Claiton de Oliveira

**Montagem e Utilização de Ambientes Virtuais
Agrícolas em um Sistema de Multiprojeção
Imersivo a partir de Cenas Naturais**

Tese apresentada à Escola de Engenharia de
São Carlos da Universidade de São Paulo,
como parte dos requisitos para obtenção do
Título de Doutor em Engenharia Mecânica.

Área de concentração: Manufatura.

Orientador: Prof. Dr. Arthur José Vieira Porto

São Carlos

2012

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

O48m Oliveira, Claiton de
 Montagem e utilização de ambientes virtuais agrícolas
 em um sistema de multiprojeção imersivo a partir de cenas
 naturais. / Claiton de Oliveira ; orientador Arthur José
 Vieira Porto. -- São Carlos, 2012.

 Tese (Doutorado - Programa de Pós-Graduação em
 Engenharia Mecânica e Área de Concentração em
 Manufatura)-- Escola de Engenharia de São Carlos da
 Universidade de São Paulo, 2012.

 1. Ambientes virtuais. 2. Sistema de multiprojeção
 imersivo. 3. Reconhecimento de cenas naturais. I. Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel **CLAITON DE OLIVEIRA**.

Título da tese: "Montagem e utilização de ambientes virtuais agrícolas em um sistema de multiprojeção imersivo a partir de cenas naturais".

Data da defesa: 05/10/2012

Comissão Julgadora:

Resultado:

Prof. Titular **Arthur José Vieira Porto (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Dr. **Mario Luiz Tronco**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Dr. **Ricardo Yassushi Inamasu**
(EMBRAPA)

Aprovado

Profa. Dra. **Jandira Guenka Palma**
(Universidade Estadual de Londrina/UEL)

Aprovado

Prof. Dr. **Emerson Carlos Pedrino**
(Universidade Federal de São Carlos/UFSCar)

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Mecânica:
Prof. Associado **Marcelo Areias Trindade**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Denis Vinicius Coury**

*Ao meu pai Daniel e à minha mãe Miriam. À minha noiva Daniele. Aos meus irmãos
Fabiano, Francielle e Franklin.*

Agradecimentos

Gostaria de agradecer aos meus pais, à minha irmã e aos meus irmãos, por serem uma fonte de inspiração e motivação em tudo que faço.

À minha querida noiva Daniele, pelo apoio, incentivo, carinho, amor e companheirismo sempre presentes.

Ao professor Drº. Arthur José Vieira Porto, pela oportunidade de realização deste trabalho, pela sua confiança, apoio e a amizade formada nestes anos de mestrado e doutorado.

Aos professores Drº. Mário Luiz Tronco e Drº. Alexandre José Tiberti pelas conversas e discussões esclarecedoras que me auxiliaram durante a realização desta pesquisa. E à professora Drª. Jandira Guenka Palma, pela sua amizade, confiança e generosidade investidas em mim desde os tempos da graduação.

Aos amigos Heleno Murilo Campeão Vale e Luciano Cássio Lulio que sempre estiveram dispostos a ajudar sem pedir nada em troca.

À Beth, do Laboratório de Simulação, por sua amizade e solicitude e à Ana Paula, da Secretaria de Pós-graduação, por sua paciência em me atender sempre que preciso.

À CAPES pelo apoio financeiro através da bolsa de estudos concedida.

Ao Sr. Viriato por gentilmente ceder o espaço de sua fazenda para a aquisição de imagens de laranjeiras.

E, claro, a todos os amigos e colegas de laboratório, pelas amizades formadas e o companheirismo existente durante esses últimos anos.

Não posso deixar de dizer, enfim, o quanto considero difícil deixar de nominar todas as pessoas a quem eu gostaria de agradecer. No entanto, quando somos abençoados em ter tantas pessoas para agradecer, a maior gratidão e reconhecimento deve ser dada a Deus.

A Ele sou grato por tudo, sempre.

"God is able to do exceedingly abundantly above all that we ask or think..."

Ephesians 3:20

Resumo

OLIVEIRA, C. **Montagem e Utilização de Ambientes Virtuais Agrícolas em um Sistema de Multiprojeção Imersivo a partir de Cenas Naturais**. 2012. 198p. Tese de Doutorado - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

A geração de ambientes virtuais de áreas urbanas ou cenas naturais coloca um grande número de problemas no âmbito da Computação Gráfica, dado que a quantidade de informação necessária para a criação de modelos realistas é dependente da dimensão e complexidade da área a modelar. A construção de um grande número de modelos de objetos naturais de forma detalhada é extremamente trabalhosa. Enquanto os modelos de estruturas artificiais, tais como máquinas ou edifícios podem ser obtidos a partir de fontes CAD, o mesmo não ocorre as plantas e outros fenômenos naturais. Embora muitos ambientes virtuais sejam criados por modelagem manual e individual de cada um dos seus componentes, os processos automáticos e semi-automáticos de reconstrução de ambientes naturais 3D provaram que podem ser muito mais eficientes, reduzindo a duração, o custo e a alocação de recursos humanos. A integração entre diferentes tecnologias e ferramentas que possibilitem a identificação de elementos em um cenário agrícola, modelagem de objetos 3D e a posterior apresentação e utilização do ambiente virtual em um sistema do tipo CAVE não é uma tarefa trivial. Dessa forma, o objetivo desta pesquisa é desenvolver uma metodologia de montagem automática de ambientes virtuais agrícolas baseada na extração de objetos de cenas naturais reais a partir de imagens de vídeo para utilização no sistema de multiprojeção imersivo do Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos (LMVI). A partir de um modelo de dados 3D projetado em um sistema que oferece um alto grau de imersão e interação como o LMVI, pode-se fazer comparações com outros modelos de dados ou com o mesmo modelo em épocas diferentes. Através da comparação entre modelos é possível identificar alterações que ocorreram no ambiente ao longo do tempo (tanto naturais como causadas pelo homem) auxiliando na tomada de decisão em processos agrícolas.

Palavras-Chaves: Ambientes Virtuais, Sistema de Multiprojeção Imersivo, Reconhecimento de Cenas Naturais.

Abstract

OLIVEIRA, C. **Generation and Use of Virtual Agricultural Environments in an Immersive Multiprojection System from Natural Scenes.** 2012. 198p. Thesis (Doctorate) - School of Engineering of São Carlos, University of São Paulo, São Carlos, 2012.

The generation of virtual environments for urban or natural scenes poses a number of problems within the Computer Graphics, since the amount of information needed to create realistic models is dependent on the size and complexity of the area to be modeled. The construction of a large number of natural object models in detail is extremely laborious. While the models of artificial structures, such as machines or buildings can be obtained from CAD sources, the same is not true plants and other natural phenomena. Although many virtual environments are created by individual and manual modeling of each of their components, the processes of automatic and semi-automatic 3D reconstruction of natural environments have proved to be much more efficient, reducing duration, cost and allocation of human resources. The integration between different technologies and tools that enable the identification of elements in an agricultural setting, modeling of 3D objects and the subsequent presentation and use of virtual environment in a CAVE-like system is not a trivial task. Thus, the objective of this research is to develop a methodology for automatic mounting of agricultural virtual environments based on the extraction of objects of natural scenes from real video images for use in the immersive multiprojection system of the Multiuser Laboratory of 3D Immersive Visualization of Sao Carlos (MLIV). From a 3d data model projected in a system that offers a high degree of immersion and interaction as MLIV, one can make comparisons with other data models or with the same model at different periods. Through the comparison between models is possible to identify changes that occurred in the environment over time (both natural and manmade) assisting the decision making in agricultural processes.

Keywords: Virtual Environments, Natural Scene Recognition, Image Segmentation.

Sumário

Lista de Figuras

Lista de Abreviaturas e Siglas

1	Introdução	29
1.1	Problema de pesquisa	30
1.2	Motivação	31
1.3	Objetivos	32
1.4	Metodologia de Pesquisa	33
1.5	Organização do Trabalho	33
2	Revisão Bibliográfica	35
2.1	Realidade Virtual	35
2.1.1	Características dos Ambientes Virtuais	37
2.1.2	Tipos de Sistemas de Realidade Virtual	40
2.1.2.1	Grau de Interação Proporcionado do Usuário	40
2.1.2.2	Grau de Imersão Proporcionado ao Usuário	41
2.1.3	Dispositivos de Realidade Virtual	44
2.1.3.1	Dispositivos de Rastreamento	44
2.1.3.2	Dispositivos de Interação	46
2.1.3.3	Dispositivos visuais	48
2.1.3.4	Dispositivos Auditivos	52
2.1.3.5	Dispositivos Hápticos	54

2.1.3.6	CAVE	56
2.1.4	Dispositivos de Telepresença	59
2.1.5	Bibliotecas de Desenvolvimento de Ambientes Virtuais	60
2.1.6	Aplicações de Ambientes Virtuais e Telepresença	62
2.2	Visão Computacional	63
2.2.1	Aquisição de Imagens	64
2.2.1.1	Câmera CCD	65
2.2.1.2	Amostragem e Quantização	66
2.2.2	Pré-processamento	66
2.2.3	Segmentação	67
2.2.4	Extração de Características	67
2.2.5	Classificação e Reconhecimento	68
2.3	Visão Computacional e Cenas Naturais	68
2.3.1	Visão Computacional e Cenas Agrícolas	69
2.4	Construção de Ambientes Virtuais a partir de Cenas Naturais	71
2.5	Considerações Finais	75

3 Laboratório Multiusuário de Visualização 3D Imersiva 77

3.1	Espaço Físico	78
3.2	Estrutura de Projeção	78
3.2.1	Telas de Projeção	79
3.2.2	Projetores	79
3.2.3	Óculos Estereoscópicos Passivos	80
3.3	Estrutura Computacional	80
3.3.1	Computadores Principais	81
3.3.2	Computadores Auxiliares	81
3.4	Sistema de Som 5.1	82

3.4.1	Disposição dos Alto falantes	83
3.5	Dispositivos de Entrada de Dados	84
3.5.1	Controle de Vídeo Game Wii	84
3.5.2	<i>Webcam</i> e Caneta <i>Laser</i>	84
3.5.3	Dispositivo de Interação por Cor	85
3.5.4	<i>Joystick</i>	85
3.6	Plataforma e Bibliotecas de Programação	86
3.6.1	VR Juggler	87
3.6.2	OpenSG	88
3.6.3	OpenCV	89
3.6.4	OpenAL	90
3.6.5	TinyXML	90
3.6.6	Curl	90
3.6.7	Importação dos Objetos 3D	91
3.6.8	Criação do Grafo de Cena	92
3.6.9	Busca de Objetos no Grafo de Cena	92
3.7	Mecanismos de Interação para Utilização do AV no LMVI	93
3.7.1	Interação por Controle Remoto Wii	93
3.7.2	Interação por Detecção de Movimentos	94
3.7.2.1	Configuração e Posicionamento da <i>Webcam</i>	96
3.7.2.2	Pré-processamento da Imagem para Filtragem do Laser	97
3.7.2.3	Detecção do Ponto de Luz Na Imagem Pré-processada	97
3.7.2.4	Mapeamento da Movimentação do Ponto de Luz no Plano	97
3.7.2.5	Criação da <i>Thread</i> de Detecção do Laser	98
3.7.2.6	Comunicação dos Dados de Movimentação	100
3.7.2.7	Desenvolvimento da Função Zoom através do Laser	100

3.7.3	Ponteiro para Seleção de Objetos na Cena	102
3.7.4	Criação e Disparo de Som 5.1	105
3.8	Considerações Finais do Capítulo	106

4 Metodologia Proposta 109

4.1	Visão Geral	109
4.2	Módulo Mundo Real	110
4.2.1	Visão Computacional	111
4.2.1.1	Aquisição das Imagens	111
4.2.1.2	Pré-processamento das Imagens	116
4.2.1.3	Segmentação das Imagens	117
4.2.1.4	Extração de Características	120
4.2.2	Modelo de Dados de Objetos Reais	128
4.2.2.1	Geração do Modelo	129
4.2.2.2	Atualização do Modelo	131
4.2.2.3	Estrutura do Modelo Gerado	131
4.3	Módulo Modelos 3D	132
4.3.1	Repositório de Objetos Virtuais 3D	132
4.3.2	Modelo de Dados de Objetos Virtuais	134
4.3.2.1	Geração do Modelo	134
4.3.2.2	Atualização do Modelo	136
4.3.2.3	Estrutura do Modelo Gerado	136
4.4	Comunicação de Dados	137
4.5	Módulo Mundo Virtual	138
4.5.1	Modelo de Dados do Ambiente Virtual	138
4.5.1.1	Geração do Modelo	139
4.5.1.2	Atualização do Modelo	140

4.5.1.3	Estrutura do Modelo Gerado	140
4.5.2	Montagem do AV em um Sistema de Multiprojeção Imersivo	141
4.5.2.1	Carregamento do arquivo 3D em memória	142
4.5.2.2	Inserção dos Modelos 3D no Grafo de Cena	144
4.5.2.3	Salvamento do Ambiente Virtual	145
4.5.3	Visualização do Ambiente Virtual	146
4.5.3.1	Interação	146
4.6	Considerações Finais do Capítulo	147
5	Testes de Aplicação e Resultados	149
5.1	Aquisição das Imagens	149
5.2	Orientação Ortogonal à Linha de Plantio	150
5.2.1	Pré-processamento	151
5.2.2	Segmentação	152
5.2.3	Seleção do Objeto de Interesse	156
5.2.4	Extração das Características	158
5.2.5	Geração do Modelo de Dados Reais	162
5.2.6	Modelo de Dados Virtuais	162
5.2.7	Modelo de Dados do Ambiente Virtual	163
5.2.8	Montagem da cena virtual	164
5.2.9	Visualização 3D Imersiva	167
5.2.10	Salvamento do Ambiente Virtual	168
5.2.11	Considerações	169
5.3	Orientação Paralela à Linha de Plantio	170
5.3.1	Pré-processamento	170
5.3.2	Segmentação	171
5.3.3	Seleção do Objeto de Interesse	174

5.4	Orientação Oblíqua à Linha de Plantio	175
5.4.1	Pré-processamento	175
5.4.2	Segmentação	176
5.4.3	Seleção do Objeto de Interesse	180
5.5	Considerações Finais do Capítulo	181
6	Considerações Finais	183
6.1	Trabalhos Futuros	185
	Referências	187

Lista de Figuras

2.1	Dispositivo de interação do tipo esteira chamado <i>CirculaFloor</i> (IWATA et al., 2004).	48
2.2	Dispositivo de interação do tipo esfera da VirtuSphere.	48
2.3	Dispositivo binocular suspenso.	50
2.4	Dispositivos hápticos fixos: Falcon da Novint (NOVINT, 2008) e Phantom da SensAble (SENSABLE, 2008) respectivamente.	55
2.5	Dispositivos hápticos CyberTouch, CyberGrasp, CyberForce e Haptic Workstation (CYBERGLOVE, 2012).	55
2.6	Projetistas experimentando e avaliando o interior de um carro dentro de um CAVE.	57
2.7	Esquema que descreve a posição dos projetores em um sistema do tipo CAVE.	58
2.8	Etapas de um sistema de visão computacional genérico (CONCI; AZEVEDO; LETA, 2008)	64
2.9	Aquisição da imagem (CONCI; AZEVEDO; LETA, 2008)	65
3.1	Paredes e chão da sala do LMVI	78
3.2	Disposição das telas de projeção do LMVI	79
3.3	Estruturas metálicas de sustentação dos projetores (a) superiores e (b) laterais	80
3.4	(a) Par estereoscópico de imagens e (b) óculos estereoscópico	81
3.5	<i>Home theater</i> que compõe o sistema de som <i>surround</i> 5.1 do LMVI	82
3.6	Conexão entre o PC e o <i>Home Theater</i>	82
3.7	Visão frontal do sistema de sonorização do LMVI	83
3.8	Visão superior do sistema de sonorização do LMVI	83
3.9	<i>Wii</i> e a barra de LEDs infravermelhos	84

3.10	<i>Webcam</i> de baixo custo e canetas <i>laser</i>	85
3.11	Dispositivo de interação por cor (azul, vermelho e verde)	86
3.12	<i>Joystick</i> Clone Vibration Água com 12 botões de funções	86
3.13	Diagrama de Classes do OpenSG combinado com o VR Juggler	89
3.14	Importação de objetos 3D a partir de uma arquivo VRML	91
3.15	Instanciação do nó raiz no grafo de cena	92
3.16	Inserção de um objeto 3D ao grafo de cena	93
3.17	Busca de um objeto 3D no grafo de cena	94
3.18	Dispositivo com LEDs infravermelhos	95
3.19	Material utilizado para uniformizar e concentrar a luz do laser	95
3.20	Posicionamento da <i>webcam</i> no sistema de multiprojeção	96
3.21	Sequência de imagens resultantes do pré-processamento do ponto luminoso	97
3.22	Deteccção do ponto de luz na imagem	98
3.23	Obtenção da posição e tamanho do ponto de luz	98
3.24	Mapeamento da movimentação do <i>laser</i> no plano	99
3.25	Criação e destruição da <i>thread</i> de deteção de movimento do <i>laser</i>	99
3.26	Função de envio de dados de movimentação para os clientes	100
3.27	Usuário afastando duas canetas <i>laser</i> para obtenção do <i>zoom</i>	101
3.28	Implementação (resumida) do <i>zoom</i> com 2 pontos de luz	102
3.29	Ponteiro 2D para seleção de objetos 3D	103
3.30	Seleção um objeto 3D a partir da posição de um ponteiro 2D	103
3.31	Mapeamento das coordenadas do ponteiro para as dimensões da tela	104
3.32	Projeção de um raio a partir da posição da câmara até o “infinito”	104
3.33	Obtenção de um objeto interceptado pelo raio projetado	105
3.34	Movimentação de um objeto 3D selecionado com o movimento do ponteiro 2D	106
4.1	Visão geral da metodologia proposta	110

4.2	Etapas do sistema de visão computacional abordadas pela metodologia proposta.	111
4.3	Enquadramento do objeto de interesse quando este se encontra no centro da imagem: (a) objeto totalmente enquadrado, (b) objeto parcialmente enquadrado.	113
4.4	Orientação da câmara: (a) paralela à linha de plantio, (b) oblíqua à linha de plantio.	113
4.5	Orientação da câmara ortogonal à linha de plantio e paralela à superfície do solo.	114
4.6	Etapas sequenciais realizadas durante a segmentação.	117
4.7	Estratégia do detector de bordas Canny.	118
4.8	Imagens de entrada e saída do detector de bordas Canny respectivamente. . .	119
4.9	Operações de fechamento e abertura respectivamente.	120
4.10	Etapas sequenciais necessárias para a extração das características do contorno.	121
4.11	Traçado da linha de contorno sobre um objeto segmentado.	123
4.12	Traçado da linha vertical que divide o objeto em dois hemisférios.	124
4.13	Traçado das linhas horizontais perpendiculares a linha vertical.	124
4.14	Delimitação da região central do quadro da imagem.	126
4.15	Fluxograma do processo de geração e atualização do modelo de dados reais. .	128
4.16	Estrutura hierárquica do modelo de dados de objetos reais.	131
4.17	Modelos 3D que representam os segmentos de linhas horizontais.	133
4.18	Modelos 3D que representam os segmentos de linhas horizontais com maior nível de detalhes.	133
4.19	Fluxograma do processo de geração e atualização do modelo de dados virtuais.	134
4.20	Estrutura hierárquica do modelo de dados de objetos virtuais.	136
4.21	Comunicação de dados entre os módulos.	137
4.22	Associação de modelos de dados reais e virtuais para geração do modelo do ambiente virtual.	138
4.23	Fluxograma do processo de geração e atualização do modelo de dados do AV.	141
4.24	Estrutura hierárquica do modelo de dados do ambiente virtual.	142

4.25 Fluxograma do processo de montagem do ambiente virtual.	143
4.26 Fluxograma do processo de carregamento dos modelos 3D.	144
4.27 Visualização de um AV através do sistema de multiprojeção imersivo.	146
5.1 Linha de referência para deslocamento entre as linhas de plantio.	150
5.2 Imagem da câmera com a orientação ortogonal à linha de plantio.	151
5.3 Imagens adquiridas nos espaços de cores RGB (esquerda) e pré-processadas no espaço de níveis de cinza com filtro Gaussiano(direita).	152
5.4 Imagens no espaço de níveis de cinza (esquerda) e imagens binárias resultantes do detector de bordas Canny (direita).	153
5.5 Detecção de bordas quadro a quadro a medida que a câmera se desloca pela cena natural.	154
5.6 Imagens resultantes do detector de bordas Canny (esquerda) e imagens resultantes de transformações morfológicas de fechamento e abertura (direita).	155
5.7 Imagens binárias resultantes de transformações morfológicas de fechamento e abertura (esquerda) e imagens no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.	156
5.8 Imagens com traçados de todos os contornos externos (esquerda) e imagens contendo apenas o maior contorno em cada quadro (direita).	157
5.9 Imagem com o traçado do contorno externo (em cima) e imagem com os traçados dos segmentos de linhas (em baixo).	158
5.10 Traçado de segmentos de linhas em uma sequência de quadros de objetos filmados consecutivamente.	160
5.11 Imagens onde o centroide do objeto de interesse se encontra no centro do quadro da imagem.	161
5.12 Modelo de dados reais (compactado) contendo as características dos objetos extraídos das cenas naturais.	163
5.13 Modelo de dados virtuais contendo as informações sobre os objetos 3D previamente modelados.	163
5.14 Modelo de dados do ambiente virtual contendo as informações sobre os modelos de dados reais e virtuais.	164

5.15	Montagem do ambiente virtual a partir das imagens da cena natural em tempo quase real.	166
5.16	Visualização do AV montado no sistema de multiprojeção imersivo do LMVI. .	167
5.17	Diferentes pontos de vista da cena virtual montada.	168
5.18	Detecção de bordas em uma imagem com (em cima) e sem profundidade de campo (em baixo).	169
5.19	Imagens adquiridas nos espaços de cores RGB (esquerda) e pré-processadas no espaço de níveis de cinza com filtro Gaussiano(direita).	170
5.20	Imagens no espaço de níveis de cinza (esquerda) e imagens binárias resultantes do detector de bordas Canny (direita).	171
5.21	Imagens resultantes do detector de bordas Canny (esquerda) e imagens resultantes de transformações morfológicas de fechamento e abertura (direita). .	172
5.22	Imagens binárias resultantes de transformações morfológicas de fechamento e abertura (esquerda) e imagens no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.	174
5.23	Imagens com traçados de todos os contornos externos (esquerda) e imagens contendo apenas o maior contorno em cada quadro (direita).	175
5.24	Imagem adquirida nos espaços de cores RGB (em cima) e pré-processada no espaço de níveis de cinza com filtro Gaussiano(em baixo).	176
5.25	Imagem no espaço de níveis de cinza (em cima) e imagem binária resultantes do detector de bordas Canny (em baixo.	177
5.26	Imagem resultante do detector de bordas Canny (em cima) e imagem resultante de transformações morfológicas de fechamento e abertura (em baixo).	178
5.27	Imagem binária resultantes de transformações morfológicas de fechamento e abertura (em cima) e imagem no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.	179
5.28	Imagem com traçados de todos os contornos externos (em cima) e imagem contendo apenas o maior contorno em cada quadro (embaixo).	180

Lista de Abreviaturas e Siglas

AMD	Arm-Mounted Display
API	Application Programmer's Interface
ASCII	American Standard Code for Information Interchange
AV	Ambiente Virtual
CAVE	CAVE Automatic Virtual Environment
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CCD	Charge Couple Device
Curl	Client Uniform Resource Locator
CIE	Commission Internationale de l'Eclairage
DHM	Dexterous Hand Master
DVI	Digital Visual Interface
EESC	Escola de Engenharia de São Carlos
FAPESP	Fundação de Amparo à Pesquisa do Estado de São Paulo
FPS	Frames Per Second
GPS	Global Positioning System
HDMI	High-Definition Multimedia Interface
HMD	Head-Mounted Display
HRTF	Head Related Transfer Function
HTTP	HyperText Transfer Protocol
HUD	Head-Up Display
LED	Light Emitting Diode
LMVI	Laboratório Multiusuário de Visualização 3D Imersiva
NEPAS	Núcleo de Ensino e Pesquisa em Automação e Simulação
NMEA	National Marine Electronics Association
RNA	Rede Neural Artificial
ROI	Region Of Interest
RV	Realidade Virtual
RGB	Red, Green, Blue
SID	Spatially Immersive Display

SIG	Sistema de Informação Geográfica
USP	Universidade de São Paulo
VC	Visão Computacional
VRML	Virtual Reality Modeling Language
WIMP	Windows, Icons, Menus and Pointing Device
X3D	Extensible 3D
XML	eXtensible Markup Language

1 Introdução

A informação sempre foi a principal base de decisões eficazes. A informação espacial é particularmente complexa, pois exige ao menos dois descritores: a localização e a identificação de algum elemento. Por anos a ligação entre esses dois descritores foram os mapas tradicionais 2D elaborados manualmente através de lápis, régua e folhas de papel. Na agricultura por exemplo, Brase (2005) descreve que antes do uso de ferramentas computacionais, os produtores rurais registravam informações básicas sobre as suas propriedades, como os limites de suas plantações, o tipo de cultura de determinada região, localização de rios e áreas de floresta nativa, em mapas feitos a mão. Essas informações não eram exatamente precisas, mas a precisão não era um atributo fundamental. Estes mapas eram utilizados estritamente como uma forma de documentar as informações básicas sobre o campo, com pouca análise.

Com o desenvolvimento da agricultura de precisão e dos sistemas de informação geográfica (SIG), a necessidade de informações espaciais mais precisas tornou-se evidente. A construção de mapas com base em fotos aéreas forneceu uma importante ferramenta de gestão de propriedades agrícolas para melhorar a aplicação de fertilizantes, distribuição das linhas de irrigação, drenagem, entre outros (BRASE, 2005). Backes e Plumer (2003) apresenta o uso de mapas de ervas daninhas para agricultura de precisão baseados em SIG, enquanto Clay e Shanahan (2011) apresenta o uso de mapas para controle de crescimento de culturas e monitoramento de produtividade. No entanto, a maioria desses mapas é apresentada na forma 2D, onde só é possível identificar o tipo e a localização de um determinado elemento. A visualização dos elementos que compõem a paisagem da forma como os enxergamos no mundo real não é disponível. Neste contexto, o uso da realidade virtual (RV) permite não apenas a visualização como a interação do usuário com o ambiente virtual 3D projetado.

O desenvolvimento de métodos e técnicas de visualização 3D de dados geográficos pode fornecer uma melhor interpretação dos dados não estruturados complexos de cenas ou paisagens reais, uma vez que permitem a transição de dados de cenas 2D para 3D. A tradução para um modelo 3D permite que uma pessoa interaja com os dados a partir de diferentes perspectivas, mais próximas da realidade física de um espaço. Neste contexto, a Realidade

Virtual (RV) combinada com informações espaciais pode oferecer não apenas a análise espacial de um ambiente, como também pode simular paisagens ou cenários do mundo real em um mundo virtual. A chave desta tecnologia concentra-se na aquisição, exibição e processamento de dados 3D.

1.1 Problema de pesquisa

A modelagem de objetos virtuais 3D é uma tarefa que exige habilidade, tempo e recursos para criação de objetos virtuais que representem objetos reais. Os projetistas precisam abstrair as características geométricas dos elementos que compõem a cena real, para que através do uso de polígonos possam construir um modelo geométrico tridimensional de uma paisagem virtual (incluindo terreno, árvores, etc). Dessa forma, a modelagem automática ou semi-automática de objetos 3D tem se mostrado como um importante avanço na virtualização de ambientes naturais (ESTEBAN; DIJK; GROEN, 2010).

A construção de ambientes virtuais a partir de cenas naturais é um fenômeno complexo que envolve parâmetros como posição, orientação, tamanho, geometria, textura, etc. Para obtenção desses parâmetros, as pesquisas envolvendo realidade virtual (RV) e sistemas de informação geográfica (SIG), tem utilizado recursos como imagens captadas por aviões ou satélites (STROTHOFF et al., 2011; CERDAT; DESCOMBES; ZERUBIA, 2003), radares e *scanners* a laser terrestres ou aéreos (HONG-WEI et al., 2011; BINNEY; SUKHATME, 2009; MOORTHY et al., 2007; ZHU et al., 2009) e sistema de posicionamento global (GPS) (CHENG et al., 2011; WENG; ZHAO; HE, 2010; VERMA; KUMAR; HSU, 2006).

Conforme será descrito no Capítulo 2, uma grande parte dos projetos de pesquisa na área tem se concentrado na reconstrução de modelos 3D individuais (uma árvore, uma estátua, um edifício) (ALIAGA; ZHANG; BOUTIN, 2007; BINNEY; SUKHATME, 2009) ao invés da montagem de uma paisagem virtual completa. Outros trabalhos descrevem a construção de paisagens virtuais, porém utilizam tecnologias de custos elevados como *scanners* a laser aéreos (HONG-WEI et al., 2011; BINNEY; SUKHATME, 2009; MOORTHY et al., 2007; ZHU et al., 2009). A possibilidade de utilização de um robô agrícola móvel ou algum outro veículo para captura de imagens em uma paisagem natural, oferece uma alternativa de aquisição de dados com baixo custo. No entanto, a extração de características de um ambiente real a partir de imagens 2D, é um dos grandes desafios encontrados na construção automática ou semi-automática de ambientes virtuais 3D.

Além disso, a montagem e utilização de ambientes virtuais em um sistema de

multiprojeção imersivo demanda uma série de conhecimentos interdisciplinares, como a implementação e controle do sistema de projeção, desenvolvimento de softwares de rede, controle de múltiplos equipamentos, renderização das imagens em tempo real, *software* de vídeo e áudio, projeção sincronizada, arquitetura e *design*, etc. Dentre os trabalhos investigados, foi possível identificar uma escassez de metodologias que descrevam todo o processo de construção de ambientes virtuais agrícolas, desde a forma de aquisição até apresentação e utilização do ambiente em um sistema de realidade virtual imersiva, utilizando ferramentas de baixo custo e implementações em código livre.

Logo, a argumentação apresentada nos parágrafos acima e que será desdobrada e explicada com mais detalhes no Capítulo 2, justifica a necessidade de desenvolver um método de montagem de ambientes virtuais agrícolas a partir de dados não estruturados complexos como cenas naturais adquiridas através de imagens 2D. Um modelo de dados 3D obtido a partir das imagens de um ambiente real oferece mais opções de análise, comparação, navegação, interação e alteração, do que apenas o uso de imagens não processadas.

1.2 Motivação

O *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos (LMVI)*, da EESC-USP, disponibiliza um sistema de multiprojeção imersivo com projeção em três faces: duas faces laterais e uma no chão. Através deste projeto, que será descrito em detalhes no Capítulo 3, os pesquisadores, da região de São Carlos, têm uma infraestrutura de visualização 3D imersiva disponível para o desenvolvimento de pesquisas científicas multidisciplinares.

A partir de um modelo de dados 3D, projetado em um sistema que oferece um alto grau de imersão e interação como o LMVI, pode-se fazer comparações com outros modelos ou com o mesmo modelo em épocas diferentes. Através dessas comparações, pode-se identificar alterações que ocorreram no ambiente ao longo do tempo. Essas alterações, que podem ser tanto naturais como causadas pelo homem, podem fornecer informações que auxiliam a tomada de decisão em processos agrícolas.

Ao longo da pesquisa bibliográfica, verificou-se que os pesquisadores se concentram no desenvolvimento de métodos e técnicas de construção de ambientes virtuais, mas poucos têm a oportunidade de utilização destes ambientes através de um sistema de multiprojeção imersivo. A transferência de uma aplicação de realidade virtual baseada em *desktop* para um sistema como o LMVI não é trivial. Logo, o desenvolvimento de procedimentos que viabilizem a utilização desses ambientes em um sistema como o LMVI, através da implementação de

mecanismos de interação de baixo custo e o uso de *software* livre, é um dos interesses dessa pesquisa.

Além disso, o Núcleo de Ensino e Pesquisa em Automação e Simulação (NEPAS), do Departamento de Engenharia Mecânica da EESC-USP, tem desenvolvido pesquisas com um robô agrícola móvel para navegação e aquisição de dados em um cenário agrícola formado principalmente por culturas brasileiras. A construção de um mapa 3D de boa qualidade a partir de cenas agrícolas adquiridas por este robô, permite sua navegação no mundo real através de um mapa virtual. Assim, a montagem e utilização de ambientes virtuais no LMVI aliado ao uso do robô agrícola móvel, complementam a motivação do desenvolvimento deste trabalho.

1.3 Objetivos

O objetivo geral desta pesquisa foi desenvolver uma metodologia de montagem automática de ambientes virtuais a partir de cenas naturais de culturas agrícolas. Para atingir esse objetivo geral, os seguintes objetivos específicos guiaram o desenvolvimento do trabalho:

- desenvolvimento de um método de aquisição de imagens, que permita a captura adequada dos dados, através da definição do posicionamento e configuração da câmera em relação aos objetos de interesse;
- desenvolvimento de uma metodologia de segmentação das imagens adquiridas e extração das características dos objetos de interesse;
- desenvolvimento de uma metodologia de identificação de tamanho, geometria e localização dos objetos de interesse;
- desenvolvimento de um modelo de dados para representação dos objetos reais extraídos das imagens;
- desenvolvimento de um modelo de dados para representação dos objetos virtuais previamente modelados;
- desenvolvimento de um método de associação entre os objetos reais extraídos das imagens e os objetos virtuais previamente modelados;
- viabilização da utilização do ambiente virtual montado no sistema de multiprojeção imersivo do LMVI através da implementação de mecanismos de interação de baixo custo e do uso de *software* livre;

- desenvolvimento de métodos de salvamento e carregamento de ambientes virtuais montados para execução e análise em períodos diferentes.

1.4 Metodologia de Pesquisa

A criação de conhecimentos científicos requer o emprego de métodos científicos apropriados. Marconi e Lakatos (2010) descreve o método científico como sendo o conjunto de atividades sistematizadas que permitem alcançar os objetivos, auxiliando o pesquisador em suas decisões. Por isso, este item se dedica a explicar os procedimentos metodológicos adotados para atingir os objetivos apresentados na Seção 1.3.

Como primeira fase dos trabalhos, foram feitas pesquisas bibliográficas com o propósito de fornecer a fundamentação teórica necessária para o prosseguimento do trabalho, bem como identificar o estado da arte da literatura ligada ao problema de pesquisa. Este raciocínio mostra que a pesquisa bibliográfica e a definição dos problemas e objetivos podem ser vistos como um processo iterativo, pois quanto mais conhecimento sobre o tema o pesquisador adquire, maior é a probabilidade de identificar os problemas e lacunas mais relevantes ligados ao tema principal. Por isso, a revisão bibliográfica se iniciou pelas metodologias existentes de construção de ambientes virtuais automáticos e semi-automáticos, direcionando-se posteriormente para tópicos correlatos, tais como construção de ambientes virtuais de paisagens naturais.

A principal fonte de pesquisas bibliográficas foram os periódicos internacionais, como forma de alinhar este trabalho aos discursos feitos por pesquisadores na literatura internacional. Adicionalmente, foram utilizados artigos publicados em anais de congressos internacionais, livros e teses de doutorado. Com isso, foi possível obter uma boa perspectiva do estado da arte em relação a construção de ambientes virtuais baseados em cenas naturais. Também foi possível identificar a lacuna na literatura que motivou a formulação do problema central deste trabalho, que diz respeito a uma metodologia bem definida de montagem e utilização de ambientes virtuais agrícolas em um sistema de multiprojeção imersivo através do reconhecimento de cenas naturais.

1.5 Organização do Trabalho

Este trabalho está organizado em uma estrutura de 5 capítulos, os quais são descritos brevemente a seguir:

Após o **Capítulo 1** que contextualiza, justifica e apresenta os objetivos do trabalho, o **Capítulo 2** apresenta os principais conceitos relacionados à realidade virtual, visão computacional e o estado da arte na construção de ambientes virtuais a partir de cenas naturais.

No **Capítulo 3** é descrito o Laboratório de Visualização 3D Imersiva de São Carlos, que refere-se a um projeto financiado pela FAPESP, sob coordenação do Professor Dr. Arthur José Vieira Porto, que oferece a comunidade científica uma infraestrutura de visualização 3D imersiva para realização de pesquisas multidisciplinares.

A metodologia de montagem dos ambientes virtuais agrícolas a partir de cenas naturais é apresentada no **Capítulo 4**, enquanto os testes de aplicação e os resultados obtidos são apresentados no **Capítulo 5**.

Por fim, o **Capítulo 6** apresenta as considerações finais da pesquisa e as perspectivas de trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo são apresentados os principais aspectos referentes a Realidade Virtual (RV), introduzindo definições e apresentando suas características e classificações, além dos dispositivos envolvidos neste tipo de tecnologia, bem como o estado da arte na construção de ambientes virtuais a partir de cenas naturais.

2.1 Realidade Virtual

Os primeiros trabalhos envolvendo a Realidade Virtual (RV) iniciaram-se na indústria de simulação, com os simuladores de voo que a força aérea do Estados Unidos passou a construir logo após a Segunda Guerra Mundial (JACOBSON, 1994). No entanto, o termo “Realidade Virtual” é creditado a Jaron Lanier, que nos anos 80 juntamente com Thomas Zimmerman, desenvolveram a primeira luva de dados, chamada *DataGlove*, usada para manipular dados em aplicações de RV (ZIMMERMAN et al., 1987), além da introdução dos primeiros capacetes de visualização (HMD *head-mounted displays*) de forma comercial (BEIER, 1998). Outros termos relacionados incluem: “Ciberespaço” (GIBSON, 1984), “Realidade Artificial” (KRUEGER; GIONFRIDDO; HINRICHSEN, 1985), e mais recentemente, “Mundos Virtuais” e “Ambientes Virtuais” (FISHER; TAZELAAR, 1990; ELLIS, 1994).

O termo, “Realidade Virtual” é utilizado em uma variedade de formas e, algumas vezes, de uma maneira confusa e distorcida. Beier (1998) afirma que, originalmente, o termo se referia a “Realidade Virtual Imersiva”, uma vez que, a utilização do termo, por Jaron Lanier, era uma forma de diferenciar as simulações computacionais tradicionais de simulações envolvendo usuários imersos em um ambiente artificial, gerado por computador (RHEINGOLD, 1991; ARAUJO, 1996).

Ao longo do tempo, o termo “Realidade Virtual” tornou-se abrangente e passou a ser utilizado não apenas para aplicações imersivas, como também para aplicações parcialmente imersivas ou ainda sem imersão alguma. Um exemplo disso é o sistema QuickTime VR

(CHEN, 1995), que utiliza fotografias para a modelagem tridimensional do mundo e fornece as capacidades de um pseudo *look-around*¹ e *walk-around*² em um monitor. Termos alternativos como “Mundos Virtuais” ou “Ambientes Virtuais” (AV), refletem mais adequadamente a vasta gama de aplicações possíveis e o fato de a tecnologia ainda estar distante de simular completamente a “realidade”. Sendo, “Ambientes Virtuais” um dos termos preferidos pela comunidade acadêmica (BEIER, 2000).

Steuer (1992), acrescenta que a definição de realidade virtual é baseada em conceitos de “presença” e “telepresença”, ou seja, a chave para a definição da realidade virtual, em termos de experiência humana ao invés de tecnologias de *hardware* e *software*, é o conceito de presença. Onde a presença, pode ser entendida como a percepção de um ambiente físico, mediada apenas pela mente humana. No entanto, quando a percepção é mediada por alguma tecnologia de comunicação, a pessoa é obrigada a perceber simultaneamente dois ambientes distintos: o ambiente físico em que o mesmo se encontra presente, e o ambiente apresentado através do meio tecnológico. Logo, o termo “telepresença” pode ser usado para descrever a última experiência como a sensação de estar em um ambiente mediado por algum meio de comunicação, ao invés do ambiente físico imediato.

Sendo a presença uma percepção natural de um ambiente e a telepresença a percepção de um ambiente mediado, este último pode ser tanto um ambiente real temporalmente ou geograficamente distante como um ambiente virtual sintetizado por computador. Dessa forma, estudos sobre a telepresença têm explorado tecnologias de realidade virtual a fim de orquestrar a percepção humana a partir de um ambiente distante, uma vez que esta proporciona a habilidade de manipular a realidade física remotamente em tempo real, através de sua representação virtual (WILSON, 2003).

