

Este exemplar corresponde à redação final da Tese/Dissertação devidamente corrigida e defendida por: SCHUBERT RIBEIRO DE CARVALHO
e aprovada pela Banca Examinadora.
Campinas, 16 de 05 de 2005

COORDENADOR DE PÓS-GRADUAÇÃO
CPG-IC

Algoritmo de Vergência para o Controle da Atenção de Humanos Virtuais
Schubert Ribeiro de Carvalho
Dissertação de Mestrado

Algoritmo de Vergência para o Controle da Atenção de Humanos Virtuais

Schubert Ribeiro de Carvalho

7 de janeiro de 2005

Banca Examinadora:

- Prof. Dr. Luiz Marcos Garcia Gonçalves
(DCA - UFRN) (Orientador)
- Neucimar Jerônimo Leite
(IC - UNICAMP) (Co-orientador)
- Prof. Dr. Siome Klein Goldenstein
(IC - UNICAMP)
- Prof. Dr. Ricardo Cordeiro de Farias
(COPPE - UFRJ)
- Prof. Dr. Jorge Stolfi
(IC - UNICAMP) (Suplente)

UNIDADE	BC
Nº CHAMADA	UNICAMP C253a
V	EX
TOMBO BC	64080
PROC.	16-P-00086-05
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	11,00
DATA	01/06/05
Nº CPD	BIB. 12351340

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Carvalho, Schubert Ribeiro de
C253a Algoritmo de vergência para o controle da atenção de humanos virtuais /
Schubert Ribeiro de Carvalho – Campinas, [S.P. :s.n.], 2005.

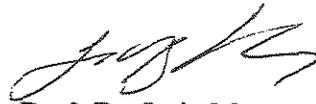
Orientadores : Luiz Marcos Garcia Gonçalves, Neucimar Jerônimo Leite
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de
Computação.

1. Realidade virtual. 2. Visão – Processamento de dados. 3. Algoritmos.
4. Computação gráfica. I. Gonçalves, Luiz Marcos Garcia. II. Leite, Neucimar
Jerônimo. III. Universidade Estadual de Campinas. Instituto de Computação.
IV. Título.

Algoritmo de Vergência para o Controle da Atenção de Humanos Virtuais

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Schubert Ribeiro de Carvalho e aprovada pela Banca Examinadora.

Campinas, 07 de janeiro de 2005.

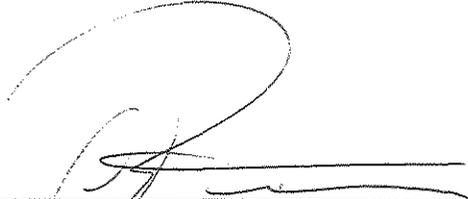


Prof. Dr. Luiz Marcos Garcia Gonçalvez
(DCA - UFRN) (Orientador)

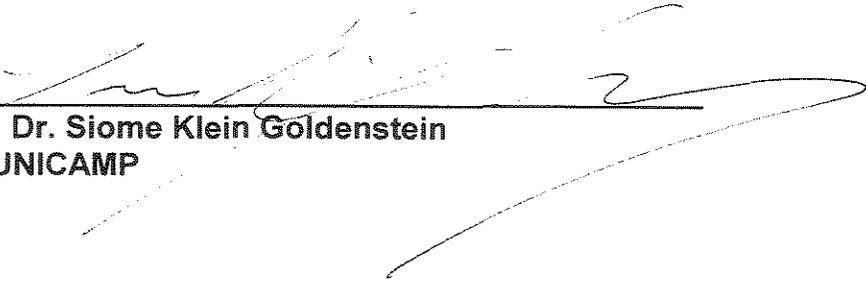
Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 07 de janeiro de 2005, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Ricardo Cordeiro de Farias
COPPE - UFRJ



Prof. Dr. Siome Klein Goldenstein
IC - UNICAMP



Prof. Dr. Luiz Marcos Garcia Gonçalves
DCA - UFRN

© Schubert Ribeiro de Carvalho, 2005.

Todos os direitos reservados.

Resumo

Os problemas de pesquisa mais comuns abordados na literatura sobre Humanos Virtuais são mais relacionados com a inserção e representação destes agentes gráficos no computador. Com o avanço das tecnologias e pesquisas nas áreas de computação gráfica e realidade virtual, novos problemas podem ser tratados, em relação as funcionalidades destes agentes virtuais. Atualmente, um dos principais desafios está associado ao desenvolvimento de humanos virtuais dotados de percepção visual realística. Um problema específico, referente a visão sintética destes agentes, é o controle da atenção, ou seja, definir e prover uma maneira eficiente e natural para que ele mude seu foco de atenção de uma posição corrente para outra no seu ambiente virtual.

A coordenação dos movimentos da cabeça do agente, fazendo com que os olhos sejam fixados no objeto desejado, não é considerada uma tarefa trivial e está relacionada com a percepção de profundidade do ambiente.

