limitado pela vista (*eye-limiting*), onde a olho nu não seria possível enxergar um pixel.

Figura I - 6 – Projetor

Para o sistema de projeção da CAVERNA Digital foram escolhidos projetores CRT (*cathode ray tube*), exibidos na Figura I - 6. A principal escolha deste tipo de projetores deve-se ao fato deles suportarem altas resoluções a altas frequências, necessária para estereoscopia ativa. Contudo apresentam uma luminosidade baixa.

As outras opções de projetores são os sistemas LCD e DLP e estes apresentam um melhor brilho, contudo os modelos convencionais não suportam frequências elevadas o suficiente para permitir um sistema de projeção estereoscópica ativo. Porém, projetores DLP mais avançados permitem a utilização destas tecnologias, suportando tanto sinais digitais quanto analógicos, porém o preço é ainda muito elevado. Para maiores detalhes de projetores DLP consultar apêndice L.

Para o gerenciamento dos projetores da CAVERNA Digital, além de controle remotos com e sem fio, os projetores possuem interfaces de comunicação serial, que

permitem o controle remoto por computador. Permitindo diversas funcionalidade como que os projetores sejam ligados e desligados remotamente, definir qual calibragem usar, dentre outras funcionalidades. Este último em especial é importante, pois permite que, conforme a origem do sinal de vídeo que esteja se usando, se auto ajuste o projetor para fazer com que a imagem se encaixe perfeitamente no sistema utilizado, no caso deste trabalho uma CAVERNA Digital. Para detalhes de calibragem consultar apêndice I.

Considerando-se faces com dimensões de 3m x 3m, a resolução mínima recomendada para cada face é de 1024 x 1024 pontos, estabelecendo um tamanho dos pontos na tela (*pixel*) da ordem de 3mm x 3mm. Os projetores utilizados suportam resoluções da ordem de 2500 x 2000 pontos, contudo os computadores ainda não são capazes de lidar com este sinal de vídeo.

Figura I - 7 – Simulação da CAVERNA Digital

Proteção acústica

Devido a sala que acomoda a CAVERNA Digital ser muito fechada e alta, problemas de ecos poderiam ocorrer. Desta forma houve um tratamento acústico nas paredes, conforme a

Figura I - 8, com uma espuma de celulose. Esta espuma absorve os ruídos, bem como possui uma cor preta e rugosidades que diminui um possível reflexão luminosa no ambiente. O telhado possui um forro em lã de vidro que também absorve ruídos. Porém estudos mostraram que seria necessário uma espessura muito maior do que a que foi utilizada para eliminar de forma mais efetiva ruídos do ambiente externo.

Figura I - 8 – Proteção acústica

Subsistema de iluminação

O sistema de iluminação foi projetado com a ajuda de arquitetos especializados para contornar a dificuldade da vista humana de se adaptar rapidamente a ambientes escuros. Todos os componentes eletrônicos são digitais para permitir um fácil sistema de desenvolvimento de circuitos de controle. Os controle são apresentado na Figura I - 9.

Figura I - 9 – Controle de luz

Refrigeração

Os espelhos são utilizados para minimizar a área de bastidores e o volume de ar-refrigerado, o tratamento na parede também permite uma melhor estabilidade térmica. Para não comprometer as características de isolamento acústico foi necessário optar-se por sistemas de ar-condicionado de alto volume (60.000 BTUs) com baixa emissão sonora, como visto na Figura I - 10.

Figura I - 10 – Ar condicionado

Óculos estereoscópicos

Os óculos estereoscópicos são necessários para fornecer a sensação de profundidade ao usuário. Caso se use óculos estereoscópicos ativos como na Figura I - 11, é necessário enviar um sinal sincronizado com o computador, até o óculos, por infra vermelho, a opção por cabo poderia interferir no ambiente. Cada sistema de computadores com capacidades de estereoscopia, no caso o aglomerado de computadores, possui um sinal de sincronismo.

32

Figura I - 11– Óculos estereoscópico ativo

Para solucionar esta questão, um comutador de vídeo é uma alternativa prática. Um ponto prático de se usar o comutador, é que estes são totalmente configuráveis por comandos seriais, o que permite o controle do sinal estereoscópico.

³² Imagem obtida em <http://www.stereographics.com/>

ANEXO II – Diagrama UML do sistema SIRIUS

A seguir é apresentado o diagrama UML do sistema SIRIUS, desenvolvido em Java.

Figura II - 1 – Diagrama de responsabilidades

Figura II - 2 – Diagrama de classes

Figura II - 3 – Diagrama de hierarquia

ANEXO III – Detalhes de implementação da JINX

IDF0

Diagrama de classes

Diagrama Principal

Entrada de Dispositivos

Comunicação

Configuração

Preparação de cena

Mundo Virtual

X3D Geometry3D

X3D Grouping component

Parser

Diagrama de sequência

Abrindo aplicação

Colisão

Síntese de Imagem e Som

Interação

Componentização

ANEXO IV – Arquivo de configuração da JINX

- DTD:

```
<!DOCTYPE jinx [  
<!ELEMENT jinx (computer+)>  
<!ELEMENT compute (window+)>  
<!ELEMENT window  
    (wand?,view?,size?,projection?,position?,screen?,frustum?)>  
<!ATTLIST computer name CDATA #REQUIRED>  
<!ATTLIST window id CDATA #IMPLIED>  
<!ATTLIST wand mode CDATA  
    (spaceball|intertrax|dataglove|flockofbirds) "spaceball">  
<!ATTLIST view mode CDATA (mono|stereo) "mono">  
<!ATTLIST size width CDATA "640">  
<!ATTLIST size height CDATA "480">  
<!ATTLIST size fullscreen CDATA (TRUE|FALSE) "TRUE">  
<!ATTLIST projection mode CDATA (planar|spherical) "planar">  
<!ATTLIST position x CDATA "0">  
<!ATTLIST position y CDATA "0">  
<!ATTLIST screen dir CDATA "0 0 -1">  
<!ATTLIST screen up CDATA "0 1 0">  
<!ATTLIST frustum left CDATA "-1">  
<!ATTLIST frustum right CDATA "1">  
<!ATTLIST frustum bottom CDATA "-1">  
<!ATTLIST frustum top CDATA "1">  
>]
```

- Exemplo:

```
<jinx>  
  <computer name="node1.midianet.intranet">  
    <window id="left">  
      <wand mode="intertrax"/>  
      <view mode="stereo"/>  
      <size width="1280" height="1024" fullscreen="TRUE"/>  
      <projection mode="planar"/>  
      <position x="0" y="0"/>  
      <screen dir="-1 0 0" up="0 1 0"/>  
      <frustum left="-1" right="1" bottom="-1" top="1"/>  
    </window>  
  </computer>  
  <computer name="node2.midianet.intranet">  
    <window id="front">  
      <size width="1280" height="1024" fullscreen="TRUE"/>  
      <projection mode="spherical"/>  
      <view mode="mono"/>  
      <screen dir="0 0 -1" up="0 1 0"/>  
      <frustum left="-1" right="1" bottom="-1" top="1"/>  
    </window>  
  </computer>  
</jinx>
```

ANEXO V - Reconhecimento de voz

A aplicação utilizada é o sistema da Nuance que, através de uma comparação com o Via-Voice da IBM, mostrou-se mais robusto e com maior escalabilidade.

Para o grau de reconhecimento da voz, existe um percentual de reconhecimento do quanto o som capturado é parecido com o do dicionário interno. Adota-se um reconhecimento de 45% como sendo capaz de executar uma ação. Utiliza-se o Inglês como língua padrão, pois o programa permite que a leitura de variáveis seja feita apenas em inglês. Futuramente se deseja implementar também em português.

A construção da gramática completa é de fácil execução, pois tendo comprovado a eficiência desta, o trabalho torna-se puramente mecânico.

Além desta gramática, temos também um arquivo vxml, onde o diálogo é feito. Assim, já testamos o diálogo onde podemos ter uma resposta da CAVERNA Digital, por exemplo:

Cave: What is your command?

User: Projectors on all.

Cave: All the projectors are on.

LISTA DE REFERÊNCIAS

- [AKELEY01] Akeley, K. e Hanrahan, P. 2001. Real-Time Graphics Architectures, Na aula CS448A <http://graphics.stanford.edu/courses/cs448a-01-fall/lectures/lecture1> 32
- [ALLARD02] Allard, J., Gouranton, V., Lecointre, L. e Raffin, B. 2002. Getting Started with NetJuggler and SoftGenLock. VR Juggler Curso no *IEEE VR*, Orlando, Florida 16, 141
- [AVANGO03] AVANGO, <http://www.avango.org/> 27
- [BATTAIOLA98] Battaiola, A. L. e Soares, L. P. 1998. Estudo e Uso Exploratório de Ferramentas de Visualização Científica, *VII Semana de Informática da UFBA*, pp. 16-30, Salvador 84
- [BEHR04] Behr, J. Dähne, P. e Roth, M. 2004. Utilizing X3D for immersive environments, Em *Proceedings of the ninth international conference on 3D Web technology*, Monterey, California, pp. 71–78 28
- [BERGEN03] Bergen, <http://www.evl.uic.edu/pape/sw/bergen/> 132
- [BERNAL99] Bernal, V. B., Kofuji, S. T., Sipahi, G. M. e Anderson, A. G. 1999. PAD Cluster: an open, modular and low cost high performance computing system. Em *Proceedings of SBAC'99*, p215-22. 11
- [BHATT02] Bhatt, A. 2002. PCI Express, Architectural Overview, Livro *Intel - IDF Fall 2002*, pp. 9-12 Setembro, <http://www.intel.com/idf/us/fall2002/presentation/OVR174PS.pdf>. 128
- [BOURKE02] Bourke, P. 3D Stereo Rendering Using OpenGL (and GLUT), <http://astronomy.swin.edu.au/~pbourke/opengl/stereogl/> 68
- [BOURKE99] Bourke, P. Calculating Stereo Pairs, <http://astronomy.swin.edu.au/~pbourke/stereographics/stereorender/> 68
- [BRESSAN04] Bressan, P. A. Visualização Volumétrica aplicada em Aglomerados de Computadores. Tese de Doutorado. Escola Politécnica da Universidade de São Paulo, São Paulo, 2004. 9
- [BUES03] Bues, M., Hinkenjann, A., Olry, T., Schupp, S. 2003.