O uso do termo “telepresença” foi cunhado por Minsky (1980), em referência aos sistemas de teleoperação para manipulação de objetos físicos remotos. Sheridan (1992) continuou esta tradição, adotando o termo presença (ao invés de telepresença), para uma revista dedicada ao estudo de teleoperação e ambientes virtuais, para referir-se à percepção de estar em um ambiente artificial ou remoto, reservando “telepresença” apenas para os casos que envolvem teleoperação. No entanto, na mesma seção da revista, Held e Durlach (1992) utilizam a expressão “telepresença” para referirem-se à experiência comum a ambos teleoperação e ambientes virtuais. Sendo assim, o termo é amplo e serve para destacar as semelhanças entre teleoperação e ambientes virtuais.

¹*Look-around* refere-se a um tipo de interação onde o usuário pode olhar o ambiente virtual em várias direções.

²*Walk-through* refere-se a um tipo de interação onde o usuário pode “andar” ou mover-se pelo ambiente virtual restrito a caminhos ou lugares pré-definidos.

A telepresença, como um refinamento da teleoperação, permite que o indivíduo controle não apenas a simulação mas a própria realidade. Neste contexto, Goldberg (2000) afirma que a essência da telepresença consiste na antipresença, ou seja, o indivíduo não tem que estar fisicamente presente em um local para afetar a realidade do mesmo. Logo, Mair (1997) afirma que isto é muito mais do que uma simples teleconferência, uma vez que permite ao usuário sentir-se imerso em um ambiente remoto e ainda ser capaz, através da teleoperação, de manipular ou controlar eventos remotos.

As fronteiras entre os sistemas de RV e sistemas de telepresença estão cada vez mais tênues e lentamente se fundem em um espectro muito amplo de tecnologias utilizadas para visualizar e interagir com mundos tridimensionais em tempo real. Através das tecnologias de RV e telepresença, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações, possibilitando simulações de situações do mundo real em um ambiente ou mundo virtual. A telepresença e operadores remotos que empregam a RV tanto para educação, treinamento, simulação como entretenimento, têm sido explorada por cientistas e desenvolvedores de sistemas complexos como apresentados em (HANÁK et al., 2007; VALLIUS; RÖNING, 2006; YANG et al., 2005).

2.1.1 Características dos Ambientes Virtuais

Os ambientes virtuais (AV), assim como qualquer tipo de sistema computacional, necessitam atender a requisitos não-funcionais (SOMMERVILLE, 2007) relacionados a aspectos de *software*, *hardware* ou fatores externos, que determinem condições ou restrições ao comportamento do sistema pretendido. Entre estes requisitos destacam-se:

- **Desempenho:** refere-se ao atendimento de exigências de processamento, tempo de resposta, consumo de recursos, *throughput* e eficiência de processos contidos no escopo do sistema;
- **Usabilidade:** compreende o atendimento de critérios relacionados ao entendimento e facilidade de uso do sistema, além do nível de satisfação dos usuários durante a navegação e interação com o AV;
- **Portabilidade:** normalmente o ambiente virtual desenvolvido é fortemente ligado ao ambiente de desenvolvimento. As aplicações devem comportar facilidades para execução em diferentes plataformas e/ou sistemas operacionais;
- **Flexibilidade:** a forma em que o ambiente foi desenvolvido deve permitir que

possíveis alterações, assim como a remoção e inserção de novos dispositivos seja implementada com o mínimo possível de alteração de código.

Além dos aspectos destacados anteriormente, Stuart (1996) e Vince (2004) acrescentam que um AV agrega peculiaridades adicionais que podem caracterizá-lo como um ambiente:

- **Sintético:** o AV é gerado em tempo real por um sistema computacional, e não pré-gravado como em sistemas multimídia;
- **Multisensorial:** significa que o ambiente pode incluir recursos que estimulem diferentes sentidos humanos, como visão, audição, tato, senso de espaço, profundidade, etc;
- **Tridimensional:** o ambiente é representado em 3 dimensões e, além disso, existem recursos que dão a ideia de que o ambiente possui profundidade e possibilita movimento através dele;
- **Imersivo:** o sistema deve permitir que o usuário sinta-se “dentro” do mundo virtual, seja com o seu corpo físico ou com uma representação qualquer (avatar³, vídeo, simulação de cabine, etc). É importante que, de alguma forma, o usuário seja “envolvido” pelo ambiente;
- **Interativo:** corresponde à capacidade do sistema detectar as entradas do usuário e, instantaneamente, promover alterações ou ações em cenas e objetos do ambiente virtual;
- **Realístico:** determina a precisão com que o AV reproduz os objetos reais, as interações com os usuários e o próprio modelo do ambiente;
- **Com Presença:** permitir a sensação de estar presente fisicamente em um ambiente artificial.

Os AV's também podem ser caracterizados como a coexistência integrada de três ideias básicas: imersão, interação e envolvimento (MORIE, 1994). Embora não seja necessário que as três coexistam em simultâneo. É importante ressaltar que nem todo AV é imersivo (LESTON, 1996; ROBERTSON; CARD; MACKINLAY, 1993). A visualização de uma cena 3D em

³Avatares são representações visuais de usuários ou agentes autônomos dentro do mundo virtual que são controlados; podem ser imagens tridimensionais ou vídeos que servem como entrada padrão para o usuário que o controla.

um monitor convencional, por exemplo, é considerada não-imersiva. Na RV não-imersiva o usuário é transportado, parcialmente, para um mundo virtual, através de uma projeção ou um monitor, mas continua a sentir-se, predominantemente, no mundo real. Entretanto, Robertson, Card e Mackinlay (1993) comentam que dispositivos baseados nos demais sentidos humanos, podem introduzir algum grau de imersão inclusive à AV's em monitores.

A ideia de imersão está ligada ao sentimento de fazer parte do ambiente - o usuário vê-se envolvido no domínio da aplicação (realidade), tendo uma sensação de presença dentro do mundo virtual. Normalmente, um sistema imersivo é obtido com o uso de capacete de visualização, ou cavernas virtuais - sistemas imersivos baseados em salas com projeções das visões nas paredes, teto, e piso (CRUZ-NEIRA, 1995). Além do fator visual, dispositivos ligados aos demais sentidos também são importantes para o sentimento de imersão, como o som (BEGAULT, 1994), o posicionamento automático da pessoa e dos movimentos da cabeça, controles reativos, etc.

A interação está ligada à capacidade do computador detectar as entradas do usuário e modificar instantaneamente o mundo virtual em função das ações efetuadas sobre ele (capacidade reativa). As pessoas são cativadas por uma boa simulação em que as cenas mudam em resposta aos seus comandos, que é a característica mais marcante dos vídeo *games*. Para que um sistema de RV pareça mais realista, o ambiente virtual inclui objetos simulados. Outros artifícios para aumentar o realismo são empregados, por exemplo, a texturização dos objetos do ambiente e a inserção de sons tanto ambientais quanto sons associados a objetos específicos (ARAUJO, 1996).

A ideia de envolvimento, por sua vez, está ligada ao grau de motivação para o engajamento de uma pessoa em determinada atividade. O envolvimento pode ser passivo, como ler um livro ou assistir televisão, ou ativo, ao participar de um jogo com algum parceiro. A RV tem potencial para os dois tipos de envolvimento ao permitir a exploração de um ambiente virtual e propiciar a interação do usuário com o mundo virtual dinâmico. Embora a percepção visual seja nosso sentido primário, outros sentidos também devem ser estimulados para proporcionar uma completa imersão; entre os quais o retorno auditivo, o tato e a força de reação.

Sob o conceito de ambientes virtuais alberga-se um vasto número de tecnologias, na sua maioria constituindo tecnologias recentes ainda sujeitas a novos desenvolvimentos sem que tenha-se, ainda, uma percepção muito clara do verdadeiro alcance que as mesmas podem ou virão a deter. Dessa forma, a seção seguinte apresenta algumas classes de RV, com o intuito de obter-se uma melhor compreensão sobre os diferentes tipos de sistemas de RV.

2.1.2 Tipos de Sistemas de Realidade Virtual

Como descrito anteriormente, Realidade Virtual (RV) é um termo usado para ambientes 3D gerados por computador, que possibilitam ao usuário entrar e interagir em um ambiente sintético. Para Draper, Kaber e Usher (1999), um ambiente sintético permite que as capacidades perceptivas, cognitivas e psicomotoras humanas sejam projetadas em um ambiente distante, hostil ou simulado.

Assim, Capin et al. (1999), a definem como uma tecnologia capaz de transportar um indivíduo para um ambiente diferente do real, sem movê-lo fisicamente, manipulando as informações destinadas aos órgãos sensoriais humanos de tal maneira que o ambiente percebido seja associado ao Ambiente Virtual (AV) desejado e não ao Ambiente Real (AR).

Diversos artigos e livros que abordam os conceitos e definições sobre realidade virtual ou ambiente virtual (KALAWSKY, 1993; LATTI; OBERG, 1994; EARNSHAW; VINCE; JONES, 1995; VINCE, 2004; KIRNER; TORI, 2004). Como mencionado, há várias definições aceitas, o que é devido, em parte, à natureza interdisciplinar da área e também à sua evolução, pois de uma maneira ou de outra, os sistemas de RV acabaram vindo de sistemas computacionais de mesa, simuladores e sistemas de teleoperação, entre outros (KIRNER; TORI, 2004).

Os sistemas de RV diferem entre si de acordo com os níveis de imersão e de interatividade proporcionado ao participante. Esses níveis são determinados pelos tipos de dispositivos de entrada e saída de dados do sistema, além da velocidade e potência do computador que o hospeda. Ainda não existe um critério claro de classificação dos sistemas de RV.

De acordo com o nível de interação, os sistemas de RV podem ser classificados como passivos, exploratórios ou interativos (ADAMS, 1994). Quanto ao grau de imersão, os sistemas ou estilos de RV podem ser classificados como RV de Simulação, RV de Projeção, *Visually Coupled Displays* ("Visores Visualmente Acoplados"), RV de Mesa, Realidade Aumentada, Virtualidade Aumentada e Hiper-realidade (PIMENTEL; TEIXEIRA, 1994; MILGRAM et al., 1994; TIFFIN; TERASHIMA, 2001). As duas próximas seções apresentam as classificações de RV com relação ao grau de interação e imersão do usuário com o AV de forma mais detalhada.

2.1.2.1 Grau de Interação Proporcionado do Usuário

Um ambiente virtual é composto por um determinado espaço e situações, ambos delimitados, incluindo todos os componentes neles inseridos, como o conjunto de condições e objetos possíveis de serem percebidos e com os quais é possível interagir. Um ambiente virtual

é, portanto, um ambiente interativo gerado por computador e disponibilizado através de um sistema de realidade virtual .

Como mencionado anteriormente, os sistemas de RV podem ser classificados de acordo com o modelo de participação do usuário, ou seja, de acordo com o nível de interação que este sistema oferece. Assim, Adams (1994) sugere que um aplicativo de RV pode proporcionar uma sessão sob três formas diferentes:

- **Passiva:** uma sessão de RV passiva proporciona ao usuário uma exploração do ambiente automática e sem interferência. A rota e os pontos de observação são explícitos e controlados exclusivamente pelo software. O usuário não tem controle algum, exceto talvez, para sair da sessão;
- **Exploratória:** uma sessão de RV exploratória proporciona uma exploração do ambiente dirigida pelo usuário. O participante pode escolher a rota e os pontos de observação, mas não pode interagir de outra forma com entidades contidas na cena;
- **Interativa:** uma sessão de RV interativa proporciona uma exploração do ambiente dirigida pelo usuário e, além disso, as entidades virtuais do ambiente respondem e reagem às ações do participante. Ou seja, ela oferece todas as possibilidades que eram restritas nas participações passiva e exploratória: o usuário tem liberdade de escolher a rota, o ponto de vista e interagir com os objetos do ambiente.

A navegação em ambientes virtuais é controlada pelo posicionamento do observador no mundo virtual, o chamado ponto de observação (*viewpoint*). A navegação é, provavelmente, a forma mais simples de interação encontrada em muitas aplicações de RV. Os dispositivos, associados a técnicas e linguagens computacionais, viabilizam a interação do usuário com a aplicação, permitindo a exploração do ambiente virtual e a modificação de objetos de maneira fácil e rápida - por exemplo, apontar, pegar, arrastar, rotacionar e visualizar os objetos sob diferentes pontos de vista.

2.1.2.2 Grau de Imersão Proporcionado ao Usuário

Dada a natureza artificial dos ambientes virtuais, é possível que os usuários e os elementos do próprio ambiente comportem-se como os complementos físicos do mundo real (HARRISON; DOURISH, 1996). Os usuários podem ser representados por avatares, que são objetos sintéticos inseridos no ambiente virtual, que podem navegar livremente nesse espaço,

encontrar artefatos, objetos ou dados e interagir utilizando canais visuais de comunicação (CHURCHILL; SNOWDON, 1998). Logo, existem diferentes formas de inserção ou imersão do usuário em relação ao ambiente ou mundo virtual.

Assim, Pimentel e Teixeira (1994) consideram que, de acordo com o grau de imersão, os sistemas ou estilos de RV podem ser classificados como RV de Simulação, RV de Projeção, *Visually Coupled Displays* ("Visores Visualmente Acoplados"), RV de Mesa e Realidade Aumentada. Milgram et al. (1994) abordam uma classe chamada Realidade Misturada como generalização da Realidade Aumentada e da Virtualidade Aumentada. Enquanto, Tiffin e Terashima (2001) descrevem uma outra classe de RV, chamada Hiper-realidade, como um paradigma para o terceiro milênio.

A RV de Simulação corresponde ao tipo mais antigo, originado com os simuladores de voo desenvolvidos pelos militares americanos após a Segunda Guerra Mundial (JACOBSON, 1994). Um sistema desse tipo basicamente imita o interior de um carro, avião ou jato, colocando o participante dentro de uma cabine com controles. Na cabine, telas de vídeo e monitores apresentam um mundo virtual que reage aos comandos do usuário. Uma vez que o sistema de RV de Simulação não processa imagens em estéreo, as imagens são geradas de forma bastante rápida. Em alguns sistemas as cabines são montadas sobre plataformas móveis, e os controles oferecem *feedback* tátil e auditivo.

A RV de Projeção também é conhecida como Realidade Artificial, e foi criada nos anos 70 por Myron Krueger. Na RV de Projeção o usuário está fora do mundo virtual, mas pode se comunicar com personagens ou objetos nele contidos. O sistema VIDEOPLACE (KRUEGER; GIONFRIDDO; HINRICHSEN, 1985), capturava a imagem do(s) usuário(s) e projetava-a em uma grande tela que representava um mundo virtual no qual os usuários podiam interagir uns com os outros ou com objetos. Krueger usou o termo Realidade Artificial para descrever o tipo de ambiente criado pelo seu sistema, que não exigia que o participante vestisse ou usasse dispositivos de entrada de dados (JACOBSON, 1994).

Os Visores Visualmente Acoplados (*Visually Coupled Displays*) correspondem a uma classe de sistemas na qual as imagens são exibidas diretamente ao usuário, que está olhando em um dispositivo que deve acompanhar os movimentos de sua cabeça. Esse dispositivo geralmente permite imagens e sons em estéreo, além de conter sensores especiais que detectam a movimentação da cabeça do usuário e usam essa informação para realimentação da imagem exibida (PIMENTEL; TEIXEIRA, 1994).

A RV de Mesa (*Desktop VR*) é um subconjunto dos sistemas tradicionais de RV em que, ao invés de capacetes de visualização (*headmounted displays* - HMD) são utilizados

grandes monitores ou algum sistema de projeção para apresentação do mundo virtual. Alguns sistemas permitem ao usuário ver imagens 3D no monitor com óculos obturadores (*shutter glasses*), polarizadores ou filtros coloridos (PIMENTEL; TEIXEIRA, 1994).

A Realidade Aumentada é usualmente definida como a sobreposição de objetos virtuais tridimensionais, gerados por computador, no ambiente real, por meio de algum dispositivo tecnológico (AZUMA, 1993; BOMAN, 1995). Onde, o ambiente predominante é real, porém enriquecido com objetos virtuais que sobrepõem o mesmo. Além de permitir que objetos virtuais possam ser introduzidos em ambientes reais, a Realidade Aumentada proporciona também ao usuário, o manuseio desses objetos com as próprias mãos, possibilitando uma interação atrativa e motivadora com o ambiente (BILLINGHURST; KATO; POUPYREV, 2001; ZHOU et al., 2004). No entanto, para que os objetos virtuais façam parte do ambiente real e sejam manuseados, deve-se utilizar um software com capacidade de visão do ambiente real e de posicionamento dos objetos virtuais, além de acionar dispositivos tecnológicos apropriados para Realidade Aumentada. Como exemplo, pode-se utilizar dispositivos visuais transparentes presos à cabeça (*heads-up-displays* (HUDs)) nos quais os dados são projetados. Pelo fato desses *displays* ou visores serem transparentes, o usuário pode ver dados, diagramas, animações e gráficos 3D sem deixar de enxergar o mundo real.

A Virtualidade Aumentada é a inserção de representações de elementos reais no mundo virtual, usando uma interface que permite ao usuário interagir com o ambiente virtual (KIRNER; KIRNER, 2007). Logo, sua principal característica consiste na predominância de um ambiente virtual envolvente. Além dos objetos estáticos pode-se, também, levar objetos reais dinâmicos para o cenário (ambiente virtual). Estes elementos reais podem ser pré-capturados ou ser capturados em tempo real por câmeras, neste caso o processo computacional faz a sua identificação e respectiva reconstrução no ambiente virtual. Milgram et al. (1994) acreditam que a conceituação de Realidade Aumentada e Virtualidade Aumentada é muito geral e só fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

A Realidade Misturada, misturando o real com o virtual, abrange duas possibilidades: a Realidade Aumentada, cujo ambiente predominante é o mundo real, e a Virtualidade Aumentada, cujo ambiente predominante é o mundo virtual. Dessa forma, Milgram et al. (1994) descrevem estes conceitos, em um contexto de Realidade-Virtualidade contínua, onde existe uma relação contínua entre o ambiente puramente virtual e o ambiente puramente real. Logo, a Realidade Misturada nada mais é que uma generalização da Virtualidade Aumentada e da Realidade Aumentada.

Por fim, a Hiper-realidade é a capacidade tecnológica de combinar realidade virtual,

realidade física, inteligência artificial e inteligência humana, integrando-as de forma natural para acesso do usuário (TIFFIN; TERASHIMA, 2001). Na Hiper-realidade o virtual é combinado com o real, incrementando-se novos elementos para facilitar e potencializar a interação do usuário com os recursos de que necessita-se para realização das tarefas.

2.1.3 Dispositivos de Realidade Virtual

Um convincente mundo virtual deve parecer natural aos participantes e a interação deve ser tanto intuitiva e transparente quanto possível. Para que isto seja alcançado, os sistemas de RV utilizam diversos dispositivos que permitam maximizar o grau de interação e imersão do usuário com o ambiente. Estes dispositivos, de modo geral, permitem a captura ou rastreamento de movimentos, além de fornecerem estímulos aos sentidos humanos. Logo, os dispositivos de RV atuam basicamente de duas formas: captando as entradas do usuário (geralmente movimentos do corpo) e estimulando sentidos como visão, audição e tato.

2.1.3.1 Dispositivos de Rastreamento

Também conhecidos como *tracking devices* ou ainda *motion trackers*, os dispositivos de rastreamento ou de trajetória permitem que um sistema de realidade virtual possa acompanhar a posição e orientação de partes selecionadas do corpo do usuário. Muitos dispositivos visuais e de interação incorporam algum dispositivo de rastreamento, a fim de medir a posição e orientação da parte do corpo ao qual estão associados, para ampliar a interação e a sensação de imersão em um AV.

Com base nas informações recebidas de um dispositivo de rastreamento, pode-se definir a perspectiva do usuário no mundo virtual, determinando qual parte do AV que deve ser “renderizada” para a visualização, bem como a geração de estímulos táteis e auditivos. Um *motion tracker* anexado a um HMD, por exemplo, mede a posição e orientação da cabeça continuamente para que a imagem gerada pelo computador se ajuste à representação da cena para a visão atual. Enquanto em uma luva, mede a posição e orientação da mão para que as respostas necessárias para a manipulação de objetos virtuais seja a mais natural possível.

Os dispositivos de rastreamento operam baseados na diferença de posição ou orientação em relação a um ponto ou estado de referência, de maneira ativa ou passiva. Enquanto no *tracking* passivo o alvo é rastreado a partir de uma ou mais câmeras que monitoram os movimentos do usuário a uma certa distância, no *tracking* ativo, são anexados sensores ao alvo a ser rastreado. Em termos de *hardware*, estes dispositivos geralmente são

formados por três componentes: uma fonte que gera um sinal, um sensor que recebe o sinal e um controle para processar o mesmo e comunicar-se com o computador. Dependendo da tecnologia empregada, ou a fonte ou o sensor é anexada ao corpo e a outra parte à um local fixo no ambiente, servindo como ponto de referência. Assim, pode-se medir as coordenadas de posição (x , y , z) e as coordenadas de orientação (*yaw*, *pitch*, e *roll*) obtendo dados suficientes para operar com dispositivos de interação com 6DOF (seis graus de liberdade).

Para localizar os movimentos do usuário com precisão dentro do espaço virtual, os atuais dispositivos de rastreamento são baseados em tecnologias mecânicas, acústicas, ópticas ou eletromagnéticas, cuja apresentação de cada uma destas abordagens segue abaixo:

- **Rastreadores Mecânicos:** estes dispositivos medem a posição e orientação através de uma conexão mecânica entre um ponto de referência e o alvo. Tipicamente, um braço mecânico articulado, conecta uma caixa de controle à cabeça, e codificadores localizadas nas juntas do braço medem a mudança de posição e orientação com relação ao ponto de referência.
- **Rastreadores Acústicos:** os rastreadores acústicos utilizam ondas sonoras ultra-sônicas (alta frequência) para medir a posição e orientação do objeto alvo. Com o uso de transmissores e receptores, pode-se medir o tempo de viagem (*time-of-flight*) do som entre a fonte emissora dos pulsos ultra-sônicos e a captação por um conjunto de receptores dispostos usualmente de maneira triangular e, assim, calcular a distância do alvo para cada um dos sensores.
- **Rastreadores Óticos:** estes rastreadores usam uma combinação de marcadores, tais como diodos emissores de luz (LEDs), câmeras de vídeo e técnicas de processamento de imagens. Uma variante utiliza algumas mini-câmeras que são posicionadas sobre o alvo a ser rastreado (cabeça ou mão do usuário) e um conjunto de LEDs (diodos emissores de luz) são fixados em uma estrutura acima do mesmo (no teto por exemplo). Em uma outra variante, ocorre o inverso, as câmeras são montadas no teto, ou uma estrutura fixa, e alguns LEDs são colocados em posições fixas e conhecidas sobre o alvo. Em ambos os casos, as projeções dos LEDs sobre os planos das imagens das câmeras permitem identificar a posição e orientação do objeto alvo que está sendo rastreado.
- **Rastreadores Eletromagnéticos:** os rastreadores eletromagnéticos possuem um transmissor ou fonte que emite campos eletromagnéticos ao longo de três eixos ortogonais que são detectados por um ou mais sensores. A partir dos sensores

pode-se determinar o tamanho e a direção destes campos e, por conseguinte, determinar a posição e orientação da fonte emissora (geralmente o usuário).

2.1.3.2 Dispositivos de Interação

As aplicações de *desktop* baseadas no paradigma WIMP (*Windows, Icons, Menus and Pointing Device*) usam dispositivos de entrada convencionais como o teclado, *mouse* e *joystick*. Estes dispositivos interativos básicos são fáceis de usar, mas eles também tendem a atrapalhar a sensação de imersão. Para que o usuário interaja com o sistema sem que perceba os dispositivos envolvidos, os pesquisadores e a indústria de RV procuram desenvolver dispositivos mais intuitivos como luvas, roupas, bolas isométricas, sensores biológicos, esteiras e esferas.

Os dispositivos isométricos ou bolas isométricas, em geral, consistem de uma bola sobre uma plataforma com botões que são configurados via *software*. Parecidos com um *mouse* ou um *joystick*, são bastante fáceis de manipular. Os sensores de entrada biológicos processam atividades denominadas indiretas, como comando de voz e sinais elétricos musculares. Em sistemas de RV o reconhecimento de comandos de voz pode facilitar a execução de tarefas no mundo virtual, principalmente quando as mãos estiverem ocupadas em outra tarefa e impossibilitadas de acessar o teclado. Já os dispositivos que utilizam sinais elétricos musculares detectam a atividade muscular por meio de eletrodos colocados sobre a pele, permitindo ao usuário movimentar-se no mundo virtual pelo simples movimento dos olhos, por exemplo.

Muitas vezes chamadas de *datagloves* ou *powergloves*, apesar de ambas as expressões se referirem a modelos particulares de luvas para interação em AVs, os diferentes tipos de luvas de RV compartilham o mesmo propósito: permitir ao usuário manipular dados de computador de uma forma intuitiva. Usando uma luva com fios, o usuário pode interagir com objetos virtuais fazendo vários gestos manuais como se estivesse interagindo com objetos reais. Geralmente o usuário é capaz de ver sua mão virtual no mundo virtual no qual se encontra inserido e assim poder coordenar seus movimentos naturalmente.

Algumas luvas medem a extensão do dedo através de uma série de cabos de fibra óptica. A luz passa através dos cabos a partir de um emissor para um sensor. A quantidade de luz que atinge o sensor muda dependendo de como o usuário mantém seus dedos: se ele curva seus dedos para o punho, menos luz atinge o sensor, que, por sua vez, envia esses dados ao computador do sistema de RV. Em geral, esse tipo de luva precisa ser calibrado para cada usuário para que funcione adequadamente. Outras luvas usam faixas de material flexível revestidas com uma tinta condutora de eletricidade para medir a posição do dedo do usuário.

À medida que o usuário dobra ou estende seus dedos, a resistência elétrica ao longo da faixa muda. O computador interpreta as mudanças na resistência e responde apropriadamente. Essas luvas são menos precisas que as luvas de fibra óptica, mas também costumam ser mais baratas. Para maior precisão e sensibilidade, foram desenvolvidas as luvas do tipo *dexterous hand master* (DHM). A DHM usa sensores anexados a cada junta dos dedos. O usuário conecta os sensores à cada uma de suas juntas através de *links* mecânicos, o que significa que a luva se parece com um exoesqueleto. Essas luvas são mais precisas do que as luvas de fibra óptica ou aquelas que usam material condutor de eletricidade, mas também são incômodas e desajeitadas.

Na tentativa de aumentar a sensação de imersão do usuário durante a navegação no ambiente virtual, alguns dispositivos de interação oferecem recursos para que o usuário possa caminhar dentro do AV sem sair de um local fixo (BOUGUILA; ISHII; SATO, 2002). Nesse contexto, dois dispositivos têm sido desenvolvidos e explorados recentemente: a esteira e a esfera.

Uma esteira é útil porque o usuário permanece parado em relação ao mundo real, mas sente-se como se estivesse, de fato, andando em um ambiente virtual. Uma limitação óbvia das esteiras normais é que você só pode andar em duas direções: para trás e para frente. Assim, algumas empresas desenvolveram esteiras multidirecionais, que ao contrário das esteiras normais que usam um único motor para exercer força tanto para frente quanto para trás em relação ao usuário, as esteiras multidirecionais usam dois motores: da perspectiva do usuário, a esteira pode exercer força para frente, para trás, para a esquerda e para a direita. Um exemplo desta tecnologia é o *CirculaFloor* desenvolvido por Iwata et al. (2004), que consiste num conjunto de placas móveis, equipadas com um mecanismo holonômico que propicia um movimento omnidirecional, como pode ser visto na Figura 2.1. O movimento dos pés do usuário é acompanhado por sensores de posição: o piso se move na direção oposta da direção do seu andar, de forma que o deslocamento do passo é cancelado. Seguindo esta linha de pesquisa, Iwata, Yano e Tomiyoshi (2007) ainda desenvolveram a *String Walker*, uma interface de locomoção que utiliza oito cordas acionadas por mecanismos de motor-polia montados sobre uma placa giratória em forma de anel, que permite ao usuário andar em um ambiente virtual em diversas direções, enquanto a sua posição é mantida.

A empresa VirtuSphere, Inc.⁴ oferece uma maneira ainda mais versátil de locomoção dentro de um ambiente virtual. O dispositivo consiste de uma esfera de plástico de gigante que se parece com uma roda de *hamster* em tamanho humano. Ao entrar na esfera, o usuário

⁴VIRTUSPHERE - A New Locomotion Technology for Immersing in Virtual Reality - Disponível em: <<http://www.virtusphere.com>>

Figura 2.1: Dispositivo de interação do tipo esteira chamado *CirculaFloor* (IWATA et al., 2004).

portando um capacete de visualização (HMD) pode andar, deitar, rolar, pular ou se arrastar sobre ela para locomover-se pelo mundo virtual. A esfera, que pode ser vista na Figura 2.2, é posicionada sobre uma plataforma estável que tem várias rodas que se apóiam contra a mesma, o que permite que ela gire em qualquer direção ao mesmo tempo que permanece em uma posição fixa. Sensores nas rodas informam ao computador em qual caminho o usuário está andando e a visão dentro do HMD do usuário muda de acordo com o caminho. Outras tecnologias semelhantes, como *Cybersphere* e o *Cyberwalk* são descritas respectivamente em (FERNANDES; RAJA; EYRE, 2003; SCHWAIGER; THÜMMEL; ULBRICH, 2007).

Figura 2.2: Dispositivo de interação do tipo esfera da VirtuSphere.

2.1.3.3 Dispositivos visuais

Os dispositivos visuais são dispositivos que apresentam visualmente para o usuário o mundo 3D e influenciam fortemente a percepção do nível de imersão de um sistema de

RV. Alguns autores chegam a dividir ou classificar as pesquisas sobre RV de acordo com os dispositivos de visualização empregados (CRUZ-NEIRA, 1995). Com base nos dispositivos visuais, os sistemas de RV podem ser monoscópicos ou estereoscópicos: no primeiro caso, uma única imagem é “renderizada”⁵ e exibida para os dois olhos, no segundo, cada olho observa uma imagem ligeiramente diferente, cada qual “renderizada” separadamente (PIMENTEL; TEIXEIRA, 1994).

Os sistemas de exibição estereoscópica ou 3D permitem ao observador/usuário ver os objetos ou o ambiente gerado além das dimensões da tela do computador, acrescentando profundidade à visualização e, conseqüentemente, tornando-a mais realista e próxima da forma que o mundo real é visto. Em linhas gerais, pode-se dizer que a estereoscopia, conceito no qual esses sistemas de exibição são baseados, trabalha com o fato de que cada um dos olhos humanos, estando ligeiramente afastados um do outro, vê imagens ligeiramente diferentes. Essas imagens são reunidas no cérebro e compreendidas com características de profundidade, distância, posição e tamanho (MACHADO, 2003).

Para a exibição estereoscópica das imagens existem diversos dispositivos utilizados, sendo que a escolha de um determinado tipo de sistema de exibição dependerá da finalidade do sistema de RV. Pimentel e Teixeira (1994) separam estes dispositivos em duas classes: a primeira composta pelos capacetes de visualização (*head-mounted displays*) e os dispositivos binoculares suspensos *head-coupled displays* ou *arm-mounted displays* e, a segunda composta pelos monitores de vídeo e telas de projeção. O que diferencia essas duas classes é que, na primeira, o dispositivo possui sensores para detectar os movimentos do usuário, enquanto que na segunda isso não ocorre, e o rastreamento depende dos comandos do usuário via outro dispositivo de entrada.

Os capacetes de visualização ou *head-mounted displays* (HMDs) são dispositivos para serem acoplados/vestidos à cabeça e são constituídos de duas telas/monitores pequenos, um à frente de cada olho, um conjunto de lentes especiais e um sistema óptico que canaliza as imagens aos olhos do observador. As lentes ajudam a focalizar imagens que estão a alguns milímetros dos olhos do usuário, ajudando também a ampliar o campo de visão do vídeo auxiliando na apresentação estéreo de um mundo virtual. Geralmente integram sistemas de áudio e detecção de trajetória, com sensores de rastreamento que medem a posição e orientação da cabeça. A imagem do mundo virtual é mostrada de acordo com o ponto de visão do usuário, enquanto o mesmo explora o ambiente. A medida que o computador recebe os dados de posição e orientação, é gerado uma sequência de imagens por quadro correspondente às

⁵“Renderização” é uma adaptação o termo em inglês “*rendering*” e, refere-se ao processo de geração de uma imagem a partir de um modelo, através de técnicas de computação gráfica.

ações e perspectivas do usuário, fornecendo um alto grau de imersão ao mesmo (GRADECKI, 1995).

Outra forma de imergir num ambiente de Realidade Virtual é utilizando os dispositivos binoculares suspensos (*head-coupled displays*), que consiste em um par de lentes binoculares montado sobre um braço mecânico articulado, com sensores ligados a este braço e controles próximos ao *display* (lentes) (PIMENTEL; TEIXEIRA, 1994) . Também conhecidos como *arm-mounted displays* (AMD's), estes dispositivos integram detecção de movimento no braço mecânico que os suporta, permitindo movimentos em até 6 graus de liberdade, com movimentos restritos pelo tamanho do braço (ARAUJO, 1996). O formato do binoculares suspensos, Figura 2.3, permite uma transição entre a visualização do mundo virtual e a interação com teclados, monitores e outros dispositivos que possam auxiliar no controle da simulação.

Figura 2.3: Dispositivo binocular suspenso.

Os dispositivos visualização baseados em monitores de vídeo representam a categoria de dispositivos do tipo *desktop displays*. Este paradigma evoluiu-se a partir dos monitores de computadores gráficos tradicionais, onde estações de trabalho gráficas acopladas com acessórios para as mãos e cabeça permitem que as imagens exibidas na tela respondam aos movimentos do usuário na frente do monitor. Diretamente derivado das aplicações gráficas computacionais interativas tradicionais, este sistema posiciona o observador externamente ao ambiente virtual, usando a tela do monitor como uma janela para o mundo virtual, por conseguinte provendo um sentido muito limitado de imersão (FISHER, 1991).

Quanto aos sistemas baseados no uso de telas de projeção, estes podem ser de constituídos de uma tela simples ou múltiplas telas. Os sistemas de projeção baseado em tela simples, em sua maioria, usam a metáfora de uma mesa (*desk-style display*), na qual os objetos são projetos em sua superfície. Em geral, sistemas de tela simples são boas opções para aplicações que requerem manipulação de objetos diretamente localizados na frente do usuário.

Estes sistemas são menores que as cavernas digitais (múltiplas telas), o que oferece uma certa mobilidade no sentido de que eles podem ser facilmente desmontados e transportados. Neste tipo de dispositivo, o usuário fica em frente a uma tela vertical inclinável, como uma tela de pintura, com tamanhos típicos entre 150-240 cm, onde o usuário pode interagir com os objetos virtuais projetados sobre e “acima” da tela (HOFFMANN; STEFANI; PATEL, 2006). Um grande exemplo deste tipo de sistema é o ImmersaDesk (CZERNUSZENKO et al., 1997).

Os sistemas de projeção baseado em múltiplas telas surgiram com uma nova proposta de interface para sistemas de RV. Neste tipo de sistema as imagens são projetadas em telas dispostas de uma maneira que envolvam o observador ou usuário do sistema, dando a sensação de imersão. Um grande exemplo deste sistema de projeção é o ambiente de realidade virtual CAVE (*Cave Automatic Virtual Environment*), onde a ilusão de imersão é criada pela projeção de gráficos estereoscópicos de computador nas paredes e chão de uma sala em forma de cubo, que envolve completamente o usuário em uma espécie de “caverna” digital (CRUZ-NEIRA, 1995). As paredes, teto e chão são telas semitransparentes onde as imagens são projetadas a partir de projetores posicionados atrás das telas. Sistemas do tipo CAVE também incorporam projeção acústica tridimensional, dispositivos de rastreamento de posição e de interação, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual e interajam com os objetos virtuais projetados neste ambiente. A projeção das imagens é feita por projetores posicionados atrás das telas e pode ser estereoscópica, exigindo dos usuários o uso de óculos obturadores ou estereoscópicos (*shutter/stereoscopic glasses*). Mais detalhes a respeito desse sistema serão abordados na Seção 2.1.3.6.

Tanto para os monitores como para as telas de projeção, existem três processos principais utilizados para a exibição de imagens estereoscópicas: anaglifo, luz polarizada ou luz intermitente. No processo por anaglifo as imagens de cada olho são exibidas em cores complementares, como vermelho e azul (ou vermelho e verde). As imagens são observadas com óculos que tem a mesma correspondência de cores (os chamados filtros), permitindo a cada olho ver a sua respectiva imagem. Esta técnica é empregada em alguns tipos de cinema 3D, onde uma componente direita da imagem, geralmente na cor vermelha, é sobreposta à componente esquerda em uma cor contrastante, geralmente o verde ou azul, para produzir um efeito 3D quando o observador olha através dos filtros coloridos (óculos). No entanto, é o conhecido que o uso prolongado desses óculos cansam os olhos e causam dor de cabeça (LIPTON, 1982; VINCE, 2004). No processo por luz polarizada, as imagens do par estereoscópico são projetadas em planos ortogonais, e no processo por luz intermitente cada imagem do par estereoscópico é apresentada de forma alternada em uma frequência rápida e imperceptível ao ser humano. Por apresentarem duas imagens diferentes no mesmo plano de exibição,

independente do método de exibição, estes dispositivos exigem o uso de algum tipo de dispositivo periférico de visualização pelo usuário para a separação das imagens, tal como óculos coloridos (ou de filtros), óculos obturadores (ou estereoscópicos) (*shutter/stereoscopic glasses*) ou óculos polarizadores (NETTO; MACHADO; OLIVEIRA, 2002).

Sistemas baseados no uso de monitores auto-estereoscópicos também já estão disponíveis (KAUFMANN; LIU; BURRIS, 2000; DODGSON, 2005). Os monitores auto-estereoscópicos por sua vez já têm um filtro aplicado à frente desdobrando a visualização da imagem de forma a apresentar uma imagem diferente para cada olho quando olhamos para qualquer ponto do monitor. Assim, esses dispositivos auto-estereoscópicos permitem visualização estéreo ou 3D sem a necessidade do uso de óculos especiais ou algum outro periférico adicional para acoplar à cabeça (BENTON, 2001).

2.1.3.4 Dispositivos Auditivos

Vivemos em um mundo com um afluxo constante de sons que nos dizem muito sobre o meio envolvente. O eco e reverberação, por exemplo, nos dão pistas sobre a direção e distância de objetos, além de uma noção a respeito das proporções do espaço físico ao qual estamos inseridos. Através do som, podemos localizar objetos que estão fora de nossa vista, como um controle remoto ou um celular. Não obstante, o som nos oferece informações sobre o tipo de material, a textura e a dureza de uma superfície. Logo, a habilidade de sintetizar o som espacial aliada a informação visual traz muitos benefícios para a realidade virtual imersiva.

Entre os sistemas de áudio existentes, encontram-se os sistemas para espacialização e os sistemas de auralização. Os primeiros sugerem campos sonoros envolventes e uma sensação de espacialidade, enquanto os últimos têm como objetivo primário simular ou reproduzirem com o máximo de fidelidade possível campos sonoros 2D/3D legítimos. Dessa forma, os sistemas de auralização são usualmente mais precisos na capacidade de prover a percepção de direção e distância (posição) dos objetos sonoros dispostos na cena, bem como reproduzir com mais realismo as características acústicas de uma cena (FARIA, 2005).

A fim de adquirir uma compreensão clara do som espacial, é importante distinguir o som mono, estéreo, *surround* e binaural de som 3D. Uma gravação de som mono é uma gravação de som com um microfone. Não existe um senso de posicionamento do som presente neste tipo de gravação. O som estéreo é gravado com dois microfones distantes um do outro, separados por um espaço vazio (ar) simulando o posicionamento dos ouvidos. Quando uma gravação estéreo é reproduzida, a gravação de um microfone entra no ouvido esquerdo, enquanto que a gravação a partir do outro microfone é canalizada para o ouvido direito. Isto

dá uma ideia da posição em que o som foi gravado pelos microfones. No entanto, geralmente tem-se a sensação de que a fonte emissora do som esteja dentro da cabeça do ouvinte - isto acontece porque os seres humanos geralmente não ouvem os sons da maneira como eles foram gravados em estéreo, separados por um espaço vazio, e a cabeça humana atua como um filtro na captura dos sons. No som envolvente (*surround*), adiciona-se mais microfones e mais alto-falantes para adquirir uma melhor ilusão de posicionamento do som, no entanto ainda não é possível obter-se a localização precisa dos objetos sonoros.