- Attacking the Network Bottleneck of Parallel Rendering with Commodity Hardware.. Em *Proceedings of 6th Symposium on Virtual Reality*, Ribeiro Preto, Brazil **23**
- [CABRAL05] Cabral, M. C. Interação baseada em gestos de mãos e cabeça para ambientes de realidade virtual utilizando visão computacional Dissertação de Mestrado. Instituto de Matemática e Estatística da Universidade de São Paulo. São Paulo. 2005. **9**
- [CAMERON02] Cameron D. e Regnier G. 2002. The Virtual Interface Architecture, Intel Press, **137**
- [CHALMERS02] Chalmers, A., Davis, T. e Reinhard, E. 2002. Pratical Parallel Rendering, A K Peters, Canadá, www.awpeters.com **22**
- [CLUSTER01] Cluster Conference, University of Utah, Fevereiro 2001. <http://www.chpc.utah.edu/cc/cc2001/talks/hargrove/hargrove.pdf> **22**
- [COFFIN03] Coffin, T., Matisse J., Jass, R., Kaspes, U., Leigh, J, Thompson, F., Venkataraman, S. e Weilinga, P.;Kites. 2003. Flying In and Out of Space, No *Workshop: Commodity Cluster for Virtual Reality IEEE VR2003*, Los Angeles; California, USA **13**
- [CRUZ92] Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V. e Hart, J. C. 1992. The Cave Áudio Visual Experience Automatic Virtual Environment. Em *Communications of the ACM Siggraph 92 Showcase*, Vol.35 N6. pp 64-72. **3, 41, 69, 95**
- [CRUZ93] Cruz-Neira, C.; D.J. Sandin; T.A. DeFanti. Surround-screen projection-based virtual reality: The design and Implementation of the CAVE. In: SIGGRAPH 1993. ACM SIGGRAPH, Anaheim, Julho 1993. **3, 3, 41**
- [CUNNINGHAN03] Cunningham, S., 2003. Computer Graphics Programming, Problem Solving and Visual Communication, Notas de curso no *1º Seminário Latino Americano de Educação em Computação Gráfica*. Escola Politécnica, Universidade de

- São Paulo, São Paulo, Brasil. 86
- [CVS] CVS <https://www.cvshome.org/> 8
- [CYCLADES01] Console Port Management August 2001 Cyclades Corporation . 46, 54
- [DAI97] Dai, P., Eckel G., Göbel, M., Hasenbrink, F., Lalioti, V., Lechner, U., Strassner, J., Tramberend, H. e Wesche, G. 1997. Virtual Spaces: VR Projection System Technologies and Applications, Curso em *Eurographics '97*, Budapest. 27
- [DISZ97] Disz, T. 1997. Introduction - The CAVE: family of Virtual Reality devices. Argonne National Laboratory. Em *Proceedings of the USENIX Windows NT Workshop*, Seattle, Washington, EUA 43
- [DIVERSE03] Diverse, <http://www.diverse.vt.edu> 28
- [DVI99] Digital Visual Interface Specification; 1999 145
- [ELDRIDGE01] Eldridge M. 2001 Designing Graphics Architectures Around Scalability and Communication, Ph.D. dissertattion, Stanford University 22
- [ELUMENS03] Elumens Corporation, <http://www.elumens.com> 42
- [FMOD03] Fmod, <http://www.fmod.org> 132
- [GENLOCK03] Genlock, <http://www.mivs.com/technical/appnotes/an005.html> 16
- [GNECCO01] Gnecco, B. B., Bressan, P. A., Lopes; R. D., Zuffo, M. K. 2001. DICELIB: A Real Time Synchronization Library for Multi-projection Virtual Reality Distributed Environments. Em *Proceedings of the 4th SBC Symposium on Virtual Reality*. pp 338-343. Florianópolis, Santa Catarina, Brasil 28
- [GNECCO05] Gnecco, B. B. Geração interativa de imagens por traçado de raios, Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo. São Paulo. 2005. No prelo. 9
- [GREENE86] Greene, N. 1986. Applications of World Projections, Em *Proceedings of Graphics Interface*. pp 108-116. 42
- [GUIMARÃES04] Guimarães, M. P. Um ambiente para o desenvolvimento de aplicações de Realidade Virtual baseadas em aglomerados

- gráficos. Tese de Doutorado. Escola Politécnica da Universidade de São Paulo. São Paulo. 2004. 9
- [HEANEY01] Heaney, S. Beowulf: A New Verse Translation, Publisher: W.W. Norton & Company; Reprint edition 11
- [HINKENJAN02] Hinkenjan, A., Bues, M., Schupp, S e Olry, T. 2002. Mixed-Mode Parallel Real-Time Rendering on Commodity Hardware; Em *Proceedings of V Symposium on Virtual Reality*. Fortaleza. Ceará. Brasil. pp. 169-178. 27
- [HUNT98] Hunt C. TCP/IP Network Administration, Seconf Edition, 1998, O'Reilly & Associates, Inc. 135
- [IPMI02] Enabling Intelligent Plataform Management Interface (IPMI) Through Standard Building Blocks, Intel Corporation; Folder; EUA 53
- [KELSO02] Kelso, J., Arsenault, L., Satterfield, S. e Kriz, R. 2002. DIVERSE: A Framework for Building Extensible and Reconfigurable Device independent Virtual Environments 28
- [KICKSTART03] Kicjstart. <http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ch-kickstart2.html> 52
- [LEWIS95] Lewis B. e Berg., D. J. 1995. Threads Primer: A Guide to Multithreaded Programming. Prentice Hall 33
- [MATSUBA96] Matsuba, S. e Roehl, B. Special Edition Using VRML, 1996, Que Corporation. 35
- [MCLAUGHLIN01] McLaughlin, B. Java & XML, Second Edition ,Agosto 2001, O'Reilly & Associates, Inc. 34
- [MOORE65] Moore, G. E. 1965. Cramminh more components onto integrated circuits; Revista *Eletronics*, Vol. 38, N. 8, <ftp://download.intel.com/research/silicon/moorespaper.pdf> 1
- [NCSA03] NCSA, Caterpillar's Distributed Virtual Reality Project, <http://www.ncsa.uiuc.edu/VEG/DVR/> 107
- [ONYX04] Onyx <http://www.sgi.com/products/visualization/onyx4/> 3
- [OPENMP03] OpenMP. <http://www.openmp.org/> 33
- [ORAD] Orad. <http://www.orad.tv/> 126

- [OSCAR03] Oscar. <http://oscar.sourceforge.net/> 53
- [PACHECO97] Pacheco, P. S. 1997. *Parallel Programming with MPI*, Morgan Kaufmann Publishers, San Francisco, California, EUA 24
- [PARK00] Park K., Cho Y.J., Krishnaprasad N., Scharver C., Lewis M., Leigh J. e Johnson A. 2000. CAVERNsoft G2: a Toolkit for High Performance Tele-Immersive Collaboration, Em *Proceedings of ACM 7th Annual Symposium on Virtual Reality Software & Technology (VRST)*, Seoul, Korea 24
- [RAFFIN03] Raffin, B.; Zuffo, M. K., Kaczmarek, H., Pan, Z., Schaeffer, B., Goudeseune C. e Soares, L. P. 2003. Introduction to Commodity Clusters for Multiprojector Displays. Em *Workshop Commodity Clusters for Virtual Reality em IEEE VR2003*, Los Angeles, California, USA. 37, 40
- [RESCHKE96] Reschke, C. Sterling, T., Ridge, D., Becker, D. Merkey, P. e Savarese, D. 1996. A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation, Em *Proceedings of the High Performance Distributed*. pp626 11
- [ROHLF94] Rohlf, J.; Helman, J. Iris Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics, SIGGRAPH'94, Orlando, Florida, 1994 pp.381-394. 131
- [SCHAEFFER03] Schaeffer, B. e Goudeseune, C. 2003. Syzygy: native PC Cluster VR, Em *Proceedings of IEEE Virtual Reality Conference* 27
- [SCHMIDT00] Schmidt, D. C. Real-time CORBA with TAO (The ACE ORB), <http://www.cs.wustl.edu/~schmidt/TAO.html>. 134
- [SCRAMNET04] Scramnet, <http://www.systran.com/scmain.html> 3
- [SCREEN00] Screen Tech, 3D Projection Screens. <http://www.screen-tech.de> 108
- [SGI03] SGI, Iris Performer. <http://www.sgi.com/software/performer/> 131
- [SINGHAL99] Singhal, S. e Zyda, M. 1999. *Networked Virtual Environments: Design and Implementation*, Addison Wesley 15

- [SOARES02a] Soares, L. P., Cabral, M. C., Bressan, P. A., Lopes, R. D. e Zuffo, M. K. 2002. Powering Multiprojection Immersive Environments with Cluster of Commodity Computers, Em *Proceedings of 1º SIACG*, Guimarães, Portugal 45
- [SOARES02b] Soares, L. P., Cabral, M. C., Bressan, P. A., Fernandes, H., Lopes, R. D. e Zuffo, M. K. 2002. Management of Commodity Computer Cluster Oriented to Virtual Reality Applications. Em *Proceedings of V Symposium on Virtual Reality*. Fortaleza, Ceará, Brasil. pp 158-168. 56, 40
- [SOFTGENLOCK03] Softgenlock,
<http://netjuggler.sourceforge.net/Download/softgenlock.pdf> 142
- [STOLL01] Stoll, G., Eldridge, M., Patterson, D., Webb, A., Berman, S., Levy, R., Caywood, C., Taveira, M., Hunt, S. e Hanrahan, P. 2001. Lightning-2: A High-Performance Display Subsystem for PC Cluster, Em *proceedings of Siggraph* 23
- [SUTHERLAND68] Sutherland, I. E. 1968. A Head Mounted Three Dimensional Display, Em *Proceedings of the AFIPS Fall Joint Computer Conferences*, Washington DC, Thompson Books, pp 757-764 41
- [TAYLOR03] Taylor, R. M., Ware, C., Interrante, V. 2003. Perceptually-Based Visualization Design, Em *Curso of Siggraph 2003* 42
- [TOP500] Top500 <http://www.top500.org/>. 12
- [UML] UML <http://www.omg.org/technology/documents/formal/uml.htm> 8
- [VIDEOSYSTEMS] Videosystems
http://videosystems.com/ar/video_dmd_field_reliability/ 155
- [VRCO03] VRCO, <http://www.vrco.com/> 26
- [VRJUGGLER03] VRJuggler,
<http://www.vrjuggler.org/html/programmer.reference/classes/> 27
- [VSS03] VSS, <http://www.isl.uiuc.edu/Software/software.htm> 131
- [WRIGHTJR96] Wright Jr., R. S., Sweet, M. OpenGL SuperBible, 1996, Waite Group Press. 130

- [X3D03] X3D, <http://www.web3d.org/x3d.html> 35
- [X3DFAQ04] Faq X3D, <http://www.lsi.usp.br/~lsoares/x3d/faq.html> 35
- [ZUFFO01] Zuffo, M. K., Soares, L. P. e Stein, A. Projeto e Construção de Mundos Virtuais em CAVERNAS Digitais, Mini Curso em *IV Symposium on Virtual Reality* 41
- [ZUFFO02] Zuffo M. K., Soares, L. P., Bressan, P. A. e Guimarães, M. P. 2002. Commodity Cluster for Immersive Environments, Fortaleza ; Ceará, Brasil, Curso em *V Symposium on Virtual Reality*. 100, 40
- [ZUFFO98] Zuffo, M. K., Lopes, R. D. e Bernal, V. B. 1998. A Distributed Shared Memory System Oriented to Volume Visualization. Em *Eurographics Workshop on Parallel Graphics and Visualization*, Rennes, pp. 155-169 48
- [ZUFFOJ01] Zuffo, J. A., Soares, L. P., Zuffo, M. K. e Lopes, R. D. 2001.CAVERNA Digital - Sistema de Multiprojeção Estereoscópico Baseado em Aglomerados de PCs para Aplicações Imersivas em Realidade Virtual, Em *Proceedings of IV Symposium on Virtual Reality* 2

APÊNDICES

APÊNDICE A - Bibliotecas gráficas

- Direct 3D

O Direct 3D é um sistema desenvolvido pela Microsoft para aplicações 3D em Windows. O Direct 3D tenta superar as dificuldades de acesso das aplicações aos recursos de um computador operando em Windows.

Devido a sua limitação de operar somente em sistemas Windows, sua utilização não se torna prática, pois se limita a apenas uma plataforma de sistema operacional. Também apresenta sérias dificuldades para a construção de programas, devido a ser uma biblioteca de desenvolvimento, com uma organização interna não bem definida.

- OpenGL

O OpenGL [WRIGHTJR96] é a biblioteca de desenvolvimento de aplicações gráficas mais utilizada atualmente. Muitos desenvolvedores têm investido nesta biblioteca de desenvolvimento, devido a sua grande portabilidade e flexibilidade. Novas versões estão surgindo permitindo um melhor uso dos recursos. Ele é encontrado em diversas plataformas, o que permite a utilização dessa tecnologia em diversos tipos de computadores, desde poderosas estações gráficas, até *video games*.

Devido a ser uma biblioteca de alto desempenho, o OpenGL possibilita a execução de diversas instruções a altas taxas. Além disso, muitas empresas disponibilizam cartões gráficos capazes de interpretar instruções OpenGL. Desta forma, os comandos são rapidamente executados, aumentando a velocidade do processamento. Porém estas instruções podem ser processadas por um processador normal, permitindo um sistema flexível o suficiente, para adaptar-se a diversos sistemas de computação gráfica.

O OpenGL permite recursos avançados em diversos campos, como controle de deformações de câmera, o que permite a utilização da estereoscopia de forma correta. Para o desenvolvimento de aplicações OpenGL, existem diversas bibliotecas

auxiliares, permitindo o desenvolvimento de programas mais complexos, em menor tempo.

Um dos inconvenientes do OpenGL, é não possuir grafo de cena, tornando-se necessário implementá-lo quando necessário.

- Iris Performer

O Iris Performer, ou simplesmente Performer, é uma biblioteca gráfica de alto desempenho, desenvolvida pela Silicon Graphics™, disponível nas plataformas IRIX, Linux e mais recentemente em Windows. O Performer foi desenvolvido para sistemas de tempo real e utiliza amplamente multiprocessamento. O Performer oferece também acesso a uma série de recursos e efeitos que podem ser produzidos pela arquitetura gráfica da Silicon Graphics™, e que não são facilmente utilizáveis e gerenciáveis por outras plataformas de desenvolvimento de baixo nível.

O Performer é programável, tanto em ANSI C, como C++, e pode ser utilizado em conjunto com as bibliotecas OpenGL e IrisGL.

O Performer possui um formato próprio para armazenamento de dados que melhora o desempenho de leitura e gravação. Suporta também a leitura de vários formatos diferentes. Contudo estes formatos estão obsoletos, não sendo, portanto, muito úteis.

Embora o Performer tenha um potencial muito bom para o desenvolvimento de aplicações para a CAVERNA Digital, ainda exige do usuário um alto grau de conhecimento de técnicas computacionais [SGI03] [ROHLF94].

APÊNDICE B - Sistema de som

- VSS

O VSS (*Virtual Sound Server* ou *Vanilla Sound Server*) é uma biblioteca criada para complementar com recursos sonoros, aplicações interativas, desenvolvidas em C ou C++ [VSS03]. O VSS opera nas plataformas IRIX, Linux e Windows.

Com o VSS é possível, controlar, coordenar e sincronizar sons sintetizados diretamente e reproduções de sons. Para se utilizar o VSS é necessário se criar um servidor que tocará as músicas. Depois do servidor ser criado, é estabelecida uma conexão entre o cliente e o servidor, e o cliente pode enviar comandos para a geração de sons.

O controle e comunicação são efetuados por intermédio de atores, estes atores podem controlar diversos sons ao mesmo tempo. A comunicação é feita diretamente por IP, esta é especificada em linha de comando.

- Bergen Sound Server & Library

A biblioteca Bergen [BERGEN03] foi desenvolvida para ser uma versão simplificada do VSS distribuída gratuitamente, assim como o VSS, existem o servidor e o cliente. Uma versão para Linux foi disponibilizada recentemente.

- Fmod

O Fmod [FMOD03] é uma biblioteca de desenvolvimento para aplicações sonoras muito poderosa e é inteiramente gratuita. Com ela é possível sintetizar sons com um baixo consumo de processamento, além de um baixo consumo de memória. Uma de suas principais vantagens é operar em diversas plataformas, dentre elas Windows, Windows CE, Linux, Macintosh, Play Station 2 e o Xbox, porém infelizmente esta ainda não opera em plataforma Silicon Graphics™. A documentação da biblioteca também é completa permitindo desenvolver sistemas com facilidade.

Como informado estes sistema opera em diversas plataformas o que facilita muito sua portabilidade. Podem ser utilizados computadores com até 16 placas de som ou até, usar dispositivos como videogames, para estes fins. Conforme a arquitetura, esta pode ser mais explorada, como por exemplo, o sistema MMX que pode aumentar a performance e qualidade da aplicação em até 2 ou 3 vezes.

Um outro ponto importante é a possibilidade da leitura de diversos formatos de áudio, por exemplo, é possível abrir arquivos nos formatos: AIFF, Oss, Ogg Vorbis,

MP2, MP3, Midi, ou até mesmo trilhas de áudio de arquivos AVI. Além de operações simples como tocar músicas de diversos CDs ao mesmo tempo.

Uma das principais vantagens do fmod para os sistemas que estão sendo desenvolvidos, é sua capacidade de executar cálculos sonoros em uma simulação em terceira dimensão, com uma excelente performance, além de todo o suporte ao som estéreo. Alguns efeitos como o Doppler podem ser calculados, com uma grande precisão física.

- VRJuggler Sound Library

O VRJuggler também possui uma biblioteca de desenvolvimento, exclusiva para sons, além de suportar os recursos de som da biblioteca Performer. Sua utilização é de relativa complexidade.

O grande inconveniente deste sistema é ela estar presa à biblioteca VRJuggler, assim sua utilização fica limitada.