A maior parte dos sistemas comerciais em utilização no mercado presta-se mais à criação de efeitos *surround* e espacialização para filmes e entretenimento de modo geral do que propriamente à reprodução de campos sonoros 2D/3D coerentes. Uma vez que, para este mercado, não é importante a localização precisa dos objetos sonoros. Já para os sistemas de realidade virtual, somente a espacialização de eventos sonoros não é suficiente para impor o realismo necessário. Falta a percepção da diretividade e da localização dos objetos no espaço, sugerindo o uso de sistemas para geração de sons 3D.

Faria (2005) destaca que entre os métodos de síntese de áudio tridimensional (3D), a percepção de bidimensionalidade e tridimensionalidade é maior na medida em que se usam técnicas de auralização binaurais ou multicanais. As técnicas de auralização permitem simular com mais eficácia a sensação de imersão em campos sonoros 3D. A gravação binaural parece mais realista, uma vez que o som é gravado de uma maneira mais próxima do sistema acústico humano, envolvendo modelos de propagação sonora formais e simulações para obter o efeito desejado. Existem várias técnicas de gravação e reprodução binaural, como apresentadas em (LOKKI et al., 2002; SUSNIK et al., 2003; MCDERMOTT, 2007). McDermott (2007) descreve uma das técnicas bastante utilizadas, que envolve o conceito de modelagem do ouvinte. Ou seja, um modelo exato da cabeça humana é usado para “ouvir” a uma variedade de sons a partir uma série de locais e orientações diferentes em torno do avatar. O modelo pode ser um “boneco” ou até mesmo uma pessoa comum. Os microfones são então embutidos nas orelhas do modelo para captar os sinais sonoros, representando a forma como o ouvido humano recebe sons. Esses sinais são analisados e o percurso dos sons entre a fonte e cada ouvido são adequadamente modelados por meio de funções de transferência relativas à cabeça (HRTF ou *Head Related Transfer Function*). Através desta técnica, procura-se obter uma série de filtros que podem ser aplicados aos sons, a fim de reproduzir a espacialização percebida.

O uso de som tridimensional pode melhorar as capacidades dos sistemas de RV dramaticamente, contribuindo para a sensação de imersividade e orientação em um ambiente virtual (DOERR et al., 2007). Os sistemas de som 3D duplicam artificialmente os ativadores

naturais que auxiliam o cérebro a localizar o som, além de recriarem eletronicamente esses efeitos em tempo real (JACOBSON, 1994). O usuário pode ser capaz de interagir de forma mais eficaz dentro de um ambiente de RV se suas ações forem acompanhados por sons apropriados que parecem ser emitidos a partir de sua própria localização (BEGAULT, 1994). Assim, os dispositivos auditivos de RV, que nada mais são do que dispositivos geradores de som 3D, auxiliam na aquisição de informações sobre sua localização além do que é fornecido pelo canal visual, tornando a experiência muito mais realista.

Segundo Jacobson (1994), as gravações de som tridimensional baseiam-se em um processo de manipulação auditiva que permite que o artista ou o engenheiro de gravação “posicione” os sons no espaço, controlando sua direção, distância e profundidade. Existem diversas placas de som projetadas para trabalhar com conjuntos de ferramentas que constroem mundos virtuais auditivos. Algumas ferramentas que são usualmente utilizadas em sistemas e ambientes voltados para aplicações de realidade virtual imersivas, como a Avango, FMOD, VSS, VRJuggler Sound Library, Bergen Sound Server & Library, Wave Engine, AuSIM3D e Lake Huron, são apresentadas em (FARIA, 2005)

2.1.3.5 Dispositivos Hápticos

Em oposição aos dispositivos convencionais como teclado, *mouse* e monitor, os dispositivos hápticos procuram ampliar a interação homem-computador na exploração de mundos virtuais, através de uma interação direta com os nossos sentidos. Para uma melhor interação com um ambiente, deve haver algum tipo de retorno ou resposta (*feedback*) ao pegar ou tocar um objeto virtual, por exemplo. Este tipo de retorno é chamado de retorno háptico (*haptic feedback*) (JEONG; HASHIMOTO; MAKOTO, 2004).

Os dispositivos hápticos incorporam sensores e atuadores que permitem o monitoramento das ações do usuário e podem fornecer-lhe um retorno tanto tátil como de força. O retorno tátil está associado à natureza do contato com o objeto virtual (OV), como a percepção de geometria, temperatura, textura e vibração. Enquanto o retorno de força refere-se ao senso de posição e movimentação, reproduzindo as forças direcionais resultantes de um movimento, o peso ou pressão e a inércia de um OV. Assim, ao tocar ou pegar os objetos do mundo 3D têm-se a sensação de estar manipulando um objeto semelhante ao objeto real, aumentando o senso de imersão no ambiente. A tecnologia de atuadores é bastante variada, sendo que os motores elétricos, pistões hidráulicos e bolsas pneumáticas são os tipos de atuadores mais utilizados e permitem o envio de sensação de força, temperatura e diversas propriedades táteis (BURDEA, 1996).

Os dispositivos hápticos que fornecem sensação de força podem ser divididos em duas categorias básicas: fixos (*ground-based*) ou móveis (*body-based*) (KUNZLER; RUNDE, 2005). Os dispositivos hápticos fixos, como os *joysticks*, são aqueles que estão fisicamente atrelados a uma plataforma ou superfície estável que permite o envio de reações de força ao usuário. Já os dispositivos hápticos móveis utilizam um ponto de conexão do próprio dispositivo para fornecer a reação de força e apresentam a vantagem de poderem ser portáteis na maioria das vezes, como as luvas, roupas e exoesqueletos.

Entre os dispositivos hápticos fixos, encontram-se os dispositivos baseados em braços mecânicos articulados, como pode ser visto na Figura 2.4. O braço mecânico rastreia a posição da terminação e é capaz de exercer força e vibração na ponta da mesma. Como alternativa à este tipo de sistema, Cai, Ishii e Sato (1996) e Kim, Koike e Sato (2002) apresentam dispositivos a base de tensão, onde ao invés da força ser aplicada através das articulações de um braço mecânico, cabos são conectados ao ponto de contacto, a fim de exercer uma força sobre o mesmo. Através de motores para criar uma tensão no cabo e codificadores para determinar o comprimento de cada cabo, obtém-se um retorno de força durante a manipulação dos objetos.

Figura 2.4: Dispositivos hápticos fixos: Falcon da Novint (NOVINT, 2008) e Phantom da SensAble (SENSABLE, 2008) respectivamente.

Com relação aos dispositivos hápticos móveis, encontram-se os dispositivos baseados em um exoesqueleto mecânico que se encaixa no corpo do usuário, fazendo com que determinados movimentos permitam-lhe sentir o peso ou a resistência do material de um objeto no mundo virtual. Alguns sistemas transmitem reação de força apenas para as mãos e braços em formas de luvas, como pode ser visto na Figura 2.5.

Figura 2.5: Dispositivos hápticos CyberTouch, CyberGrasp, CyberForce e Haptic Workstation (CYBERGLOVE, 2012).

2.1.3.6 CAVE

Os sistemas de exibição estereoscópica⁶ ou 3D permitem ao observador/usuário ver os objetos ou o ambiente gerado além das dimensões da tela do computador, acrescentando profundidade à visualização e, conseqüentemente, tornando-a mais realista e próxima da forma que o mundo real é visto. Desde o desenvolvimento dos primeiros capacetes de visualização (HMDs) no fim dos anos 80 até o início dos anos 90, vários dispositivos de exibição de imersão espacial, denominados SIDs (*Spatially Immersive Displays*), foram desenvolvidos como uma proposta de interface para sistemas de RV.

No início de 1991, os pesquisadores Carolina Cruz-Neira, Thomas A. DeFanti e Daniel J. Sandin, do Laboratório de Visualização Eletrônica (*Electronic Visualization Laboratory*) da Universidade de Illinois em Chicago, projetaram um ambiente de multiprojeção estereoscópico voltado para aplicações imersivas de realidade virtual, baseado em grandes telas de projeção. Pelo projeto, as imagens seriam projetadas em telas dispostas na forma de um cubo, para envolver o observador ou usuário do sistema e prover a sensação de imersão. Este sistema foi anunciado e demonstrado em uma conferência de computação gráfica e técnicas interativas, o SIGGRAPH'92, com o título: *"The CAVE: Audio Visual Experience Automatic Virtual Environment"* (CRUZ-NEIRA et al., 1992).

Cruz-Neira, Sandin e DeFanti (1993) afirmam que o nome "CAVE" é tanto um acrônimo recursivo (*CAVE Automatic Virtual Environment*) quanto uma referência ao "Mito da Caverna", narrado por Platão no livro VII de sua obra "A República", que retrata a exploração da percepção, ilusão e realidade através do uso de metáforas, como a projeção de sombras nas paredes da caverna. Logo, a concepção deste sistema, consiste de uma estrutura em formato cúbico, como uma sala ou uma "caverna digital", onde as paredes e o chão são formadas por telas de projeção. Através de projetores posicionados atrás das telas que formam as paredes e acima do chão deste ambiente, são projetadas imagens estereoscópicas de alta resolução, que quando vistas através de óculos obturadores ou estereoscópicos (*shutter/stereoscopic glasses*), os usuários que encontram-se dentro do volume das telas obtêm uma sensação imediata e intensa de imersão em um ambiente virtual. Além da projeção de imagens 3D, a CAVE também incorpora projeção acústica tridimensional, dispositivos de rastreamento de posição e de interação, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual e interajam com os objetos virtuais projetados neste ambiente.

Os sistemas de visualização do tipo CAVE apresentam ao usuário uma representação estereoscópica em escala natural de um ambiente tridimensional gerado por computador. Logo,

⁶A visualização estereoscópica permite a percepção de profundidade e a sensação de espaço.

o ambiente possui uma relação direta com as dimensões humanas que através de uma visão referenciada pelo *tracking* da cabeça e a interação realística com objetos virtuais, via *data glove* e outros dispositivos, viabiliza a manipulação, operação e controle de mundos virtuais. Ou seja, o sistema oferece uma interface para a navegação no espaço 3D com as capacidades de *look-around*⁷, *walk-around*⁸ e *fly-through*⁹ em um mundo virtual.

Apesar do objetivo inicial do desenvolvimento da CAVE ter sido traçado como uma ferramenta útil de visualização científica (CRUZ-NEIRA, 1995), ela acabou sendo utilizada não apenas no meio acadêmico como também na indústria, governos, museus e entretenimento. Por ser uma das mais completas tecnologias de visualização de dados (SHERMAN; CRAIG, 2002), a CAVE tornou-se uma importante ferramenta de pesquisa, cuja vasta gama de aplicações abrange as áreas de engenharia, arquitetura, medicina, arte, negócios e muitas outras. Algumas empresas de produção, por exemplo, utilizam os recursos de um dispositivo CAVE para exibirem seus produtos à clientes potenciais ou então corrigir eventuais problemas relacionados com seus produtos antes que eles sejam submetidos ao processo de manufatura (CREAGH, 2003). A Figura 2.6, exibe projetistas experimentando e avaliando o interior de um carro, dentro de um sistema CAVE no Centro de Desenvolvimento da Mercedes-Benz¹⁰.

Figura 2.6: Projetistas experimentando e avaliando o interior de um carro dentro de um CAVE.

Uma grande vantagem da CAVE em relação a outros sistemas imersivos é a possibilidade de compartilhamento do mundo virtual por vários usuários simultaneamente. A quantidade de pessoas que podem compartilhar do mesmo ambiente, está limitada diretamente

⁷*Look-around* refere-se a um tipo de interação onde o usuário pode olhar o ambiente virtual em várias direções.

⁸*Walk-through* refere-se a um tipo de interação onde o usuário pode “andar” ou mover-se pelo ambiente virtual restrito a caminhos ou lugares pré-definidos.

⁹*Fly-through* é semelhante ao *walk-around*, porém sem a restrição dos movimentos quanto a caminhos ou lugares específicos no mundo virtual.

¹⁰<http://www.emercedesbenz.com>

às dimensões físicas da caverna utilizada. A estrutura computacional envolvida no acionamento de CAVEs é bastante avançada e deve processar os pares estereoscópicos das imagens, além de gerenciar os dispositivos de interação, auditivos e projetores. A Figura 2.7 mostra um esquema que descreve um sistema CAVE com o posicionamento dos projetores. Na figura apresentada, tem-se o uso de espelhos para ajustar a distância dos projetores em relação às telas.

Figura 2.7: Esquema que descreve a posição dos projetores em um sistema do tipo CAVE.

Por ser um sistema de projeção cúbica, estes sistemas podem ser implementados a partir de diferentes configurações, que podem variar de 3 a 6 faces (ou paredes) de projeção. Raskar et al. (1998) propuseram a criação do “escritório do futuro”, onde as paredes do escritório seriam utilizadas para projeção das imagens e o usuário poderia interagir tanto com seu próprio ambiente de trabalho como com outros ambientes remotos, servindo como estudo para aplicações na telepresença e realidade aumentada. Gross et al. (2003) descrevem um sistema de telepresença chamado *blue-c*, que utilizando um sistema CAVE, um conjunto de câmeras 3D e telas de vidro semitransparentes podem criar a impressão de total imersão para desenvolvimento de produtos e colaboração. Nesta mesma linha, Rhee e Park (2007) adicionam câmeras de vídeo de baixo custo a um sistema CAVE existente, com o auxílio de algoritmos de imagem sofisticados que viabilizem a sensação de presença e colaboração por um custo reduzido. Modelos mais modestos como o de Jacobson (2002), por exemplo, são formados por apenas 3 faces, onde as imagens são projetadas em 2 paredes posicionadas em forma de V, além de uma projeção direta no chão. Este tipo de caverna é mais simples e barata, porém a sensação de imersão é menor (do que um sistema com 6 faces) e oferece menos mobilidade ao usuário, pois o mesmo fica limitado a um ponto central do ambiente.

A construção de um sistema de multiprojeção imersivo demanda uma série de conhecimentos interdisciplinares, como a implementação e controle do sistema de projeção, desenvolvimento de softwares de rede, controle de múltiplos equipamentos, renderização das

imagens em tempo real, *softwares* de vídeo e áudio, projeção sincronizada, arquitetura e design, etc. Além disso, por se tratar de uma tecnologia recente e de alto custo, muitos trabalhos envolvendo esta tecnologia ainda estão em fase exploração e desenvolvimento.

O *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos* disponibiliza um dispositivo do tipo CAVE cuja infraestrutura é descrita no Capítulo 3.

2.1.4 Dispositivos de Telepresença

Os sistemas de telepresença referem-se a um conjunto de tecnologias que permitem a sensação de presença em um local remoto, a partir de uma localização diferente da que pretende-se interagir (STEUER, 1992; FISHER, 1991). Muito similar à realidade virtual, onde luta-se para alcançar a ilusão de presença dentro de uma simulação gerada por computador, a telepresença luta para alcançar a ilusão de presença em um local remoto. Logo a ideia da telepresença, lembra a RV, exceto que na RV o usuário está imerso em um mundo virtual, ao passo que na telepresença o usuário está imerso em um mundo real remoto. Assim, Rosenberg (1995) afirma que no fim, tanto a telepresença como a realidade virtual são essencialmente o mesmo, uma interface humano-computador que permite ao usuário tirar vantagem de suas habilidades naturais quando interage com um outro ambiente que não aquele que o rodeia.

Como a telepresença é a habilidade de perceber e agir em um ambiente remoto e não apenas percebê-lo, o usuário precisa receber estímulos a partir do local remoto para que o mesmo possa ter a habilidade de exercer o controle ou a manipulação de objetos dentro deste ambiente distante (CHAPIN; LACEY; LEIFER, 1994). Dessa forma, recursos como vídeo, áudio, dispositivos hápticos, sistemas de visualização, como HMDs (*Head Mounted Displays*), telas de exibição auto-estereoscópicas¹¹, auto-falantes estéreo, luvas e outros dispositivos de RV, podem ser utilizados pelo operador do sistema remoto.

As tecnologias dos sistemas de telepresença, do ponto de vista do usuário, requerem que o controle seja exercido pelo operador humano e dispositivos tais como sistemas de rastreamento (*tracking*) de cabeça e corpo, *joysticks*, luvas e estruturas em forma de exoesqueleto, entre outros controladores de aplicação específica. Quanto a tecnologia utilizada no ambiente remoto, grande parte desta é semelhante ao encontrado na área da robótica, mais precisamente a tele-robótica, uma vez que os sistemas de telepresença não exigem operação autônoma. Os sistemas de telepresença por definição requerem um laço humano na operação ou interação com o ambiente. Dessa forma, os equipamentos que podem ser encontrados no

¹¹Dispositivos de exibição autoestereoscópicos permitem a visualização e percepção de imagens 3D sem a necessidade de óculos especiais.

ambiente remoto incluem sensores, microfones, câmeras, manipuladores entre outros.

2.1.5 Bibliotecas de Desenvolvimento de Ambientes Virtuais

Com a avanço no estudo de tecnologias de realidade virtual, programas e bibliotecas vêm sendo desenvolvidos para os mais diversos tipos de sistemas de RV. Algumas interfaces de programação de aplicação (API - *Application Programmer's Interface*), como a CAVELib e a VR Juggler, são específicas para dispositivos de exibição imersivos, enquanto outras como a OpenGL e a Direct3D são de propósito geral para programação 3D.

A OpenGL (*Open Graphics Library*) é uma biblioteca gráfica aberta composta por funções e estruturas de dados que constitui uma interface portátil de alto nível para acesso ao *hardware* gráfico. Essa interface consiste basicamente de funções que podem ser utilizadas para especificar objetos e operações necessárias para produzir aplicações 3D interativas, oferecendo ao desenvolvedor um pequeno conjunto de primitivas geométricas. A biblioteca Direct3D, faz parte da tecnologia DirectX da Microsoft, e fornece um conjunto de funcionalidades de grande utilidade para o desenvolvimento de aplicações 3D, sendo considerada padrão na plataforma Windows. Ela apresenta uma interface de alto nível, permitindo acesso aos recursos avançados de hardware das placas gráficas 3D, sem a necessidade de conhecer as particularidades de cada *hardware*.

Como a OpenGL e a Direct3D, são bibliotecas de baixo nível, que se concentram basicamente na renderização das cenas, foram desenvolvidas bibliotecas baseadas nesta tecnologia que se concentram na estruturação das cenas 3D, como a API Java 3D. O Java 3D consiste em uma API baseada nas bibliotecas gráficas OpenGL e DirectX, e refere-se a uma hierarquia de classes Java que serve como interface para o desenvolvimento de sistemas gráficos tridimensionais, onde o programador necessita trabalhar apenas com construtores de alto nível para criar, manipular e visualizar objetos geométricos, tendo os detalhes de visualização gerenciados automaticamente (BROWN; PETERSEN, 1999). Ainda com base na OpenGL e a Direct3D, foram desenvolvidas as chamadas *engines*, um pacote de software orientado à cena, que permite a construção de aplicações 3D de um modo mais fácil e flexível. As *engines* abstraem os detalhes e funcionalidades de bibliotecas de baixo nível como a OpenGL e a Direct3D e oferece uma interface baseada em objetos e classes mais intuitivas.

Um grande exemplo de uso das *engines*, está no desenvolvimento de jogos de computadores. Através de um *game engine* (motor de jogo ou motor 3D), têm-se um pacote de *software* que lida com as principais funcionalidades do jogo. Ou seja, um código que define e processa todas as possibilidades de um determinado jogo, tais como: gráficos, sons, inteligência

artificial, rede, controle entre outras. Algumas das particularidades deste conceito são a sua modularidade, extensibilidade e facilidade de reutilização. Além disso, os melhores motores de jogos permitem ainda que os usuários possam modificar cenários, modelos, sons e outras componentes do jogo (os chamados *mods*), tornando as possibilidades ainda maiores. Dessa forma, alguns autores e pesquisadores de aplicações de realidade virtual, como Jacobson e Lewis (2005), Tan, Hashimoto e Sato (2005), Ryu et al. (2006), utilizam ferramentas baseadas em motores 3D de jogos para implementação de mundos virtuais em um dispositivo de exibição imersivo.

Com relação as linguagens de RV destacam-se a VRML e o X3D. A VRML (*Virtual Reality Modeling Language*), como o próprio nome sugere é uma linguagem para modelagem de realidade virtual, que consiste em um formato de arquivo onde é possível criar objetos tridimensionais através da descrição ou definição de cor, transparência, brilho e textura. Os objetos podem ser formas básicas, como esferas, cubos, ovoides, hexaedros, cones, cilindros, ou formas criadas pelo próprio programador, como as extrusões (CAREY, 1998). Tido como o sucessor da VRML, o X3D (*Extensible 3D*) tem como sua principal característica a capacidade de codificar a cena usando a sintaxe da XML (*eXtensible Markup Language*). O X3D é um padrão aberto para distribuir conteúdo 3D. Ele não é uma API de programação, tão pouco um formato de arquivo para troca de geometrias. Ele combina ambos, geometria e descrições de comportamentos instantâneos em um simples arquivo que tem inúmeros formatos de arquivos disponíveis para isso, incluindo o XML (DALY; BRUTZMAN, 2007).

Inicialmente, muitas aplicações 3D para RV eram programadas diretamente a partir da biblioteca gráfica OpenGL ou alguma outra biblioteca similar. Posteriormente, foram desenvolvidos *kits* ou sistemas de autoria de ambientes virtuais ou ainda bibliotecas específicas pra RV, como o ARTToolkit. O ARTToolkit¹² é uma biblioteca desenvolvida no Laboratório de Tecnologias de Interface Humana da Universidade de Washington, que permite a criação de aplicações de realidade aumentada. Para isso, ela utiliza as capacidades de rastreamento de vídeo a fim de calcular a posição e orientação de uma câmera com relação a marcadores físicos em tempo real.

O ambiente de desenvolvimento de programas aplicativos para sistemas de multi-projeção estereoscópicas é complexo, pois leva em conta aspectos como a representação de grandes estruturas de dados, o gerenciamento de periféricos e o mapeamento das imagens nos projetores. Neste contexto, encontra-se a CAVELib¹³, que é a API original desenvolvida pelo Laboratório de Visualização da Universidade de Illinois em Chicago, criadores

¹²<http://www.hitl.washington.edu/artoolkit/>

¹³<http://www.mechdyne.com>

da CAVE. Esta API provê o suporte geral para a construção de ambientes virtuais para sistemas de multiexibição imersivos e equipamentos de realidade virtual. A CAVELib permite a configuração dos dispositivos de exibição, a sincronização dos múltiplos processos, mapeamento das vistas estereoscópicas, comunicação com os dispositivos de rastreamento, suporte ao modelo de câmera virtual baseado no ponto de vista do observador e os recursos básicos de comunicação em rede entre ambientes de realidade virtual remotos.

Em alternativa à CAVELib, foram desenvolvidas outras bibliotecas para os sistemas de exibição imersivos, em cujo cenário destaca-se a VR Juggler (BIERBAUM et al., 2001). O VR Juggler é um projeto de pesquisa do Centro de Aplicações para RV da Iowa State University, e trata-se de uma biblioteca de realidade virtual, de código aberto, multiplataforma e independente de dispositivo. Ela fornece uma abstração dos dispositivos de entrada e saída, podendo ser utilizada em conjunto com qualquer biblioteca de grafo de cena (OpenGL Performer, OpenSceneGraph, OpenSG) para renderizar a cena.

Com o uso do VR Juggler, o desenvolvedor não precisa se preocupar com o dispositivo de entrada que o usuário irá utilizar para interagir com a aplicação, ou como ele irá visualizá-la. A escolha dos dispositivos é feita através de arquivos de configuração. Baseado nesses arquivos, a biblioteca carrega os *drivers* adequados. A grande vantagem dos arquivos de configuração é permitir que diferentes dispositivos sejam combinados de várias maneiras sem necessidade de recompilar a aplicação. Assim, a aplicação que usa o VRJuggler não precisa ser desenvolvida levando em consideração o tipo ou o número de projetores ou telas usadas para a exibição, nem da disposição delas, nem precisa ser previamente preparada para ser controlada por um determinado dispositivo. O usuário pode configurar a aplicação para rodar com um *joystick* em uma CAVE ou utilizar um *mouse* 3-D imerso em um capacete de realidade virtual, apenas mudando alguns parâmetros do arquivo. Além de sistemas do tipo CAVE, o VR Juggler suporta uma variedade de dispositivos de RV, como mesas de projeção, monitores simples de estações de trabalho e os tradicionais HMDs.

2.1.6 Aplicações de Ambientes Virtuais e Telepresença

Um ambiente virtual pode representar qualquer mundo tridimensional real ou abstrato. Isso inclui sistemas reais como edifícios, paisagens, submarinos, naves espaciais, locais de escavação arqueológica, anatomia humana, esculturas, reconstrução da cena de um crime, sistemas solares, e assim por diante. Estes mundos virtuais podem ser animados, interativos, compartilhados e podem expor comportamento e funcionalidade.

Muito similar aos ambientes virtuais, onde luta-se para alcançar a ilusão de presença

dentro de uma simulação gerada por computador, a telepresença luta para alcançar a ilusão de presença em um local remoto com a habilidade de perceber e agir em um ambiente distante através da manipulação e controle dos objetos reais remotos (CHAPIN; LACEY; LEIFER, 1994). Assim, a combinação de realidade virtual e telepresença (teleoperação e tele-robótica) permitem a criação e utilização de um vasto número de aplicativos que podem ser utilizados nos mais diversos campos do pesquisa, trabalho, exploração, inspeção e manutenção.

Portanto, tal como as tecnologias de realidade virtual e telepresença evoluem, as aplicações de RV e telepresença tornam-se literalmente ilimitadas. Adams (1994), acredita que a RV é uma reformulação da interface entre pessoas e tecnologia da informação, oferecendo novas formas de comunicação de informações, a visualização de processos, bem como a expressão criativa das ideias. Assim, as aplicações da RV podem incluir a formação de uma variedade de áreas: militar, médica, equipamentos de operação, educação, concepção de avaliação (prototipagem virtual), estudos ergonômicos e arquitetônicos, simulação de sequências de montagem e manutenção tarefas, estudo e tratamento de fobias (por exemplo, medo de altura), entretenimento, manufatura virtual (PORTO et al., 2002) e muito mais.

Publicações de trabalhos que empregam os sistemas de realidade virtual e telepresença têm sido descritos para os mais diversos fins, como a simulação de missões críticas na medicina (SORID; MOORE, 2000; PARK et al., 2007); modelagem e testes de colisão na indústria automobilística (FELEZ; VERA; MARTINEZ, 1999; SCHWEBELA; GAINESA; SEVERSON, 2008); estudos de biologia molecular (GILLET et al., 2004); educação e entretenimento (HUGHES et al., 2005); tratamento de fobias (JUAN et al., 2005; TAKACS; SIMON, 2007); simulação do comportamento de satélites no espaço (KIMURA; IBARAKI, 1999) e simulação, inspeção e treinamento com aeronaves (VORAA et al., 2002).

Buscando novas formas de obter uma melhoria organizacional das empresas e propiciar a modernização do sistema produtivo, alguns pesquisadores desenvolveram aplicações baseadas em tecnologia de RV e telepresença para treinamento de funcionários (PARK; JANG; CHAI, 2006); controle e planejamento da produção (SCHENK; STRASSBURGER; KISSNER, 2005); visualização de *layouts* de fábricas e prototipação de produtos (HUANG et al., 2006)); processos de engenharia (CORSEUIL et al., 2003) e controle de estoques (MONTEIRO; PAIVA, 2002).

2.2 Visão Computacional

De modo geral, a Visão Computacional (VC) pode ser definida como o estudo de métodos e técnicas que permitam que algum agente possa “ver” o ambiente em que ele

atua através da aquisição de dados desse ambiente. Por exemplo, um sistema que verifica automaticamente a situação legal de veículos, através do reconhecimento de suas placas através de imagens.

A modelagem e construção de grandes modelos 3D a partir de imagens é uma meta ambiciosa da computação gráfica e da visão computacional. O objetivo final é interagir diretamente com modelos digitais 3D criados a partir de imagens de objetos reais. O alcance deste objetivo permite a telepresença, a realidade virtual, a simulação e o treinamento (ALIAGA; ZHANG; BOUTIN, 2007).

Embora os sistemas de VC podem diferir no tipo de entrada e saída dos dados, bem como o tipo de algoritmo utilizado, muitos seguem uma arquitetura comum, como apresentada na Figura 2.8.

Figura 2.8: Etapas de um sistema de visão computacional genérico (CONCI; AZEVEDO; LETA, 2008)

A etapa de aquisição, geralmente refere-se à aquisição de imagens, no entanto, pode-se adquirir outros tipos de dados como de sensores *laser*, infravermelhos, etc. Neste trabalho, será abordado somente imagens como dados de entrada do sistema.

2.2.1 Aquisição de Imagens

Segundo Conci, Azevedo e Leta (2008), o primeiro passo na conversão de uma cena real tridimensional em uma imagem eletrônica é a redução de dimensionalidade. A câmera fotográfica ou a câmera de vídeo converte cenas 3D em representações 2D da imagem. A Figura 2.9 apresenta uma ilustração desse processo.

Uma imagem pode ser considerada uma distribuição de energia luminosa em uma

Figura 2.9: Aquisição da imagem (CONCI; AZEVEDO; LETA, 2008)

posição espacial. Na Figura 2.9, pode-se observar que a iluminação distribui energia sobre o objeto. Parte dessa energia luminosa é absorvida pelo objeto, parte é transmitida e parte é refletida, sendo capturada pela câmera.

Observando-se a Figura 2.9, tem-se que a imagem formada na câmera pode ser expressa por uma função de cada ponto (x, y) , conforme apresentado na Equação 2.1.

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (2.1)$$

onde $i(x, y)$ é a função de iluminação e $r(x, y)$ é a função de reflexão do objeto. A função $f(x, y)$ descreve a energia luminosa da imagem na coordenada espacial (x, y) . Essa função é convertida em sinal elétrico pelo dispositivo de captura e transmitida para um computador. Com base nessa informação, torna-se possível desenvolver algoritmos de processamento de imagens, análise de imagens, visão computacional e e outras áreas da computação gráfica que usam imagens digitais como dado de entrada.

2.2.1.1 Câmera CCD

O dispositivo de aquisição de imagens mais utilizado atualmente é a câmera CCD (*Charge Couple Device*), que consiste em uma matriz de células semicondutoras fotossensíveis que atuam como capacitores, armazenando carga elétrica proporcional à energia luminosa incidente.

Uma câmera CCD monocromática simples possui um conjunto de lentes que focalizam a imagem sobre a área fotossensível do CCD, um sensor CCD e seus circuitos complementares. Para aquisição de imagens coloridas utilizando-se CCD, é necessário um conjunto de prismas

e filtros de cor encarregados de decompor a imagem colorida em suas componentes R (*red*), G (*green*) e B (*blue*), cada qual capturada por um CCD independente.

2.2.1.2 Amostragem e Quantização

Existem dois conceitos importantes relacionados a imagem digital: amostragem da imagem (*sampling*) e quantização da imagem (*quantization*). Ambos são propriedades relativas ao processamento computacional de uma imagem, sendo o primeiro referente ao número de pontos amostrados de uma imagem digitalizada e o segundo referente à quantidade de níveis que podem ser atribuídos a cada ponto digitalizado (CONCI; AZEVEDO; LETA, 2008). Quanto maior é a resolução ou a quantidade de pontos (x, y) da imagem, mais detalhes podem ser armazenados e, conseqüentemente, maior será o espaço gasto em armazenamento.

A imagem digital pode ser representada por uma matriz bidimensional de *pixels*. Cada *pixel* é representado no computador como um número inteiro que corresponde à intensidade de luz no ponto. Frequentemente, a cor do *pixel* é representada como um inteiro de 8 *bits* variando entre 0 e 255. Em uma imagem em níveis de cinza por exemplo, o valor 0 correspondente à cor preta, 255 à cor branca e os outros níveis são distribuídos entre esses valores.

A resolução e os níveis de cor servem para tornar uma imagem mais nítida e definida. Portanto, imagens com poucos detalhes podem ser digitalizadas com poucos níveis e com baixa resolução, ao passo que imagens com grande riqueza de detalhes devem ser digitalizadas com alta resolução e uma quantidade de níveis de cor bem elevada.

2.2.2 Pré-processamento

As câmeras digitais de hoje podem capturar imagens com uma resolução de 10 *megapixels* ou mais. Dessa forma, a imagem a ser analisada pode conter muitos dados para serem processados. Como resultado, um dos passos mais comuns de pré-processamento é diminuir a resolução da imagem para uma resolução que seja mais fácil de operar.

Outros passos do pré-processamento incluem a conversão de imagens coloridas em níveis de cinza, ou a filtragem de uma imagem para a remoção de ruídos através de técnicas de restauração. Além disso, a etapa de pré-processamento pode também ajustar uma imagem para corrigir um pouco a iluminação, falta de contraste, correção de foco, imagens borradas por movimento, etc.

2.2.3 Segmentação

Como o próprio nome sugere, a segmentação de imagens consiste em segmentar uma imagem em partes menores para posterior extração de atributos e cálculo de parâmetros descritivos. Existem dois tipos principais de segmentação: em regiões e em pontos de interesse. A primeira agrupa *pixels* em regiões que cobrem toda a imagem. Já a segunda tenta produzir uma representação esparsa da imagem, com base em “pontos de interesse”, ou em locais da imagem onde é provável que se encontra algum elemento de interesse na imagem.

Ao longo dos anos, diferentes abordagens têm sido desenvolvidas e uma descrição completa das principais abordagens pode ser encontrada em Fu e Mui (1981) e Pal e Pal (1993).

Dentre os métodos abordados na literatura, destacam-se os métodos de segmentação de imagens baseados em bordas. Estes têm sido utilizados desde os primeiros trabalhos de Roberts (1963). Estes métodos baseiam-se na intensidade dos valores dos *pixels* para orientar o processo de segmentação. Os detectores de bordas, como são comumente chamados esses tipos de métodos, calculam os pontos de fronteira (bordas) utilizando um operador diferencial local. Entre os operadores diferenciais mais utilizados, Gonzalez e Woods (2002) descreve em detalhes uma variedade deles como: Robert, Sobel, Canny, Prewitt, entre outros. Entre os detectores de bordas existentes, Gonzalez e Woods (2002) destaca o detector de bordas Canny como um detector de bordas ótimo por tentar otimizar o processo de detecção e localização das bordas, produzindo uma resposta única para cada ponto de borda.

2.2.4 Extração de Características

Utilizando-se imagens já segmentadas, pode-se extrair características relevantes a respeito das regiões ou dos pontos de interesse destacados, como: quantidade total de elementos, dimensões, geometria, propriedades de luminosidade e textura (CONCI; AZEVEDO; LETA, 2008).

Assim, a etapa de extração de características permite a descoberta de valores de tamanho, área, perímetro, centro de massa, etc. Além disso, a forma dos objetos pode ser avaliada, com base em parâmetros de circularidade, retilineidade, concavidade, entre outros. A partir desses valores, pode-se tentar classificar e reconhecer os segmentos da imagem na etapa seguinte.

2.2.5 Classificação e Reconhecimento

Uma vez que as características das regiões ou pontos de interesse da imagem encontram-se disponíveis, pode-se utilizar uma base de imagens apenas para classificar essas regiões de acordo com sua semelhança. A partir dessa base de imagens, segmentos de novas imagens podem ser reconhecidos como pertencentes a uma das classes previamente definidas.

Existem diversas técnicas de classificação que podem ser encontradas no trabalho de Lulio (2011). As mais simples implicam processos de agrupamento estatístico, para os quais se necessita de alguma intervenção humana, e são chamadas de técnicas supervisionadas. As mais sofisticadas permitem ao computador reconhecer diferentes objetos através de técnicas com pouca ou nenhuma intervenção humana, conhecidas como técnicas não-supervisionadas.

2.3 Visão Computacional e Cenas Naturais

Soh e Tsatsoulis (1999) diferencia as cenas naturais de cenas artificiais pelo tipo de objetos presentes na cena. As cenas naturais são compostas de elementos como florestas, culturas agrícolas, rios, rochas, solos, montanhas, nuvens, etc. Enquanto as cenas artificiais são compostas de elementos criados pelo homem como edifícios, veículos, rodovias, áreas urbanas, etc.

A análise destes dois tipos de cenas se diferencia exatamente pelo tipo de elementos que as compõem. As bordas em cenas artificiais são geralmente mais nítidas e menos complexas do que aquela em cenas naturais, e isso geralmente facilita o processo de segmentação destas cenas. Já as cenas naturais correspondem a um ambiente não estruturado onde a aparência dos elementos pode variar grandemente com base na área geográfica, na estação, nas condições do clima atual ou nas condições do clima em épocas passadas.

A distribuição dos objetos naturais em um determinado ambiente, pode ser afetada pela poluição química, a erosão do solo, precipitação e umidade do solo, seca ou perturbação de regimes hidrológicos, queimadas, etc. Todas essas peculiaridades das cenas naturais juntamente com a exigência de análise automática ou semi-automática de dados como imagens terrestres ou de satélite, têm sido um desafio para os pesquisadores que investigam novas abordagens para a segmentação e análise de imagens destes ambientes.

2.3.1 Visão Computacional e Cenas Agrícolas

No contexto de ambientes agrícolas, o uso da visão computacional, para analisar imagens destes cenários, tem muitas aplicações potenciais como: identificação de pragas, monitoramento de crescimento e detecção de doenças (TILLET, 1991). Desta forma, Dunn (2007) divide a análise de imagens de cenas naturais na agricultura em quatro principais áreas:

- localização;
- rastreamento;
- classificação / reconhecimento;
- navegação/medição.

A localização de um objeto em uma imagem é uma das formas mais básicas de análise de imagens. O método mais simples de localização de objetos é através da aplicação de um limiar para segmentar uma imagem em regiões-alvo e não-alvo. Figueiredo et al. (2004) descreve um sistema de análise de comportamento de aves, baseado neste método.

A localização pode também ser conseguida através do conhecimento prévio sobre o objeto de interesse, como a aplicação descrita por Philipp e Rath (2002), que implementou um sistema de detecção de haste e região de cálice de uma maçã. Esta pesquisa utilizou a luz como forma de auxiliar na detecção de objetos por reflexões geradas em torno da área do tronco. Assim a haste era detectada a partir da forma das reflexões. Outros trabalhos envolvendo a localização de frutos em pomares de laranjas e maçãs podem ser encontrados em Plebe e Grasso (2001) e Stajniko, Lakota e Hocevar (2004), que utilizam imagens RGB e imagens térmicas respectivamente.

O rastreamento de um objeto de interesse tem importantes implicações em estudos com animais, como análise de seu bem-estar e investigações comportamentais. Sergeant, Boyle e Forbes (1998) apresenta um trabalho sobre análise do comportamento de aves baseado no rastreamento de centroides, enquanto Zelek, Bruce e Kanwar (2000) e Frost et al. (2004) descrevem o monitoramento de suínos utilizando detecção de bordas e delineamento de contornos.

Na área de classificação de objetos de cenas agrícolas, existem trabalhos na classificação de frutas e vegetais, detecção de ervas daninhas e produtos hortifrutícolas, como descrito ao longo desta seção. O conceito básico da maioria das aplicações é classificar o objeto de interesse em classes pré-definidas, com base em alguns elementos ou características

presentes e detectáveis. Bennedsen, Peterson e Tabb (2005) identifica defeitos em imagens de maçãs rotativas utilizando limiares, enquanto Davies et al. (2003) utilizou um detector de segmentos de linhas vetorial para detecção da presença de insetos grãos de arroz.

Yadav e Jindal (2001) investigou a utilização de imagens em nível de cinza para determinar a percentagem de grãos de arroz quebrados após a moagem, enquanto comprimentos de onda de vermelho, verde e infravermelho foram usados por Marchant, Andersen e Onyango (2001) para classificar vegetação a partir de uma cena de fundo. Enquanto Maxwell et al. (2004) utiliza o processamento de imagem de grande escala proveniente de satélite para geração de mapas de cobertura do solo.

O espaço de cores CIE-Luv¹⁴ é utilizado por Tantaswadi et al. (1999) para medir a distância euclidiana entre as cores para discriminar defeitos no algodão. Já Hasankhani e Navid (2012) descreve a classificação de batatas baseado no tamanho e cor dos objetos segmentados das imagens.

Jr et al. (2003) utilizou limiares para classificação de plantas de milho danificadas, baseados no histograma de níveis de verde. E Lee, Slaughter e Giles (1999) relata um sistema de visão artificial para detecção de alvos de ervas daninhas. Neste último, as imagens foram segmentadas em uma imagem binária utilizando classificadores Bayesianos.

Kondo et al. (2000), Guyer e Yang (2000) utilizaram redes neurais artificiais para classificação de frutos de laranjeiras e cerejeiras respectivamente. Enquanto Goel et al. (2003) utilizou RNA e árvores de decisão sobre imagens aéreas (com 72 comprimentos de onda) de cultivo de milho para determinar a taxa de aplicação de nitrogênio e infestação de plantas daninhas, e, Huang et al. (2007) usou informação multi-espectral para identificar a doença ferrugem amarela em culturas de trigo.

E por fim, uma importante área relacionada com a análise de cenas naturais, refere-se a navegação de máquinas ou veículos autônomos através ao longo de um cenário agrícola. Neste tipo de análise, as soluções de visão de máquina procuram identificar e rastrear um alvo vegetativo, ao invés de uma linha pré-definida. Isto permite gerar desvios do veículo quando necessário, para assegurar que a integridade da cultura seja mantida.

Slaughter, Chen e Curley (1999) utilizou uma janela (ROI - *Region Of Interest*) trapezoidal para determinar a posição da linha de plantio, baseando-se na média das posições obtidas para localização da linha de interesse dentro da região de interesse. Seguindo a mesma estratégia Billingsley (2000) descreve um sistema de visão implementado em um trator

¹⁴CIE-Luv é um espaço de cores adotado pela Comissão Internacional de Iluminação (CIE - *Commission Internationale de l'Eclairage*)

utilizando também janelas em uma região de interesse (ROI) trapezoidal para o rastreamento de linhas de plantio.

Tillett, Onyango e Marchant (1997) utilizou um modelo de filtro de Kalman de uma série de linhas de cultura, para executar o controle preditivo de máquinas agrícolas. Ao passo que Sogaard e Olsen (2003) descreve um sistema que utiliza o histograma dos níveis de cinza das imagens aplicado a uma regressão dos mínimos quadrados para detectar linhas de plantio.

A implementação de um filtro de Kalman estendido que é atualizado com a posição do objeto de interesse detectado com uma modificação da Transformada de Hough é apresentado por Hague, Marchant e Tillett (2000). A metodologia adotada por Hague, Marchant e Tillett (2000) usa o conhecimento prévio do tipo de linha e espaçamento entre linhas para auxiliar na detecção e rastreamento.

Ainda na área da navegação, Lulio (2011) apresenta o desenvolvimento uma arquitetura de aquisição, processamento e análise de imagens com o objetivo de segmentar, classificar e reconhecer padrões de navegação das linhas de plantio, como referências de guiagem de um robô móvel, entre plantações de laranja, milho e cana.

No campo da medição, as análises de cenas naturais oferecem a possibilidade de obtenção de dados sobre objetos de interesse sem a necessidade do contato físico direto. Schofield et al. (1999), por exemplo, utilizou múltiplas medições da área de superfície visível de suínos, com vista de cima, para estimar o peso atual e o ganho de peso de suínos ao longo de diferentes dias de amostragem. A área de superfície dos suínos foi obtida com a medição de intensidades dos *pixels* utilizando limiares. Enquanto Greenand e Whittemore (2003) relatou a extensão deste trabalho para a detecção do tipo de suíno usando uma RNA com funções de base radial.

2.4 Construção de Ambientes Virtuais a partir de Cenas Naturais

A geração de ambientes virtuais de áreas urbanas ou cenas naturais coloca um grande número de problemas no âmbito da Computação Gráfica, dado que a quantidade de informação necessária para a criação de modelos realistas é dependente da dimensão e complexidade da área a modelar. A construção de um grande número de modelos de objetos naturais de forma detalhada é extremamente trabalhosa. Enquanto os modelos de estruturas artificiais, tais como máquinas ou edifícios podem ser obtidos a partir de fontes CAD, o mesmo não ocorre

as plantas e outros fenômenos naturais.

O Google tem um projeto chamado Your World in 3D¹⁵ (Seu Mundo em 3D) que visa transformar todas as cidades do planeta em modelos tridimensionais para serem exibidas através do Google Earth e Google Maps. No entanto, essa transformação não é automática. Através de uma comunidade de desenvolvedores de objetos 3D, o Google convoca modeladores do mundo todo para que através de seu *software*, o Google Sketchup, partes de cidades sejam modeladas e enviadas ao Google Earth e Google Maps. A localização de cada objeto é fornecida pelo próprio Google, bem como sua escala, mas a modelagem geométrica dos objetos vai depender de cada desenvolvedor voluntário.

Embora muitos ambientes virtuais sejam criados por modelagem manual e individual de cada um dos seus componentes, a pesquisa de processos automáticos e semi-automáticos de reconstrução de ambientes naturais 3D provaram que podem ser muito mais eficientes, reduzindo a duração, o custo e a alocação de recursos humanos (PIMENTEL et al., 2001). Neste contexto, os trabalhos de pesquisa tem se concentrado no processamento de imagens 2D do ambiente real para construção do ambiente virtual 3D. Através desta abordagem, o uso de fotografias ou imagens de vídeo são utilizadas para extração de informações geométricas 3D de prédios, casas e paisagens.

O processo de reconstrução virtual baseado em imagens pode ser dividido em duas partes: na primeira buscam-se formas automáticas ou semi-automáticas de extração de características (ou informações) das imagens para detectar as construções, e na segunda, efetua-se a reconstrução propriamente dita.

Com relação a modelagem de paisagens naturais, os trabalhos encontrados na literatura se concentram na reconstrução de elementos individuais destas paisagens, como árvores e rochas. Reche, Martin e Drettakis (2004), por exemplo, apresentam um método de reconstrução volumétrica de árvores 3D a partir da segmentação de imagens 2D reais. Já Shlyakhter et al. (2001) apresentam um método de reconstrução de plantas baseado em silhuetas extraídas dessas mesmas imagens. Outras abordagens de construção de modelos de plantas 3D baseado no uso de fotografias pode ser visto em (ZENG; ZHANG; ZHAN, 2006; HAOPENG et al., 2010). Em todos os casos apresentados, as abordagens limitam-se ao reconhecimento e reconstrução de árvores isoladas, ou seja, onde o fundo contém apenas o céu ou algum outro objeto que não se confunda com a árvore.

Outros trabalhos como de Yang e Jie (2010) e Yongjian et al. (2009) realizam a segmentação manual, através do uso de ferramentas de edição de imagens, para posterior

¹⁵<http://sketchup.google.com/intl/en/yourworldin3d/index.html>

construção ou modelagem de objetos 3D de um paisagem natural. Já Xu et al. (2010) apresentam um projeto de uma base de dados de modelos 3D para simplificar o processo de reconstrução de cenas naturais com base num sistema de informação geográfica (SIG). No entanto, através desta abordagem além da necessidade de configurar uma arquivo com os atributos de consulta manualmente, o usuário deve participar da escolha do modelo final a ser inserido na cena de modo interativo.

Devido a complexidade de modelagem dos componentes de um paisagem natural, Feng-quan e Gui-gang (2009) e Zeng e Wang (2008) utilizam algoritmos fractais para construção do objetos como vegetação de forma interativa. No entanto, essa abordagem não se preocupa com a localização dos objetos, apenas com a quantidade a ser distribuída no ambiente. Já Wang et al. (2008) primeiro um método de reconstrução de plantas 3D, através de fotografias de fatias de plantas individuais para posterior reconstrução do modelo virtual.

(TIAN-YANG; JING; XIN-PEI, 2008) apresentam um estudo sobre a modelagem de árvores de uma paisagem natural para o acompanhamento do crescimento das plantas, porém o uso de imagens é utilizado apenas para a modelagem inicial, enquanto o ambiente virtual reproduz o crescimento através da simulação do fenômeno. Enquanto Wei-dong, Chen e Ping-ping (2010) realizam a modelagem de plantas 3D, também para estudos de crescimento, mas através de dados experimentais.

Com o intuito de reconstruir grandes áreas urbanas a partir de imagens reais, as pesquisas na área de realidade virtual (RV), outra estratégia tem sido o uso de imagens aéreas obtidas por aeronaves e satélites (STROTHOFF et al., 2011). A justificativa é que imagens aéreas e imagens obtidas por satélites com alta resolução permitem reconstruir grandes áreas com poucas imagens. No entanto, como as imagens aéreas fornecem somente o topo das edificações e paisagens, as pesquisas desenvolvidas nesta área ainda não apresentam resultados muito precisos e são objetos de interesse de diferentes pesquisadores. Cerdas, Descombes e Zerubia (2003) apresentam uma metodologia de construção de cenas urbanas e paisagens naturais através de imagens aéreas, combinada com o uso de conhecimento prévio sobre a geometria dos objetos na imagem.

Alguns trabalhos de construção de ambientes virtuais (AVs) a partir de cenas naturais, referem-se ao uso de uma tecnologia para sensoriamento remoto chamada LiDAR, sigla para *Light Detection and Ranging*. Em português, esta tecnologia vem sendo utilizada com a terminologia de Sistema de Varredoura a Laser (CENTENO; MITISHITA, 2007) ou de Perfilamento a Laser (SILVA; RIBAS, 2007).

A tecnologia LiDAR é baseada nos mesmos princípios utilizados no sistema de radar,

com a diferença que, ao invés do uso de ondas de rádio para localizar os objetos de interesse, o sistema LiDAR utiliza pulsos laser. Os sistemas que utilizam esta tecnologia são chamados de sistemas *scanner laser* aerotransportado (ALS - *Airborne Laser Scanner*). Eles são compostos por um sensor laser que essencialmente registra o tempo entre o sinal emitido e recebido de um determinado ponto, um Sistema Inercial de Navegação (INS - *Inertial Navigation System*) e um receptor GPS (Sistema de Posicionamento Global) em uma plataforma aerotransportada.

Uma vantagem dos sensores de varredura a laser em relação aos clássicos sensores passivos (fotografias aéreas e imagens de satélite) é que os *scanners* a laser não dependem do sol como uma fonte de iluminação. Sendo assim, a análise dos dados não é prejudicada por sombras causadas por nuvens ou objetos vizinhos (WAGNER et al., 2004). Outra característica importante deste sensor é que os pulsos de laser podem penetrar por pequenas aberturas no dossel de florestas e fornecer informações sobre a estrutura florestal e também sobre o relevo destas áreas.

Giongo et al. (2007) cita o uso desta tecnologia na área florestal, para estimativas de alturas, delimitação de árvores individuais em uma área de reflorestamento e estimativas volumétricas de povoamento de eucaliptos para inventário florestal. No entanto, no caso de vegetações as medições diretas obtidas dos dados LiDAR são as alturas do dossel da floresta, ou seja, a topografia das copas, volume e diâmetro de copa. A identificação de espécies requer a fusão com outros tipos de sensores.

Trabalhos nesta área podem ser encontrados tanto para reconstrução de edifícios de paisagens urbanas (CHENG et al., 2011; WENG; ZHAO; HE, 2010; VERMA; KUMAR; HSU, 2006), como para reconstrução de paisagens naturais como florestas (AHLBERG et al., 2004). Apesar dos avanços nesta área de pesquisa, a construção dos modelos virtuais se resume a identificação de características do topo dos edifícios ou do topo das copas das árvores. No caso dos edifícios, utiliza-se diferentes técnicas de processamento dos dados para reconstrução dos telhados e o restante do objeto é formado por um bloco geométrico para imprimir a altura correta do objeto na cena. Os detalhes de silhueta do objeto acabam sendo estimados ou ignorados.

Ainda na área de dispositivos a laser, Hong-wei et al. (2011), Binney e Sukhatme (2009) e Moorthy et al. (2007) utilizam dados de um *scanner* a laser terrestre para reconstrução de plantas de paisagens naturais, através da identificação de galhos e folhas da copa das árvores. No entanto, Zhu et al. (2009) relatam que o processo de aquisição dos dados baseados em *scanners* é um pouco demorado e a construção de modelos virtuais em tempo real.

2.5 Considerações Finais

Este capítulo buscou introduzir os principais conceitos de realidade virtual, além de apresentar e discutir os principais trabalhos relacionados à construção de ambientes virtuais a partir de cenas naturais. Dada a grande quantidade e diversidade de trabalhos publicados sobre o tema, não é possível atingir o esgotamento da literatura. No entanto, com base no conteúdo apresentado até aqui, uma conclusão preliminar é de que não há uma metodologia completa de montagem de ambientes virtuais agrícolas a partir de cenas naturais, utilizando uma busca de objetos previamente modelados. Como apresentado na Seção 2.4, a grande maioria dos trabalhos pesquisados na literatura se concentram na área de reconstrução de objetos 3D a partir de imagens reais ou outras fontes de dados como *scanners* a laser. Essa reconstrução pode ser realizada de forma automática ou de forma semi-automática através da intervenção do usuário.

Neste ponto é possível identificar uma lacuna na literatura, que diz respeito à escassez de metodologias de montagem de ambientes virtuais, que permitam o uso da visão computacional apenas para extração de características dos objetos de interesse sem a necessidade de reconstruir os objetos segmentados das imagens. A integração entre diferentes tecnologias e ferramentas que possibilitem a identificação de elementos em um cenário agrícola e a posterior apresentação e utilização em um sistema do tipo CAVE não é uma tarefa trivial. Dessa forma, outra lacuna identificada, foi a ausência de metodologias que descrevem todo o processo de montagem de ambientes virtuais a partir de cenas naturais, do momento da aquisição dos dados (da imagem) até a apresentação e utilização em um sistema de multiprojeção imersivo do tipo CAVE.

3 Laboratório Multiusuário de Visualização 3D Imersiva

O Laboratório Multiusuário de Visualização Imersiva de São Carlos (LMVI) é um projeto financiado pela FAPESP, sob coordenação do Professor Dr. Arthur José Vieira Porto. O LMVI disponibiliza um dispositivo do tipo CAVE formado por três telas de projeção: chão, lado e frente. Através deste projeto, os pesquisadores da região de São Carlos têm uma infraestrutura de visualização 3D imersiva disponível para o desenvolvimento de pesquisas multidisciplinares.

Esta estrutura foi utilizada como ferramenta para realização dos testes de aplicação da metodologia de montagem de ambientes virtuais agrícolas, tanto para visualização como interação com os ambientes projetados. No entanto, após a aquisição e instalação da infraestrutura física do LMVI, uma das etapas de desenvolvimento deste trabalho, foi o estudo e implementação de toda a infraestrutura de *software* desse sistema com o uso de ferramentas de código livre. Além da pesquisa e da escolha das ferramentas e bibliotecas de programação necessárias para o desenvolvimento deste projeto, algumas soluções de interação e disparo de som 3D foram criadas para utilização do sistema.

O desenvolvimento da infraestrutura de *software* e as soluções criadas especificamente para o LMVI foi fruto de um trabalho em conjunto com um outro pesquisador membro do projeto. Os resultados desse desenvolvimento foram publicados pelos dois autores em Oliveira et al. (2009) e Vale et al. (2009).

Neste capítulo será apresentada uma descrição detalhada deste dispositivo, abordando desde sua estrutura física às bibliotecas utilizadas para implementação e controle do sistema.

3.1 Espaço Físico

O Laboratório Multiusuário de Visualização Imersiva (LMVI) está localizado junto ao prédio do NEPAS (Núcleo de Ensino e Pesquisa em Automação e Simulação) do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos da Universidade de São Paulo.

A estrutura física que abriga o LMVI é composta por um prédio construído exclusivamente para comportar o sistema de multiprojeção imersivo. Dentro deste prédio, foi montada uma sala com 10 metros de largura, 10 metros de comprimento e 5 metros de altura. As paredes da sala são compostas por divisórias em *eucatex* e o teto por forro de PVC com luminárias fluorescentes. O chão é forrado com carpete para auxiliar na absorção do som 5.1, reduzindo o nível reverberação no ambiente.

Como as telas de projeção oferecem melhores resultados em um ambiente escuro, aumentando o contraste preto/branco e a resolução de imagem, tanto as paredes quanto o teto da sala que abriga o sistema foram pintadas de preto, que junto com o carpete cinza escuro colaboram para que o ambiente fique o mais escuro possível durante a projeção, como pode ser visto na Figura 3.1.

Figura 3.1: Paredes e chão da sala do LMVI

3.2 Estrutura de Projeção

Dentre as várias possibilidades de sistemas de multiprojeção imersivos, o NEPAS (Núcleo de Ensino e Pesquisa em Automação e Simulação) da Escola de Engenharia de São Carlos optou por montar um dispositivo do tipo CAVE de 3 faces, onde 2 delas correspondem à 2 paredes laterais e a terceira corresponde ao chão.

As paredes são formadas por telas de projeção posicionadas em forma de V, com um ângulo de 90 graus entre elas. Para cada face existe um par de projetores responsáveis pela

projeção das imagens estereoscópicas. A projeção das imagens nas paredes laterais é realizada a pela parte de trás das telas, enquanto o chão recebe uma projeção pela parte superior com o auxílio de um espelho para regular a distância entre o projetor e o solo.

3.2.1 Telas de Projeção

As telas de projeção frontal e lateral são flexíveis e sem emenda com dimensões: 3,2m x 2,4m para a tela frontal, 2,4m x 2,4m para a tela lateral. A tela de projeção do chão é rígida para que possa suportar o peso de um ou mais usuários do sistema e possui dimensões de 3,2m x 2,4m. A estrutura que sustenta as telas de projeção é fixa e feita em madeira de lei, como pode ser visto através da Figura 3.2.

Figura 3.2: Disposição das telas de projeção do LMVI

3.2.2 Projetores

O sistema é composto por 6 projetores Christie modelo DS+60/DW30/Matrix 3000 de alta intensidade luminosa e alta resolução. Cada projetor é composto por um sistema de iluminação com 2 lâmpadas de 300W que oferecem altos níveis de brilho e contraste. A resolução nativa de cada dispositivo é de 1400 x 1050 *pixels*.

Para a recepção dos dados enviados pelas saídas das placas de vídeo, as conexões entre os computadores e os projetores são estabelecidas através de cabos DVI (Digital Visual Interface).

A distância entre os projetores e as telas de projeção, bem como sua posição em relação às telas, devem ser ajustadas de forma a obter-se o melhor alinhamento possível das

imagens divididas nas 3 telas que compõem o sistema.

Os projetores responsáveis pela tela do chão são mantidos em uma estrutura metálica fixada no topo da estrutura de madeira que sustenta as 3 telas, como pode ser visto na Figura 3.3(a). Para ajustar a distância entre estes projetores e o solo, suas lentes são voltadas para um espelho fixado em uma ângulo de aproximadamente 45 graus que reflete as imagens para a tela do chão.

Os projetores responsáveis pela tela frontal e tela lateral são mantidos em estantes metálicas e posicionados com suas lentes voltadas diretamente para as telas de projeção, como pode ser visto na Figura 3.3(b).

Figura 3.3: Estruturas metálicas de sustentação dos projetores (a) superiores e (b) laterais

3.2.3 Óculos Estereoscópicos Passivos

Para cada tela de projeção, dois projetores com filtros polarizadores projetam um par estereoscópico de imagens, que ao olho nu o espectador consegue visualizar uma imagem com predominância da cor verde e a outra com predominância da cor roxa, sobrepostas e desfocadas, como pode ser visto na Figura 3.4(a). Somente através do uso de óculos estereoscópicos passivos com filtros polarizadores de mesma cor que as imagens projetadas, apresentado na Figura 3.4(b), o espectador visualiza a imagem tridimensional.

3.3 Estrutura Computacional

O desenvolvimento e execução de aplicações para os sistema de multiprojeção depende de um conjunto amplo e sempre evolutivo de tecnologias, que inclui desde a infraestrutura de comunicação até a capacidade de geração e apresentação de imagens de maneira sincronizada pelos computadores que compõem o sistema.

Figura 3.4: (a) Par estereoscópico de imagens e (b) óculos estereoscópico

A estrutura computacional do LMVI é composta por computadores principais responsáveis pela geração e apresentação das imagens para cada tela e computadores auxiliares de apoio ao desenvolvimento das aplicações.

3.3.1 Computadores Principais

Os computadores principais referem-se a um aglomerado (cluster) de 3 computadores PC responsáveis pela geração e apresentação das imagens em cada uma das 3 telas. Cada computador é equipado com processador AMD Opteron 850 de 2.41 GHz, 4 GB de memória RAM e placa de vídeo Nvidia Quadro FX4500 com 512 MB de memória interna. Um dos computadores do cluster possui ainda uma placa de som Creative PCle SB X-Fi Titanium, com saída de áudio digital 7.1.

3.3.2 Computadores Auxiliares

Os computadores auxiliares referem-se aos demais computadores do laboratório responsáveis pelo apoio na implementação de aplicações, modelagem de objetos 3D e interface com os computadores principais do sistema. Essas estações de trabalho possuem as mesmas configurações das máquinas do cluster principal e permitem que as aplicações sejam desenvolvidas e testadas localmente (no desktop) antes de serem enviadas ao cluster do sistema de multiprojeção.

O aglomerado de 3 computadores principais descrito anteriormente é o único responsável pela geração e apresentação das imagens no sistema de multiprojeção. Além de apoiar no desenvolvimento das aplicações, um dos computadores auxiliares é utilizado como interface/acesso aos computadores do aglomerado sem participação na projeção.

3.4 Sistema de Som 5.1

Em sistemas de realidade virtual multiusuário, como os sistemas do tipo CAVE, é necessária a utilização de matrizes de alto-falantes, para que todos os usuários possam, ao mesmo tempo, ser imersos no ambiente virtual tanto visualmente, quanto com relação ao som tridimensional.

O LMVI possui um *Home Theater* Samsung HT-C460 com 850W RMS que proporciona um sistema de som *surround* 5.1 composto por cinco auto-falantes e um *subwoofer*, como mostrado na Figura 3.5.

Figura 3.5: *Home theater* que compõe o sistema de som *surround* 5.1 do LMVI

A conexão entre o PC e o *home theater* foi realizada por meio de cabo óptico, que liga a saída óptica da placa de som à entrada óptica digital do *home theater*, como pode ser visto na Figura 3.6.

Figura 3.6: Conexão entre o PC e o *Home Theater*

Um sistema de sonorização como este possibilita a utilização de áudio padrão (estéreo ou mono) ou áudio 3D. Além disso, alguns algoritmos permitem ainda o uso de som *surround* 5.1 juntamente com *head tracking*, criando uma simulação sonora completa (SOARES et al., 2010).

Alguns cuidados com o ambiente real devem ser tomados para atingir a melhor qualidade de sonorização, como por exemplo, o uso de carpete no chão e/ou forro nas paredes, para a eliminação da reverberação inadequada. O excesso de reverberação pode confundir os

usuários quanto ao posicionamento correto dos sons disparados.

3.4.1 Disposição dos Alto falantes

As telas de projeção do LMVI foram montadas entre duas salas, a sala de utilização e a sala de equipamentos. Os projetores, *home theater* e computadores controladores de cada uma das telas de projeção ficam dispostos na sala de equipamentos. A Figura 3.7 mostra que ficam visíveis ao usuário, na sala de utilização, apenas os alto-falantes traseiros do *home theater*.

Figura 3.7: Visão frontal do sistema de sonorização do LMVI

Todos os auto-falantes do *home theater* são suspensos em suportes adequados, com exceção do *subwoofer*, que se recomenda manter no chão, para melhor aproveitamento dos sons graves. Os auto-falantes frontais, situados na mesma sala dos projetores, são mantidos fora da área de projeção, posicionados próximos às junções das telas frontal e lateral. O auto-falante frontal central, que também se situa na sala de equipamentos, é posicionado abaixo da área de projeção. A Figura 3.8 mostra a visão superior dos equipamentos de som do LMVI.

Figura 3.8: Visão superior do sistema de sonorização do LMVI

3.5 Dispositivos de Entrada de Dados

Como forma de interagir com o ambiente virtual projetado é necessário o uso de alguns dispositivos de entrada de dados, que possam ser lidos e interpretados como comandos do usuário para realizar alguma ação no ambiente 3D. Além dos dispositivos mais comuns como o mouse e o teclado, o LMVI está equipado com outros dispositivos de interação que serão descritos nesta seção.

3.5.1 Controle de Vídeo Game Wii

O Controle Wii (também chamado de *Wiimote*) é o controle primário do vídeo game Wii da Nintendo. Ele é conectado ao PC via *bluetooth* e associado a uma barra de LEDs infravermelhos permite que seus movimentos sejam captados e transmitidos ao computador, funcionando como uma espécie de “mouse aéreo”. A Figura 3.9 apresenta o controle Wii e barra de LEDs.

Figura 3.9: *Wiimote* e a barra de LEDs infravermelhos

3.5.2 Webcam e Caneta Laser

Como forma de interagir de uma maneira simples, intuitiva e barata com o sistema de multiprojeção, o LMVI dispõe de uma *webcam* de baixo custo e 3 canetas *laser* (*Laser Pen*) que servirão como ponto de referência para a movimentação do usuário, como pode ser visto na Figura 3.10.

A ideia deste tipo de interação é simples: a *webcam* capta as imagens do usuário movimentando a caneta para manipular os objetos 3D, utiliza funções de processamento de imagem para filtrar o ponto luminoso da caneta e calcula o deslocamento deste ponto no campo de visão da *webcam*. Com base no deslocamento da caneta no campo de visão da *webcam*, o desenvolvedor seleciona comandos de interação do usuário com o ambiente tridimensional.

A *webcam* foi posicionada no ponto de junção das 3 telas, com a lente voltada para

Figura 3.10: Webcam de baixo custo e canetas laser

os olhos do usuário. Dessa forma, é possível obter um amplo campo de visão onde o usuário pode movimentar a caneta laser para interagir com o ambiente.

3.5.3 Dispositivo de Interação por Cor

Como alternativa às canetas laser, foi montado um dispositivo que permitisse a detecção de movimentação do usuário com base na cor do ponto luminoso. Assim como o uso da caneta laser descrito anteriormente, utiliza-se a *webcam* para captar as imagens do usuário movimentando o dispositivo para manipular os objetos 3D. A diferença é que ao invés de calcular o deslocamento de um ponto luminoso qualquer na imagem, identifica-se primeiro a cor desse ponto para posterior realização de uma ação.

O dispositivo dispõe de LEDs coloridos (azul, vermelho e verde) e 3 botões que permitem acender a cor desejada. Para facilitar a detecção do ponto luminoso através da *webcam*, foi utilizado um filtro de material plástico que uniformiza e concentra o brilho dos LEDs em um mesmo ponto, como pode ser vista através da Figura 3.11.

3.5.4 Joystick

Para a interação com simuladores de voo ou outras aplicações que envolvem um grande número de funções ou uma grande área a ser explorada, é utilizado um *Joystick Clone Vibration Águia* com 12 botões de funções, como pode ser visto na Figura 3.12. Com um design ergonômico é um controle preciso para as aplicações de simulação dessa natureza, oferecendo um maior realismo ao usuário do sistema.

Figura 3.11: Dispositivo de interação por cor (azul, vermelho e verde)

Figura 3.12: Joystick Clone Vibration Águia com 12 botões de funções

3.6 Plataforma e Bibliotecas de Programação

O desenvolvimento de aplicações gráficas tridimensionais para sistemas de multiprojeção envolve estudos e recursos ligados com hardware, software, percepção, interface do usuário, fatores humanos e aplicações. Durante a implementação do sistema, o desenvolvedor precisa trabalhar somente com ferramentas de alto nível de abstração, para que este possa criar, manipular e visualizar objetos geométricos, deixando os detalhes de visualização de baixo nível para bibliotecas especializadas.

Para auxiliar o desenvolvimento, diversas bibliotecas de programação estão sendo desenvolvidas, mas, até o momento, nenhuma oferece todos os recursos necessários para

este domínio de aplicação. Dessa forma, o desenvolvedor necessita utilizar um conjunto de bibliotecas que combinadas oferecem soluções para o desenvolvimento de aplicações de Realidade Virtual Imersiva.

A plataforma de programação utilizada no LMVI é o Visual C++ 2005 Express Edition. Para obter o controle e execução das aplicações no sistema de multiprojeção sem se preocupar com a separação das imagens, sincronismos e gerenciamento de processamento das máquinas do cluster é utilizado o *framework* VRJuggler 2.2.1. Para a importação dos modelos 3D e a criação do grafo de cena é utilizado a biblioteca gráfica OpenSG 1.6.0. Para o desenvolvimento de soluções de interação por detecção de movimentos é utilizado a biblioteca OpenCV 2.0. E por fim, para manipulação de som 5.1 é utilizado a biblioteca OpenAL 2.0. Nesta seção são apresentados maiores detalhes sobre essas bibliotecas.

3.6.1 VR Juggler

O VR Juggler é um projeto de pesquisa do Virtual Reality Applications Center da Iowa State University, e refere-se à uma coleção de tecnologias, as quais fornecem as ferramentas necessárias para o desenvolvimento de aplicações em realidade virtual, permitindo ao usuário executar as aplicações em sistemas de realidade virtual.

Este projeto constitui-se basicamente de uma plataforma para o desenvolvimento e execução de aplicações baseadas em realidade virtual, constituída por um conjunto de bibliotecas em C++. Essas bibliotecas oferecem um conjunto de classes interligadas que facilitam a programação. A plataforma permite que a aplicação seja independente da arquitetura do hardware e do sistema operacional. Esta plataforma é quem gerencia esses aspectos e torna acessível a configuração do hardware dedicado por meio da configuração visual de seus atributos sem que haja a necessidade de modificar o código da aplicação. Isto disponibiliza um ambiente operacional unificado garantindo a portabilidade da aplicação entre diferentes recursos, o que caracteriza esta ferramenta.

Toda ação que ocorre em um ambiente virtual é controlada pelo VR Juggler mediante um dos seus componentes denominado *kernel*. O *kernel* atua como interface entre a aplicação e a plataforma, o que proporciona a abstração de hardware para a plataforma virtual. Isso é, a interface do *kernel* da plataforma disponibiliza para toda aplicação as funcionalidades acessíveis exceto funções específicas de APIs gráficas e é o único modo da aplicação acessar o hardware.

O conceito de plataforma virtual facilita e simplifica os esforços do desenvolvimento

de aplicações em sistemas de ambientes virtuais complexos. Propicia um ambiente de trabalho unificado que suporta o desenvolvimento e execução de aplicações, independente das tecnologias básicas. Uma plataforma virtual garante a longevidade das aplicações, e permite que desenvolvedores de aplicação acompanhem os avanços tecnológicos sem investir tempo e recursos modificando aplicações para suportar novas tecnologias. Neste trabalho, foi utilizado a versão 2.2.1.

3.6.2 OpenSG

Como as interfaces baseadas em ambientes virtuais precisam suportar cenas complexas tridimensionais (3D) com centenas ou milhares de formas separadas, precisamos utilizar métodos eficientes para gerenciar tal quantidade de informações de forma efetiva, especialmente levando em consideração que se deve prover interatividade em tempo real.

OpenSG é um gerenciador de grafos de cena para o desenvolvimento de aplicações gráficas de alto desempenho. Um grafo de cena é um arranjo de objetos 3D em uma estrutura de árvore que especifica o conteúdo de um universo virtual e indica como este será renderizado. Nele é possível definir som, luz, orientação, geometria, localização e aparência visual dos objetos. Descrever uma cena usando um grafo de cena é uma tarefa mais simples do que construir a cena utilizando linhas de comando que especificam primitivas gráficas, tais como as do OpenGL.

O grafo de cena permite ordenar dados em forma hierárquica onde nós pai afetam seus nós filho. Logo, ele permite estruturar os dados do ambiente virtual em forma de uma árvore, onde o nó raiz agrupa todos os objetos do ambiente, e os nós nos níveis inferiores armazenam informações, agrupam objetos relacionados e finalmente armazenam a geometria dos objetos a serem representados no ambiente virtual.

O OpenSG provê um *framework* orientado a objetos em que em um nível de menor abstração encontra-se a biblioteca OpenGL, ou seja, o *OpenSG* é compatível com aplicações que suportam OpenGL. Sua função é livrar o desenvolvedor de implementar e otimizar chamadas gráficas de baixo nível, facilitando o desenvolvimento rápido de aplicações gráficas poupando o desenvolvedor de programar rotinas complexas de computação gráfica 3D e renderização. A Figura 3.13 apresenta um diagrama de classes que abrange a utilização do OpenSG em conjunto com o *framework* VR Juggler. Neste trabalho, foi utilizado a versão 1.8.

Figura 3.13: Diagrama de Classes do OpenSG combinado com o VR Juggler

3.6.3 OpenCV

O OpenCV (*Open Source Computer Vision*) é uma biblioteca de programação, de código aberto, desenvolvida inicialmente pela Intel Corporation. O OpenCV implementa uma variedade de ferramentas de interpretação de imagens, indo desde operações simples como um filtro de ruído, até operações complexas, tais como a análise de movimentos, reconhecimento de padrões e reconstrução em 3D.

A biblioteca OpenCV foi idealizada com o objetivo de tornar a visão computacional acessível a usuários e programadores em áreas tais como a interação humano-computador em tempo real e a robótica. A biblioteca está disponível com o código fonte e os executáveis (binários) otimizados para os processadores Intel.

Um programa OpenCV, ao ser executado, invoca automaticamente uma DLL (*Dynamic Linked Library*) que detecta o tipo de processador e carrega, por sua vez, a DLL otimizada para este. Juntamente com o pacote OpenCV é oferecida a biblioteca IPL (*Image Processing Library*), da qual a OpenCV depende parcialmente, além de documentação e um conjunto de códigos exemplos.

A biblioteca está dividida em cinco grupos de funções: Processamento de imagens; Análise estrutural; Análise de movimento e rastreamento de objetos; Reconhecimento de

padrões e Calibração de câmera e reconstrução 3D. Neste trabalho, foi utilizado a versão 2.1.0.

3.6.4 OpenAL

OpenAL (*Open Audio Library*) é uma interface de software para hardwares de áudio. Em geral, trata-se de uma maneira de gerar áudio para simulações tridimensionais. Essa interface consiste em diversas funções que afetam o CPU e o hardware de som, permitindo que programadores criem saída multicanal para som tridimensional em torno do usuário. Se o hardware de som consiste em apenas um buffer de saída, então a aplicação OpenAL deve ser totalmente direcionada ao CPU. O estado das informações OpenAL são explícitas ao usuário.

O OpenAL foi criado no final dos anos 90 como um complemento para OpenGL, seguindo o mesmo estilo de codificação. Inicialmente a biblioteca havia sido criada apenas para a plataforma LINUX, porém, a Creative Labs a padronizou para que se tornasse uma biblioteca independente de plataforma. Neste trabalho, foi utilizado a versão 1.0.

3.6.5 TinyXML

TinyXML é uma biblioteca livre e de código-aberto, específica para análise de documentos XML (*eXtensible Markup Language*). O nome *Tiny* (minúsculo em português) vem do fato de ser uma biblioteca simples, leve e que pode ser facilmente integrada a aplicações escritas em C++.

Essa biblioteca permite criar objetos C++ e carregar o conteúdo de um documento XML para dentro desses objetos. Esses objetos, que representam o documento XML carregado, podem ser manipulados, alterados e salvos novamente como um documento XML. Neste trabalho, foi utilizado a versão 2.6.2.

3.6.6 Curl

Curl (Client URLs¹) é um projeto de *software* livre que fornece bibliotecas que permitem que um servidor *web* possa transferir arquivos para um computador remoto utilizando uma variedade de protocolos de Internet. As bibliotecas fornecidas pelo Curl são altamente configuráveis, permitindo que praticamente qualquer tipo de solicitação cliente-servidor possa ser atendida.

¹URL (*Universal Resource Locator*) refere-se ao endereço de uma página da Internet.

Ao usar essas bibliotecas, um servidor pode atuar como um cliente, criando e respondendo às solicitações usando qualquer tecnologia construída sobre HTTP, como o XML-RPC, SOAP, ou WebDAV. Neste trabalho, essa ferramenta foi importante para o acesso a arquivos de dados remotos, armazenados em um servidor *web*, facilitando a comunicação do LMVI com computadores remotos através da Internet. A versão utilizada foi a 7.18.

3.6.7 Importação dos Objetos 3D

Existem diversas maneiras de criar e exibir objetos tridimensionais nas telas de projeção de um sistema de multiprojeção. Usando-se a biblioteca OpenGL ou DirectX, pode-se criar primitivas 3D, como esferas, torus, cones, cubos etc, exibindo-as nas telas de projeção. Porém, o método mais comum de adição de objetos virtuais ao mundo virtual é a importação de objetos 3D.

Dentre os vários formatos de arquivos de objetos 3D, o OpenSG (responsável pelo grafo de cena) permite a importação de arquivos VRML97 (com extensão WRL) ou arquivos com extensão 3DS. Para o sistema de multiprojeção do LMVI optou-se pela utilização de arquivos VRML97, pois a disposição dos eixos exportados nesses arquivos é a mesma disposição encontrada no sistema OpenSG. A disposição dos eixos X,Y e Z dos arquivos exportados em 3DS é diferente da encontrada em OpenSG, o que obriga a execução de modificações no momento da importação de objetos.

O OpenSG, dentro do VRJuggler, é inicializado através dos procedimentos: “*void OpenSG::initScene()*” e “*void OpenSG::init()*”. Esses procedimentos são executados uma única vez durante a execução do sistema. A importação de objetos 3D, dentro do OpenSG, é feita dentro do procedimento “*void OpenSG::initScene()*”, onde são inicializados, também, outros componentes da cena, como as câmeras importadas de VRML. Para a importação direta dos arquivos é utilizado o comando “*SceneFileHandler*”, que associa um arquivo VRML a um nó, que pode posteriormente ser adicionado ao grafo de cena, como pode ser visto na Figura 3.14.

```
NodePtr wrlCamModel = SceneFileHandler.the().read(file);  
NodePtr wrlModel = SceneFileHandler.the().read(file);
```

Figura 3.14: Importação de objetos 3D a partir de uma arquivo VRML

3.6.8 Criação do Grafo de Cena

De uma maneira simplificada, o grafo de cena nada mais é do que uma estrutura de dados em forma de árvore, onde as folhas dessa árvore (chamadas de nós do grafo) representam os objetos 3D, a textura e iluminação da cena virtual. Portanto, após a importação dos modelos ou objetos 3D, é necessário inseri-los nesta árvore para posterior manipulação e *renderização* (apresentação dos objetos na tela) dos mesmos.

A criação do grafo de cena, ou árvore de objetos que farão parte do mundo virtual, tem início no procedimento “*void OpenSG::initScene()*”, onde são importados todos os arquivos VRML iniciais e onde é criada a raiz da árvore. O procedimento “*createRoot()*”, mostrado na Figura 3.15, cria o primeiro nó da árvore, onde serão adicionados todos os nós restantes.

```
NodePtr OpenSG::createRoot() {  
  
    //create Root Node  
    NodePtr root = Node::create();  
    OSG::setName(root, "ROOT");  
  
    //create Root Core (empty)  
    TransformPtr rootCore = Transform::create();  
  
    //add rootCore to root node  
    beginEditCP(root, Node::CoreFieldMask);  
        root->setCore(rootCore);  
    endEditCP(root, Node::CoreFieldMask);  
  
    return root;  
}
```

Figura 3.15: Instanciação do nó raiz no grafo de cena

Para a criação de cada nó da árvore de objetos 3D, deve-se criar um nó adicional, chamado de “*core*”, responsável pelo gerenciamento do objeto. Um “*core*” muito utilizado é o “*TransformCore*”, responsável pela posição e orientação das geometrias associadas. A Figura 3.16 apresenta de forma resumida a função de inserção de novos nós na árvore de objetos.