APÊNDICE C - Ferramentas de comunicação

- *Socket*

Este é o padrão mais simples de comunicação, utilizado em diversas plataformas de computadores. Neste, é aberto um canal de comunicação entre dois ou mais computadores, e mensagens são enviadas de um lado para outro. Este sistema é altamente eficiente, contudo, devido a não possuir um protocolo refinado, a transferência de certos tipos de dados fica comprometida a uma programação mais avançada. Uma das grandes vantagens do *socket* é a flexibilidade, porém ele está sujeito aos mesmos tipos de problemas de outros meios de comunicação, como, por exemplo, erros de sincronismo.

- CORBA

O CORBA (*Common Object Request Broker Architecture*) é o resultado do consórcio chamado OMG (*Object Management Group*), incluindo, mais de 800

empresas. O objetivo principal do CORBA é unificar as aplicações de transferência de dados, utilizando-se de uma linguagem orientada a objetos.

O CORBA é um sistema independente de arquitetura, sistema operacional e linguagem de programação. A principal vantagem dele é sua capacidade de camuflar as peculiaridades das camadas mais baixas de programação de rede para o usuário final.

O CORBA divide-se, basicamente, em três partes. A primeira seria o ORB (Object Request Broker), camada de programação (*middleware*) responsável pela comunicação dos diversos objetos do sistema. Logo acima dele existem os CORBA services, que implementam os serviços básicos para a construção de aplicações, e mais acima existem as CORBA facilities, que incorporam os serviços específicos para determinadas aplicações, que são definidas por vários comitês dentro da OMG.

Existe uma implementação do CORBA desenvolvida na Universidade de Washington, chamada de TAO [SCHMIDT00], que é destinada a aplicações de tempo real, como por exemplo, às necessárias para realidade virtual. Esta possui a vantagem de ser gratuita.

Embora o CORBA se mostre muito eficiente, está mais dirigido a transporte de grandes quantidades de dados entre computadores, e não tanto ao sistema de sincronismo. Isto o coloca como alternativa interessante para transferência de dados, mas não para sincronismos.

- MPI

A biblioteca MPI tem sido muito usada para solucionar a paralelização de programas numéricos, e tem obtido grande sucesso [PACHECO97]. Devido a seus mecanismos de transferência de mensagem não dependerem de nenhuma implementação específica, o porte de soluções é muito rápido. Além disso o MPI é suportado em diversas plataformas de computadores.

- MPI-VIA

Uma das formas de utilizar-se o sistema VIA é por intermédio do MPI-VIA. Desta forma, utilizando os comandos padrões do MPI, todas as vantagens do VIA podem ser exploradas facilmente. Além disso, programas já desenvolvidos para MPI são automaticamente portados para esta plataforma.

O MPI-VIA baseia-se na distribuição MPICH, sendo chamado, assim, de MVICH., a utilização do MPI-VIA cria um aumento de 5 microssegundos na transferência das mensagens em relação ao uso direto do VIA. Existem restrições de exportação deste sistema, pelos EUA, devido ser considerado de segurança nacional.

- PVM

O PVM (Parallel Virtual Machine) originalmente desenvolvido no Oak Ridge National Labs, é um sistema que fez muito sucesso nos primeiros sistemas paralelos. Uma de suas grandes vantagens é suportar um sistema completamente heterogêneo. A distribuição dos processos para os nós do aglomerado é feita de maneira transparente, disto surgiu o seu nome. Contudo com a constante evolução das outras bibliotecas de desenvolvimento acabou tornando o PVM um sistema obsoleto.

APÊNDICE D - Protocolos de Transferência de Dados

- TCP/IP

O TCP/IP [HUNT98] é um conjunto de protocolos que opera em praticamente todos os tipos de sistema operacionais e meios de comunicações (Token-ring, Ethernet, FDDI, etc.). Foi concebido como um padrão militar em 1983 e, com o surgimento da Internet, tornou-se um padrão. Tanto que este possui um endereçamento único, utilizado ao redor do mundo, o que não impede que endereços virtuais sejam usados localmente.

O padrão TCP/IP é totalmente aberto, e sua forma de endereçamento consiste de 32 bits, que normalmente são separados de 2 em 2 bytes. Esta separação é normalmente utilizada para a criação de sub-redes. Isto é feito por intermédio de máscaras, que fazem parte do endereçamento ser considerado de equipamentos na mesma rede interna, e qualquer endereço diferente, pertencente a outra rede, no caso a Internet.

O protocolo TCP, pertencente ao TCP/IP, possui um sistema que permite determinar se a mensagem enviada chegou no destino. Porém, muitas vezes, não há necessidade desta confirmação. Nestes casos é utilizado o protocolo UDP, também pertencente ao TCP/IP, e que não possui mecanismo de verificação de recebimento de mensagem, e por isso torna-se um protocolo mais rápido.

- Myrinet

A Myrinet é uma empresa com diversas soluções dirigidas à comunicação de dados com grande banda de transmissão e baixa latência, possuindo uma boa relação custo-benefício de seus produtos. As soluções Myrinet são muito utilizadas em sistemas computacionais de alta performance, em especial, aglomerados.

A soluções da Myrinet apresentam alta performance e alta disponibilidade ao mesmo tempo. Suas velocidades podem chegar a 2.0 Gigabits/segundo nos dois sentidos de transmissão (*full-duplex*), e sua latência é muito baixa, além de possuir grande escalabilidade, podendo chegar até dezena de milhares de pontos de rede.

O sistema da Myrinet também possui grande flexibilidade, além de robustez. Pode enviar pacotes TCP/IP de forma totalmente transparente, além de usar a filosofia nível de usuário (*user level*), na qual a aplicação tem acesso direto aos recursos de rede. Neste caso as diversas operações com a pilha de rede, e as constantes cópias das informações são eliminados.

Um dos grandes inconvenientes do sistema Myrinet, são as restrições do governo norte americano, o qual limita as vendas deste equipamento, com receio de possível utilização militar. Por isso, muitos centros no mundo não conseguem acesso a esta tecnologia.

- VIA

O protocolo VIA (Virtual Interface Architecture) foi concebido em 1997 por uma parceria entre Intel™, Compaq™ e Microsoft™. Atualmente disponível em diversas placas de rede e comutadores de rede (*switchs*) de mercado, que também suportam o padrão Ethernet, possibilitando a convivência dos dois padrões.

O objetivo do VIA é aumentar a eficiência de comunicação entre a aplicação e o sistema de rede, sendo uma de suas vantagens a de não ser proprietário. Assim, tanto a eletrônica, quanto os programas podem seguir este padrão, livremente.

Os principais focos do VIA [CAMERON02] são diminuir a latência, utilizar melhor a largura de banda do sistema e economizar ciclos de CPU para processamento de rede, sem a necessidade de trocar a eletrônica utilizada. Devido ao Ethernet não utilizar o sistema de nível de usuário (*user level*), a pilha de rede é gerenciada pelo sistema operacional, que consome certo tempo e cria a necessidade de cópias adicionais de mensagens. No caso do VIA não há a necessidade de chamadas ao sistema operacional, assim a aplicação consegue acesso direto ao cartão de rede. Como exemplo, uma transação padrão TCP/IP consome cerca de 25000 ciclos de processador, em uma estação de 400MHz, o que acarreta uma latência fixa de 62 microssegundos. O que é muito bom comparado a outros sistemas de transmissão.

O VIA é um sistema robusto, capaz de separar comandos de controle de dados Além disso utiliza a filosofia de passagem de mensagens (*message-passing*). Assim, o sistema é orientado a conexão, atuando ponto a ponto. A utilização do VIA em programação de baixo nível é relativamente complexa. Assim pode-se optar por alguma biblioteca de desenvolvimento de nível mais alto, para a utilização destes recursos.

APÊNDICE E - Estereoscopia

Existem diversas formas de simular o efeito de estereoscopia artificialmente, aqui serão mostradas as principais técnicas disponíveis.

Profundidade de Cor (*ChromaDepth*): Este método usa a diferença do comprimento e lentes refrativas. Em geral o vermelho aparecerá próximo e o azul afastado, as outras cores seguem a seqüência do arco-iris.

33

Exemplo de Profundidade de Cor e Anaglifo

Anaglifo (Anaglyph): Vem de grego significando relevo, sendo as figuras esculpidas em paredes. Para ver elas é necessário usar um óculos com lentes vermelhas-azul ou variantes, e geralmente a lente vermelha esta sobre a vista esquerda. Neste caso as cores da imagem são quebradas na cores primárias exibindo somente o vermelhor para o olho vermelho e as outras corem para a outra vista. Os óculos não permitem que uma vista veja sua imagem errada. Assim o cérebro junta as imagem criando a sensação de profundidade. Com uma aproximação da cor final.

34

Polarização Linear e Circular

Polarização Linear :As partículas eletro-magnéticas chamadas de fótons é dividido em componentes elétricos e magnéticos que são perpendicular entre elas, como exemplo acima. A luz refletida normalmente é polarizada na direção da superfície que ela refletiu. Um filtro polarizador, permite que somente a luz que esteja polarizada em certo angulo passe. Devido a luz poder estar orientada em qualquer direção, um filtro polarizante pode bloquear até 60% da luz que esta sendo usada. A polarização pode

³³ Imagem obtida em <http://www.scec.org/geowall/leftright.html>

³⁴ Imagem obtida em <http://www.barco.com/>

se norte-sul, leste-oeste, ou mesmo na diagonal. Este método apresenta problema se o usuário rotacionar a cabeça, pois a polarização se perde. É barato. O alinhamento do filtro é perpendicular a luz.