3.6.9 Busca de Objetos no Grafo de Cena

Para transladar, rotacionar, editar, inserir ou remover nós de um grafo de cena foi criado uma rotina de buscas de objetos no grafo. Utilizando as funções nativas da biblioteca OpenSG foi possível desenvolver uma função que busca qualquer nó na árvore que seja filho de um nó utilizado como argumento da função (pode ser o nó raiz) e que possua um determinado nome como identificador único, como pode ser visto no código apresentado na Figura 3.17.

```
NodePtr OpenSG::addNode(NodePtr wrlModel, const Char8* nodeName, NodePtr root){  
  
    // create wrl transf node to be added to the root  
    NodePtr wrlTransf = Node::create();  
    OSG::setName(wrlTransf, nodeName);  
  
    //create core for wrl transf node  
    TransformPtr wrlCore = Transform::create();  
  
    //add wrlTransf node to root  
    beginEditCP(root, Node::ChildrenFieldMask);  
    root->addChild(wrlTransf);  
    endEditCP(root, Node::ChildrenFieldMask);  
  
    //add wrlCore to wrlTransf  
    beginEditCP(wrlTransf, Node::CoreFieldMask);  
    wrlTransf->setCore(wrlCore);  
    endEditCP(wrlTransf, Node::CoreFieldMask);  
  
    //add wrlModel to wrlTransf  
    beginEditCP(wrlTransf, Node::ChildrenFieldMask);  
    wrlTransf->addChild(wrlModel);  
    endEditCP(wrlTransf, Node::ChildrenFieldMask);  
  
    return root;  
}
```

Figura 3.16: Inserção de um objeto 3D ao grafo de cena

3.7 Mecanismos de Interação para Utilização do AV no LMVI

Para utilização do ambiente virtual projetado no LMVI, além dos óculos estereoscópicos é necessário o uso de algum tipo de dispositivo de interação. Para isso, é necessário o uso de alguns dispositivos de entrada de dados, que possam ser lidos e interpretados como comandos do usuário para realizar alguma ação no ambiente 3D.

Inicialmente, a única forma de interação entre o usuário e o ambiente virtual projetado no sistema de multiprojeção imersivo do LMVI era através do teclado. Para que a interação fosse um pouco mais intuitiva e oferecesse um mais liberdade ao usuário para movimentar-se dentro do sistema de multiprojeção outras formas de interação foram desenvolvidas. O desenvolvimento e a utilização destas soluções são apresentados ao longo desta seção.

3.7.1 Interação por Controle Remoto Wii

Optou-se por testar tal dispositivo, pois o Controle Remoto Wii (*Wiimote*) é considerado um dispositivo de baixo custo, se comparado aos dispositivos mais comuns que são usados em sistemas imersivos. Para acessar o dispositivo e realizar a leitura dos dados de

```
NodePtr OpenSG::findNode(NodePtr node, std::string nodeName) {  
  
    UInt32 children = node->getNChildren();  
  
    if (getName(node)) {  
        if (getName(node) == nodeName) {  
            return node;  
        }  
    }  
  
    for (int i=0; i < children; i++) {  
        NodePtr child = findNode(node->getChild(i), nodeName);  
        if (child != NullFC) {  
            return child;  
        }  
    }  
  
    return NullFC;  
}
```

Figura 3.17: Busca de um objeto 3D no grafo de cena

entrada foi preciso instalar e configurar uma biblioteca chamada VRPN e configurá-la junto ao arquivo de configuração principal do VR Juggler.

Apesar dos testes com o Wiimote se mostrarem satisfatórios em um primeiro momento, foram encontrados alguns problemas provenientes do uso de “*bluetooth*”. O sinal do controle é perdido muito facilmente, obrigando a reinicialização do sistema de controle, o que inviabiliza a mobilidade do usuário, durante a utilização do dispositivo.

3.7.2 Interação por Detecção de Movimentos

Como forma de superar os problemas encontrados na interação pelo controle remoto Wii, iniciou-se o desenvolvimento de interação por detecção de movimentos através de uma *webcam*. A princípio, foram adquiridas *webcams* de baixo custo e 2 LEDs infravermelhos de alta intensidade. Os LEDs foram fixados em uma caixa de plástico para fins de testes antes de sua utilização com o usuário, como pode ser visto em destaque na Figura 3.18.

Uma *webcam* foi desmontada e um filtro de luz foi anexado à sua lente. O filtro de luz era formado por folhas de filme de fotos, com o intuito de permitir apenas a passagem da luz infravermelha, bloqueando a luz natural. Logo percebeu-se que a captura da luz infravermelha não era suficientemente boa para permitir a detecção correta dos movimentos dos LEDs.

Em seguida, como uma ideia derivada do uso de LEDs infravermelhos e *webcams*, optou-se por testar a captura de luz proveniente de canetas *laser* e lanternas de baixo custo.

Figura 3.18: Dispositivo com LEDs infravermelhos

Foram adquiridas algumas canetas *laser* e lanternas e filtros de luz foram anexados às mesmas, para corrigir o ruído da luz captada pela câmera. Esses novos filtros eram formados por um material plástico branco translúcido, que são utilizados para embalar eletrônicos, como pode ser visto na Figura 3.19.

Figura 3.19: Material utilizado para uniformizar e concentrar a luz do laser

A detecção de movimentos do usuário foi realizada através do uso de uma ou mais canetas *laser* dentro do campo de visão de uma *webcam*. Através do deslocamento do ponto luminoso do *laser* é possível movimentar objetos 3D no ambiente virtual de uma maneira simples e intuitiva. No entanto, o maior desafio dessa solução consiste na identificação correta do ponto luminoso da caneta.

As imagens de onde queremos extrair alguma informação em alguns casos precisam ser convertidas para um determinado formato ou tamanho e precisam ainda ser filtradas para remover ruídos provenientes do processo de aquisição da imagem. Os ruídos podem aparecer de diversas fontes, como por exemplo, o tipo de sensor utilizado, a iluminação do ambiente, as condições climáticas no momento da aquisição da imagem, a posição relativa entre o objeto de interesse e a câmera. Note que ruído não é apenas interferência no sinal de captura da imagem, mas também interferências que possam atrapalhar a interpretação ou o reconhecimento de objetos na imagem.

3.7.2.1 Configuração e Posicionamento da Webcam

Para que a *webcam* captasse corretamente, apenas a luz transmitida pelo dispositivo e não toda a luz ambiente, foi necessário alterar a configuração da *webcam* da seguinte forma:

- Brilho: 0;
- Contraste: MÁXIMO ;
- Saturação: MÁXIMO ;
- Proporção de branco: MÁXIMO ;
- Comp. luz de fundo: 0;
- Exposição à luz: desmarcada a opção “automático” e configurado o valor MÍNIMO.

Todas as alterações foram feitas para que a *webcam* exibisse uma imagem mais escura possível, captando somente a luz do *laser* ou lanterna. Apesar da biblioteca OpenCV permitir o uso de mais de uma câmera foi utilizada somente uma *webcam* para uso do menor processamento possível.

Com o intuito de captar todos os movimentos do usuário com apenas uma *webcam*, diversas maneiras de posicionamento da *webcam* foram testadas. A melhor maneira encontrada foi posicioná-la sobre a tela do chão, exatamente na junção das três telas de projeção, apontada para os olhos do usuário, em um ângulo próximo a 60 graus, como pode ser visto na Figura 3.20. As demais posições atrapalhavam a detecção do *laser* ou lanterna, por captar a luz dos projetores superiores que refletem no chão.

Figura 3.20: Posicionamento da *webcam* no sistema de multiprojeção

3.7.2.2 Pré-processamento da Imagem para Filtragem do Laser

Para que apenas o ponto luminoso do *laser* ou lanterna fosse identificado em um primeiro momento, utilizando a biblioteca OpenCV, a imagem foi convertida de colorida para escala de cinza. Em seguida, um “*threshold*” foi aplicado nessa imagem, gerando uma nova imagem, com valores preto ou branco (luz = branco, preto = restante). A Figura 3.21 apresenta uma sequência com de 3 imagens que ilustram o processam aqui descrito. Como pode ser visto na última imagem da Figura 3.21, foi possível obter-se uma imagem sem ruídos, com fundo preto e somente o ponto de luz em destaque.

Figura 3.21: Sequência de imagens resultantes do pré-processamento do ponto luminoso

3.7.2.3 Detecção do Ponto de Luz Na Imagem Pré-processada

Para a detecção do ponto de luz, varreu-se a imagem inteira (640x480 *pixels*), verificando *pixel* a *pixel*, por um valor maior que zero (que não seja preto). Ao encontrá-lo, utilizou-se uma função “*cvfloodfill*” para verificar a existência de outros *pixels*, de mesma cor, conectados ao *pixel* em questão. Caso existissem, seria possível descobrir a altura, largura, área, x e y do componente encontrado (conjunto de *pixels* de mesma cor, conectados entre si), através do parâmetro “*comp*”, da função “*cvfloodfill*” do OpenCV.

Caso existissem mais de um componente na imagem, comparava-se o componente atual, com o componente anterior e armazenava-se o maior. Através dos dados do componente foi possível obter sua posição (centro do objeto) e calcular seu raio (tamanho do objeto). Os trechos de código condensado que implementam a detecção e obtenção da posição e tamanho do ponto de luz na imagem podem ser vistos nas Figuras 3.22 e 3.23.

3.7.2.4 Mapeamento da Movimentação do Ponto de Luz no Plano

Com relação à movimentação, armazenou-se a posição X e Y do centro do componente e calculou-se a diferença da próxima posição (X, Y) , a cada *frame*. Em seguida, essa diferença foi multiplicada por uma variável chamada “*velocidade*” e o resultado foi utilizado como incremento da posição (X, Y) do objeto, no mundo 3D. A variável “*velocidade*”

```

void MoveObject::detect(IplImage **_img)
{
    //scan the image / find the connected components / store the 2 biggest comp

    for (int i = 0; i < width*height; i++) {
        pixval = cvGet2D(grayThresh,i/width,i%width);
        if (pixval.val[0] > 240) {
            cvFloodFill(grayThresh,cvPoint(i%width, i/width),
                cvScalar(255), cvScalar(0), cvScalar(0), &comp,flags,mask);
            if (comp.area > 5) {
                if (comp.area != comp1.area) {
                    comp2.rect.x = comp1.rect.x;
                    comp2.rect.y = comp1.rect.y;
                    comp2.rect.width = comp1.rect.width;
                    comp2.rect.height = comp1.rect.height;
                    comp2.area = comp1.area;
                    comp1.rect.x = comp.rect.x;
                    comp1.rect.y = comp.rect.y;
                    comp1.rect.width = comp.rect.width;
                    comp1.rect.height = comp.rect.height;
                    comp1.area = comp.area;
                } //end if
            } //end if
        } //end if
    } //end for
}

```

Figura 3.22: Detecção do ponto de luz na imagem

```

void MoveObject::getCenter(comp1)
{
    //If flashlight is on get its center
    if (comp1.area > 0) {
        //get center
        x = comp1.rect.x + comp1.rect.width/2;
        y = comp1.rect.y + comp1.rect.height/2;
        //get radius
        if (comp1.rect.width > comp1.rect.height) {
            r = comp1.rect.width/2;
        } else {
            r = comp1.rect.height/2;
        }
    }
}

```

Figura 3.23: Obtenção da posição e tamanho do ponto de luz

(*moveSpeed*) foi utilizada para aumentar ou diminuir a velocidade de movimentação dos objetos 3D. Essa variável é alterada pelas teclas “+” e “-” do teclado do operador. Um trecho condensado do código que implementa o mapeamento da movimentação do *laser* pode ser visto na Figura 3.24.

3.7.2.5 Criação da Thread de Detecção do Laser

Todo processamento da biblioteca OpenCV é realizado através de uma “*thread*” do VRJuggler. Essa “*thread*” é chamada dentro da biblioteca OpenSG para captura ininterrupta dos dados da câmera e detecção do movimento do *laser*. Um pequeno trecho de código que

```

void MoveObject::detect(IplImage **_img)
{
    if (comp2.area < 6) { //just one light is on
        int temp = (frameLightON - lastFrameLightON);
        if ((temp < 5) && (temp > 0)) {
            printf("piscou \n\n");
            laserClicked = true;
        }
        this->lastFrameLightON = this->frameLightON;
        if ((lastX==0)&&(lastY==0)) {
            lastX = x;
            lastY = y;
        }
        float deltax = x - lastX;
        float deltay = y - lastY;
        float tempx = deltax*moveSpeed, tempy = deltay*moveSpeed;
        application->distX += tempx/100;
        application->distY += (tempy/100)*-1;
        lastX = x;
        lastY = y;
    }
}

```

Figura 3.24: Mapeamento da movimentação do *laser* no plano

demonstra como esta “*thread*” é criada e destruída pode ser visto na Figura 3.25.

```

void MoveObject::startThread() {
    opencvThread = new vpr::Thread(boost::bind(&MoveObject::start_cam,this));
}

void MoveObject::stopThread() {
    opencvThread->kill();
    printf("OpenCV stopped and Thread killed\n");
}

```

Figura 3.25: Criação e destruição da *thread* de detecção de movimento do *laser*

Ao final do processamento e calculo dos valores necessários para a movimentação, descobriu-se que somente o computador com acesso a *webcam* atualizava as variáveis a cada *frame*. Portanto, a movimentação dos objetos 3D só ocorria na tela da frente, “*front wall*”, onde a *webcam* estava conectada.

As demais máquinas do *cluster*, que processam o OpenSG independentemente umas das outras, não eram capazes de identificar a entrada de dados da *webcam*, pois esse dispositivo não pode ser configurado diretamente nos arquivos de configuração do VRJuggler, onde são configurados, por exemplo, o teclado e outros dispositivos de entrada.

3.7.2.6 Comunicação dos Dados de Movimentação

Qualquer dispositivo conectado a um dos computadores do *cluster*, que não esteja configurado no arquivo de configuração do VRJuggler, não pode ser compartilhado entre as máquinas, através do próprio VR Juggler. Dessa forma, o VRJuggler não controla automaticamente a transmissão de dados da *webcam* para todas as máquinas do *cluster*.

Para solucionar esse problema, foi necessário implementar uma comunicação de rede por *Socket* (TCP/IP), onde o computador contendo a *webcam* foi designado como *servidor* e os demais como *clientes*. A cada *frame*, os valores obtidos pelo *servidor*, com relação a imagem da *webcam*, são enviados para os *clientes*, para que os mesmos atualizem suas respectivas variáveis e possam movimentar os objetos 3D em suas telas de projeção, ao mesmo tempo que a tela da frente. Uma amostra do código de envio condensado pode ser visto na Figura 3.26.

```
void Server::sendData(){  
  
    //DATA BUFFER  
    printf(sendbuf, "%iF%4.2fX%4.2fY%4.2fZ%4.2fA%4.2fB%sH", flag, distX, distY, distZ);  
  
    //SEND DATA TO CLIENT  
    bytesSent = send(acceptSocket, sendbuf, len, 0);  
    if (bytesSent == SOCKET_ERROR){  
        printf("Server: send() error %ld.\n", WSAGetLastError());  
    } else {  
        printf("Server: send() to client OK.\n");  
        printf("Server: Bytes Sent: %ld.\n", bytesSent);  
        printf("Server: Data sent is: \"%s\"\n", sendbuf);  
    }  
}
```

Figura 3.26: Função de envio de dados de movimentação para os clientes

3.7.2.7 Desenvolvimento da Função Zoom através do Laser

A movimentação com utilização de apenas uma caneta *laser* ou lanterna permite apenas a movimentação dos objetos em seus eixos *X* e *Y*. Para a implementação do *zoom* (eixo *Z*), seria necessário a utilização de um outro ponto de luz, ou então, a utilização de menus, ativados em alguma região da tela, para que o usuário pudesse alternar entre a movimentação (*X, Y*) e o *zoom* (*Z*).

Optou-se pela utilização de um segundo *laser*, que seria responsável somente pela movimentação no plano *Z*, enquanto o primeiro continuaria com a movimentação no plano (*X, Y*). Para que os *lasers* pudessem ser diferenciados, foram utilizados *laser* e lanternas

com cores distintas. A partir da cor, escolhe-se o tipo de movimentação que o mesmo irá desempenhar.

Um grave problema foi encontrado. Dependendo da posição ou ângulo do *laser* ou lanterna em relação à *webcam*, pode haver alteração da cor do ponto luminoso na imagem colorida. Logo, torna-se difícil trabalhar com a detecção do *laser* por cor, em uma imagem que sofre alterações de cor o tempo todo. Para resolver esse problema, a cor do *laser* foi identificada dentro de uma certa tolerância e, em seguida, a imagem era convertida em escala de cinza, e depois, em preto e branco, eliminando ruídos de *pixels* fora de um “*threshold*”, como descrito anteriormente.

A movimentação em *Z* (*zoom*) era obtida de acordo com a posição do *laser* para cima de um determinado ponto da imagem, deslocava-se o objeto para frente e para baixo de um determinado ponto da imagem, deslocava-se o objeto para trás. A movimentação do objeto teve que ser feita utilizando somente um dos *lasers* por vez. Caso os dois estivessem ligados, somente um deles seria levado em consideração.

Com o tempo, percebeu-se que a identificação dos *lasers* ou lanternas por cor não era suficientemente eficiente. Optou-se por trabalhar com dois *laser* ou lanternas, porém, ambos em escala de cinza. Obviamente, o sistema não conseguia mais detectar quando um ou outro *laser* estavam acesos, portanto foi necessário mudar o sistema de *zoom*.

Ao invés de trabalhar com um *laser* para *zoom* e com outro *laser* para movimentação (*X, Y*), optou-se por implementar o *zoom* positivo quando os dois pontos luminosos se distanciavam e o *zoom* negativo quando os dois pontos luminosos se aproximavam. A Figura 3.27 apresenta um usuário segurando as canetas *laser* inicialmente próximas uma da outra e na sequência afastadas para obtenção do *zoom*.

Figura 3.27: Usuário afastando duas canetas *laser* para obtenção do *zoom*

O sistema identifica quando há apenas um *laser* ou lanterna acesa, para a movimentação (*X, Y*) e quando há dois *lasers* ou lanternas acesas para movimentação *Z*. Atualmente, este é o sistema de movimentação em funcionamento no LMVI. Um trecho

condensado com a implementação do *zoom* com 2 *lasers* pode ser visto na Figura 3.28.

```
void MoveObject::detect(IplImage **_img)
{
    ...
    else { //two lights are on
        laserClicked = false;
        frameCount = 0;
        lastX = 0.0;
        lastY = 0.0;
        float distCenter = abs(x2-x);
        if (lastDistCenter > 0) {
            float deltaz = distCenter - lastDistCenter;
            float tempz = deltaz*moveSpeed*0.3;
            application->distZ += tempz/100;
        }
        lastDistCenter = distCenter;
    }
    ...
}
```

Figura 3.28: Implementação (resumida) do *zoom* com 2 pontos de luz

3.7.3 Ponteiro para Seleção de Objetos na Cena

Após o desenvolvimento da movimentação dos objetos, com o uso de uma caneta *laser*, foi necessário desenvolver um ponteiro que se comportasse como a “seta do *mouse*” para permitir a seleção de objetos específicos em uma cena para posterior manipulação.

Escrever na tela de projeção (2D), ao invés do mundo 3D, usando as bibliotecas VRJuggler e OpenSG, não é uma tarefa fácil. Portanto, foi modelado um ponteiro 2D no formato VRML que pode ser incorporado à qualquer mundo virtual, através da importação de objetos descrita na Seção 3.6.7. Ao ser incorporado ao ambiente virtual, este ponteiro permite selecionar objetos na tela de projeção, com o uso de algum dispositivo de interação (caneta *laser*, *joystick*, etc.). A Figura 3.29 apresenta um exemplo do ponteiro 2D projetado no ambiente 3D.

O ponteiro criado é tratado pelo sistema como um nó comum da árvore de objetos, porém se movimenta somente no plano (X, Y) , para simular o movimento de um *mouse* em um *desktop*. Porém, como o ponteiro tem movimentação 2D e os objetos possuem posição 3D, foi necessário trabalhar com raios de intersecções para a correta localização dos objetos.

Para selecionar um objeto a partir da localização do ponteiro, foi necessário obter as coordenadas do centro do ponteiro e as coordenadas da cena a partir da visão da câmera, ou seja, as coordenadas “*WorldToScreen*” do mundo virtual, como pode ser visto

Figura 3.29: Ponteiro 2D para seleção de objetos 3D

na implementação (resumida) apresentada na Figura 3.30. Após obter-se as coordenadas do centro do ponteiro no mundo virtual, essas coordenadas foram multiplicadas pelas coordenadas da cena, a partir da visão da câmera, obtendo-se as coordenadas do ponteiro no plano da tela.

```
NodePtr OpenSG::pickObject(NodePtr pointer) {

    //First, we get the pointer_center in world coords
    DynamicVolume pointer_vol;
    pointer->getWorldVolume(pointer_vol);
    Pnt3f pointer_center;
    pointer_vol.getCenter(pointer_center);

    //Map the camera view points from world coordinates to screen coords
    Matrix wts;
    mContextData->mWin->getPort(0)->getCamera()->
        getWorldToScreen(wts, *mContextData->mViewport);

    printf("Pointer_Center World Coords, x : %f, y : %f\n",
        pointer_center.x(), pointer_center.y());

    //Get the pointer_center in screen coords through wts x pointer_center
    Pnt3f result;
    wts.multFullMatrixPnt(pointer_center, result);
    Real32 x = result[0];
    Real32 y = result[1];

    printf("Pointer_Center Screen Coords, x : %f, y : %f\n", x, y);

}
```

Figura 3.30: Seleção um objeto 3D a partir da posição de um ponteiro 2D

As coordenadas do ponteiro no plano da tela inicialmente se encontravam no intervalo entre $[-1, 1]$ onde o canto inferior esquerdo da tela correspondia ao valor $(-1, -1)$ e o canto superior direito correspondia ao valor $(1, 1)$. No OpenSG, assim como no OpenGL, o ponto $(0, 0)$ se encontra no canto inferior esquerdo, logo, as coordenadas (X, Y) variam em X , de

0 até a largura da tela e em Y de 0 até a altura da tela, de baixo para cima e da esquerda para a direita. Através do mapeamento deste intervalo para as dimensões da tela, variável do *desktop* (640x480) para a LMVI (1024x768), obteve-se o ponto (X, Y) do ponteiro em *pixels*. A implementação (resumida) deste procedimento pode ser vista na Figura 3.31.

```
Int32 width = mContextData->mWin->getPort(0)->getPixelWidth();
Int32 height = mContextData->mWin->getPort(0)->getPixelHeight();

//The screen coords are between -1 and 1. Where: left bottom=(-1,-1) and right top=(1,1)
//Map (x,y) from -1..1 to a 0..1 interval

x = (x + 1)/2.0;
y = (y + 1)/2.0;
//Get the (x,y) in pixel coords
x = x * width;
y = (y*height)*-1;
```

Figura 3.31: Mapeamento das coordenadas do ponteiro para as dimensões da tela

Em seguida, calculou-se um raio (linha) de intersecção, que parte da posição da câmera até o “infinito” da cena, passando pelo ponto (X, Y) do ponteiro, em coordenadas da tela. Esse raio é disparado paralelamente ao eixo Z da cena. A partir do raio calculado, verificou-se o mesmo interceptava algum objeto. Em caso positivo, retorna-se o primeiro objeto interceptado como resultado, bem como suas propriedades. A implementação (resumida) deste procedimento pode ser vista nas Figuras 3.32 e 3.33.

```
//Calculate a ray that starts at the camera position and goes through the pixel x, y in the
viewport port

Line ray;
mContextData->mWin->getPort(0)->getCamera()->
calcViewRay(ray,x,y,*mContextData->mWin->getPort(0));
```

Figura 3.32: Projeção de um raio a partir da posição da câmera até o “infinito”

Com o objeto interceptado identificado, foi necessário desenvolver um método e seleção de objetos 3D com o ponteiro 2D a partir da movimentação das canetas *lasers*. Foi definido que quando o usuário acender e apagar o *laser* rapidamente, o sistema irá tratar este evento como um clique duplo do *mouse*. Após o clique duplo do *laser*, foi estabelecido que o objeto selecionado deve seguir o ponteiro 2D sem alterar sua posição em Z . Dessa forma o usuário pode arrastar algum o objeto do mundo virtual por meio do ponteiro 2D aqui apresentado. A implementação (resumida) deste procedimento pode ser vista na Figura 3.34.


```
if (iAct->didHit()) {
    // get the hit point
    fixedHitPoint = iAct->getHitPoint();
    //get the node that was hit (it will return a GeometryNode)
    NodePtr hitNode = iAct->getHitObject();

    //get the hitNode_center in world coords
    DynamicVolume hitNode_vol;
    hitNode->getWorldVolume(hitNode_vol);
    Pnt3f hitNode_center;
    hitNode_vol.getCenter(hitNode_center);

    this->initialHitNodeCenterX = hitNode_center.x();
    this->initialHitNodeCenterY = hitNode_center.y();

    distHitPoint2CenterX = fixedHitPoint.x() - hitNode_center.x();
    distHitPoint2CenterY = fixedHitPoint.y() - hitNode_center.y();

    this->hitNodeName = (char*)getName(hitNode);

    delete iAct;
    return hitNode;
} //end if didHit()
delete iAct;
return NullFC;
```

Figura 3.33: Obtenção de um objeto interceptado pelo raio projetado

3.7.4 Criação e Disparo de Som 5.1

Para gerar um maior realismo nas cenas virtuais projetadas, métodos de criação e disparo de som 5.1 foram implementados. O software usado para a criação de arquivos de sons com 6 canais de áudio foi o Adobe Cool Edit 2.1. O formato de arquivo utilizado foi o Windows PCM WAVE 16 bits 6-channel interleaved, um arquivo com extensão WAV, com suporte a gravação de 6 canais de áudio num mesmo arquivo. A biblioteca de áudio OpenAL, compatível com o framework VRJuggler, foi utilizada para reprodução do arquivo multicanal, durante a execução dos ambientes virtuais.

Foi desenvolvido um sistema de disparo (*trigger*) para sons via teclado ou algum outro dispositivo de interação. Tanto o procedimento de leitura da entrada de dados, quanto o procedimento de disparo dos sons 5.1 foram colocados dentro de *threads*, para que não houvesse interrupção ou atrasos da simulação do AV, durante a utilização dos mesmos.

```

void OpenSG::followPointer(NodePtr pointer, NodePtr hitNode) {
    if (hitNode != NullFC) {
        //First, we get the pointer_center in world coords
        DynamicVolume pointer_vol; pointer->getWorldVolume(pointer_vol);
        Pnt3f pointer_center; pointer_vol.getCenter(pointer_center);

        //Map the camera view points from world coordinates to screen coords
        Matrix wtsMatrix;
        mContextData->mWin->getPort(0)->getCamera()-
        >getWorldToScreen(wtsMatrix, *mContextData->mViewport);

        //Get the pointer_center in screen coords
        Pnt3f pointer_centerScreenCoords;
        wtsMatrix.multFullMatrixPnt(pointer_center, pointer_centerScreenCoords);
        //Get the hit point in screen coords
        Pnt3f hitPointScreenCoords;
        wtsMatrix.multFullMatrixPnt(fixedHitPoint, hitPointScreenCoords);

        //Update hitPointScreenCoords with the pointer_centerScreenCoords
        hitPointScreenCoords[0] = pointer_centerScreenCoords.x();
        hitPointScreenCoords[1] = pointer_centerScreenCoords.y();

        //Get the updated hitPointScreenCoords in world coords
        wtsMatrix.invert();
        wtsMatrix.multFullMatrixPnt(hitPointScreenCoords, fixedHitPoint);
        this->distNodeX = (fixedHitPoint.x() - distHitPoint2CenterX);
        this->distNodeY = (fixedHitPoint.y() - distHitPoint2CenterY);
    }
}

```

Figura 3.34: Movimentação de um objeto 3D selecionado com o movimento do ponteiro 2D

3.8 Considerações Finais do Capítulo

Este capítulo apresentou uma descrição detalhada do LMVI, abordando desde sua estrutura física às bibliotecas utilizadas para implementação e controle do sistema.

A integração de diferentes técnicas e ferramentas de realidade virtual para atingir um objetivo maior é o processo de interligar diferentes modelos fisicamente ou funcionalmente através de uma mesma plataforma. Devido a complexidade das implementações das bibliotecas e *frameworks* para desenvolvimento de realidade virtual 3D imersiva, a integração entre diferentes tecnologias como o OpenSG, VRJuggler e OpenCv em um sistema de multiprojeção imersivo, foi um dos desafios encontrados durante a etapa de desenvolvimento. Um sistema de multiprojeção com três lados como o do LMVI, torna complicado a implementação de soluções que seriam aparentemente simples.

A infraestrutura de software desenvolvida para o LMVI, bem como as soluções de interação e disparo de som 5.1 tornaram o dispositivo de visualização operacional, permitindo tanto a visualização como a interação com o sistema de maneira fácil, simples e de baixo

custo.

A partir de todo desenvolvimento de soluções específicas para o LMVI outras pesquisadores podem utilizar o sistema de multiprojeção imersivo, sem se preocupar com os detalhes técnicos e computacionais envolvidos na projeção, visualização e interação.

4 Metodologia Proposta

Este capítulo apresenta a metodologia de montagem automática de ambientes virtuais a partir de cenas naturais de culturas agrícolas. A metodologia desenvolvida ainda aborda os aspectos referentes a visualização e utilização dos ambientes virtuais montados em um sistema de multiprojeção imersivo. Inicialmente é apresentada uma visão geral da metodologia e em seguida os módulos que a compõem são detalhados.

4.1 Visão Geral

A metodologia desenvolvida nesta pesquisa foi dividida três módulos principais: **mundo real**, **modelos 3D** e **mundo virtual**. Uma visão geral desses módulos é apresentada através de um diagrama exibido na Figura 4.1.

No módulo **mundo real** são desenvolvidos os estágios referente a visão computacional, aquisição de dados de GPS e a geração de um modelo de dados de objetos reais extraídos das cenas naturais. Durante a etapa de visão computacional ocorre a aquisição das cenas naturais na forma de imagens de vídeo, pré-processamento e segmentação das imagens, seleção dos objetos de interesse e a extração de características dos objetos segmentados. Com a obtenção das características dos objetos de interesse das cenas associadas aos dados obtidos do GPS, um modelo de dados de objetos reais é gerado e disponibilizado para o uso do módulo mundo virtual.

No módulo **modelos 3D** são desenvolvidos modelos de objetos 3D pré-definidos e disponibilizados para o uso do módulo mundo virtual através de um repositório de objetos virtuais de culturas agrícolas. Para facilitar o acesso a este repositório, um modelo de dados de objetos virtuais é gerado durante a modelagem dos objetos 3D, contendo a descrição de cada modelo e localização física do mesmo.

O módulo **mundo virtual** é o módulo responsável pela integração dos dados disponibilizados pelos módulos mundo real e modelos 3D, gerando um modelo de dados do

mundo virtual. A partir deste modelo de dados, a cena virtual pode ser gerada e disponibilizada para visualização e interação do usuário com o sistema de multiprojeção imersivo do LMVI.

O detalhamento de cada um desses módulos é apresentado nas seções seguintes.

Figura 4.1: Visão geral da metodologia proposta

4.2 Módulo Mundo Real

Como descrito anteriormente, este módulo é responsável pela geração do modelo de dados dos objetos reais extraídos das cenas naturais. A geração desse modelo é obtida com aplicação da uso da visão computacional associada aos dados obtidos de um GPS. O detalhamento de cada uma das etapas envolvidas neste módulo é descrito ao longo desta

seção.

4.2.1 Visão Computacional

Como descrito no Capítulo 2, Visão Computacional é a área da ciência que se dedica a desenvolver teorias e métodos voltados à extração automática de informações úteis contidas em imagens. Neste trabalho, as informações obtidas pela visão computacional permitem a geração de modelos de dados que devem ser utilizados para a montagem automática dos ambientes virtuais a partir de cenas naturais.

Para o desenvolvimento desta metodologia, foram abordadas algumas etapas fundamentais de um sistema de visão computacional, como pode ser visto no diagrama apresentado na Figura 4.2.

Figura 4.2: Etapas do sistema de visão computacional abordadas pela metodologia proposta.

A etapa de classificação e reconhecimento dos objetos de interesse não foi abordada, pois, logo após a segmentação das imagens a metodologia proposta prevê a escolha de um elemento específico - entre os elementos segmentados - para extração de suas características.

Todas as etapas apresentadas na Figura 4.2 são executadas quadro a quadro a partir das imagens de vídeo das cenas naturais. Os detalhes referentes ao desenvolvimento dessas etapas são apresentados ao longo desta seção.

4.2.1.1 Aquisição das Imagens

A fase de aquisição de imagens consiste em obter uma representação da informação visual, que seja a mais fiel possível e ao mesmo tempo processável por um computador. Tal

representação pode ser construída a partir de alguma fonte de radiação, como calor, raios X, luz visível e micro-ondas, captada por dispositivos sensíveis a tais radiações.

A metodologia desenvolvida nesta pesquisa baseia-se no uso de câmeras de vídeo CCD (*Charge Coupled Device*) como dispositivo de captação das imagens no espaço de cores RGB. Estas câmeras podem ser acopladas a um robô agrícola móvel ou algum outro elemento móvel que permita a movimentação da câmera pelo ambiente real. No entanto, alguns parâmetros devem ser levados em consideração durante a aquisição das imagens, como:

- o posicionamento da câmera no ambiente real;
- a orientação da câmera em relação aos objetos de interesse;
- a distância da câmera até o objeto de interesse

Posição e Distância da Câmera

Uma vez que a câmera se movimenta pela cena paralelamente ao solo, o posicionamento da câmera no ambiente real refere-se apenas a altura da câmera com relação ao solo.

Para que as características relativas a forma e o tamanho dos objetos de interesse nas imagens sejam preservadas durante a aquisição das imagens, definiu-se que a tanto a altura quando a distância da câmera em relação ao objeto devem ser suficientes para a captura da totalidade do objeto em um mesmo quadro da imagem. Ou seja, não existem cortes de partes do objeto quando este está enquadrado no centro da imagem. Logo, a distância e a altura da câmera com relação ao objeto de interesse devem ser suficientes para que o objeto seja enquadrado da base até o topo e de um lado até o outro, quando este estiver no centro da imagem, como pode ser visto na Figura 4.3.

Outro ponto importante, é que durante a movimentação da câmera pela cena agrícola a distância da câmera até os objetos de interesse deve ser razoavelmente constante ou com poucas variações. Do contrário, as estimativas de tamanho do objeto segmentado podem levar a resultados incompatíveis com a realidade.

Orientação da Câmera

Quanto a orientação da câmera, se esta for paralela ou oblíqua à linha de plantio, como apresentado na Figura 4.4, os objetos de interesse se sobrepõem, dificultando a identificação

Figura 4.3: Enquadramento do objeto de interesse quando este se encontra no centro da imagem: (a) objeto totalmente enquadrado, (b) objeto parcialmente enquadrado.

de objetos individuais.

Figura 4.4: Orientação da câmera: (a) paralela à linha de plantio, (b) oblíqua à linha de plantio.

A fim de distinguir ou individualizar os elementos de uma cultura agrícola mais facilmente, a orientação da câmera de vídeo deve ser ortogonal à linha de plantio dos objetos de interesse e paralela à superfície do solo, como pode ser visto na Figura 4.5.

Configurações Básicas da Câmera

A metodologia adotada considera como configurações básicas da câmera de vídeo a taxa de quadros, a resolução e a profundidade de campo.

Um de vídeo digital nada mais é do que uma sequência quadros (imagens) de uma determinada cena capturados durante um intervalo de tempo. A taxa de quadros se refere ao número de quadros que são capturados por segundo durante a aquisição da imagem. Para a aquisição das imagens das naturais, 24 quadros por segundo (24 *fps*) é uma taxa adequada

Figura 4.5: Orientação da câmera ortogonal à linha de plantio e paralela à superfície do solo.

para a captura das cenas enquanto a câmera é deslocada pelo cenário agrícola.

A resolução das imagens se refere quantidade de *pixels* (pontos) que a imagem contém. Quanto maior a quantidade de *pixels* mais precisão e detalhes podem ser observados nas imagens. Todavia, resoluções muito altas exigem uma maior capacidade de armazenamento das imagens captadas. No contexto das cenas naturais, 1920 x 1080 *pixels* (alta definição) é uma resolução adequada para capturar a maior parte dos detalhes dos objetos.

Profundidade de campo (*Depth of field*) é a relação entre o objeto focado e tudo o que está ao seu redor. Ou seja, de acordo com o valor da profundidade de campo estabelecida antes da captura da imagem, os elementos da imagem que estiverem a frente ou atrás do plano de foco começam a perder nitidez. Quanto maior a profundidade de campo, maior é o foco em toda a imagem. Quanto menor a profundidade de campo, menor é o foco à frente e no fundo da imagem, com relação ao objeto que se encontra no plano de foco.

A configuração da profundidade de campo depende de pelo menos dois fatores: abertura do diafragma da lente e distância focal (*zoom*). Quanto mais aberto estiver o diafragma, menor será a profundidade de campo. Na medida que a abertura do diafragma diminui a profundidade de campo aumenta. Dependendo do e da distância entre os objetos contidos nas cenas naturais, a profundidade de campo deve ser ajustada para que os objetos de interesse sejam capturados com mais nitidez do que o fundo da imagem.

Iluminação da Cena Natural

As fontes de iluminação em cenas naturais, podem causar variação no reconhecimento de objetos de interesse. No caso de cenas naturais agrícolas, o sol é a principal, senão a única fonte de iluminação na cena. Devido às propriedades refletoras da superfície de cada objeto, o sol pode causar uma distribuição desigual de intensidade entre diferentes partes de um objeto, deixando algumas áreas mais claras e outras mais escuras.

As partes mais claras de um objeto podem ganhar mais destaque durante a segmentação e acabar sendo reconhecidas como objetos distintos das áreas com pouca luz. Assim, um objeto que deveria ser segmentado por inteiro, acaba sendo segmentado em partes separadas. Para minimizar este problema, as imagens devem ser preferencialmente capturadas em um dia e horário em que a distribuição de luz seja a mais uniforme possível nos objetos de interesse na cena.

Parâmetro de Identificação da Cultura

Antes de iniciar a captura das imagens, um operador deve informar ao sistema de visão computacional que tipo de cultura está sendo analisada. Ao final do módulo mundo real, este parâmetro é incorporado ao modelo de dados reais, facilitando a integração com o modelo de dados virtuais do módulo modelos 3D. O resultado dessa integração permite a localização física dos modelos 3D necessários para a montagem do ambiente virtual. Caso o parâmetro não seja informado, um parâmetro padrão (genérico) será utilizado pelo sistema de visão computacional.

Referência de Tamanho dos Objetos

Para que o tamanho de cada objeto segmentado seja estimado durante a extração das características, é necessário alguma referência de tamanho. Essa referência pode ser fornecida somente uma vez, no início da captura das imagens e pode ser feita de duas formas:

- utilizando um marcador de tamanho conhecido no campo de visão da câmera e na mesma distância do objeto enquadrado em relação a câmera;
- mensurando o primeiro objeto enquadrado no mundo real.

O tamanho da referência no mundo real (em metros ou cm) é convertido em unidades de *pixels* de acordo com o tamanho do quadro da imagem. Logo, para cada objeto identificado

durante a fase de segmentação, pode-se utilizar seu tamanho em *pixels* para encontrar a proporção entre ele e a referência fornecida. Em seguida, esta proporção é multiplicada pelo tamanho real da referência para obter-se o tamanho real do objeto em unidades como metros ou centímetros.

Disponibilização dos Dados de um GPS

Durante a captura de cada quadro das imagens, o robô agrícola móvel ou qualquer outro elemento que estiver movimentando a câmera pela cena agrícola deve disponibilizar dados de um GPS, como posição, orientação e altura. Dessa forma, durante a segmentação, no instante em que objeto de interesse é identificado, os dados do GPS atuais são geo-referenciados para o sistema de coordenadas do mundo real para obtenção da posição o objeto.

4.2.1.2 Pré-processamento das Imagens

As imagens adquiridas no estágio anterior podem conter erros ou ruídos que influenciam durante a segmentação das mesmas. Logo, o estágio de pré-processamento tem como principais objetivos a correção e a preparação da imagem para o estágio de segmentação.

Para corrigir os ruídos de uma imagem, deve-se aplicar algum filtro que possibilite a eliminação dos mesmos. O filtro Gaussiano adéqua-se a esta aplicação, uma vez que sua propriedade é de suavização das imagens adquiridas. Assim, neste estágio, as imagens são suavizadas utilizando o filtro Gaussiano, definido por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.1)$$

Onde, σ é o desvio padrão.

Dada a escolha do algoritmo de segmentação adotado nesta metodologia, antes do processamento das imagens estas devem ser transformadas do espaço de cores RGB para o espaço de níveis de cinza.

Uma maneira simples de obter-se o nível de cinza de um determinado *pixel* é através da média simples dos valores R, G e B. No entanto, os olhos humanos não detectam brilho linearmente com a cor. Basicamente, vemos o verde mais brilhante do que azul. Assim, uma maneira comumente utilizada em aplicações científicas é dada pela fórmula:

$$Cinza = R * 0.299 + G * 0.587 + B * 0.114 \quad (4.2)$$

Onde, os valores que multiplicam as componentes de cores primárias são ajustes de brilho.

A imagem resultante é uma imagem com apenas um canal de 8 *bits*, isto é, uma imagem com 256 níveis de cinza por *pixel*. Onde, 0 corresponde ao preto, 255 ao branco.

As operações realizadas durante o pré-processamento das imagens podem ser realizadas sobre uma cópia da imagem original para preservar os dados capturados no espaço de cores RGB.

4.2.1.3 Segmentação das Imagens

O estágio de segmentação das imagens consiste em dividir uma imagem de entrada em regiões que dizem respeito ao mesmo conteúdo, ou seja, em regiões homogêneas. Ao dividir-se uma imagem em partes constituintes, é possível caracterizar as regiões de interesse.

No processo de segmentação das cenas naturais, deve-se considerar que as imagens já foram transformadas do espaço de cores RGB para o espaço de níveis de cinza. Pois, a metodologia proposta utiliza os valores de descontinuidade dos níveis de cinza para a detecção de bordas na imagem.

Durante o processo de segmentação das imagens, as etapas sequenciais apresentadas na Figura 4.6 devem ser realizadas.

Figura 4.6: Etapas sequenciais realizadas durante a segmentação.

Detecção de Bordas

Para realizar esta tarefa, deve-se empregar um bom detector de bordas que possua a habilidade para:

- localizar e marcar todas as bordas realmente existentes na imagem;

- minimizar a distância entre a borda detectada e a borda verdadeira (real);
- fornecer respostas únicas, ou seja, não deve identificar múltiplos *pixels* de borda onde só existir uma borda.

O detector de bordas Canny (CANNY, 1986) atende a esses requisitos. Assim, o primeiro passo após o estágio de pré-processamento das imagens é o emprego desse algoritmo. Esse método de processamento de imagens identifica as bordas de uma imagem, retraindo apenas os pontos que são máximos locais do gradiente da imagem ao longo da direção do gradiente. O gradiente funciona como um detector de borda, enquanto a identificação do ponto de máximo localiza o evento com precisão. Canny utiliza a derivada combinada com o filtro Gaussiano. Essa filtragem reduz o efeito do ruído presente na imagem.

Após ter identificado os pontos de máximo local, o detector de bordas Canny precisa ligá-los formando uma linha de borda. Para isso, é utilizado um processo chamado limiarização por histerese. Este processo utiliza dois valores de limiares: o inferior e o superior. Se a intensidade de um *pixel* for maior do que o limiar superior ela é preservada na saída. Se a intensidade for menor do que o limiar inferior ela é descartada, sendo zerada a saída. Se a intensidade estiver entre os dois limiares, ela é preservada apenas se o *pixel* está topologicamente conectado a um *pixel* com intensidade maior do que o limiar superior. A Figura 4.7 mostra esquematicamente o funcionamento do detector Canny.

Figura 4.7: Estratégia do detector de bordas Canny.

Como pode ser visto na Figura 4.7, a entrada do detector Canny é uma imagem em níveis de cinza, enquanto a imagem resultante é uma imagem binária ¹ contendo as bordas detectadas, como pode ser visto na Figura 4.8.

¹Em uma imagem binária os valores possíveis de cada *pixel* são apenas dois: preto ou branco.

Figura 4.8: Imagens de entrada e saída do detector de bordas Canny respectivamente.

Ligação de Bordas Desconexas

Apesar do detector Canny ser considerado um detector de bordas ótimo, exceto sob circunstâncias ideais, a imagem resultante do algoritmo pode apresentar múltiplas bordas desconexas. Ou seja, a imagem resultante pode não fornecer uma borda completa que representa o contorno do objeto perfeitamente. Portanto, para uma melhor segmentação dos objetos de interesse é necessário ligar as bordas desconexas para obter-se o contorno do objeto. A estratégia utilizada nesta pesquisa combina o uso de duas operações morfológicas na imagem resultante do detector de bordas: erosão e dilatação.

O efeito da erosão em uma imagem binária é reduzir gradualmente os limites das regiões de *pixels* em primeiro plano (*pixels* brancos), enquanto a dilatação aumenta gradualmente os limites das regiões desses *pixels*. A combinação destas duas operações permite a ligação de bordas desconexas na imagem.

Ao final desta etapa, a imagem binária contém os contornos de todos os elementos identificados pelo detector de bordas, como pode ser visto através da Figura 4.9. No entanto, um mesmo objeto pode conter contornos externos, que representam a sua silhueta, e contornos internos que representam regiões de relevo. Como o objetivo da pesquisa é extrair o objeto completo da imagem, é necessário aplicar um algoritmo de extração de contornos externos.

Extração de Contornos Externos

Com o objetivo de extrair somente a silhueta dos objetos de interesse, a imagem binária proveniente do estágio anterior deve ser submetida a um algoritmo capaz de extrair somente os contornos externos. Para isto, pode ser utilizado um algoritmo clássico para extração de componentes conectados. Neste contexto, o algoritmo de Suzuki (SUZUKI; BE, 1985) é um filtro adequado para este tipo de operação.

Figura 4.9: Operações de fechamento e abertura respectivamente.

O algoritmo percorre os *pixels* da imagem a partir do canto superior esquerdo, linha a linha, procurando por um *pixel* branco. Quando este *pixel* é encontrado ele é rotulado com um valor inteiro e todos os *pixels* vizinhos a ele que também sejam brancos são rotulados com o mesmo valor. Em seguida, toda vez que um *pixel* é examinado, é verificado se o mesmo já possui algum rótulo, caso não possua, é verificado se algum de seus vizinhos possui algum rótulo, caso verdadeiro o rótulo do *pixel* vizinho é atribuído ao *pixel* analisado, senão um novo rótulo é atribuído.

Cada contorno detectado pelo algoritmo é armazenado em uma estrutura de dados como um vetor de pontos (*pixels*). Como cada contorno representa uma determinada região homogênea da imagem, a escolha de um desses contornos, implica na escolha de um objeto de interesse da imagem.

4.2.1.4 Extração de Características

Os dados resultantes da segmentação descrita na etapa anterior é uma estrutura de dados com os vetores de pontos de todos os contornos obtidos. Devido as características das culturas agrícolas, durante a movimentação da câmera pelo ambiente real agrícola, no momento em que o objeto de interesse se encontra no centro da imagem, ele é o elemento predominante. Sendo assim, o maior contorno da imagem é o objeto de interesse a ser analisado. Logo, somente o maior contorno detectado é considerado para a extração de suas características. Além disso, caso o maior contorno encontrado seja menor do que 10% do tamanho da imagem ele é descartado, pois assume-se que objetos muito pequenos não são objetos de interesse na cena natural.

Para encontrar o maior contorno da imagem, basta percorrer a estrutura de dados que contém todos os contornos da imagem e comparar a quantidade de *pixels* de cada um deles e armazenar sempre o maior valor encontrado. A partir do maior contorno encontrado a

etapa de extração de características deve ser iniciada.

Entre as características mais comuns, como: cor, textura, tamanho, posição e forma, a metodologia proposta utiliza somente a forma, o tamanho e a posição como parâmetros para a montagem automática dos ambientes virtuais. Para obtenção destes dois parâmetros, as etapas sequenciais apresentadas na Figura 4.10 devem ser realizadas.

Figura 4.10: Etapas sequenciais necessárias para a extração das características do contorno.

Localização do Centroide

Para extração das características do contorno que representa o objeto de interesse da cena natural, primeiro deve-se localizar um ponto especial em sua superfície plana, chamado centroide. Também conhecido como centro de gravidade de uma figura geométrica, este ponto possui uma distribuição homogênea de área em torno de si.

Nesta metodologia, a técnica utilizada para a determinação do centroide é em função dos momentos geométricos do contorno (TEAGUE, 1980). Os momentos são obtidos pela integração de todos os *pixels* do limite do contorno, utilizando a equação:

$$m_{p,q} = \int \int x^p y^q f(x,y) dx dy \quad (4.3)$$

Onde, $(p + q)$ determinam a ordem dos momentos e x e y são as coordenadas de cada ponto do contorno.

Para o cálculo do centroide, necessita-se apenas do momento de ordem zero e dos momentos de primeira ordem, como segue:

$$m_{0,0} = \int \int dx dy f(x,y) \quad (4.4)$$

$$m_{1,0} = \int \int dx dy x f(x,y) \quad (4.5)$$

$$m_{0,1} = \int \int dx dy y f(x,y) \quad (4.6)$$

O momento de ordem zero $m_{0,0}$ descreve a área do contorno. Através dos momentos de primeira ordem $m_{1,0}$ e $m_{0,1}$ e a área do contorno obtém-se as coordenadas do centroide do objeto como:

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad (4.7)$$

$$y_c = \frac{m_{0,1}}{m_{0,0}} \quad (4.8)$$

Traçado de Linhas de Interesse

Como a imagem resultante da etapa de segmentação é binária e descreve apenas os contornos dos objetos segmentados, para facilitar a extração das características desses objetos, deve-se traçar algumas linhas de interesse sobre a imagem original adquirida. As linhas de interesse, são linhas que permitem estimar o tamanho e a forma do objeto de interesse da cena natural e são definidas como:

- **Linha de Contorno:** deve-se traçar a linha do contorno externo selecionado utilizando as coordenadas x e y do vetor que armazena os pontos do contorno, como pode ser visto na Figura 4.11.

Figura 4.11: Traçado da linha de contorno sobre um objeto segmentado.

- **Segmento de Linha Vertical:** deve-se traçar um segmento de linha reta vertical que passa pelo centroide e divide o objeto de interesse em duas partes (Figura 4.12), aqui chamadas de hemisférios (esquerdo e direito), da seguinte forma:

Seja x_c e y_c as coordenadas do centroide e X , Y os vetores de coordenadas do contorno:

- a extremidade superior do segmento de linha vertical é determinada por $(x_c, \min(Y))$, onde $\min(Y)$ é o menor valor de y pertencente ao contorno.
- a extremidade inferior do segmento de linha vertical é determinada por $(x_c, \max(Y))$, onde $\max(Y)$ é o maior valor de y pertencente ao contorno.

- **Segmentos de Linhas Horizontais:** para cada hemisfério (esquerdo e direito) do contorno, deve-se traçar n segmentos de linhas retas horizontais perpendiculares ao segmento de linha reta vertical, da seguinte forma:

- a distância entre os segmentos de linhas horizontais deve ser constante para todas as linhas. A escolha do valor da distância depende do nível de detalhes da silhueta que deseja-se extrair para geração do modelo de dados reais. O valor padrão adotado é de 10 *pixels*.

- a quantidade n segmentos de linhas horizontais deve ser determinada pela razão entre a altura do objeto (em *pixels*) e a distância entre os segmentos horizontais a serem traçados. A altura do objeto pode ser obtida através da distância entre as extremidades do segmento de linha vertical que divide o contorno em dois hemisférios.

Figura 4.12: Traçado da linha vertical que divide o objeto em dois hemisférios.

- iniciando-se a partir do topo do segmento de linha vertical, para cada hemisfério, traça-se um segmento de linha reta horizontal que parte do segmento de linha vertical até o ponto de contorno mais distante encontrado na imagem, como mostra a Figura 4.13.

Figura 4.13: Traçado das linhas horizontais perpendiculares a linha vertical.

O papel dos segmentos de linhas horizontais é o mesmo que dividir o objeto real segmentado em várias fatias para facilitar a montagem do objeto virtual correspondente, com base nas informações sobre estas fatias. Como os segmentos de linhas horizontais respeitam os limites do contorno, eles permitem a obtenção tanto do tamanho como da forma do objeto de interesse.

Seleção do Frame Para Extração das Características

Uma vez que todas as etapas do sistema de visão computacional estão sendo executadas quadro a quadro a partir das imagens de vídeo, para obtenção do tamanho e posição do objeto de interesse, é necessário selecionar quadros (*frames*) específicos para obtenção destes dados.

Conforme definido na etapa de aquisição das imagens, assume-se que quando o objeto de interesse estiver no centro do quadro da imagem, é possível segmentá-lo por inteiro. Ou seja, não existem cortes de partes do objeto quando este está enquadrado no centro da imagem. Dessa forma, a obtenção do tamanho e posição do objeto de interesse, só ocorre quando o centroide do contorno estiver no centro da imagem.

Como a câmera de vídeo está em movimento pela cena natural e o cálculo para localização do centroide é realizado quadro a quadro, não há garantias de que o centroide do contorno do objeto coincida com o centro da imagem. Isso ocorre, porque de um quadro para outro da imagem pode haver pequenas variações de iluminação e posicionamento da câmera em relação ao objeto, ocasionando uma variação na localização do centroide de um quadro para outro. Para solucionar este problema, ao invés de detectar se o centroide está ou não no centro da imagem, deve-se detectar se o centroide está contido em uma região (janela) no centro da imagem, como pode ser visto na Figura 4.14.

Como pode ser visto na Figura 4.14. A região central da imagem é delimita por dois segmentos de linhas verticais paralelas que ocupam toda a altura do quadro da imagem. Esta região aumenta a tolerância do processo de detecção da posição do centroide em relação ao centro da imagem. A distância entre as linhas que delimitam a região central da imagem pode ser parametrizada. O valor padrão adotado foi de 15% da largura do quadro.

Obtenção do Tamanho

Como descrito na Seção 4.2.1.1, para que o tamanho de cada objeto segmentado seja estimado durante a extração das características, é necessário alguma referência de tamanho.

Figura 4.14: Delimitação da região central do quadro da imagem.

Essa referência pode ser fornecida somente uma vez, no início da captura das imagens e pode ser feita de duas formas:

- utilizando um marcador de tamanho conhecido no campo de visão da câmera e na mesma distância do objeto enquadrado em relação a câmera;
- mensurando o primeiro objeto enquadrado no mundo real.

O tamanho da referência no mundo real (em metros ou cm) deve ser convertido em unidades de *pixels* de acordo com o tamanho do quadro da imagem. Logo, para cada objeto identificado durante a fase de segmentação, pode-se utilizar seu tamanho em *pixels* para encontrar a proporção entre ele e a referência fornecida. Em seguida, esta proporção deve ser multiplicada pelo tamanho real da referência para obter-se o tamanho real do objeto em unidades como metros ou cm.

A altura do objeto de interesse deve ser obtida pelo tamanho do segmento de linha vertical que divide o contorno em dois hemisfério. Os tamanhos de cada segmento de linha horizontal traçado ao longo do segmento de linha vertical até os limites do contorno, definem a largura do objeto tanto à esquerda quanto a direita.

Como a imagem adquirida e processada é uma representação 2D da cena natural, não existem dados que refletem à profundidade do objeto de interesse. No entanto, segundo Huang e Mayer (2009), a maioria culturas agrícolas apresentam simetria radial ou parcialmente radial. Assumindo que os objetos de interesse da cena natural possuem este tipo de simetria,

pode-se utilizar a média aritmética do tamanho dos segmentos de linhas horizontais à esquerda e direita do contorno, para estimar a profundidade do objeto em cada ponto onde foi traçada uma linha horizontal.

Obtenção da Posição

Como mencionado na Seção 4.2.1.1, durante a captura de cada quadro das imagens, o robô agrícola móvel ou qualquer outro elemento que estiver movimentando a câmera pela cena agrícola deve disponibilizar dados de um GPS, como posição, orientação e altura. Dessa forma, durante a segmentação, no instante em que objeto de interesse é identificado, os dados do GPS atuais são geo-referenciados para o sistema de coordenadas do mundo real para obtenção da posição o objeto.

Os dispositivos de GPS enviam mensagens num formato próprio, variando de dispositivo para dispositivo. No entanto, segundo Ramos (2007) a maioria deles transmitem mensagens no padrão NMEA (*National Marine Electronics Association*). As mensagens possuem formato ASCII (*American Standard Code for Information Interchange*), com um comprimento máximo de 82 caracteres, subdivididas em campos delimitados por vírgulas, contendo os identificadores da mensagem, os dados específicos para cada tipo de mensagem e os caracteres de término da *string*. Desta forma, seguindo o formato da norma, é possível extrair os dados de interesse da mensagem recebida.

A partir dos dados de latitude, longitude e altura em coordenadas geográficas, pode-se calcular a posição dos objetos em termos de coordenadas locais (X,Y,Z), conforme descrito por (ADAMCHUK, 2001). O primeiro passo é converter os valores de latitude e longitude de graus, minutos e segundos para graus decimais com o sinal correspondente (negativo para latitudes sul e longitudes oeste), da seguinte forma:

$$graus_decimais = graus + \frac{min}{60} + \frac{seg}{3600} \quad (4.9)$$

A partir dos dados em graus decimais, Adamchuk (2001) obtém um fator de conversão de latitude F_{lat} e longitude F_{lon} em função de uma latitude específica da seguinte forma:

$$F_{lat} = \frac{\pi}{180^\circ} \left(\frac{a^2 b^2}{(a^2 \cos^2 \varphi + b^2 \sin^2 \varphi)^{3/2}} + h \right) \quad (4.10)$$

$$F_{lon} = \frac{\pi}{180^\circ} \left(\frac{a^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}} + h \right) \cos \varphi \quad (4.11)$$

Onde, $a = 6.378.137m$ que corresponde ao valor do raio equatorial (em metros) e $b = 6.356.752,3142m$ que corresponde ao valor do raio polar (em metros) de um elipsoide que representa a superfície da terrestre. φ é o valor da latitude em graus decimais e h é altura acima do elipsoide.

Após a conversão dos dados do GPS em graus decimais com o sinal correspondente e a obtenção dos fatores de conversão F_{lat} e F_{lon} , a distância entre dois pontos pode ser obtida como:

$$Distancia = \sqrt{(F_{lat}(\varphi_1 - \varphi_2))^2 + (F_{lon}(\lambda_1 - \lambda_2))^2} \quad (4.12)$$

4.2.2 Modelo de Dados de Objetos Reais

A última etapa do módulo mundo real consiste na geração de um modelo de dados de objetos reais. Este modelo contém informações que descrevem os objetos reais segmentados das cenas naturais em termos das características extraídas a partir de seu contorno. O processo de geração e atualização desse modelo pode ser visto através da Figura 4.15.

Figura 4.15: Fluxograma do processo de geração e atualização do modelo de dados reais.

A geração e atualização desse modelo é de grande importância, pois é através da associação entre o modelo de dados de objetos reais e o modelo de dados de objetos virtuais (modelo que descreve os objetos virtuais previamente modelados) que o ambiente virtual agrícola pode ser montado.

4.2.2.1 Geração do Modelo

Como as informações referentes aos objetos extraídos das cenas naturais são formadas por campos e estruturas repetidas e numerosas, o modelo de dados de objetos reais deve possuir um formato simples, legível, extensível e que ofereça facilidade de compartilhamento de informações. Neste contexto, o padrão XML (*eXtensible Markup Language*) mostra-se como um formato adequado. Pois, segundo Teague, Palmer e Jackson (2003), o padrão XML permite a geração de um arquivo texto neutro em termos de *software*, capaz de armazenar qualquer tipo de informação estruturada ou semi-estruturada, encapsulando os dados e permitindo a troca de informações entre diferentes sistemas de computação. Logo, a geração do modelo de dados de objetos reais consiste na geração de um documento XML contendo as informações dos objetos reais obtidas com a visualização computacional.

O padrão XML define uma representação natural para estruturas hierárquicas com campos repetidos. A estrutura de um documento XML é uma árvore de elementos representados por *tags*. Esses elementos podem conter outros elementos ou atributos organizados hierarquicamente. Dessa forma, o processo de geração do documento XML com base nas características extraídas de cada objeto de interesse das cenas naturais é formado pelos seguintes passos:

1. geração do cabeçalho do documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

2. geração do elemento raiz do documento XML:

```
<objetos_reais>
```

3. geração do elemento que define o objeto real extraído e seus atributos:

```
<objeto id="0" tipo="padrao">
```

- o atributo “id” deve conter um número inteiro único para identificação de cada objeto extraído da cena.
- o atributo “tipo” deve conter uma *string* que identifica o tipo de cultura da qual este objeto foi extraído. Como descrito na Seção 4.2.1.1, esta informação pode ser informada por um operador ao sistema de visão computacional. Caso nenhum parâmetro de identificação da cultura seja informado, um parâmetro padrão deve ser utilizado.

4. geração do elemento que define a posição do centro do objeto real e sua respectiva estrutura hierárquica:

```
<centro>
  <posicao x="0" y="0" z="0" />
</centro>
```

- os atributos “x,y,z” do elemento “posicao” devem conter números em ponto flutuante obtidos através dos dados de posição do GPS.
5. geração do elemento principal que abriga as informações referentes as características de todos os segmentos de linhas horizontais traçados sobre o contorno do objeto de interesse:

```
<segmentos qtde="50">
```

- o atributo “qtde” deve conter um número inteiro referente a quantidade de segmentos de linha que foi utilizado para extrair as características do objeto de interesse da cena natural. Como a distância entre os segmentos de linha é fixa, a quantidade de segmentos varia de acordo com o tamanho do objeto.
6. geração dos elementos que contêm as informações sobre os segmentos de cada hemisfério objeto:

```
<hemisferio_esquerdo>
<hemisferio_direito>
```

- durante a etapa de extração de características do objeto de interesse, foi traçado um segmento de linha vertical passando pelo centroide do contorno que dividiu o objeto de interesse em duas partes, aqui chamadas de hemisférios (esquerdo e direito).
7. geração dos elementos que contêm as informações sobre cada segmento de linha horizontal traçado no contorno do objeto de interesse :

```
<segmento_esquerdo id="0" tamX="1" tamY="1" tamZ="0.1" />
<segmento_direito id="0" tamX="0" tamY="0" tamZ="0.1" />
```

- as informações sobre cada segmento de linha são organizadas de acordo com o hemisfério ao qual pertencem.

- os atributos “tamX, tamY, tamZ” devem conter números em ponto flutuante que definem a largura, profundidade e altura de cada segmento do objeto de interesse.

4.2.2.2 Atualização do Modelo

O documento XML contendo o modelo de dados de objetos reais é gerado uma única vez no início da execução da visão computacional. No entanto, os dados deste documento são atualizados para cada objeto de interesse obtido das cenas naturais, conforme o processo descrito na Figura 4.15, apresentada anteriormente.

A atualização do documento XML não sobrescreve os dados já inseridos, apenas adiciona novos dados no arquivo a medida que mais informações são extraídas das cenas. O parâmetro “id” do elemento XML “objeto” permite a distinção dos elementos inseridos no documento. Dessa forma, não é preciso aguardar o término da execução da visão computacional para acessar o modelo de dados de objetos reais.

4.2.2.3 Estrutura do Modelo Gerado

Os elementos XML descritos na seção anterior devem ser organizados de uma maneira hierárquica para geração do modelo de dados de objetos reais e disponibilizados como um documento XML bem formado para o uso do módulo mundo virtual. A hierarquia dos elementos do documento XML gerado durante os estágios da visão computacional pode ser vista através da Figura 4.16.

```
<?xml version="1.0" encoding="UTF-8" ?>
<objetos_reais>
  <objeto id="0" tipo="padrao">
    <centro>
      <posicao x="0" y="0" z="0" />
    </centro>
    <segmentos qtde="2" espacamento="0.1">
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" tamX="4.0016" tamY="7.0008" tamZ="0.1" />
        <segmento_esquerdo id="1" tamX="10.002" tamY="5.0008" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" tamX="0" tamY="0" tamZ="0.1" />
        <segmento_direito id="1" tamX="0" tamY="0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
</objetos_reais>
```

Figura 4.16: Estrutura hierárquica do modelo de dados de objetos reais.

4.3 Módulo Modelos 3D

No módulo **modelos 3D** são desenvolvidos modelos de objetos 3D pré-definidos e disponibilizados para o uso do módulo mundo virtual através de um repositório de objetos virtuais de culturas agrícolas. Para facilitar o acesso a este repositório, um modelo de dados de objetos virtuais é gerado durante a modelagem dos objetos 3D, contendo a descrição de cada modelo e localização física do mesmo.

4.3.1 Repositório de Objetos Virtuais 3D

O repositório de objetos virtuais 3D é um local onde os modelos 3D previamente modelados são armazenados e disponibilizados para o compartilhamento entre sistemas. O acesso a este repositório é realizado através de um modelo de dados de objetos virtuais que descreve o tipo as características e o arquivo correspondente de cada modelo 3D.

A modelagem dos objetos 3D pode ser realizada através de qualquer ferramenta de criação de objetos 3D que permita a exportação dos modelos para um arquivo 3D, como o formato VRML (*Virtual Reality Modeling Language*).

Como a metodologia de proposta neste trabalho utiliza segmentos de linhas horizontais para extração das características dos objetos de interesse nas cenas naturais, os modelos 3D devem ser gerados de forma que representem estes segmentos. No contexto de culturas agrícolas, devem ser gerados modelos 3D em forma de semicírculos que representam os segmentos de linhas horizontais esquerdos e direitos, como pode ser visto na Figura 4.17.

Os semicírculos apresentados na Figura 4.17, devem ser modelados com o diâmetro de um metro. Dessa forma, os valores de largura e profundidade de cada segmento de linha horizontal do modelo de dados reais, podem ser utilizados como fatores de multiplicação de escala desses objetos. Assim, um mesmo modelo 3D pode ser utilizado em diferentes partes do objeto virtual a ser montado, apenas ajustando a sua escala.

Durante a modelagem, os efeitos visuais não podem ser os únicos requisitos considerados. O desempenho da aplicação responsável pela exibição destes modelos é um importante requisito que deve ser levado em consideração. Dessa forma, os modelos podem ser gerados com diferentes níveis de detalhe. A Figura 4.18 apresenta outros modelos 3D que também representam os segmentos de linhas horizontais utilizados metodologia, porém com outro nível de detalhe.

Os níveis de detalhes (*Level of Detail - LOD*) são conjuntos de modelos que

Figura 4.17: Modelos 3D que representam os segmentos de linhas horizontais.

Figura 4.18: Modelos 3D que representam os segmentos de linhas horizontais com maior nível de detalhes.

representam o mesmo objeto com uma variação de graus de complexidade. Durante a execução de uma aplicação de RV, o sistema seleciona um nível detalhe para ser exibido, dependendo da distância do ponto de vista do observador (*eyepoint*) até o objeto.

A variação de complexidade se refere a quantidade de polígonos utilizados para processar o modelo. Se o ponto de vista está longe de um objeto, este objeto é exibido com um nível de detalhe baixo, contendo poucos polígonos. Conforme o ponto de vista se move em direção ao objeto, o sistema substitui este objeto por outro com nível de detalhe mais complexo. Assim, seu uso desempenha um importante papel na otimização do processamento da cena virtual.

4.3.2 Modelo de Dados de Objetos Virtuais

O modelo de dados de objetos virtuais contém informações sobre os modelos 3D disponíveis no repositório de objetos virtuais 3D. O processo de geração e atualização desse modelo, pode ser visto através da Figura 4.19.

Figura 4.19: Fluxograma do processo de geração e atualização do modelo de dados virtuais.

Através da associação entre o modelo de dados de objetos reais e o modelo de dados de objetos virtuais o ambiente virtual agrícola pode ser montado.

4.3.2.1 Geração do Modelo

Do mesmo modo que a geração do modelo de dados de objetos reais, a geração do modelo de dados de objetos virtuais consiste na geração de um documento XML contendo as informações dos objetos virtuais disponíveis no repositório de objetos virtuais 3D. O processo de geração do documento XML com base nas informações de cada modelo 3D é formado pelos seguintes passos:

1. geração do cabeçalho do documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

2. geração do elemento raiz do documento XML:

```
<objetos_virtuais>
```

3. geração do elemento que define o objeto virtual previamente modelado:

```
<objeto id="0" tipo="padrao">
```

- o atributo “id” deve conter um número inteiro único para identificação de cada objeto modelado e disponibilizado no repositório.
- o atributo “tipo” deve conter uma *string* que identifica o tipo de cultura que este objeto representa. Caso o objeto não represente um tipo de cultura específico, um parâmetro padrão deve ser utilizado.

4. geração do elemento que define a posição do centro do objeto virtual e sua respectiva estrutura hierárquica:

```
<centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">  
  <posicao x="0" y="0" z="0" />  
</centro>
```

- o atributo “modelo” do elemento “centro” deve conter uma *string* que indica o nome e o caminho do arquivo que contém o modelo 3D.
- os atributos “tamX, tamY, tamZ” do elemento “centro” devem conter números em ponto flutuante que definem a escala do objeto virtual previamente modelado.
- os atributos “x,y,z” do elemento “posicao” devem conter números em ponto flutuante que indicam as coordenadas do mundo virtual em que o objeto foi criado.

5. geração do elemento principal que abriga as informações referentes a todos os segmentos de linhas horizontais utilizados pela metodologia:

```
<segmentos>
```

6. geração dos elementos que contêm as informações sobre os segmentos de cada hemisfério objeto:

```
<hemisferio_esquerdo>  
<hemisferio_direito>
```

7. geração dos elementos que contêm as informações sobre cada modelo 3D que representa um segmento de linha horizontal, utilizado pela metodologia:

```
<segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="0.5" tamY="1.0" tamZ="0.1"/>
<segmento_direito id="0" modelo="seg_dir.wrl" tamX="0.5" tamY="1.0" tamZ="0.1" />
```

- o atributo “modelo” de cada elemento deve conter uma *string* que indica o nome e o caminho do arquivo que contém o modelo 3D.
- os atributos “tamX, tamY, tamZ” de cada elemento devem conter números em ponto flutuante que definem a escala do objeto virtual previamente modelado.

4.3.2.2 Atualização do Modelo

O documento XML contendo o modelo de dados de objetos virtuais deve ser atualizado a cada novo modelo 3D construído e disponibilizado no repositório de objetos virtuais 3D, conforme o processo descrito na Figura 4.19, apresentada anteriormente.

A atualização do documento XML não sobrescreve os dados já inseridos, apenas adiciona novos dados no arquivo a medida que novos modelos são construídos. O parâmetro “id” do elemento XML “objeto” permite a distinção dos modelos 3D inseridos no documento.

4.3.2.3 Estrutura do Modelo Gerado

Os elementos XML descritos na seção anterior devem ser organizados de uma maneira hierárquica para geração do modelo de dados de objetos virtuais e disponibilizados como um documento XML bem formado para o uso do módulo mundo virtual. A hierarquia dos elementos desse documento XML pode ser vista através da Figura 4.20.

```
<?xml version="1.0" encoding="UTF-8" ?>
<objetos_virtuais>
  <objeto id="0" tipo="padrao">
    <centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">
      <posicao x="0" y="0" z="0" />
    </centro>
    <segmentos>
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="0.5" tamY="1.0" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" modelo="seg_dir.wrl" tamX="0.5" tamY="1.0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
</objetos_virtuais>
```

Figura 4.20: Estrutura hierárquica do modelo de dados de objetos virtuais.

4.4 Comunicação de Dados

Como apresentado anteriormente, os dados gerados pelos módulo mundo real e modelos 3D são: o modelo de dados de objetos reais e o modelo de dados de objetos virtuais respectivamente. Estes dados devem ser disponibilizados para o acesso do módulo mundo virtual para que o ambiente virtual com os objetos de interesse das cenas naturais seja montado.

Como descrito na Seção 4.2.2, o documento XML contendo o modelo de dados de objetos reais é atualizado para cada objeto de interesse obtido das cenas naturais. Essa atualização não sobrescreve os dados já inseridos, apenas adiciona novos dados no arquivo a medida que mais informações são extraídas das cenas. O parâmetro “id” do elemento XML “objeto” permite a distinção dos elementos que já foram lidos em acessos anteriores de elementos adicionados e ainda não lidos no acesso corrente. Dessa forma, não é preciso aguardar o término da execução da visão computacional para acessar o modelo de dados de objetos reais.

A comunicação de dados através da Internet permite que durante a execução da visão computacional, o ambiente virtual possa ser montado remotamente em tempo quase real (modo *on-line*) a medida que o modelo de dados de objetos reais é gerado e disponibilizado na rede, como pode ser visto na Figura 4.21.

Figura 4.21: Comunicação de dados entre os módulos.

Dependendo da distância e os meios de comunicação disponíveis no ambiente real remoto, a disponibilização dos dados na Internet pode ser realizada através da transmissão do arquivo XML para uma estação de dados mais próxima que permita a disponibilização dos dados na *web*.

4.5.1.1 Geração do Modelo

O conteúdo das informações desse modelo de dados é obtido com a associação do modelo de dados dos objetos reais com o modelo de dados dos objetos virtuais. Logo, a geração do modelo de dados do ambiente virtual consiste na geração de um documento XML que representa esta associação. O processo de geração do documento XML com base nas informações de cada modelo de dados é formado pelos seguintes passos:

1. geração do cabeçalho do documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

2. geração do elemento raiz do documento XML:

```
<ambiente_virtual>
```

3. geração do elemento que define o objeto virtual a ser inserido no AV:

```
<objeto id="0" tipo="padrao">
```

- o atributo “id” deve conter um número inteiro único para identificação de cada objeto virtual.
- o atributo “tipo” deve conter uma *string* que identifica o tipo de cultura que este objeto representa. Caso o objeto não represente um tipo de cultura específico, um parâmetro padrão deve ser utilizado.

4. geração do elemento que define a posição do centro do objeto virtual e sua respectiva estrutura hierárquica:

```
<centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">  
  <posicao x="0" y="0" z="0" />  
</centro>
```

- somente o valor do atributo “modelo” deve ser obtido através do modelo de dados de objetos virtuais, os valores dos demais atributos são originários do modelo de dados de objetos virtuais.

5. geração do elemento principal que abriga as informações referentes a todos os segmentos de linhas horizontais utilizados pela metodologia:

```
<segmentos qtde="2" espacamento="0.1">
```

- todos os atributos deste elemento são originários do modelo de dados de objetos reais.

6. geração dos elementos que contêm as informações sobre os segmentos de cada hemisfério objeto:

```
<hemisferio_esquerdo>
```

```
<hemisferio_direito>
```

7. geração dos elementos que contêm as informações tanto dos objetos reais como dos virtuais:

```
<segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="4.0" tamY="7.0" tamZ="0.1"/>
```

```
<segmento_direito id="0" modelo="seg_dir.wrl" tamX="0" tamY="37.0" tamZ="0.1"/>
```

- somente o valor do atributo “modelo” foi obtido através do modelo de dados de objetos virtuais, os valores dos demais atributos são originários do modelo de dados de objetos reais.

4.5.1.2 Atualização do Modelo

O documento XML contendo o modelo de dados do ambiente virtual é gerado uma única vez no início da montagem do ambiente virtual. No entanto, sempre que o modelo de dados de objetos reais for atualizado este modelo deve consequentemente ser atualizado também, como pode ser visto na Figura 4.23.

A atualização do documento XML não sobrescreve os dados já inseridos, apenas adiciona novos dados no arquivo a medida que mais informações são obtidas a partir do modelo de dados reais. O parâmetro “id” do elemento XML “objeto” permite a distinção dos elementos que já foram lidos inseridos no documento, bem como os objetos 3D que já foram inseridos no ambiente virtual. Dessa forma, não é preciso aguardar o término da geração do modelo de dados virtuais completo para a montagem do AV. A medida que o documento é atualizado um novo objeto 3D é inserido na cena virtual.

4.5.1.3 Estrutura do Modelo Gerado

Assim como os demais modelos de dados, os elementos XML descritos na seção anterior devem ser organizados de uma maneira hierárquica para geração do modelo de dados

Figura 4.23: Fluxograma do processo de geração e atualização do modelo de dados do AV.

do ambiente virtual. A hierarquia dos elementos desse documento XML pode ser vista através da Figura 4.24.

4.5.2 Montagem do AV em um Sistema de Multiprojeção Imersivo

A modelagem de ambientes virtuais 3D é uma tarefa que exige habilidade, tempo e recursos para criação de objetos virtuais que representem objetos reais. Os projetistas precisam abstrair as características geométricas dos elementos que compõem a cena real, para

```

<?xml version="1.0" encoding="UTF-8" ?>
<ambiente_virtual>
  <objeto id="0" tipo="padrao">
    <centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">
      <posicao x="335" y="0" z="0" />
    </centro>
    <segmentos qtde="2" espacamento="0.1">
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="4.0016" tamY="7.0008" tamZ="0.1" />
        <segmento_esquerdo id="1" modelo="seg_esq.wrl" tamX="10.002" tamY="5.0008" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" modelo="seg_dir.wrl" tamX="0" tamY="37.0008" tamZ="0.1" />
        <segmento_direito id="1" modelo="seg_dir.wrl" tamX="0" tamY="0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
</ambiente_virtual>

```

Figura 4.24: Estrutura hierárquica do modelo de dados do ambiente virtual.

que através do uso de polígonos possam construir um modelo geométrico tridimensional de uma paisagem virtual (incluindo terreno, árvores, etc). Mas isso não é o suficiente, uma vez que a montagem da cena final exige o conhecimento da localização, orientação e tamanho de cada elemento da cena.

A orientação, posição e dimensionamento dos modelos 3D na cena virtual são problemas que devem ser superados para se obter uma construção automatizada de um ambiente virtual livre de intervenção do usuário. Através da metodologia proposta, estas informações já estão disponíveis no modelo de dados do ambiente virtual. Logo, a montagem do ambiente virtual é realizada com base na descrição da cena virtual registrada no documento XML que descreve a estrutura do AV a ser montado.

O processo de montagem, inicia-se através do *parse* do documento XML, identificando qual modelo 3D deverá ser selecionado no repositório de objetos virtuais 3D. Através do nome do arquivo 3D do modelo descrito, a aplicação carrega o respectivo arquivo e insere no grafo de cena para posterior *renderização*. Este processo se estende enquanto houverem informações no documento XML, seja em modo *on-line* ou *off-line*. Para não inserir o mesmo objeto do modelo de dados mais de uma vez, deve ser verificado se o mesmo já foi inserido ou não através de seu identificador.

O objetivo desta etapa é sintetizar uma cena natural real a partir de imagens 2D em um mundo 3D apenas com as informações contidas no modelo de dados do ambiente virtual. O fluxograma que descreve o processo de montagem do AV, pode ser visto na Figura 4.25.

4.5.2.1 Carregamento do arquivo 3D em memória

Como descrito anteriormente, o atributo “modelo” dos elementos XML que compõem o objeto real extraído das cenas naturais, contém o nome e o caminho do arquivo 3D correspondente no repositório de objetos virtuais 3D. Como cada objeto a ser montado na cena

Figura 4.25: Fluxograma do processo de montagem do ambiente virtual.

virtual pode requisitar a leitura e o carregamento de um mesmo arquivo 3D repetidamente, após a seleção do arquivo 3D a ser inserido na cena virtual, primeiro verifica-se se este arquivo já foi carregado anteriormente e se encontra armazenando em memória. Se for verdade, usa-se apenas uma cópia (clone) do modelo carregado em memória otimizando tempo e recursos envolvidos em uma nova leitura e carregamento de dados do arquivo 3D. O fluxograma que descreve este processo pode ser visto na Figura 4.26.

A montagem do ambiente virtual é realizada independente do tipo de objeto que

Figura 4.26: Fluxograma do processo de carregamento dos modelos 3D.

deverá ser inserido. A partir do processamento das imagens de uma ambiente real e a extração dos objetos de interesse, pode-se montar um mundo virtual que representa cenas naturais de forma automática.