Polarização Circular : O conceito desta ppolarização se deve ao fato do fótons estar rotacionando ou no sentido horário ou no sentido anti horário. Desta forma podem haver dois filtro um para cada sentido. Os olhos humanos não conseguem perceber diferença na polarização da luz. O alinhamento dos filtros é na direção da luz. Passiva é bom para audiências.

Infitec: Polariza a luz dividindo o esperctro luminoso em dois, assim um filtro somente opera com certaz cores e o outro com as complementares.

Estereoscopia é o recurso que nos permite enxergar imagens sintetizadas por computador, com efeitos de profundidade. Este efeito se deve ao fato da disparidade binocular, que faz com que cada olho enxergue imagens ligeiramente diferentes. Baseado nos pontos de diferenças entre as imagens, o cérebro humano é capaz de distinguir se um determinado ponto está próximo ou afastado.

35

Primórdios da estereoscopia

³⁵ Imagem obtida em: <http://www.stereo3d.org>

Há muitos anos sistemas de estereoscopia têm sido usados, contudo só recentemente se tornaram um dispositivo mais comum. Estes já podem ser encontrados em *video-games* ou mesmo em sistemas domésticos de televisão. Para sistemas de estereoscopia tradicional é necessária a utilização de alguns óculos especiais, permitindo que imagens distintas cheguem a cada olho. Para isto existem duas técnicas principais: a primeira é o sistema passivo que explora alguma técnica de polarização da luz e a segunda é o sistema ativo que utiliza algum sistema automático, que seleciona o que cada olho deve enxergar.

- Estereoscopia Passiva:

No sistema passivo de estereoscopia, as imagens para cada olho são projetadas ao mesmo tempo e posteriormente se utiliza algum recurso físico passivo para separar as imagens geradas para cada olho, sem a necessidade de nenhum mecanismo para sincronização.

O sistema mais comum é baseado na polarização direcional da luz. Nos sistemas de projeção são acoplados filtros. Um projetor com um filtro posicionado na vertical e outro com o filtro posicionado na posição horizontal. Desta forma, em um olho se utiliza um filtro vertical, que permite que somente a luz do projetor com filtro vertical atravessar e na outra vista se utiliza um filtro horizontal. Assim, cada vista só verá a imagem correspondente.

36

Exemplo de Sistema de Projetores para Projeção Passiva

³⁶ Imagem obtida em: <http://www.berezin.com>

- Estereoscopia Ativa:

No caso do sistema ativo, os óculos possuem um sistema, normalmente baseado em cristal líquido, no qual as lentes podem ser controladas, para permitir ou não que a luz atravesse. O objetivo destes óculos é exibir a cada momento uma imagem referente a um olho, baseado na oclusão de cada lente, permitindo que cada olho receba sua respectiva imagem. Existem vários tipos de óculos estereoscópicos, sendo o mais comum o sistema óculos obturador (*shutter glasses*).

O controle para as lentes obstruírem a luz ou não é por meio de um emissor eletrônico de pulsos, presente em alguns cartões gráficos. Este sinal é enviado para os óculos, sendo devidamente sincronizado com as imagens que estão sendo geradas na tela. Uma particularidade deste sistema é a necessidade de atualização da imagem projetada na tela em uma frequência de apresentação relativamente alta, normalmente da ordem de 120 Hz, o que limita a utilização para somente projetores ou monitores que suportem esta frequência.

37

Óculos para estereoscopia ativa acionados por infravermelho

APÊNDICE F - SoftGenLock

O SoftGenLock [ALLARD02] é todo um sistema desenvolvido para disponibilizar o estéreo ativo a qualquer computador baseado no sistema operacional Linux, independente de placa de vídeo ou de acessar os gerenciadores (*driver*) da mesma. Para isto o SoftGenLock utiliza a seguinte técnica:

³⁷ Imagem obtida em <http://www.stereographics.com>

A memória de vídeo (*frame-buffer*) deve ser configurada para o dobro do tamanho da resolução que será apresentada na tela. Na aplicação gráfica, será necessário desenhar as imagens respectivas para cada olho, ao mesmo tempo na memória de vídeo, em duas regiões distintas. O SoftGenLock fará a comutação entre as duas porções de memória, cada vez que for necessário exibir na tela a imagem respectiva de cada olho.

O recurso de comutação das imagens do olho esquerdo e do direito, ao invés de ser controlado pelo cartão gráfico, será executado pelo sistema operacional. Para tanto, é necessário que o mesmo trabalhe em tempo real. Ou seja, não poderá haver latências nesta operação, o que comprometeria toda a operação. Assim é necessário que a estação que se deseja utilizar o SoftGenLock esteja com um núcleo (*kernel*), que trabalhe em tempo real. Posteriormente, somente será necessária a leitura do módulo de sistema operacional “softgenlock.o”, para que o sistema apresente uma imagem estereoscópica.

Porém, existe um inconveniente em usar um núcleo Linux de tempo real. Este não permite acesso em tempo real a certos recursos, como a placa de rede. Assim, é necessário o uso da porta paralela para a comunicação. Para fazer a comunicação entre os diversos computadores pela porta paralela, é necessário então a utilização de algum dispositivo físico, ou seja, um sistema que faça a conexão das portas. O sistema que faz a interconexão das portas paralelas é bem simples, somente necessitando fazer conexões de certos sinais dos cabos paralelos, [SOFTGENLOCK03].

Módulo do SoftGenlock Produzido

Embora o Softgenlock seja uma solução muito prática, existem alguns problemas. Sua estabilidade é baixa, desta forma para aplicações que necessitam de alta disponibilidade, esta solução não é adequada. Além disso apenas as atualizações de quadros são sincronizadas, níveis de iluminação não são. Assim estes ajustes devem ser feitos manualmente, quando os projetores ou monitores permitem.

APÊNDICE G - Sistemas de Interconexão de Vídeo

Neste apêndice se pretende mostrar os diversos tipos de sistema de interconexão disponíveis, que são essenciais para a transmissão das imagens entre o sistema de geração até o sistema de visualização.

VGA

O cabo VGA conhecido também como VGA-15 pinos é o cabo mais popular para monitores convencionais e profissionais. Ele é normalmente macho-macho e cinco dos seus pinos são os sinais de vermelho, verde, azul, sincronismo horizontal e sincronismo vertical (RGBHV).

RCA

Os cabos RCA são mais usados para equipamentos padrão VCR. Tipicamente ele é composto de três cabos com as seguintes cores: Vermelho, branco e amarelo, sendo que o cabo vermelho é responsável pelo canal direito do áudio, o branco é o canal esquerdo e o amarelo é o vídeo.

Como o sinal de vídeo é transportado apenas por um cabo, temos então uma qualidade comprometida de vídeo. Porém esse tipo de conector é o mais comum no mercado devido a estar presente em televisões, câmera de vídeo, vídeo game, vídeo cassetes, dentre outros dispositivos domésticos.

BNC

O cabo BNC é equivalente ao RCA, temos que a diferença entre os dois é apenas a sua terminação, sendo que a mudança de um padrão para outro pode ser feita com uma simples adaptação. O cabo BNC é preferido para aplicações mais profissionais pois a sua conexão é melhor e mais segura devido ao seu encaixe especial de conectar e girar.

Cabos para vídeo componente

Os cabos componentes são parecidos com RCA sendo que a diferença é que o sinal de vídeo transmitido é dividido em três. Esse sinal é referido também como Y,Cr,Cb, ou Y,Pb,Pr. Este tipo de conexão não deve ser confundido com o padrão RGB pois utiliza o mesmo sistema de cabeamento. Muitos DVDs e televisões de alta qualidade têm este tipo de saída.

Super Vídeo ou Y/C

O cabo Super Vídeo pode se referir como um SVHS e é encontrado como um sistema de qualidade superior. O super vídeo difere do RCA pois ele é capaz de transmitir uma qualidade mais alta de imagem pois separa o sinal de vídeo em dois diferente componentes: luminância e crominância.

RGBHV

O RGBHV é parecido com o BNC porém o sinal de vídeo é dividido em cinco canais. Cada cabo transmite respectivamente a cor vermelha, a verde e a azul além do sincronismo vertical e horizontal. Seguindo esse princípio temos mais duas variações o RGB H/V no qual temos apenas quatro cabos

sendo que o sincronismo é transportado por apenas um cabo, e ainda o RGB no qual o sincronismo é transmitido por um mesmo cabo de sinal de cor, normalmente o cabo verde.

O RGBHV é o cabo que transmite a melhor qualidade de vídeo para sistemas analógicos é encontrado em monitores profissionais de alta performance. Uma outra particularidade é que se pode facilmente ter a saída VGA 15-pinos convertida para RGBHV.

DVI

A interface DVI, definida pela DDWG (*Digital Display Working Group*) [DVI99] foi especificada para ser uma interface simples de baixo custo, tanto para o cartão gráfico, como para o sistema de visualização. O conector DVI existe em dois formatos, um puramente digital, e outro múltiplo que opera com sinais digitais e analógicos, além de ser *hot plug detectin*, ou seja, os dispositivos podem ser detectados no momento da conexão. Esta interface se usa do padrão existente VESA para permitir uma implementação simples de baixo custo.

Com a tecnologia DVI podemos ter ganhos de performance dependendo do sistema de visualização, sistemas como projetores CRT, necessitam de um intervalo de tempo para fazer a re-varredura, que simplesmente desperdiça uma quantidade de tempo sem estar transmitindo nenhuma informação, com os sistemas mais modernos, nem sempre é necessário este intervalo, o que resulta em um melhor desempenho.