4.5.2.2 Inserção dos Modelos 3D no Grafo de Cena

Após a leitura e o carregamento dos dados do arquivo 3D, o modelo 3D descrito no arquivo deve ser inserido no grafo de cena do ambiente virtual, da seguinte forma:

1. o modelo 3D que corresponde a parte central do objeto é inserido no ponto (0,0,0) no grafo de cena como filho da raiz do grafo;
2. os modelos 3D que correspondem aos segmentos de linhas horizontais são inseridos como filhos da parte central de seu respectivo objeto;
3. através dos atributos de tamanho em X, Y e Z de cada elemento do modelo de dados, cada modelo inserido no grafo de cena tem sua escala ajustada multiplicando-se pelos valores descritos nesses atributos.
4. os modelos que correspondem aos segmentos de linhas horizontais são transladados em relação a parte central do objeto de acordo com o atributo “espaçamento” do documento

XML que descreve o modelo de dados. O processo de translação desses modelos produz um efeito de empilhamento de objetos em torno da parte central, formando o volume do objeto virtual 3D final após o término da montagem.

5. após a inserção de todas as partes que compõem o objeto virtual 3D, a parte central do objeto é translada para a posição indicada pelo atributo “posicao” do elemento “centro” do modelo de dados. Ao transladar a parte central do objeto, todos os elementos que são filhos dela (elementos que correspondem aos segmentos) serão translados simultaneamente para a posição desejada.

Se o processo de montagem do AV estiver ocorrendo de modo *on-line* acessando o modelo de dados reais pela Internet ou uma rede local, o processo de inserção dos modelos 3D no grafo de cena, ocorre simultaneamente ao processo de atualização do modelo de dados do ambiente virtual, que por sua vez é atualizado sempre o que o modelo de dados reais também é atualizado. Ou seja, toda vez que um novo objeto é extraído da cena natural e inserido no modelo de dados reais, o modelo de dados do ambiente virtual é atualizado e o processo de inserção dos modelos 3D no grafo de cena é repetido novamente.

Caso o processo de montagem do AV ocorra de modo *off-line*, todos os objetos descritos no modelo de dados do ambiente virtual são inseridos em lote na cena virtual.

4.5.2.3 Salvamento do Ambiente Virtual

Como o conteúdo do ambiente virtual pode ser gerado dinamicamente, é de suma importância que o ambiente montado possa ser salvo e armazenado em algum repositório para uso posterior. Este requisito pode ser facilmente obtido através do salvamento do documento XML que descreve o modelo de dados do ambiente virtual. Através desta abordagem, obtém-se uma modularidade do ambiente virtual montado, de forma que os modelos individuais 3D que compõem o AV podem ser alterados ou substituídos sem afetarem o processo de montagem. Além disso, o ambiente virtual pode ser recarregado no dispositivo de visualização sem a necessidade de acesso ao modelo de dados de objetos reais tanto remoto como local.

Se o ambiente gerado for todo armazenado em um único arquivo 3D, o processo de alteração do conteúdo do ambiente não deve apresentar a mesma flexibilidade da metodologia proposta. Além disso, o volume de dados de arquivo 3D é bem maior do que do documento XML proposto, otimizando tempo e recursos durante o recarregamento do AV no dispositivo de projeção.

4.5.3 Visualização do Ambiente Virtual

Como descrito no Capítulo 3, o Laboratório Multiusuário de Visualização Imersiva de São Carlos (LMVI) disponibiliza um sistema de multiprojeção imersivo do tipo CAVE de 3 faces, onde 2 delas correspondem à 2 paredes laterais e a terceira corresponde ao chão. Logo, a visualização do ambiente virtual montado pode ser realizada através desse sistema.

Para cada tela de projeção, dois projetores com filtros polarizadores projetam um par estereoscópico de imagens, que ao olho nu o espectador consegue visualizar uma imagem com predominância da cor verde e a outra com predominância da cor roxa, sobrepostas e desfocadas, como pode ser visto na Seção 3.2.3. Somente através do uso de óculos estereoscópicos passivos com filtros polarizadores de mesma cor que as imagens projetadas, o espectador visualiza a imagem tridimensional.

Para que a cena virtual montada possa ser realmente visualizada através do sistema de multiprojeção imersivo, algumas fontes de luz devem ser inseridas no grafo de cena durante a inicialização do sistema. A iluminação do ambiente virtual pode ser localizada em objetos ou grupos de objetos ou em todo o universo virtual de acordo com a posição que os nós de iluminação são inseridos no grafo. Um exemplo de visualização de um ambiente virtual através do sistema do LMVI pode ser visto na Figura 4.27.

Figura 4.27: Visualização de um AV através do sistema de multiprojeção imersivo.

4.5.3.1 Interação

Para utilização do ambiente virtual projetado no LMVI, além dos óculos estereoscópicos é necessário o uso de algum tipo de dispositivo de interação. Para isso, é necessário o

uso de alguns dispositivos de entrada de dados, que possam ser lidos e interpretados como comandos do usuário para realizar alguma ação no ambiente 3D.

Como forma de interagir com o ambiente virtual projetado o usuário pode utilizar algum dos dispositivos de interação disponíveis no sistema de multiprojeção imersivo, como: mouse, teclado, *joystick*, controle de vídeo game Wii e dispositivos de interação por luz e detecção de movimento com o auxílio de uma *webcam*. A descrição de cada um desses dispositivos, suas implementações e uso encontra-se detalhada na Seção 3.7.

4.6 Considerações Finais do Capítulo

Este capítulo apresentou a metodologia de montagem automática de ambientes virtuais a partir de cenas naturais de culturas agrícolas abordando os aspectos referentes a visualização e utilização dos ambientes virtuais montados em um sistema de multiprojeção imersivo. O metodologia aqui apresentada foi dividida três módulos principais: **mundo real**, **modelos 3D** e **mundo virtual**.

Os modelos de dados reais, virtuais e do mundo virtual oferecem uma grande extensibilidade para metodologia proposta, pois além de objetos de culturas agrícolas, outros objetos podem ser facilmente incorporados nestes modelos respeitando a estrutura apresentada. Além disso, a estrutura hierárquica dos modelos de dados permite adicionar novos elementos aos objetos extraídos da cenas naturais, de acordo com o nível de segmentação aplicado no processamento das imagens.

A comunicação de dados utilizando arquivos XML ao invés das imagens de vídeo, torna a comunicação entre os módulos mais leve e mais rápida pelo menor volume de dados a serem acessados. Além disso, como os dados podem ser disponibilizados na Internet ou em uma Rede Local, o módulo mundo real não precisa enviar dados para nenhum destinatário específico. Os módulos ou aplicações que requerem acesso ao modelo de dados reais é que devem realizar o *download* do mesmo. Isso permite que os módulos que compõem a metodologia proposta possam ser executados de maneira independente um dos outros.

Como a execução do módulo mundo virtual pode ocorrer em paralelo ou sequência com o módulo mundo real, a montagem automática dos ambientes virtuais pode ser realizada tando em modo *on-line* (tempo quase real) como *off-line* (em lote).

A visualização e interação dos ambientes virtuais montados em um sistema de multiprojeção imersivo permitem a percepção de um maior realismo da cena gerada, bem como

a utilização para diferentes propósitos como análise de crescimento de culturas, navegação de robôs agrícolas móveis, distribuição de pragas, alterações causadas por fatores climáticos, entre outros.

5 Testes de Aplicação e Resultados

Neste capítulo são apresentados os testes de aplicação da metodologia proposta, bem como os resultados obtidos. Inicialmente, a etapa de aquisição das imagens é apresentada, em seguida, os testes são divididos de acordo com três ângulos de orientação da câmera em relação aos objetos de interesse nas cenas naturais: orientação ortogonal, paralela e oblíqua.

A apresentação dos resultados obedece a mesma sequência de etapas apresentada no Capítulo 4: visão computacional, geração dos modelos de dados, integração dos modelos, montagem do ambiente virtual e visualização 3D imersiva.

As ferramentas computacionais utilizadas para a implementação de cada uma das etapas da metodologia proposta são descritas na Seção 3.6.

5.1 Aquisição das Imagens

Para realização dos testes, foram adquiridas imagens de cenas naturais de laranjeiras. Com a permissão do proprietário, Sr. Viriato, as imagens foram adquiridas na Fazenda Jangada Brava no município de Boa Esperança do Sul, Estado de São Paulo.

As coordenadas geográficas da área são: latitude 21°56' S e longitude 48°19' O. A altitude média é de 490 metros em relação ao nível do mar.

Os objetos de interesse para extração de características e posterior montagem do ambiente virtual foram laranjeiras “Valência”, enxertadas em citrumeleiro “*Swingle*”.

O dispositivo de captura foi uma câmera digital Nikon 3100 com lente 18-105 mm e suas configurações durante as filmagens foram:

- foco automático;
- modo filme HD (1920x1080);
- 24 fps (quadros por segundo).

Durante a movimentação da câmera pelo cenário real agrícola, duas pessoas (alternadamente) simularam a movimentação de um robô agrícola móvel se locomovendo entre as linhas de plantio em baixa velocidade (aprox. 1 m/s), segurando a câmera digital voltada para as laranjeiras.

Para referência de tamanho durante a etapa de extração de características das imagens, as primeiras árvores de cada vídeo tiveram suas áreas de copas mensuradas em campo, permitindo assim encontrar uma correspondência de tamanho em *pixels* para estimativa do tamanho real desses objetos apenas a partir das imagens de vídeo.

Para que a distância entre a câmera e as laranjeiras permanecesse razoavelmente constante ou com poucas variações, foi fixado uma linha de referência (Figura 5.1) que auxiliou o deslocamento durante as filmagens.

Figura 5.1: Linha de referência para deslocamento entre as linhas de plantio.

Durante a aquisição das imagens, foram gravados 20 vídeos de aproximadamente 40 s cada (960 quadros de imagem por vídeo). Cada vídeo contém o registro de 10 árvores em média, totalizando um banco de imagens de 19200 quadros de pelo menos 200 árvores de laranjeiras com diferentes ângulos de orientação da câmera. Nas seções seguintes são apresentados os testes e os resultados obtidos a partir dessas imagens.

5.2 Orientação Ortogonal à Linha de Plantio

Neste primeiro caso de teste, a orientação da câmera foi ortogonal à linha de plantio dos objetos de interesse e paralela à superfície do solo. A altura média da câmera em relação ao solo foi de 1,70 m e a distância em relação as laranjeiras foi de 1,35 m. Tanto a altura

como a distância foram suficientes para a captura da totalidade dos objetos em um mesmo quadro da imagem, como pode ser visto na Figura 5.2.

Figura 5.2: Imagem da câmera com a orientação ortogonal à linha de plantio.

5.2.1 Pré-processamento

Conforme definido na metodologia proposta, as imagens adquiridas foram transformadas do espaço de cores RGB para o espaço de níveis de cinza. Além disso, foi aplicado o filtro Gaussiano para correção de ruídos, através da suavização das imagens adquiridas. As imagens originais e as imagens resultantes podem ser vistas na Figura 5.3.

Figura 5.3: Imagens adquiridas nos espaços de cores RGB (esquerda) e pré-processadas no espaço de níveis de cinza com filtro Gaussiano(direita).

Antes da transformação das imagens do espaço de cores RGB para o espaço de níveis de cinza, as imagens originais foram preservadas e apenas cópias dessas imagens sofreram essas transformações para serem submetidas à etapa de segmentação.

5.2.2 Segmentação

A partir das imagens pré-processadas, foi aplicado o detector de bordas Canny para localizar regiões onde há mudança brusca nos níveis de cinza da imagem. Ou seja, o detector procura destacar arestas na imagem, analisando descontinuidades nos níveis de cinza e atenuando ruídos.

As imagens resultantes do algoritmo Canny são imagens binárias contendo apenas bordas brancas em um fundo preto, como pode ser visto na Figura 5.4.

Figura 5.4: Imagens no espaço de níveis de cinza (esquerda) e imagens binárias resultantes do detector de bordas Canny (direita).

Observe que regiões com maior tonalidade (sombra) são agrupadas separadas de regiões de maior incidência de luz. Além disso, como cada etapa da metodologia é aplicada quadro a quadro na imagem, a detecção de bordas dos objetos das imagens varia de quadro para quadro, como pode ser visto na Figura 5.5. Isso se explica pela diferença de tonalidades na imagem a medida que a câmera se desloca pelo ambiente real.

Figura 5.5: Detecção de bordas quadro a quadro a medida que a câmera se desloca pela cena natural.

Apesar do detector Canny ser considerado um detector de bordas ótimo, as imagens resultantes do algoritmo e apresentadas nas Figuras 5.4 e 5.5, apresentam múltiplas bordas desconexas. Com o intuito de encontrar uma borda completa que representa o contorno do objeto o mais próximo de sua silhueta, foram aplicadas as transformações morfológicas de fechamento e abertura, nessa ordem.

Como descrito no Capítulo 4, as transformações morfológicas de fechamento e abertura utilizam operadores morfológicos de erosão e dilatação. A operação de erosão é responsável pela redução gradual dos limites das regiões de *pixels* em primeiro plano, enquanto a dilatação é responsável pelo aumento gradual desses *pixels*. A combinação destas duas operações permite a ligação de bordas desconexas na imagem, como pode ser visto na Figura 5.6.

Figura 5.6: Imagens resultantes do detector de bordas Canny (esquerda) e imagens resultantes de transformações morfológicas de fechamento e abertura (direita).

Os resultados das transformações morfológicas demonstrados na Figura 5.6, apresentam tantos contornos externos quanto internos dos elementos identificados pelo detector de bordas. Como a metodologia propõe a extração da silhueta dos objetos de interesse para então extrair suas características, foi aplicado o algoritmo de Suzuki (SUZUKI; BE, 1985) para extração dos contornos externos das imagens. O resultado desse algoritmo são vetores de contornos contendo apenas os pontos (x, y) de cada *pixel* dos contornos externos.

Nesta etapa, a imagem original que havia sido preservada durante o pré-processamento foi utilizada para desenhar os contornos externos a partir dos pontos fornecidos pelos vetores resultantes do algoritmo de Suzuki.

Os resultados do traçado de contornos externos nas imagens no espaço de cores RGB podem ser visto na Figura 5.7.

Figura 5.7: Imagens binárias resultantes de transformações morfológicas de fechamento e abertura (esquerda) e imagens no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.

O traçado dos contornos externos foram realizados nas imagens originais adquiridas (no espaço de cores RGB) e preservadas durante o pré-processamento, para fins de comparação direta com o objeto segmentado. Caso o traçado fosse realizado nas imagens binárias resultantes das transformações morfológicas de fechamento e abertura, não seria possível visualizar os objetos reais capturados e os traçados externos encontrados, em uma mesma imagem.

5.2.3 Seleção do Objeto de Interesse

A Figura 5.7, apresenta imagens com mais de um contorno em um mesmo quadro. Seguindo a proposta da metodologia, nesta etapa, somente o maior contorno de cada quadro foi preservado, enquanto os demais foram desconsiderados para as fases seguintes.

Caso o maior contorno encontrado fosse menor do que 10% do tamanho da imagem ele também era descartado, pois assumiu-se que objetos muito pequenos não eram objetos de interesse na cena natural. As imagens resultantes podem ser vistas na Figura 5.8.

Figura 5.8: Imagens com traçados de todos os contornos externos (esquerda) e imagens contendo apenas o maior contorno em cada quadro (direita).

As diferenças de luminosidade geraram concavidades verticais (de baixo para cima ou o inverso) que não deveriam existir na imagem resultante. Este problema foi minimizado durante o traçado de segmentos de linhas de interesse e apresentado na seção 5.2.4.

5.2.4 Extração das Características

Nesta etapa, foi realizada a localização do centroide utilizando momentos geométricos. Em seguida, foram traçados os segmentos de linha, conforme descrito na metodologia proposta.

O primeiro segmento traçado foi a linha vertical que passa pelo centroide e divide o objeto de interesse em dois hemisférios: esquerdo e direito. Em seguida, foram traçados os segmentos de linhas horizontais perpendiculares ao segmento de linha vertical, que partem deste segmento até o ponto de contorno mais distante de sua origem. A distância entre os segmentos de linha horizontais foi constante, com o valor de 10 *pixels*.

A imagem resultante com o traçado dos segmentos de linhas estratégicas pode ser vista na Figura 5.9.

Figura 5.9: Imagem com o traçado do contorno externo (em cima) e imagem com os traçados dos segmentos de linhas (em baixo).

Na imagem de cima da Figura 5.9, observa-se que as diferenças de luminosidade em partes do objeto geraram concavidades verticais (de baixo para cima ou o inverso) ignorando partes do objeto que possuem certa intensidade de sombra ou nitidez. A estratégia de traçar os segmentos de linhas horizontais partindo do segmento de linha vertical central até o ponto mais distante do contorno permitiu o fechamento dessas concavidades (verticais) sem afetar as concavidades horizontais, como pode ser visto na imagem de baixo da Figura 5.9.

Os segmentos de linhas horizontais estão simplesmente dividindo o objeto real segmentado em várias fatias para facilitar a montagem do objeto virtual correspondente. Como os segmentos de linhas horizontais respeitam os limites do contorno, eles permitem a obtenção tanto do tamanho como da forma do objeto de interesse.

Como as operações aqui descritas ocorrem quadro a quadro, foram selecionados os principais quadros de um trecho das filmagens, onde objetos filmados consecutivamente, aparecem no início (da esquerda para direita), no centro e no fim do quadro. Esta sequência de imagens pode ser vista na Figura 5.10.

Figura 5.10: Traçado de segmentos de linhas em uma sequência de quadros de objetos filmados consecutivamente.

A Figura 5.10 mostra que em determinados quadros os objetos de interesse estão cortados, pois ainda não foram totalmente enquadrados enquanto a câmera se locomove pelo cenário. Logo, a Figura 5.11 apresenta somente os quadros onde os centroides dos objetos de interesse se encontram no centro da imagem.

Figura 5.11: Imagens onde o centroide do objeto de interesse se encontra no centro do quadro da imagem.

Note que, o centro da imagem foi delimitado por uma região ou janela que define se um objeto se encontra no centro do quadro. Dessa forma, no primeiro quadro em que o

centroide do objeto foi detectado nesta região, este quadro foi escolhido para extração das características.

As características de cada objeto foram obtidas mensurando o tamanho de cada segmento de linha traçado no contorno. Dessa forma foi possível a obtenção das dimensões do objeto no plano XY. Para obtenção do tamanho em Z (profundidade), foi calculado a média aritmética do tamanho dos segmentos de linhas horizontais à esquerda e direita do contorno, em cada ponto onde foi traçada uma linha horizontal. Esta estratégia se justifica pelo fato que segundo Ramos et al. (2010), o crescimento vegetativo da copa de laranjeiras tende a ser homogêneo quando consideradas as faces leste e oeste em plantios no sentido norte-sul. E segundo Huang e Mayer (2009), a maioria culturas agrícolas apresentam simetria radial ou parcialmente radial.

Como durante a captura as imagens não havia um dispositivo GPS disponível, para a realização destes testes a posição de cada objeto foi estipulada com base nos quadros em os mesmos eram detectados.

5.2.5 Geração do Modelo de Dados Reais

Para cada objeto detectado e analisado, as informações de tamanho e posição foram inseridas em um documento XML que representa o modelo de dados reais. O modelo resultante (compactado) para as imagens analisadas pode ser visto na Figura 5.12.

A medida em que os quadros de imagens são analisados, toda vez que um objeto é detectado e suas características são extraídas, este documento XML é atualizado e disponibilizado na Internet. Assim, o módulo mundo virtual não precisa esperar a segmentação de todos os quadros da imagem para iniciar a montagem do ambiente virtual.

5.2.6 Modelo de Dados Virtuais

Como a metodologia propõe o uso de modelos 3D previamente modelados, para serem utilizados somente para montagem da cena virtual, foram utilizados 3 modelos no formato VRML, que representam os segmentos de linhas traçados no contorno da imagem. As informações necessárias para utilização desses segmentos durante a montagem do ambiente virtual, foram reunidas em documento XML que representa o modelo de dados virtuais proposto. O documento XML resultante pode ser visto na Figura 5.13.

```

<?xml version="1.0" encoding="UTF-8" ?>
<objetos_reais>
  <objeto id="0" tipo="laranjeira">
    <centro>
      <posicao x="0" y="0" z="0" />
    </centro>
    <segmentos qtde="67">
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" tamX="74" tamY="37" tamZ="0.1" />
        <segmento_esquerdo id="1" tamX="110" tamY="55" tamZ="0.1" />
        <segmento_esquerdo id="2" tamX="110" tamY="56" tamZ="0.1" />
        <segmento_esquerdo id="3" tamX="113" tamY="56.5" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" tamX="0" tamY="0" tamZ="0.1" />
        <segmento_direito id="1" tamX="0" tamY="0" tamZ="0.1" />
        <segmento_direito id="2" tamX="1.9" tamY="56" tamZ="0.1" />
        <segmento_direito id="3" tamX="0" tamY="0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
  <objeto id="1" tipo="laranjeira">
    ...
  </objeto>
  <objeto id="2" tipo="laranjeira">
    ...
  </objeto>
  <objeto id="3" tipo="laranjeira">
    ...
  </objeto>
</objetos_reais>

```

Figura 5.12: Modelo de dados reais (compactado) contendo as características dos objetos extraídos das cenas naturais.

```

<?xml version="1.0" encoding="UTF-8" ?>
<objetos_virtuais>
  <objeto id="0" tipo="laranjeira">
    <centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">
      <posicao x="0" y="0" z="0" />
    </centro>
    <segmentos>
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="0.5" tamY="1.0" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" modelo="seg_dir.wrl" tamX="0.5" tamY="1.0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
</objetos_virtuais>

```

Figura 5.13: Modelo de dados virtuais contendo as informações sobre os objetos 3D previamente modelados.

5.2.7 Modelo de Dados do Ambiente Virtual

Nesta fase, o modelo de dados reais e o modelo de dados virtuais foram associados para geração do modelo de dados do ambiente virtual, a medida que o modelo de dados reais era atualizado e disponibilizado na Internet. Para descobrir se o modelo foi atualizado, o módulo mundo virtual realizou constantes consultas ao modelo de dados reais, através de solicitações *http*. O modelo resultante pode ser visto através do documento XML apresentado na Figura 5.14.


```

<?xml version="1.0" encoding="UTF-8" ?>
<ambiente_virtual>
  <objeto id="0" tipo="laranjeira">
    <centro modelo="seg_central.wrl" tamX="0.5" tamY="1.0" tamZ="0.1">
      <posicao x="0" y="0" z="0" />
    </centro>
    <segmentos qtde="67" espacamento="0.1">
      <hemisferio_esquerdo>
        <segmento_esquerdo id="0" modelo="seg_esq.wrl" tamX="74" tamY="37" tamZ="0.1" />
        <segmento_esquerdo id="1" modelo="seg_esq.wrl" tamX="110" tamY="55" tamZ="0.1" />
        <segmento_esquerdo id="2" modelo="seg_esq.wrl" tamX="110" tamY="56" tamZ="0.1" />
        <segmento_esquerdo id="3" modelo="seg_esq.wrl" tamX="113" tamY="56.5" tamZ="0.1" />
      </hemisferio_esquerdo>
      <hemisferio_direito>
        <segmento_direito id="0" modelo="seg_esq.wrl" tamX="0" tamY="0" tamZ="0.1" />
        <segmento_direito id="1" modelo="seg_esq.wrl" tamX="0" tamY="0" tamZ="0.1" />
        <segmento_direito id="2" modelo="seg_esq.wrl" tamX="1.9" tamY="56" tamZ="0.1" />
        <segmento_direito id="3" modelo="seg_esq.wrl" tamX="0" tamY="0" tamZ="0.1" />
      </hemisferio_direito>
    </segmentos>
  </objeto>
  <objeto id="1" tipo="laranjeira">
    ...
  </objeto>
  <objeto id="2" tipo="laranjeira">
    ...
  </objeto>
  <objeto id="3" tipo="laranjeira">
    ...
  </objeto>
</ambiente_virtual>

```

Figura 5.14: Modelo de dados do ambiente virtual contendo as informações sobre os modelos de dados reais e virtuais.

5.2.8 Montagem da cena virtual

Os módulos mundo real e mundo virtual foram executados simultaneamente em máquinas separadas, comunicando-se apenas pela Internet. Durante a execução do módulo mundo virtual, foram realizadas consultas constantes ao modelo de dados reais disponibilizado na Internet, para verificar se um novo objeto havia sido detectado e suas características extraídas e inseridas no modelo.

Utilizando os parâmetros de tamanho e posição do modelo de dados, os modelos VRML lidos e carregados em memória foram inseridos no grafo de cena, tiveram suas escalas ajustadas e suas posições transladadas de acordo com os valores obtidos do documento XML. A Figura 5.15 apresenta os resultados obtidos com a montagem do ambiente virtual a partir das imagens apresentadas na Figura 5.11.

Figura 5.15: Montagem do ambiente virtual a partir das imagens da cena natural em tempo quase real.

Através do parâmetro identificador de cada objeto foi possível verificar se o mesmo já havia sido utilizado na montagem do AV ou se era um novo objeto. Para cada novo objeto inserido no modelo de dados reais, o módulo mundo virtual associou suas informações com o modelo de dados virtuais para atualização do modelo de dados do ambiente virtual. A cada atualização do modelo de dados do AV, os modelos referentes a esta atualização foram inseridos no AV. Dessa forma, o ambiente virtual pôde ser montado e visualizado em tempo quase real a medida em que os objetos eram segmentados nas imagens das cenas naturais. Essa característica pode ser explorada em aplicações de telepresença.

5.2.9 Visualização 3D Imersiva

A visualização das cenas virtuais foi disponibilizada durante toda a montagem do AV. O observador pôde assistir a montagem do ambiente virtual a partir das cenas naturais em tempo quase real, a medida que as imagens eram segmentadas remotamente e as informações pertinentes disponibilizadas na Internet. A estrutura de visualização utilizada foi o sistema de multiprojeção imersivo do LMVI, como pode ser visto na Figura 5.16.

Figura 5.16: Visualização do AV montado no sistema de multiprojeção imersivo do LMVI.

De posse dos dispositivos de interação disponíveis, foi possível transladar e rotacionar o ambiente virtual para observação a partir de diferentes pontos de vista, como pode ser visto

através da Figura 5.17.

Figura 5.17: Diferentes pontos de vista da cena virtual montada.

5.2.10 Salvamento do Ambiente Virtual

As informações referente o ambiente virtual montado e visualizado no sistema de multiprojeção imersivo do LMVI, foram salvas e armazenadas em algum repositório para uso posterior. Para salvar a cena virtual, bastou salvar documento XML que descreve o modelo de dados do ambiente virtual. Através desta abordagem, obteve-se uma modularidade do ambiente virtual montado, de forma que os modelos individuais 3D que compõe o AV podem ser alterados ou substituídos sem afetarem o processo de montagem.

Para fins de testes, o ambiente virtual gravado foi recarregado no dispositivo de visualização e o resultado foi exatamente o mesmo ambiente que havia sido visualizado anteriormente. Isto comprovou a independência do módulo mundo virtual dos demais módulos em termos de execução.

Com a cena salva em XML o volume da dados referente a cena foi drasticamente reduzido em comparação com uma cena salva em VRML, otimizando tempo e recursos durante o recarregamento do AV no dispositivo de projeção.

5.2.11 Considerações

Como descrito na Seção 5.1, as imagens das cenas naturais de laranjeiras foram adquiridas com o foco da câmera configurado no automático. Neste tipo de configuração, observou-se que em determinados quadros da imagem, objetos que estão no fundo se apresentam com foco semelhante a objetos que estão em primeiro plano. Nestes casos, o algoritmo de detecção de bordas Canny identifica os objetos de fundo como constituintes do primeiro plano.

Uma maneira de evitar a identificação dos objetos que estão no fundo da imagem é reduzindo a profundidade de campo durante a captura das imagens, como descrito na Seção 4.2.1.1. Quanto menor a profundidade de campo, menor é o foco à frente e no fundo da imagem, com relação ao objeto que se encontra no plano de foco. Para fins de testes desta afirmação, foi aplicado o detector de bordas Canny em um mesmo quadro de imagem sem profundidade de campo definida e com profundidade de campo simulada manualmente, com o tratamento da imagem. O resultado pode ser visto na Figura 5.18.

Figura 5.18: Detecção de bordas em uma imagem com (em cima) e sem profundidade de campo (em baixo).

Observe que na imagem da parte de cima da Figura 5.18, o objeto que se encontra no fundo foi detectado como parte do primeiro plano da imagem. Na imagem da parte de baixo, o mesmo quadro de imagem com profundidade de campo simulada, este objeto foi ignorado pelo detector de bordas Canny. Logo, como definido na metodologia proposta, isso é um

importante dado a ser considerado durante a aquisição das imagens.

5.3 Orientação Paralela à Linha de Plantio

A metodologia proposta define que para a extração de objetos de interesse das cenas naturais, a orientação da câmera em relação às linhas de plantio seja ortogonal, como descrito na Seção 4.2.1.1. No entanto, para embasar a escolha desse ângulo de orientação da câmera, foram analisadas imagens em diferentes ângulos de aquisição.

Neste segundo caso de teste, a orientação da câmera foi paralela à linha de plantio dos objetos de interesse e paralela à superfície do solo. A altura média da câmera em relação ao solo foi de 1,70 m e a distância em relação as laranjeiras foi de 1,35 m.

5.3.1 Pré-processamento

Conforme definido na metodologia proposta, as imagens adquiridas foram transformadas do espaço de cores RGB para o espaço de níveis de cinza. Além disso, foi aplicado o filtro Gaussiano para correção de ruídos, através da suavização das imagens adquiridas. As imagens originais e as imagens resultantes podem ser vistas na Figura 5.19.

Figura 5.19: Imagens adquiridas nos espaços de cores RGB (esquerda) e pré-processadas no espaço de níveis de cinza com filtro Gaussiano(direita).

Antes da transformação das imagens do espaço de cores RGB para o espaço de níveis

de cinza, as imagens originais foram preservadas e apenas cópias dessas imagens sofreram essas transformações para serem submetidas à etapa de segmentação.

5.3.2 Segmentação

A partir das imagens pré-processadas, foi aplicado o detector de bordas Canny para localizar regiões onde há mudança brusca nos níveis de cinza da imagem. Ou seja, o detector procura destacar arestas na imagem, analisando descontinuidades nos níveis de cinza e atenuando ruídos.

As imagens resultantes do algoritmo Canny são imagens binárias contendo apenas bordas brancas em um fundo preto, como pode ser visto na Figura 5.20.

Figura 5.20: Imagens no espaço de níveis de cinza (esquerda) e imagens binárias resultantes do detector de bordas Canny (direita).

Observe que regiões com maior tonalidade (sombra) são agrupadas separadas de regiões de maior incidência de luz. Além disso, os objetos que apresentam maior nitidez em relação aos seus vizinhos, também têm suas bordas destacadas, como é o caso de parte da grama que está mais próxima da câmera.

Outro ponto importante a ser observado, é que neste ângulo de orientação da câmera, não é possível distinguir objetos individualmente, pois os dosséis se sobrepõem gerando um povoamento de bordas por toda a extensão das linhas de plantio.

Como pode ser visto na Figura 5.20, as imagens resultantes do algoritmo Canny apresentam múltiplas bordas desconexas. Com o intuito de encontrar apenas bordas completas, foram aplicadas as transformações morfológicas de fechamento e abertura, nessa ordem.

Como descrito no Capítulo 4, as transformações morfológicas de fechamento e abertura utilizam operadores morfológicos de erosão e dilatação. A operação de erosão é responsável pela redução gradual dos limites das regiões de *pixels* em primeiro plano, enquanto a dilatação é responsável pelo aumento gradual desses *pixels*. A combinação destas duas operações permite a ligação de bordas desconexas na imagem, como pode ser visto na Figura 5.21.

Figura 5.21: Imagens resultantes do detector de bordas Canny (esquerda) e imagens resultantes de transformações morfológicas de fechamento e abertura (direita).

Os resultados das transformações morfológicas demonstrados na Figura 5.21, apresentam tantos contornos externos quanto internos dos elementos identificados pelo detector de bordas. No entanto, para extração da linha de plantio como um único objeto, foi aplicado o algoritmo de Suzuki (SUZUKI; BE, 1985) para extração dos contornos externos das imagens. O resultado desse algoritmo são vetores de contornos contendo apenas os pontos (x, y) de cada *pixel* dos contornos externos.

Nesta etapa, a imagem original que havia sido preservada durante o pré-processamento foi utilizada para desenhar os contornos externos a partir dos pontos fornecidos pelos vetores resultantes do algoritmo de Suzuki.

Os resultados do traçado de contornos externos nas imagens no espaço de cores RGB podem ser visto na Figura 5.22.

Figura 5.22: Imagens binárias resultantes de transformações morfológicas de fechamento e abertura (esquerda) e imagens no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.

O traçado dos contornos externos foram realizados nas imagens originais adquiridas (no espaço de cores RGB) e preservadas durante o pré-processamento, para fins de comparação direta com o objeto segmentado. Caso o traçado fosse realizado nas imagens binárias resultantes das transformações morfológicas de fechamento e abertura, não seria possível visualizar os objetos reais capturados e os traçados externos encontrados, em uma mesma imagem.

5.3.3 Seleção do Objeto de Interesse

A Figura 5.22, apresenta imagens com mais de um contorno em um mesmo quadro. Seguindo a proposta da metodologia, nesta etapa, somente o maior contorno de cada quadro foi preservado, enquanto os demais foram desconsiderados, como pode ser visto na Figura 5.23.

Os resultados encontrados indicam que a mesma metodologia adotada para extração de objetos individuais, pode ser utilizada em trabalhos futuros para navegação do robô agrícola através das linhas de plantio. No entanto, para a montagem do ambiente virtual, a orientação paralela às linhas de plantio não se mostrou adequada. Pois com a metodologia proposta, às linhas de plantio seriam interpretadas pelo sistema como um único objeto a ser montado na cena virtual, ou seja, como se fosse uma única laranja.

Figura 5.23: Imagens com traçados de todos os contornos externos (esquerda) e imagens contendo apenas o maior contorno em cada quadro (direita).

5.4 Orientação Oblíqua à Linha de Plantio

Como mencionado na seção anterior, a metodologia proposta define que para a extração de objetos de interesse das cenas naturais, a orientação da câmera em relação às linhas de plantio seja ortogonal, como descrito na Seção 4.2.1.1. No entanto, para embasar a escolha desse ângulo de orientação da câmera, foram analisadas imagens em diferentes ângulos de aquisição.

Neste terceiro caso de teste, a orientação da câmera foi oblíqua à linha de plantio dos objetos de interesse e paralela à superfície do solo. A altura média da câmera em relação ao solo foi de 1,70 m e a distância em relação as laranjeiras foi de 1,35 m.

5.4.1 Pré-processamento

Conforme definido na metodologia proposta, as imagens adquiridas foram transformadas do espaço de cores RGB para o espaço de níveis de cinza. Além disso, foi aplicado o filtro Gaussiano para correção de ruídos, através da suavização das imagens adquiridas. A imagem original e a imagem resultante podem ser vistas na Figura 5.24.

Antes da transformação das imagens do espaço de cores RGB para o espaço de níveis de cinza, as imagens originais foram preservadas e apenas cópias dessas imagens sofreram

essas transformações para serem submetidas à etapa de segmentação.

Figura 5.24: Imagem adquirida nos espaços de cores RGB (em cima) e pré-processada no espaço de níveis de cinza com filtro Gaussiano(em baixo).

5.4.2 Segmentação

A partir das imagens pré-processadas, foi aplicado o detector de bordas Canny para localizar regiões onde há mudança brusca nos níveis de cinza da imagem. Ou seja, o detector procura destacar arestas na imagem, analisando descontinuidades nos níveis de cinza e atenuando ruídos.

As imagens resultantes do algoritmo Canny são imagens binárias contendo apenas bordas brancas em um fundo preto, como pode ser visto na Figura 5.25.

Figura 5.25: Imagem no espaço de níveis de cinza (em cima) e imagem binária resultantes do detector de bordas Canny (em baixo).

Assim como nos testes anteriores, as regiões com maior tonalidade (sombra) foram agrupadas separadas de regiões de maior incidência de luz. E, os objetos que apresentam maior nitidez em relação aos seus vizinhos, também tiveram suas bordas destacadas, como é o caso da grama que está mais próxima da câmera.

Da mesma forma que apresentado no teste com a câmera na orientação paralela, neste ângulo de orientação da câmera, não é possível distinguir objetos individualmente, pois os dosséis se sobrepõem gerando um povoamento de bordas por toda a extensão das linhas de plantio.

Como pode ser visto na Figura 5.25, as imagens resultantes do algoritmo Canny apresentam múltiplas bordas desconexas. Com o intuito de encontrar apenas bordas completas,

foram aplicadas as transformações morfológicas de fechamento e abertura, nessa ordem.

Como descrito no Capítulo 4, as transformações morfológicas de fechamento e abertura utilizam operadores morfológicos de erosão e dilatação. A operação de erosão é responsável pela redução gradual dos limites das regiões de *pixels* em primeiro plano, enquanto a dilatação é responsável pelo aumento gradual desses *pixels*. A combinação destas duas operações permite a ligação de bordas desconexas na imagem, como pode ser visto na Figura 5.26.

Figura 5.26: Imagem resultante do detector de bordas Canny (em cima) e imagem resultante de transformações morfológicas de fechamento e abertura (em baixo).

Os resultados das transformações morfológicas demonstrados na Figura 5.26, apresentam tantos contornos externos quanto internos dos elementos identificados pelo detector de bordas. No entanto, para extração da linha de plantio como um único objeto, foi aplicado

o algoritmo de Suzuki (SUZUKI; BE, 1985) para extração dos contornos externos das imagens. O resultado desse algoritmo são vetores de contornos contendo apenas os pontos (x, y) de cada *pixel* dos contornos externos.

Nesta etapa, a imagem original que havia sido preservada durante o pré-processamento foi utilizada para desenhar os contornos externos a partir dos pontos fornecidos pelos vetores resultantes do algoritmo de Suzuki.

Os resultados do traçado de contornos externos nas imagens no espaço de cores RGB podem ser visto na Figura 5.27.

Figura 5.27: Imagem binária resultantes de transformações morfológicas de fechamento e abertura (em cima) e imagem no espaço de cores RGB com traçados de contornos externos obtidos pelo algoritmo de Suzuki.

Assim como nos testes anteriores, o traçado dos contornos externos foram realizados

nas imagens originais adquiridas (no espaço de cores RGB) e preservadas durante o pré-processamento, para fins de comparação direta com o objeto segmentado. Caso o traçado fosse realizado nas imagens binárias resultantes das transformações morfológicas de fechamento e abertura, não seria possível visualizar os objetos reais capturados e os traçados externos encontrados, em uma mesma imagem.

5.4.3 Seleção do Objeto de Interesse

A Figura 5.27 apresenta imagens com mais de um contorno em um mesmo quadro. Seguindo a proposta da metodologia, nesta etapa, somente o maior contorno de cada quadro foi preservado, enquanto os demais foram desconsiderados, como pode ser visto na Figura 5.28.

Figura 5.28: Imagem com traçados de todos os contornos externos (em cima) e imagem contendo apenas o maior contorno em cada quadro (embaixo).

Os resultados encontrados são similares aos resultados obtidos com o ângulo de

orientação paralelo às linhas de plantio. A mesma metodologia adotada para extração de objetos individuais, pode ser utilizada em trabalhos futuros para navegação do robô agrícola através das linhas de plantio. No entanto, para a montagem do ambiente virtual, a orientação oblíqua, assim como a orientação paralela às linhas de plantio, não se mostrou adequada. Pois com a metodologia proposta, às linhas de plantio seriam interpretadas pelo sistema como um único objeto a ser montado na cena virtual, ou seja, como se fosse uma única laranjeira.

5.5 Considerações Finais do Capítulo

Neste capítulo foram apresentados os testes de aplicação da metodologia proposta, bem como os resultados obtidos. Inicialmente, foi apresentada a etapa de aquisição das imagens, e em seguida, foram apresentados os resultados de testes com três ângulos de orientação da câmera em relação aos objetos de interesse nas cenas naturais: orientação ortogonal, paralela e oblíqua.

Os testes realizados com a orientação da câmera ortogonal às linhas de plantio mostraram que este é o melhor ângulo para extração de objetos individuais das cenas naturais. Pois através desse ângulo foi possível distinguir bem os objetos capturados pelas imagens e obter seus respectivos contornos para posterior montagem do ambiente virtual.

O fato da metodologia proposta apresentar uma arquitetura modular, permitiu a execução e manutenção dos módulos que a constituem de maneira independente uns dos outros. Ou seja, o módulo mundo real pode realizar o reconhecimento das imagens e disponibilizar suas informações sem que a montagem do mundo virtual esteja acontecendo em paralelo. Ou ainda, o módulo mundo virtual pode carregar um ambiente virtual pré-gravado independente de haver ou não dados disponíveis pelo módulo mundo real. E por fim, o ambos os módulos podem ser executados em paralelo, disponibilizando e requisitando dados enquanto o reconhecimento das cenas e a montagem do ambiente virtual ocorre em tempo quase real.

A restrição encontrada, ocorreu quando os objetos estão muito próximos uns dos outros. O contorno de um objeto acaba se fundindo com o contorno de seu vizinho, dando uma falsa interpretação de que múltiplos objetos constituem um único elemento. Todavia, trabalhos futuros podem tratar este tipo de situação aprimorando apenas a etapa de extração de características da metodologia proposta sem a necessidade de alterar os demais módulos.

Os testes realizados com as orientações da câmera paralela e oblíqua às linhas de plantio indicaram que a metodologia proposta não seria suficiente para extração destas linhas

e montagem no ambiente virtual, uma vez que a sobreposição de objetos faz com que o algoritmo de detecção de bordas reconheça toda a linha de plantio como um único objeto de interesse. No entanto, como mencionado anteriormente, trabalhos futuros podem aprimorar o módulo visão computacional presente na metodologia proposta sem comprometer os demais módulos que a constituem.

Os testes de comunicação entre os módulos, demonstraram que a montagem automática de ambientes virtuais a partir de cenas naturais pode ocorrer enquanto o reconhecimento das cenas estiver acontecendo em qualquer parte do mundo, desde que os dados sejam disponibilizados pela Internet.

A utilização dos modelos de dados através de documentos XML, permitiu uma troca rápida de informações entre os módulos, uma vez que o documento XML por ser um arquivo texto é muito menor e mais leve do que as imagens de vídeo em alta definição como as que foram adquiridas para a realização dos testes de aplicação.

Por fim, o desenvolvimento de um método de montagem de ambientes virtuais agrícolas a partir de dados não estruturados complexos como as cenas naturais adquiridas através de imagens 2D, possibilitou a geração de um modelo de dados 3D obtido a partir das imagens de um ambiente real oferecendo mais opções de análise, comparação, navegação, interação e alteração, do que apenas o uso de imagens não processadas.

6 Considerações Finais

Este trabalho apresentou uma metodologia de montagem e utilização de ambientes virtuais agrícolas em um sistema de multiprojeção imersivo a partir de cenas naturais. O desenvolvimento desta metodologia possibilitou uma melhor interpretação dos dados não estruturados complexos de cenas ou paisagens reais, uma vez que viabilizou-se a transição de dados de cenas 2D para 3D.

A tradução de um modelo 2D para um modelo 3D permitiu a interação com os dados a partir de diferentes perspectivas, mais próximas da realidade física do espaço analisado. Os pontos chave desta metodologia concentraram-se na aquisição, exibição e processamento de dados em duas e três dimensões.

A modelagem automática dos ambientes virtuais agrícolas através do uso de modelos 3D previamente concebidos, permitiu que pesquisadores de diferentes áreas do conhecimento, possam utilizar a metodologia proposta sem a necessidade de modelarem individualmente modelos geométricos tridimensionais complexos para construção de ambientes virtuais.

O uso de imagens a partir de uma câmera digital se locomovendo pelo cenário agrícola seja por meio de um robô agrícola móvel ou algum outro veículo que a transporte, mostrou-se como uma alternativa de baixo custo se comparada a outras tecnologias como captação de imagens por aviões ou satélites, radares e *scanners* a laser.

Diferente de outras metodologias encontradas na literatura e descritas no capítulo de revisão bibliográfica, a metodologia proposta não se concentrou na reconstrução de modelos 3D precisos (porém com alto custo computacional), mas sim na montagem de ambientes virtuais a partir de modelos 3D simples (com poucos polígonos) e rápidos de serem lidos, carregados e *renderizados*. No entanto, apesar da simplicidade dos modelos 3D, a metodologia de montagem permitiu a obtenção de representações 3D muito próximas dos objetos reais correspondentes.

Esta pesquisa ainda contribuiu com a formalização todo o processo de montagem dos ambientes virtuais agrícolas, desde a forma de aquisição das imagens até a apresentação e

utilização do usuário em sistema do tipo CAVE, através de uma arquitetura modular, vindo de encontro a escassez de métodos dessa natureza disponíveis na literatura.

A partir de um modelo de dados 3D projetado em um sistema que oferece um alto grau de imersão e interação como o LMVI, permitiu-se comparações com outros modelos de dados ou com o mesmo modelo salvo em épocas diferentes. Através da comparação entre modelos possibilitou-se identificar alterações que podem ocorrer no ambiente real ao longo do tempo, tanto naturais como causadas pelo homem, auxiliando na tomada de decisão em processos agrícolas. Além disso, permitiu-se validar o modelos de dado obtido em relação ao ambiente real que ele representa.

O estudo e desenvolvimento de mecanismos de utilização e interação dos ambientes virtuais no sistema de multiprojeção imersivo do LMVI, integrando diferentes tecnologias e ferramentas computacionais, permitiu não apenas a montagem e utilização dos ambiente virtuais agrícolas, mas a disponibilização de uma ferramenta de visualização 3D imersiva para diferentes pesquisadores com interesse no uso desta tecnologia.

A utilização dos modelos de dados através de documentos XML, permitiu uma troca rápida de informações entre os módulos, uma vez que o documento XML por ser um arquivo texto é muito menor e mais leve do que as imagens de vídeo em alta definição como as que foram adquiridas para a realização dos testes de aplicação.

O uso da Internet como meio de comunicação entre os módulos, permitiu a montagem automática de ambientes virtuais a partir de cenas naturais adquiridas e processadas em qualquer parte do mundo, desde que o modelo de dados reais esteja disponibilizado na rede mundial de computadores.

Por fim, o fato da metodologia proposta apresentar uma arquitetura modular, permitiu a execução e manutenção dos módulos que a constituem de maneira independente uns dos outros. O módulo mundo real pode realizar o reconhecimento das imagens e disponibilizar suas informações sem que a montagem do mundo virtual esteja acontecendo em paralelo. Ou ainda, o módulo mundo virtual pode carregar um ambiente virtual pré-gravado independente de haver ou não dados disponíveis pelo módulo mundo real. E, ambos os módulos podem ser executados em paralelo, disponibilizando e requisitando dados enquanto o reconhecimento das cenas e a montagem do ambiente virtual ocorre em tempo quase real.

6.1 Trabalhos Futuros

A metodologia proposta mostrou-se adequada para a montagem e utilização de ambientes virtuais agrícolas em um sistema de multiprojeção imersivo a partir de cenas naturais. No entanto, trabalhos futuros podem estendê-la para a extração de características de detalhes internos dos objetos de interesse, como frutos por exemplo.

Ainda com relação a etapa de visão computacional, outras técnicas de segmentação e extração de características podem ser incorporadas para a extração de objetos cujas linhas de plantio são densas (objetos muito próximos de seus vizinhos).

Por fim, como perspectiva de trabalho futuro, pode-se aprimorar a extração de características de objetos cuja simetria é previamente conhecida, como veículos, equipamentos agrícolas, pessoas, animais, entre outros.

Referências

- ADAMCHUK, V. I. *Untangling the GPS Data String*. Lincoln, NE, 2001.
- ADAMS, L. *Visualização e Realidade Virtual*. São Paulo, SP: Makron Books, 1994.
- AHLBERG, S. et al. On modeling and visualization of high resolution virtual environments using lidar data. *Proceedings of the 12th Int. Conf. on Geoinformatics*, p. 299–306, 2004.
- Simplifying the Reconstruction of 3D Models using Parameter Elimination*. 1–8 p.
- ARAUJO, R. B. *Especificação e Análise de um Sistema Distribuído de Realidade Virtual*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 1996.
- AZUMA, R. Tracking requirements for augmented reality. *Communications of the ACM*, v. 36, n. 7, p. 50–51, 1993.
- BACKES, M.; PLUMER, L. On the adequacy of gis-generated weed maps for precision farming. *ScanGIS - Scandinavian Research Conference on Geographical Information Science*, 2003.
- BEGAULT, D. R. *3-D Sound for Virtual Reality and Multimedia*. San Diego, CA, USA: Academic Press Professional, Inc., 1994.
- BEIER, K.-P. A virtual reality check for manufacturers - a short introduction to virtual reality. *ASTD - American Society for Training and Development*, v. 9, n. 1, p. 14, February 1998.
- BEIER, K.-P. Web-based virtual reality in design and manufacturing applications. *Proceedings of COMPIT'2000 - 1st International EuroConference on Computer Applications and Information Technology in the Maritime Industry*, Potsdam, Germany, p. 45–55, March 2000.
- BENNEDSEN, B. S.; PETERSON, D. L.; TABB, A. Identifying defects in images of rotating apples. *Computers and Electronics in Agriculture*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 48, n. 2, p. 92–102, ago. 2005.
- BENTON, S. A. *Selected Papers on Three-Dimensional Displays*. Bellingham, WA - USA: SPIE Optical Engineering Press, 2001.
- BIERBAUM, A. et al. Vr juggler: A virtual platform for virtual reality application development. *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*, p. 89, 2001.
- BILLINGHURST, M.; KATO, H.; POUPYREV, I. The magicbook - moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, v. 21, n. 3, p. 6–8, 2001.
- BILLINGSLEY, J. Automatic guidance of agricultural mobiles at the ncea. *Industrial Robot: An International Journal*, MCB UP Ltd, v. 27, p. 449 – 457, 2000.

- BINNEY, J.; SUKHATME, G. 3d tree reconstruction from laser range data. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 1321–1326.
- BOMAN, D. K. International survey: Virtual-environment research. *IEEE Computer*, v. 28, n. 6, p. 57–65, 1995.
- BOUGUILA, L.; ISHII, M.; SATO, M. Realizing a new step-in-place locomotion interface for virtual environment with large display system. *EGVE '02: Proceedings of the Workshop on Virtual environments 2002*, p. 197–207, 2002.
- BRASE, T. *Precision Agriculture*. 1. ed. [S.l.]: Delmar Cengage Learning, 2005. 256 p.
- BROWN, K.; PETERSEN, D. *Ready-to-Run Java 3D*. New York, NY: John Wiley & Sons, 1999.
- BURDEA, G. C. *Force and Touch Feedback for Virtual Reality*. New York, NY, USA: John Wiley & Sons, Inc., 1996.
- CAI, Y.; ISHII, M.; SATO, M. A human interface device for cave: Size virtual workspace. *IEEE International Conference on Systems, Man, and Cybernetics*, v. 3, p. 2084–2089, 1996.
- CANNY, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8, n. 6, p. 679–698, nov. 1986. ISSN 0162-8828.
- CAPIN, T. K. et al. *Avatars in Networked Virtual Environments*. New York, NY, USA: John Wiley & Sons, 1999.
- CAREY, R. The virtual reality modeling language explained. *IEEE MultiMedia*, v. 5, n. 3, p. 84–93, 1998.
- CENTENO, J. A.; MITISHITA, E. A. Laser scanner aerotransportado no estudo de áreas urbanas: A experiencia da ufpr. *Simposio Brasileiro de Sensoriamento Remoto*, p. 3645–3562, 2007.
- CERDAT, F.; DESCOMBES, X.; ZERUBIA, J. Urban scene rendering using object description. In: *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*. [S.l.: s.n.], 2003. v. 1, p. 62 – 64 vol.1.
- CHAPIN, W.; LACEY, T.; LEIFER, L. Designspace: A manual interaction environment for computer aided design. *Proceedings of CHI '94*, p. 24–28, 1994.
- CHEN, S. E. Quicktime vr: An image-based approach to virtual environment navigation. *Siggraph '95: Proceedings of The 22nd Annual Conference on Computer Graphics and Interactive Techniques*, p. 29–38, 1995.
- CHENG, L. et al. Dynamic triangle based method for 3d building rooftop reconstruction from lidar data. In: *Geoinformatics, 2011 19th International Conference on*. [S.l.: s.n.], 2011. p. 1–4.
- CHURCHILL, E. F.; SNOWDON, D. Collaborative virtual environments: An introductory review of issues and systems. *Virtual Reality: Research, Development and Applications*, v. 3, n. 1, p. 3–15, 1998.

- CLAY, D. E.; SHANAHAN, J. F. *GIS Applications in Agriculture, Volume Two: Nutrient Management for Energy Efficiency*. [S.l.]: CRC Press - Taylor and Francis, 2011. 472 p.
- CONCI, A.; AZEVEDO, E.; LETA, F. *Computação Gráfica - Volume 2 - Teoria e Prática*. Rio de Janeiro, RJ: Campus, 2008.
- CORSEUIL, E. T. L. et al. Buscando o uso operacional de realidade virtual em grandes modelos de engenharia. *VI Symposium on Virtual Reality - SVR*, p. 187–198, 2003.
- CREAGH, H. Cave automatic virtual environment. *Proceedings of Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference*, p. 499–504, 2003.
- CRUZ-NEIRA, C. *Virtual Reality Based on Multiple Projection Screens: The Cave and its Applications to Computational Science and Engineering*. Tese (Doutorado) — University of Illinois, Chicago, IL, USA, 1995.
- CRUZ-NEIRA, C.; SANDIN, D. J.; DEFANTI, T. A. Surround-screen projection-based virtual reality: The design and implementation of the cave. *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, p. 135–142, 1993.
- CRUZ-NEIRA, C. et al. The cave: Audio visual experience automatic virtual environment. *Communications of the ACM*, v. 35, n. 6, p. 64–72, 1992.
- CYBERGLOVE, S. 3d interaction technologies. *Disponível em: <http://www.cyberglovesystems.com/all-products/>. Acesso em: 6 jun. 2012*, 2012.
- CZERNUSZENKO, M. et al. The immersadesk and infinity wall projection-based virtual reality displays. *ACM SIGGRAPH Computer Graphics*, v. 31, n. 2, p. 46–49, 1997.
- DALY, L.; BRUTZMAN, D. X3d: Extensible 3d graphics standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, v. 24, n. 6, p. 130–135, 2007.
- DAVIES, E. R. et al. Design of efficient line segment detectors for cereal grain inspection. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, USA, v. 24, n. 1-3, p. 413–428, jan. 2003.
- DODGSON, N. A. Autostereoscopic 3d displays. *Computer (IEEE)*, v. 38, n. 8, p. 31–36, 2005.
- DOERR, K.-U. et al. Evaluation of a low-cost 3d sound system for immersive virtual reality training systems. *IEEE Transactions on Visualization and Computer Graphics*, v. 13, n. 2, p. 204–212, 2007.
- DRAPER, J. V.; KABER, D. B.; USHER, J. M. Speculations on the value of telepresence. *CyberPsychology & Behavior*, v. 2, n. 4, p. 349–362, 1999.
- DUNN, M. T. *Application of Vision Sensing in Agriculture*. Tese (Doutorado) — The University of Southern Queensland, Australia, 2007.
- EARNSHAW, R. A.; VINCE, J. A.; JONES, H. *Virtual Reality Applications*. London, UK, UK: Academic Press Ltd., 1995.

- ELLIS, S. R. What are virtual environments? *IEEE Computer Graphics and Applications*, v. 14, n. 1, p. 17–22, 1994.
- ESTEBAN, I.; DIJK, J.; GROEN, F. Automatic 3d modeling of the urban landscape. In: *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*. [S.l.: s.n.], 2010. p. 421–428.
- FARIA, R. R. A. *Auralização em Ambientes Audiovisuais Imersivos*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, São Paulo, SP, 2005.
- FELEZ, J.; VERA, C.; MARTINEZ, M. L. Virtual reality applied to traffic accident analysis. *Computer Networks*, v. 30, n. 20, p. 1907–1914, 1999.
- FENG-QUAN, J.; GUI-GANG, S. Simulating the landscape plants by the fractal algorithm. In: *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*. [S.l.: s.n.], 2009. v. 2, p. 197–200.
- FERNANDES, K. J.; RAJA, V.; EYRE, J. Cybersphere: The fully immersive spherical projection system. *Communications of the ACM*, v. 46, n. 9, p. 141–146, 2003.
- FIGUEIREDO, G. F. et al. Advancement in whole house machine vision based poultry behavior analysis. *ASAE Annual International Meeting*, St. Joseph, MI, 2004.
- FISHER, S. S. Virtual environments, personal simulation & telepresence. *Virtual Reality: Theory, Practice and Promise*, p. 101–110, 1991.
- FISHER, S. S.; TAZELAAR, J. M. Living in a virtual world. *Byte*, v. 15, n. 7, p. 215–221, 1990.
- FROST, A. R. et al. Localization of spherical fruits for robotic harvesting. *Computers and electronics in agriculture*, v. 44, p. 93–106, 2004.
- FU, K.; MUI, J. A survey on image segmentation. *Pattern Recognition*, v. 13, n. 1, p. 3–16, 1981.
- GIBSON, W. *Neuromancer*. New York, USA: ACE Books, 1984.
- GIL, A. C. *Como Elaborar Projetos de Pesquisa*. 5. ed. São Paulo, SP: Atlas, 2011.
- GILLET, A. et al. Augmented reality with tangible auto-fabricated models for molecular biology applications. *Proceedings of The 15th IEEE Visualization 2004 (VIS'04)*, p. 235–242, 2004.
- GIONGO, M. et al. Lidar: princípios e aplicações florestais. *Pesquisa Florestal Brasileira*, v. 30, n. 63, p. 231–244, 2007.
- GOEL, P. et al. Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn. *Computers and Electronics in Agriculture*, v. 39, n. 2, p. 67–93, 2003.
- GOLDBERG, K. *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*. Cambridge, MA, USA: The MIT Press, 2000.

- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. [S.l.]: Prentice Hall, 2002.
- GRADECKI, J. D. *The Virtual Reality Construction Kit*. New York, NY - USA: John Wiley & Sons Inc, 1995.
- GREENAND, R. P. W. D. J. P. C. P. S. D. M.; WHITTEMORE, C. T. Monitoring pig growth using a prototype imaging system. *British Society of Animal Science*, 2003.
- GROSS, M. et al. blue-c: A spatially immersive display and 3d video portal for telepresence. *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, San Diego, CA, USA, p. 819–827, 2003.
- GUYER, D.; YANG, X. Use of genetic artificial neural networks and spectral imaging for defect detection on cherries. *Computers and Electronics in Agriculture*, v. 29, n. 3, p. 179 – 194, 2000.
- HAGUE, T.; MARCHANT, J. A.; TILLET, N. D. Ground based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture*, v. 25, n. 1-2, p. 11–28, 2000.
- HANÁK, D. et al. Building a panoramic recording and presentation system for telepresence. *(VRIC'2007) 9th Virtual Reality International Conference*, Laval, France, p. 185–190, April 2007.
- HAOPENG, W. et al. Virtual reality simulation of plants based on remote sensing information. In: *Computational and Information Sciences (ICCIS), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 936 –939.
- HARRISON, S.; DOURISH, P. Re-place-ing space: The roles of place and space in collaborative systems. *CSCW '96: Proceedings of the 1996 ACM conference on Computer Supported Cooperative Work*, New York, NY, USA, p. 67–76, 1996.
- HASANKHANI, R.; NAVID, H. Potato sorting based on size and color in machine vision system. *Journal of Agricultural Science*, v. 4, n. 5, 2012.
- HELD, R. M.; DURLACH, N. I. Telepresence. *Presence: Teleoperators and Virtual Environments*, v. 1, n. 1, p. 109–112, 1992.
- HOFFMANN, H.; STEFANI, O.; PATEL, H. Extending the desktop workplace by a portable virtual reality system. *International Journal of Human-Computer Studies*, v. 64, n. 3, p. 170–181, 2006.
- HONG-WEI, L. et al. The reconstruction of three-dimensional tree models from terrestrial lidar. In: *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. v. 4, p. 371 –374.
- HUANG, H.; MAYER, H. Generative statistical 3d reconstruction of unfoliated trees from terrestrial images. *Annals of GIS*, v. 15, n. 2, p. 97–105, 2009.
- HUANG, T. et al. A virtual prototyping system for simulating construction processes. *Automation in Construction*, v. 16, n. 5, p. 576–585, 2006.

- HUANG, W. et al. Identification of yellow rust in wheat using in-situ spectral reflectance measurements and airborne hyperspectral imaging. *Precision Agriculture*, Springer US, v. 8, p. 187–197, 2007.
- HUGHES, C. E. et al. Mixed reality in education, entertainment, and training. *IEEE Computer Graphics and Applications*, v. 25, n. 6, p. 24–30, 2005.
- IWATA, H. et al. Circulafloor. *SIGGRAPH '04: ACM SIGGRAPH 2004 Emerging Technologies*, 2004.
- IWATA, H.; YANO, H.; TOMIYOSHI, M. String walker. *ACM SIGGRAPH 2007 Emerging Technologies*, 2007.
- JACOBSON, J. Configuring multiscreen displays with existing computer equipment. *Proceedings of Conference on Human Factors HFES'2002*, p. 760–765, 2002.
- JACOBSON, J.; LEWIS, M. Game engine virtual reality with caveat. *Computer (IEEE)*, v. 38, n. 4, p. 79–82, 2005.
- JACOBSON, L. *Realidade Virtual em Casa*. Rio de Janeiro, RJ: Berkeley, 1994.
- JEONG, S.; HASHIMOTO, N.; MAKOTO, S. A novel interaction system with force feedback between real - and virtual human: An entertainment system: "virtual catch ball". *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, p. 61–66, 2004.
- JR, D. S. et al. Fall armyworm damaged maize plant identification using digital images. *Biosystems Engineering*, v. 85, n. 4, p. 449 – 454, 2003.
- JUAN, M. C. et al. Using augmented reality to treat phobias. *IEEE Computer Graphics and Applications*, v. 25, n. 6, p. 31–37, 2005.
- KALAWSKY, R. *The Science of Virtual Reality and Virtual Environments*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1993.
- KAUFMANN, C.; LIU, A.; BURRIS, D. Dti autostereographic display: Initial evaluation. In: WESTWOOD, J. D. et al. (Ed.). *Medicine Meets Virtual Reality 2000*. Amsterdam, Netherlands: IOS Press, 2000. p. 156–158.
- KIM, S.; KOIKE, Y.; SATO, M. Tension based 7 dofs force feedback device: Spidar-g. *Transactions on Control, Automation and Systems Engineering*, v. 4, n. 1, p. 9–16, 2002.
- KIMURA, K.; IBARAKI, J. Visualization of satellite orbital motions by virtual reality. *Proceedings of the Symposium on Celestial Mechanics*, v. 31, p. 413–419, 1999.
- KIRNER, C.; KIRNER, T. G. Virtual reality and augmented reality applied to simulation visualization. In: *Simulation and Modeling: Current Technologies and Applications*. Hershey, NY, USA: IGI Publishing, 2007. p. 391–419.
- KIRNER, C.; TORI, R. *Realidade Virtual: Conceitos e Tendências*. São Paulo, SP: Editora Mania de Livro, 2004.
- KONDO, N. et al. Machine vision based quality evaluation of iyokan orange fruit using neural networks. *Computers and Electronics in Agriculture*, v. 29, n. 1, p. 135 – 147, 2000.

- KRUEGER, M. W.; GIONFRIDDO, T.; HINRICHSEN, K. Videoplace - an artificial reality. *CHI '85: Proceedings of The Sigchi Conference On Human Factors In Computing Systems*, p. 35–40, 1985.
- KUNZLER, U.; RUNDE, C. Kinesthetic haptics integration into large-scale virtual environments. *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, p. 551–556, 2005.
- LATTA, J. N.; OBERG, D. J. A conceptual virtual reality model. *IEEE Computer Graphics and Applications*, v. 14, n. 1, p. 23–29, 1994.
- LEE, W. S.; SLAUGHTER, D. C.; GILES, D. K. Robotic weed control system for tomatoes. *Precision Agriculture*, v. 1, p. 95–113, 1999.
- LESTON, J. Virtual reality: The it perspective. *The Computer Bulletin*, v. 38, n. 3, p. 12–13, June 1996.
- LIPTON, L. *Foundations of the Stereoscopic Cinema*. New York, NY - USA: Van Nostrand Reinhold, 1982.
- LOKKI, T. et al. Creating interactive virtual auditory environments. *IEEE Computer Graphics and Applications*, v. 22, n. 4, p. 49–57, 2002.
- LULIO, L. C. *Técnicas de Visão Computacional Aplicadas ao Reconhecimento de Cenas Naturais e Locomoção Autônoma em Robôs Agrícolas Móveis*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2011.
- MACHADO, L. S. *A Realidade Virtual no Modelamento e Simulação de Procedimentos Invasivos em Oncologia Pediátrica: Um Estudo De Caso no Transplante de Medula Óssea*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2003.
- MAIR, G. M. Telepresence - the technology and its economic and social implications. *Proceedings of the IEEE International Symposium on Technology and Society*, Glasgow, UK, p. 118–124, June 1997.
- MARCHANT, J.; ANDERSEN, H.; ONYANGO, C. Evaluation of an imaging sensor for detecting vegetation using different waveband combinations. *Computers and Electronics in Agriculture*, v. 32, n. 2, p. 101–117, 2001.
- MARCONI, M. A.; LAKATOS, E. M. *Metodologia Científica*. 5. ed. São Paulo, SP: Atlas, 2010.
- MAXWELL, S. et al. An automated approach to mapping corn from landsat imagery. *Computers and Electronics in Agriculture*, v. 43, n. 1, p. 43 – 54, 2004.
- MCDERMOTT, S. An exploration of virtual 3d sound. *Acoustical Society of America*, New Orleans, Louisiana, 2007.
- MILGRAM, P. et al. Augmented reality: A class of displays on the reality-virtuality continuum. *SPIE - Telemanipulator and Telepresence Technologies*, v. 2351, p. 282–292, 1994.
- MINSKY, M. Telepresence. *Omni*, p. 45–51, June 1980.

MONTEIRO, E. F.; PAIVA, A. C. de. Um ambiente para a visualização de informações baseado em realidade virtual não-imersiva. *Proceedings of V Synposium on Virtual Reality*, p. 395–397, 2002.

MOORTHY, I. et al. Extracting tree crown properties from ground-based scanning laser data. In: *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*. [S.l.: s.n.], 2007. p. 2830 –2832.

MORIE, J. F. Inspiring the future: Merging mass communication, art, entertainment and virtual environments. *ACM SIGGRAPH Computer Graphics*, v. 28, n. 2, p. 135–138, 1994.

NETTO, A. V.; MACHADO, L. S.; OLIVEIRA, M. C. F. *Realidade Virtual: Fundamentos e Aplicações*. Florianópolis, SC: Visual Books, 2002.

NOVINT, T. I. Novint falcon. <http://home.novint.com/>, 2008.

OLIVEIRA, C. de et al. Low-cost gesture detection as a form of interaction in a virtual reality multi-projection system. In: AL, P. J. da Silva Bartolo et (Ed.). *Innovative Developments in Design and Manufacturing : Advanced Research in Virtual and Rapid Prototyping*. London: Taylor & Francis Ltd, 2009. p. 615–618.

PAL, N. R.; PAL, S. K. A review on image segmentation techniques. *Pattern Recognition*, v. 26, n. 9, p. 1277 – 1294, 1993.

PARK, C.-H.; JANG, G.; CHAI, Y.-H. Development of a virtual reality training system for live-line workers. *International Journal of Human-Computer Interaction*, v. 20, n. 3, p. 285–303, 2006.

PARK, S. H. et al. Virtual knee joint replacement surgery system. *Geometric Modeling and Imaging*, v. 4, n. 6, p. 79–84, 2007.

PHILIPP, I.; RATH, T. Improving plant discrimination in image processing by use of different colour space transformations. *Computers and Electronics in Agriculture*, v. 35, n. 1, p. 1 – 15, 2002.

PILAU, F. G. *Saldo de Radiação da Copa de Laranjeira num Pomar e de Renques de Cafeeiros: Medidas e Estimativas*. Tese (Doutorado) — Universidade de São Paulo - Escola Superior de Agricultura Luiz de Queiroz, 2005.

PIMENTEL, J. et al. Construção e gestão da complexidade de cenários urbanos 3d em ambientes virtuais imersivos. *Actas do X Encontro Português de Computação Gráfica*, p. 165–174, 2001.

PIMENTEL, K.; TEIXEIRA, K. *Virtual Reality: Through the New Looking Glass*. New York, NY, USA: McGraw-Hill, Inc., 1994.

PLEBE, A.; GRASSO, G. Localization of spherical fruits for robotic harvesting. *Machine Vision and Applications*, Springer Berlin Heidelberg, v. 13, p. 70–79, 2001.

PORTO, A. J. V. et al. Manufatura virtual: Conceituação e desafios. *Gestão & Produção*, v. 9, n. 3, p. 297–312, 2002.

- RAMOS, A. M. *Aplicação, investigação e análise da metodologia de reduções*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2007.
- RAMOS, R. A. et al. Variação sazonal do crescimento vegetativo de laranjeiras hamlin enxertadas em citrumeleiro swingle no município de limeira, estado de são paulo. *Acta Scientiarum. Agronomy*, Editora da Universidade Estadual de Maringa - EDUEM, Maringa, v. 32, n. 3, p. 539–545, 2010.
- RASKAR, R. et al. The office of the future: A unified approach to image-based modeling and spatially immersive displays. *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, p. 179–188, 1998.
- RECHE, A.; MARTIN, I.; DRETTAKIS, G. Volumetric reconstruction and interactive rendering of trees from photographs. In: *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings*. [S.l.: s.n.], 2004. p. 720–727.
- RHEE, S.-M.; PARK, J. Low-cost telepresence for collaborative virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, Piscataway, NJ, USA, v. 13, n. 1, p. 156–166, 2007.
- RHEINGOLD, H. *Virtual Reality*. New York: Summit Books, 1991.
- ROBERTS, L. G. *Machine perception of three-dimensional solids*. Tese (Doutorado) — Massachusetts Institute of Technology. Dept. of Electrical Engineering, 1963.
- ROBERTSON, G. G.; CARD, S. K.; MACKINLAY, J. D. Three views of virtual reality: Nonimmersive virtual reality. *Computer (IEEE)*, v. 26, n. 2, p. 81–83, February 1993.
- ROSENBERG, L. Virtual haptic overlays enhance performance in telepresence tasks. *Proceedings of SPIE Telemanipulator and Telepresence Technologies*, v. 2351, p. 99–108, 1995.
- RYU, J. et al. A game engine based architectural simulator on multi-projector displays. *Proceedings of The 11th Conference on Computer-Aided Architectural Design Research in Asia(CAADRIA)*, v. 4, p. 613–615, 2006.
- SCHENK, M.; STRASSBURGER, S.; KISSNER, H. Combining virtual reality and assembly simulation for production planning and worker qualification. *Proc. of The International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, p. 411–414, 2005.
- SCHOFIELD, C. et al. Monitoring pig growth using a prototype imaging system. *Journal of Agricultural Engineering Research*, v. 72, n. 3, p. 205 – 210, 1999.
- SCHWAIGER, M. C.; THÜMMEL, T.; ULBRICH, H. Cyberwalk: Implementation of a ball bearing platform for humans. *Lecture Notes in Computer Science*, v. 4551, p. 926–935, 2007.
- SCHWEBELA, D. C.; GAINESA, J.; SEVERSON, J. Validation of virtual reality as a tool to understand and prevent child pedestrian injury. *Accident Analysis & Prevention*, 2008.
- SENSABLE, T. I. Haptic devices. <http://www.sensable.com/>, 2008.
- SERGEANT, D.; BOYLE, R.; FORBES, M. Computer visual tracking of poultry. *Computers and Electronics in Agriculture*, v. 21, n. 1, p. 1 – 18, 1998.

- SHERIDAN, T. B. Musings on telepresence and virtual presence. *Presence: Teleoperators and Virtual Environments*, v. 1, n. 1, p. 120–126, 1992.
- SHERMAN, W. R.; CRAIG, A. *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- SHLYAKHTER, I. et al. Reconstructing 3d tree models from instrumented photographs. *Computer Graphics and Applications, IEEE*, v. 21, n. 3, p. 53–61, may/jun 2001.
- SILVA, F. A.; RIBAS, W. K. Perfilamento laser em áreas densamente florestadas. *Rio Pipeline Conference and Exposition*, 2007.
- SLAUGHTER, D. C.; CHEN, P.; CURLEY, R. G. Vision guided precision cultivation. *Precision Agriculture*, Springer Netherlands, v. 1, p. 199–217, 1999.
- SOARES, L. P. et al. Designing a highly immersive interactive environment: The virtual mine. *Computer Graphics Forum*, v. 29, n. 6, p. 1756–1769, 2010.
- SOGAARD, H. T.; OLSEN, H. J. Determination of crop rows by image analysis without segmentation. In: *Computers and Electronics in Agriculture*. [S.l.: s.n.], 2003. p. 141–158.
- SOH, L. K.; TSATSOLIS, C. Segmentation of satellite imagery of natural scenes using data mining. *IEEE Transactions on Geoscience and Remote Sensing*, v. 37, n. 2, 1999.
- SOMMERVILLE, I. *Engenharia de Software*. 8. ed. Sao Paulo, SP, Brasil: Pearson Education, 2007.
- SORID, D.; MOORE, S. K. The virtual surgeon. *IEEE Spectrum*, v. 37, n. 7, p. 26–31, 2000.
- STAJNKO, D.; LAKOTA, M.; HOCEVAR, M. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. *Computers and Electronics in Agriculture*, v. 42, n. 1, p. 31–42, 2004.
- STEUER, J. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, v. 42, n. 4, p. 73–93, 1992.
- STROTHOFF, S. et al. Interactive generation of virtual environments using muavs. In: *Proceedings of International Symposium on VR innovation (ISVRI) 2011*. [S.l.: s.n.], 2011.
- STUART, R. *The Design of Virtual Environments*. New York: McGraw-Hill, Inc., 1996.
- SUSNIK, R. et al. Spatial sound generation using hrtf created by the use of recursive filters. *Proceedings of EUROCON 2003. Computer as a Tool. The IEEE Region 8*, v. 1, p. 449–453, 2003.
- SUZUKI, S.; BE, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, v. 30, n. 1, p. 32–46, apr 1985.
- TAKACS, B.; SIMON, L. A clinical virtual reality rehabilitation system for phobia treatment. *Proceedings of the 11th International Conference Information Visualization*, p. 798–806, 2007.

- TAN, P. S.; HASHIMOTO, N.; SATO, M. Game-engine based virtual environments for immersive projection display systems. *Eizo Joho Media Gakkai Gijutsu Hokoku*, v. 29, n. 17, p. 29–32, 2005.
- TANTASWADI, P. et al. Machine vision for automated visual inspection of cotton quality in textile industries using color isodiscrimination contour. *Computers and Industrial Engineering*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 37, n. 1-2, p. 347–350, out. 1999.
- TEAGUE, M. R. Image analysis via the general theory of moments. *Optical Society of America*, v. 70, p. 920–930, 1980.
- TEAGUE, T.; PALMER, M.; JACKSON, R. Xml for capital facilities. *Leadership and Management in Engineering*, v. 3, n. 2, p. 82–85, 2003.
- TIAN-YANG, D.; JING, F.; XIN-PEI, Z. Virtual tree model based on environment influence and its application. In: *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on*. [S.l.: s.n.], 2008. p. 582.
- TIFFIN, J.; TERASHIMA, N. *HyperReality: Paradigm for the Third Millennium*. London, UK: Routledge, 2001.
- TILLET, R. Image analysis for agricultural processes: a review of potential opportunities. *Journal of Agricultural Engineering Research*, v. 50, n. 0, p. 247 – 258, 1991.
- TILLET, R.; ONYANGO, C.; MARCHANT, J. Using model-based image processing to track animal movements. *Computers and Electronics in Agriculture*, v. 17, n. 2, p. 249 – 261, 1997.
- VALE, H. M. C. et al. Construction of immersive multi-projection environments for treatment of phobia of heights. In: AL, P. J. da Silva Bartolo et (Ed.). *Innovative Developments in Design and Manufacturing : Advanced Research in Virtual and Rapid Prototyping*. London: Taylor & Francis Ltd, 2009. p. 619–623.
- VALLIUS, T.; RÖNING, J. A telepresence robot system realized by embedded object concept. *Proceedings of SPIE, The International Society for Optical Engineering*, Boston, MA, USA, v. 6384, p. 638403.1–638403.13, October 2006.
- VERMA, V.; KUMAR, R.; HSU, S. 3d building detection and modeling from aerial lidar data. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. [S.l.: s.n.], 2006. v. 2, p. 2213 – 2220.
- VINCE, J. *Introduction to Virtual Reality*. London: Springer-Verlag, 2004.
- VORAA, J. et al. Using virtual reality technology for aircraft visual inspection training: Presence and comparison studies. *Applied Ergonomics*, v. 33, n. 6, p. 559–570, 2002.
- WAGNER, W. et al. From single-pulse to full-waveform airborne laser scanners: potential and practical challenge. *International Archives of Photogrammetry and Remote Sensing*, v. 35, p. 201–206, 2004.
- WANG, C. et al. Three dimensional reconstruction from optical microscopic plant slice images. In: *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*. [S.l.: s.n.], 2008. p. 94 –97.

- WEI-DONG, T.; CHEN, G.; PING-PING, L. Study on modeling virtual plant development process based on growth model. In: *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*. [S.l.: s.n.], 2010. v. 3, p. 432 –435.
- WENG, S.; ZHAO, G.; HE, B. Rapid reconstruction of 3d building models from aerial images and lidar data. In: *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. [S.l.: s.n.], 2010. v. 2, p. V2–195 –V2–198.
- WILSON, S. *Information Arts: Intersections of Art, Science, and Technology*. Cambridge, MA, USA: The MIT Press, 2003.
- XU, W. et al. Design and implementation of 3D model database for general-purpose 3D GIS. *Geo-Spatial Information Science*, Wuhan University, co-published with Springer, v. 13, n. 3, p. 210–215–215, sep 2010.
- YADAV, B.; JINDAL, V. Monitoring milling quality of rice by image analysis. *Computers and Electronics in Agriculture*, v. 33, n. 1, p. 19–33, 2001.
- YANG, D. et al. The teleoperation system of space robot based on virtual technology. *Mechatronics and Automation, 2005 IEEE International Conference*, Niagara Falls, Canada, v. 4, p. 2156–2159, August 2005.
- YANG, K.; JIE, J. Design and implementation of virtual scene based on vrml. In: *Audio Language and Image Processing (ICALIP), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 1022 –1025.
- YONGJIAN, H. et al. Real-time rendering of large-scale tree scene. In: *Computer Science Education, 2009. ICCSE '09. 4th International Conference on*. [S.l.: s.n.], 2009. p. 748 –752.
- ZELEK, J. S.; BRUCE, N.; KANWAR, R. *Watching Pigs*. Ontario, Canada: [s.n.], 2000.
- ZENG, J.; ZHANG, Y.; ZHAN, S. 3d tree models reconstruction from a single image. In: *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*. [S.l.: s.n.], 2006. v. 2, p. 445 –450.
- ZENG, L.; WANG, G. Interactive plant modeling based on fractal. In: *Natural Computation, 2008. ICNC '08. Fourth International Conference on*. [S.l.: s.n.], 2008. v. 4, p. 644 –648.
- ZHOU, Z. et al. Interactive entertainment systems using tangible cubes. *Proceedings of Australian Workshop on Interactive Entertainment*, p. 19–24, 2004.
- ZHU, C. et al. Cluster-based construction of tree crown from scanned data. In: *Plant Growth Modeling, Simulation, Visualization and Applications (PMA), 2009 Third International Symposium on*. [S.l.: s.n.], 2009. p. 352 –359.
- ZIMMERMAN, T. G. et al. A hand gesture interface device. *Chi '87: Proceedings Of The SIGCHI/GI - Conference on Human Factors in Computing Systems and Graphics Interface*, p. 189–192, 1987.