Estimação de largura de banda:

$$\text{pixel/sec} = \text{pixelsHorizontal} \times \text{pixelsVertical} \times \text{Frequência} \times (1 + \text{Overhead\%})$$

Os monitores do futuro possuirão uma memória interna, somente quando os dados da tela mudarem estes serão enviados para o monitor.

O padrão DVI referencia o VESA para baixo custo e interoperabilidade. Os padrões que o DVI adota do VESA são:

- VESA *Display Data Channel (DCC) Specification*.
- VESA *Extended Display Identification Data (EDID) Specification*.
- VESA *Video Signal Standard (VSIS) Specification*.
- VESA *Monitor Timing Specification (DMT)*.
- VESA *Generalized Timing Formula Specification (GTF)*.
- VESA *Timing Definition for LCD Monitors Specification*.
- Compatibility com outras implementações baseadas em T.M.D.S.

Como forma de transmissão de dados é utilizado o T.M.D.S., uma técnica que especifica padrões de transmissão e converte os dados de 8 bits para 10 bits de transição minimizados. Na inicialização do sistema e o próprio sistema operacional o monitor DVI deve suportar uma resolução baixa, e ambos a BIOS e SO deve poder descobrir que monitor se trata pelo protocolo DDC2B, para isto o DVI se utiliza da estrutura EDID. DVI permite 1 ou 2 conexões T.M.D.S. A escolha entre uma conexão ou duas se faz pela frequência fornecida na primeira conexão, caso a velocidade seja de até 165MHz será utilizado só a primeira conexão caso seja maior se utilizara as duas conexões cada usando metade da frequência necessária. Se utilizar cores com mais de 24bits será necessário utiliza a segunda conexão.

Devido a necessidades de BIOS o sistema devera suportar o *Industry Standard Timings* que se refere a 640x480 pixels a 60Hz de atualização de quadros o pixel clock de 25.175 e a frequência horizontal de 31.5 kHz. Para um dispositivo de compatível com o DVI este deverá possuir a estrutura de dados EDID nas versões 1.2 e 2.0. É necessário o suporte DDC2B fornecendo +5 volts com o pino de força enviando 55mA e o monitor sugando até 50mA ligado e 10mA desligado. O gamma também é algo que deve ser algo suportado em alguma parte do sistema.

O *Digital Video Interface* ou DVI é parecido com o VGA porem ele é um pouco maior. O principal motivo da criação desse conector e do cabo é para ter uma interface digital-digital entre um computador e o seu display. Porém essa situação

ideal raramente ocorre atualmente por não encontrarmos essa interface em muitos devices.

O DVI encontra-se ainda em desenvolvimento pelas empresas como a Sony, Epson entre outras, com isso não temos um padrão totalmente definido, situação essa que poderá ser definida nos próximos anos.

DFP

O DFP, ou Mdr -20, era o tipo de conector inicial especificado pelo grupo de trabalho do *Digital Flat Painel*, membro da *Video Electronics Standards Association*. Como o DVI-D, esse conector suporta somente sinais digitais. O DFP ainda é usado em algumas aplicações, entretanto o seu uso está se tornando cada vez mais escasso.

EVC ou P&D

Outro tipo de conector que alguns fabricantes estão usando em seus projetores chama-se EVC ou P&D. Esse tipo de conector é similar ao DVI-I, mas o seu tamanho é ligeiramente maior. Como o DVI-I, esse Conector suporta sinais digitais e analógicos. Alguns fabricantes utilizam o EVC para USB.

APÊNDICE H - Evolução do padrão AGP

No processo de evolução do AGP passou por vários padrões. O primeiro modelo de AGP, chamado de AGP 1x, permitia transferir até 266 MB de dados por segundo, o padrão seguinte foi o AGP 2x, que é duas vezes mais rápido, permitindo que o cartão gráfico receba ou transmita até 533 MB de dados por segundo, o AGP 4x permite transferências de dados de até 1066 MB por segundo, AGP 8X transferência de 2132 MB por segundo, ou seja aproximadamente 2.1GB/seg.

Dependendo do modelo de placa mãe utilizada, o conector AGP pode comportar apenas um certo tipo de placas neste formato.

Se o pino estiver próximo do fundo do gabinete, do lado da fonte, então são suportadas apenas cartões gráficos AGP 1x ou 2x, se o pino estiver na posição contrária, mais próximo da frente do gabinete, então são suportadas apenas cartões gráficos AGP 4x. Mas, se não existir pino algum então temos um barramento AGP universal, onde podem ser encaixados cartões gráficos de qualquer um dos quatro padrões. Esta é a configuração mais comum nas placas mães atuais.

AGP 1x/2x e Universal

Naturalmente, assim como muda o encaixe na placa mãe, também muda o formato do conector do cartão gráfico.

Apesar de permitir um barramento de dados largo o suficiente para fornecer dados mesmo para os cartões gráficos 3D mais poderosos, os sistemas AGP possuem um grave problema, que dificulta a produção de cartões gráficos, devido ao limite de potencia elétrica disponível, conforme o poder de processamento um cartão gráfico aumenta, mais transistores ele terá, aumentando o consumo elétrico.

Um conector AGP comum, não é capaz de suprir estavelmente mais de 20 ou 25 Watts, o que limita bastante o potencial dos cartões gráficos. Neste caso, a solução encontrada pelos projetistas da 3DFx foi usar uma fonte externa.

O AGP Pro é na verdade um conector AGP com 48 contatos a mais, 20 de um lado e mais 28 do outro. Estes contatos adicionais são usados para aumentar a capacidade de fornecimento elétrico.

Existem dois tipos de interface física de barramento AGP Pro: o AGP Pro50 e o AGP Pro110. O nome indica a capacidade de fornecimento elétrico de ambos os padrões: o AGP Pro50 é certificado para fornecer até 50 Watts, enquanto o AGP Pro110 pode fornecer até 110 Watts.

O formato do encaixe não informa se o conector é Pro50 ou Pro110, apenas mostra quais tipos de placas são suportadas. O que muda do Pro50 para o Pro110 são os capacitores de alimentação da placa mãe, que devem ser capaz de manter uma corrente maior, e não o formato do encaixe. Nesse conector você pode instalar tanto placas de vídeo AGP convencionais quanto placas de vídeo AGP Pro.

AGP tradicional e AGP Pro

A Nvidia está apresentando uma solução muito prática em relação a controlador de dispositivos (driver) para todas as suas placas, chamada de UDA. O que permite um código de máquina padrão, facilitando os produtores de cartões gráficos.

APÊNDICE I Calibragem de Projetores

São apresentadas os padrões de imagens para calibração de projetores analógicos

Correto
Keystone
Side Pin
Top Key
Top Cushion
Botton Key
Botton Cushion
Bow Vertical
Bow Horizontal
Clinearity
S Linearity
Skew

Padrões de imagens de calibragem gerados pela JINX:

Barra de cores

Grades para alinhamento

Granulado e gradiente de cor e intensidade

APÊNDICE J - PRIMEIRO ARQUIVO X3D EXEMPLO

```
<?xml version="1.0" encoding="UTF-8"?>  
<X3D>  
  <Scene>  
    <Sound spatialize="false">  
      <AudioClip DEF="MYSOUND" loop="true" stopTime="0"  
url="sound/ICEFRONT.S3M"/>  
      <!--AudioClip DEF="MYSOUND" loop="true" stopTime="0"  
url="sound/invtro94.s3m"/-->  
      <!--AudioClip DEF="MYSOUND" loop="true" stopTime="0"  
url="sound/ILWU.S3M"/-->  
    </Sound>  
    <Sound location="0 -2 -6" spatialize="true">  
      <AudioClip DEF="MYSOUND2" loop="true" stopTime="0"  
url="sound/drumloop.wav"/>  
    </Sound>  
    <Sound location="0 0 4" spatialize="true">  
      <!--AudioClip DEF="MYSOUND2" loop="true" stopTime="0"  
url="sound/jungle.wav"/-->  
    </Sound>  
    <DirectionalLight direction="0 -1 0" color="1 1 0.9" ambientIntensity="1.0"/>  
    <SpotLight color="0 0 1" location="-2 -2 -2.5" /> <!--meu parse atual tem bug  
e nao funciona cut-->  
    <PointLight location="8 10 8" color="0.29 0.29 0.4" ambientIntensity="1.0"/>  
    <PointLight location="-8 -10 -8" color="1 1 1" ambientIntensity="1.0"/>  
    <TimeSensor DEF="TIME" cycleInterval="8" loop="true"/>  
    <PositionInterpolator DEF="INTERP" key="0 0.25 0.5 0.75 1" keyValue="0 0 6 3 0  
6 0 0 6 -3 0 6 0 0 6"/>  
    <OrientationInterpolator DEF='ORIENT' key='0 0.5 1.0' keyValue='0 1 0 0 0 1  
0 3.14 0 1 0 6.28'/>  
    <CoordinateInterpolator DEF='ci' key='0 .5 1' keyValue='4 6 0 7 12 0 11 6 0  
15 2 0 20 6 0 4 6 0 7 0 0 11 6 0 15 10 0 20 6 0 4 6 0 7 12 0 11 6 0 15 2 0 20 6 0'/>  
    <Transform translation="2 1 -5">  
      <Shape>  
        <Appearance DEF='App'>  
          <Material specularColor="0.0 0.0 1.0" diffuseColor="0.0  
0.0 1.0" />  
        </Appearance>  
        <NurbsCurve DEF='nc' containerField='geometry' controlPoint='-4  
0 0 -6 4 0 6 -4 0 4 0 0 ' order='4' tessellation='30'/>  
      </Shape>  
    </Transform>  
    <Transform DEF="ELEVATION" translation="2 -2 -5">  
      <Shape DEF="ELEV">  
        <Appearance DEF='App'>  
          <Material specularColor="1.0 1.0 1.0" diffuseColor="1.0  
1.0 1.0" />  
        </Appearance>  
        <ElevationGrid colorPerVertex='false' xSpacing="0.5"  
zSpacing="0.3" xDimension='6' zDimension='8' height='0 0 0 0 0 0 0 0 0 0 0 0 0  
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0' />  
      </Shape>  
    </Transform>
```

```
<Transform DEF="PIRAMID" translation="0 0 -4">
  <Shape DEF="PI">
    <Appearance DEF='App'>
      <Material specularColor="0.0 1.0 1.0" diffuseColor="0.0
1.0 1.0" />
    </Appearance>
    <IndexedFaceSet coordIndex="0 1 2 -1, 1 3 2 -1, 2 3 0 -1, 3 0
1">
      <Coordinate point="0 0 0, 1 0 0, 0.5 0 0.8, 0.5 0.8
0.2"/>
    </IndexedFaceSet>
  </Shape>
</Transform>
<Transform DEF="LETTER" translation="-1 -2 2">
  <Shape DEF="SPHERE">
    <Appearance DEF='App'>
      <Material specularColor="0.0 1.0 0.0" diffuseColor="0.0
1.0 0.0" />
    </Appearance>
    <Text string="aa" />
  </Shape>
</Transform>
<Transform DEF="SPHERE" translation="-4 0 2">
  <Shape DEF="SPHERE">
    <Appearance DEF='App'>
      <Material specularColor="0.1 0.1 1.0" diffuseColor="0.0
0.0 1.0" />
    </Appearance>
    <Sphere radius="1"/>
  </Shape>
</Transform>
<Transform DEF="CONE1" translation="2 0 -2">
  <Shape DEF="CONE1">
    <Appearance DEF='App'>
      <Material diffuseColor='1.0 0.2 0.2' />
    </Appearance>
    <Cone BottomRadius="1"/>
  </Shape>
</Transform>
<Transform DEF="CONE2" translation="-2 0 -4">
  <Shape DEF="CONE2">
    <Appearance DEF='App'>
      <Material diffuseColor='1.0 0.2 0.2' />
    </Appearance>
    <Cone BottomRadius="1" bottom="FALSE"/>
  </Shape>
</Transform>
<Transform DEF="BOX" translation="0 2 5">
  <Shape DEF="BOX">
    <Appearance DEF='App'>
      <Material diffuseColor='1.0 1.0 0.1' />
    </Appearance>
    <Box />
  </Shape>
</Transform>
<Transform DEF="CYLINDER" rotation='0 1 0 0.3' translation="-2 -2 -6">
  <Shape DEF="CYLINDER2">
    <Appearance DEF='App'>
      <ImageTexture url="luciano.bmp.old"/>
    </Appearance>
    <Cylinder />
  </Shape>
</Transform>
<Transform DEF="CYLINDER" rotation='0 1 0 0.1' translation="2 -2 -6">
  <Shape DEF="CYLINDER2">
    <Appearance DEF='App'>
      <ImageTexture url="earth.bmp"/>
    </Appearance>
    <Cylinder top="FALSE"/>
  </Shape>
</Transform>
<Transform DEF="SPHERE" translation="-3 -2 4">
  <Shape DEF="SPHERE">
    <Appearance DEF='App'>
```

```

1.0 1.0" />
    <Material specularColor="1.0 1.0 1.0" diffuseColor="1.0
    </Appearance>
    <Sphere radius="1"/>
  </Shape>
</Transform>
<Transform DEF="SPHERE" translation="3 5 4">
  <Shape DEF="SPHERE">
    <Appearance DEF='App'>
      <Material specularColor="1.0 1.0 1.0" diffuseColor="1.0
1.0 1.0" />
      </Appearance>
      <Sphere radius="1"/>
    </Shape>
  </Transform>
  <Transform DEF="BALL">
    <Shape DEF="SPHERE">
      <Appearance DEF='App'>
        <Material specularColor="1.0 0.0 0.0" diffuseColor="1.0
0.0 0.0" />
        </Appearance>
        <Sphere radius="1"/>
      </Shape>
    </Transform>
    <Transform DEF="WIRECUBE" translation="2 0 -2">
      <Shape>
        <Appearance>
          <Material emissiveColor='0.8 0.8 0.8' />
        </Appearance>
        <IndexedLineSet coordIndex='0 1 -1 1 2 -1 2 3 -1 3 0 -1 4 5 -1
5 6 -1 6 7 -1 7 4 -1 0 6 -1 1 7 -1 2 4 -1 3 5 -1' set_coordIndex=''
set_colorIndex=''>
          <Coordinate point='-3 -2 2 3 -2 2 3 2 2 -3 2 2 3 2 -2 -3
2 -2 -3 -2 -2 3 -2 -2' />
        </IndexedLineSet>
      </Shape>
    </Transform>
    <Transform DEF="SPIRAL" translation="-3 -3 -3">
      <Shape>
        <IndexedLineSet coordIndex='0 1 2 3 4 -1' set_coordIndex=''
set_colorIndex=''>
          <Coordinate point='0.0000 0.0000 0.0000 -0.0017 0.0125
-0.0002 -0.0033 0.0250 -0.0007 -0.0048 0.0375 -0.0015 -0.0061 0.0500 -0.0026' />
        </IndexedLineSet>
      </Shape>
    </Transform>
    <Transform DEF="RAL2" translation="0 -3 -6">
      <Shape>
        <Extrusion beginCap='false' crossSection='-6.225 4.8 -6.225 -
4.8 6.225 -4.8 6.225 2.1 1.425 2.1 1.425 4.8 -6.225 4.8' endCap='false' solid='false'
spine='0 0 0 0.2 0 0 0.4 0' set_orientation='' set_spine='' set_scale=''
set_crossSection='' />
      </Shape>
    </Transform>
  </Transform>
  <ROUTE fromNode="TIME" fromField="fraction_changed" toNode="INTERP"
toField="set_fraction" />
  <ROUTE fromNode="INTERP" fromField="value_changed" toNode="BALL"
toField="translation" />
  <ROUTE fromNode="TIME" fromField="fraction_changed" toNode="ORIENT"
toField="set_fraction" />
  <ROUTE fromNode="ORIENT" fromField="value_changed" toNode="SPIRAL"
toField="rotation" />
  <ROUTE fromNode="TIME" fromField="fraction_changed" toNode="ci"
toField="set_fraction" />
  <ROUTE fromNode="ci" fromField="value_changed" toNode="nc"
toField="set_controlPoint" />
</Scene>
</X3D>

```

Um dos inconvenientes de se trabalhar com sistemas tipo CAVERNA Digital, é a dificuldade de acesso a dispositivos de controle. O rastreador é um deles, contudo uma forma alternativa prática é a utilização de equipamentos sem fio, como computadores portáteis (*notebooks*) ou computadores de mão (*palm-top*). Em especial os computadores de mão com rede sem fio são portáteis o suficiente para poderem ser levados para o

Computador de mão

ambiente imersivo sem grandes transtornos. O sistema sem fio opera sob banda de rádio, assim ele não será capaz de interferir com o sinal de infravermelho utilizado pelos óculos, e o sistema de rastreamento também não terá grandes problemas de ruídos. Uma vantagem é a possibilidade de diversos dispositivos sem fio coexistirem no mesmo ambiente. Sua transmissão é lenta, porém suficiente para a maioria das rotinas de gerenciamento.

Uma das possibilidades de se operar com sistema de computadores de mão com o opcional sem fio, é prendê-lo ao sistema de rastreamento. Assim, os comandos fornecidos pelo sistema sem fio podem ser combinados com os valores de posição e rotação, fornecendo novas possibilidades de navegação e operação.

APÊNDICE L - Projetores DLP

A Texas Instruments lançou um estudo [VIDEOSYSTEMS] em maio de 2002 para investigar os DMD (*Digital Micromirror Device*) e aprender como o processamento deste sistema DLP (*Digital Light Processing*) se comporta e verificar a performance deste sistema em relação ao sistema LCD. O sistema DLP apresentou uma vida útil e mais confiável que o sistema LCD, pois o sistema de filtros se consumia e o contraste e precisão das imagens ia caindo com o tempo. Em testes se verificou que os projetores LCD começam a comprometer a cor em média após 1368 horas de uso.

APÊNDICE M - PCI-Express

O PCI-Express [BHATT02] é um barramento novo que apresenta alta velocidade e baixa latência e tem o objetivo de ser de propósitos gerais, desta forma ele unificaria os padrões em barramento para todos os tipos de plataformas, desde micro computadores até servidores. Ele concorre diretamente com o AGP. As principais empresas já estão mostrando o seu interesse neste padrão, como a 3DLabs, a ATI e a Nvidia, todos eles prometem que lançarão cartões gráficos em um curto espaço de tempo. Mais potência elétrica, mas não total 60Watts 25 AGP 8X, computadores antigos não vão suportar. Vale ressaltar que existem pinos de controle. Além disso, serve para rede, apresentando uma baixa latência, o que o PCI-X não apresenta. InfiniBand™ Architecture, 10 Gigabit funciona em sua plenitude.

Barramento	Pinos	Banda (GB/s)
AGP 8X	108	2,1
PCI X16	62	8

PCI EXPRESS CARD ELECTROMECHANICAL SPECIFICATION, REV 1.0

04747239

38

Vertical Edge-Card Connector

³⁸ imagem obtida em <http://www.extremetech.com/>

GLOSSÁRIO

Aglomerado de computadores convencionais	Conjunto de computadores convencionais conectados em rede, utilizados para a implementação de um sistema de computação de alto desempenho.
Aglomerados de computadores	São um conjunto de computadores, capazes de compartilhar seus recursos, para um objetivo comum.
Alta disponibilidade	Sistema capaz de se manter funcionando, caso certo nível de problemas aconteçam.
Alto desempenho	Sistema capaz de oferecer um desempenho elevado em consideração a tecnologia atual.
Armazenamento duplo	Técnica que consiste em definir uma região de memória para ser apresentada em uma tela, enquanto outra região é utilizada para a produção da próxima tela.
Áudio espacializado	Simulação de som, na qual caixas de som pré-definidas trazem a sensação de pontos de som definidos no ambiente virtual.
CAVE	(CAVE Automatic Virtual Environment) Sistema de Realidade Virtual, que oferece uma sensação total de imersão, baseado em múltiplas projeções dispostas ao redor do usuário, permitindo uma grande liberdade de movimento aos usuários
CAVERNA Digital	Sistema de projeção cúbico acionado por um aglomerado de computadores
Coerência visual	Capacidade que das imagens de ter continuidade, tanto geométrica como temporal.
Data-lock	Responsável pela coerência da cena, assegurando que as informações em cada nó do aglomerado sejam idênticas em cada instante.
Driver	Programa utilizado em conjunto com o sistema operacional, capaz de permitir a comunicação do sistema principal com algum dispositivo acoplado ao computador
Escalabilidade	
Estereoscopia	Recurso que nos permite enxergar imagens sintetizadas por computador, com efeitos de profundidade.

Filas de Processamento Gráfico	Sistemas compostos de vários processadores e dispositivos eletrônicos dirigidos a cálculos gráficos, dispostos em sequência.
<i>Frame-lock</i>	Sistema para manter a continuidade da imagem, executando a troca de quadros ao mesmo tempo em um sistema de multiprojeção.
<i>Genlock</i>	Sinal compartilhado entre os sistemas de geração de vídeo, que é utilizado como referencia para a taxa e momento das atualizações das imagens bem como cor e brilho.
<i>In-Band Networking Management</i>	Gerenciamento de computadores pelo sistema padrão. Normalmente a rede de computadores
JINX	Sistema de desenvolvimento de aplicações de realidade virtual imersiva para aglomerados de computadores
Latência	Tempo que uma mensagem demora para chegar de um ponto ao outro do sistema
Modem	Usado para comunicação, principalmente por telefone
Mouses espaciais	Sistema de navegação 3D similar a um mouse, porém conte 6 graus de liberdade.
Multiprojeção	Sistema de projeção com diversos projetores devidamente calibrados.
<i>No-break</i>	Dispositivo capaz de manter a tensão em caso de ausência de energia
<i>Out-of-Band Networking Management</i>	Gerenciamento de computadores por um sistema alternativo. Normalmente pela porta serial
P43	Os tubos de projetores CRT possuem uma camada de fósforo para que a imagem seja produzida, existem vários padrões para o fósforo de projeção, o fósforo mais antigo P31 possui um decaimento muito lento o que inviabilizava seu uso com estereoscopia, já o P43 possui um decaimento muito mais rápido (1.2 ms), permitindo tal uso.
PANDEMONIUM	Mesa de controle, para gerenciamento de dispositivos de realidade virtual

Parallax	Ângulo que faz com que os olhos separados tenham um foco virtualmente.
<i>PowerWall</i>	Sistema de multiprojeções paralelas, produzindo telas de grandes dimensões e de alta resolução.
Rasterização	Síntese de imagens por computador.
Rastreamento	Recurso que permite a determinação interativa da posição e orientação relativas de um dispositivo no mundo virtual.
<i>Rate-lock</i>	Sistema que define a velocidade das atualizações.
Realidade virtual imersiva	Sistema de simulação por computador de um ambiente, porém fazendo o usuário se sentir dentro do ambiente, não somente uma imagem.
Reconhecimento de voz	Sistema capaz de interpretar voz humana e retornar instruções.
RS232	Sistema de conexão serial
<i>Shading Languages</i>	Linguagem de programação utilizada para descrever cenas gráficas, muito útil para aumentar o realismo das imagens
SIRIUS	Interface gráfica do sistema de gerenciamento de ambientes de realidade virtual
Sistemas de projeção semi-esféricos	que ao invés de possuir uma tela plana, possui uma tela esférica e a projeção é feita por um projetor disposto entre o usuário e a tela. Estes sistemas apresentam algumas vantagens por não
<i>Sort-first</i>	Sistema no qual o paralelismo gráfico ocorre antes do processamento geométrico.
<i>Sort-last</i>	Sistema no qual o paralelismo gráfico ocorre depois da rasterização (rasterization), mas antes da fragmentação.
<i>Sort-middle</i>	Sistema no qual o paralelismo gráfico ocorre entre o processamento geométrico e a rasterização (rasterization) do processador geométrico.
Supercomputação	Computadores mais evoluídos para as condições da tecnologia atual.
Telnet	Sistema que permite a comunicação entre dois computador por comandos textuais através de uma rede

<i>Vview frustum culling</i>	Técnica usada para desenhar somente os objetos que estão dentro do campo de visão, que é representado como uma pirâmide.
Visualização Volumétrica	Conjunto de técnicas e algoritmos para a síntese de imagens a partir de dados amostrados no espaço n-dimensional (normalmente $n \geq 3$).
Volume Lógico	Técnica usada para gerenciar diversos discos a fim do sistema operacional usar eles como um local único. Muitas vezes vem acompanhado de um sistema RAID para redundância e versatilidade.
VRCluster	Sistema completo para realidade virtual imersiva, baseado em aglomerados de computadores convencionais para aplicações de realidade virtual.
Wi-Fi	wireless fidelity, basicamente se refere as redes sem fio utilizadas em computação baseadas no protocolo IEEE 802.11
ZEUS	Principal programa do sistema de gerenciamento, responsável pelo transporte das mensagens através do sistema

ÍNDICE

3D Studio	81	Fmod	79
3DX Demo or Die	83	frame-lock.....	19, 75
acuidade visual	96	<i>genlock</i>	17, 49
aglomerados de computadores	1	Glass	29
AGP.....	22	glxgears.....	91
alta disponibilidade	57	heterogêneo.....	38
armazenamento duplo	20	HMD	42
árvore de visita	72	homogêneo.....	38
assimétrico	38	Hyper-cubes.....	25
Avalon	29	IC:IDO	30
AVANGO	28	IDE.....	50
barramento	22, 37	iluminação	45
Beowulf	12	IPMI.....	54
cabos.....	51	IRIX	33
câmeras robotizadas	63	Itautec	4, 105
campo de visão.....	6	JINX.....	68, 79
CAVE.....	3, 42	KickStart.....	53
CAVELib	27	KVM	46, 56
CAVERNA Digital	2, 5, 44	lâmina	13
CAVERNsoft	27	latência.....	16, 33
centralizado	38	Linux.....	32, 49, 55, 91
coerência visual.....	2, 6	memória compartilhada	16
COMDEX	82	Microsoft Windows	32
Computação em Grade.....	14	molecular	83
contensão da rede	93	Moore.....	1
controle de iluminação	60	MPI	75, 79
controle dos projetores	61	MPICH.....	75
cursos	102	multiprojeção	5
CVS	9	multiserial	55
daemons	59	navegar.....	73
data-lock.....	21	Nvidia	89
DeltaGen	30	óleo e gás	5
DHCP	53	Onyx	89
disjuntores	84	OpenGL	72, 79, 88, 90
distribuído	38	OpenInventor	36
Diverse	29	OpenMP.....	34
DTD	35	OpenSG	27
DVI.....	24, 37, 97	ORAD.....	24
escalabilidade horizontal.....	46	Oscar.....	54
escalabilidade vertical	46	paralax	69
Escher.....	80	paralelização gráfica.....	87
estação de controle	57	PCI	23
estereoscopia	18, 57, 69	Perfect cluster	92
fibra óptica	51	PowerWall	82
filas de processamento gráfico	88	projeção esférica.....	81

Pthread	75	SoftGenlock	17
Pthreads	34	som	72
publicações	102	sort-first	23
PXE	53	sort-last	23, 72
RAM	50	sort-middle	23
rastreador	66	SPADE	54
rastreamento	64	SPEC	90
rastreamento de cabeça	70	Syzygy	28
rate-lock	21	Terminal Server	47
realidade virtual	3	threads	34
reconhecimento de voz	56, 65	TimeLock	77
RedHat	53	Top 500	13
réguas de força	56	transacionais	14
resolução	22	tubo de raios catódicos	18
RGB	36	UML	8
Rio de Janeiro	81	USB	74
RS232	55, 74	VIA	47
SBPC	82	volume lógico	50
Schema	35	VRCluster	2, 5
SCSI	50	VRJuggler	28
segurança	55	VRML	36
<i>Shading Languages</i>	79	Web3D	36
SIBGRAPI	82	WildCat	89
Silicon Graphics	3, 25, 46, 87	Windows	91
simétrico	38	X3D	36, 68, 77
SIRIUS	59, 104	XML	34, 50, 58, 68, 78
SMP	34	ZEUS	58