Neste trabalho, propomos um algoritmo de vergência, usado para o controle da atenção de humanos virtuais, onde o parâmetro de profundidade é obtido do *Z-buffer*. O controle dos movimentos da cabeça do humano virtual é executado com a realização de simples cálculos geométricos e no final do processo de vergência, os olhos são fixados corretamente em qualquer região de interesse do ambiente, passada previamente para o agente virtual.

Palavras-chave: Humanos Virtuais, Algoritmo de Vergência, Visão Sintética e Z-buffer.

Abstract

Common research problems treated in literature about Virtual Humans are most related with insertion and representation of these graphic agents in computers. With technological advance and new researches in Computer Graphics and Virtual Reality, new problems can have place, related to the functionality of the virtual agents. Currently, one of the main challenges is associated to development of virtual humans with realistic virtual perception and a specific problem that appears, related to synthetic vision in these agents, is the proper control of attention. That is, to define and provide an efficient and natural manner for it to change attentional focus from a current position to another one in its virtual environment.

Coordination of head movements by way of fixation of eyes in a desired object is not considered a trivial task, being related to depth perception.

In this work, we propose an algorithm for vergence, used for attention control in virtual humans, where the depth parameter is obtained from *Z-buffer*. Control of virtual human head movements is performed with simple geometric calculations and at the end of vergence process the eyes are correctly fixed in any interest region of the environment, previously given to the virtual agent.

Keywords: Virtual Humans, Vergence Algorithm, Synthetic Vision and "Z-buffer".

À minha mãe Arneide e ao meu pai Schubert.

Agradecimentos

Imagino que a grande maioria dos pesquisadores acredite que a última coisa que se escreve em uma dissertação ou tese é o resumo, pelo menos é isso que vejo nos livros de metodologia da pesquisa. Bom, sou obrigado a discordar disso por que eu passei todo meu mestrado imaginando quando chegaria este momento, o da escrita dos agradecimentos. Bom, agora que cheguei aqui estou com uma dúvida. Não sei se começo por ordem alfabética ou por ordem cronológica :-). Espero não esquecer de ninguém.

Em primeiro lugar quero agradecer aos meus pais Arneide e Schubert por nunca deixarem com que eu perca as esperanças ou fique desanimado, fazendo com que eu sempre acredite em mim e no que eu sou capaz de fazer. A minha tia Christina por ter sempre me ajudado em vários aspectos importantes de minha vida, se hoje sou o que sou é por que ela também contribuiu para isso. A minha avó Inácia por também ter me apoiado quando cheguei em Campinas e estava duro sem bolsa :-). A Milene Carvalho por ter que me dividir com o mestrado e várias vezes entender que não posso ir ao cinema ou pra balada por que tenho que terminar este mestrado, ufa! acho que agora consegui. É, só que agora vem o Doutorado!

Quero agradecer também aos meus familiares, tios e tias Amarildo, Janete, Claudette, Ivone, Paulo, Anibal, Milton, a minha avó Beth, por me apoiarem. Ao meu bisavô Leonideo pelos poemas. Aos meus primos, Mikio, Sandro, Paulo, Leandro, Tiago, Michiko, Daniele, Lais. Aos meus mais novos familiares, Ildemar, Socorro e Any Caroline (minha afilhada).

A Luiz Marcos (Super LM) por sempre responder aos meus e-mails, tentando dar explicações nos mínimos detalhes às minhas dúvidas, por ter tido muita paciência durante todo o desenvolvimento deste trabalho e principalmente por ter acreditado que eu iria conseguir. Obri-

gado Eloi, Affonso e Ting pelas cartas de recomendações. Valeu Bianchi por ter me mostrado as maravilhas da realidade virtual, e principalmente por ter me orientado e me ajudado com o meu TCC.

Eu estava começando a escrever este parágrafo para agradecer a grande amizade dos meus dois grandes amigos Alexandre (meu irmão) e Aldenor, quando o Aldenor me ligou dizendo que estava indo pra Manaus ser um dos diretores de evento da coca-cola e que vai se casar daqui a três meses, que notícia ótima! Boa sorte Aldenor! E boa sorte no mestrado em biologia Alexandre.

Ao Djou (Fernando) e Guilherme pelas momentos de alegria, pelo convívio civilizado na república :-), pelas faxinas organizadas e pelas pizzas com coca-cola nos finais de semana (os vizinhos que o digam). Valeu Bob Esponja (Edmar) pelas conversas animadas e por ter ido a Belém me visitar, ainda estou devendo!

Aos amigos do IC Douglas e Lasaro (pelas conversas filosóficas) ao Pará (Eduardo), Silvana, Borin, Juliana, Gustavo (baiano), Jé Augusto, Jean. Ao povo de Belém que nunca desiste, aos que ainda estão por aqui e também aos que já voltaram, Fábio, Heider, Carmem, Tatágia (Aroldo), Márcia e a todos os outros que vieram pra cá para estudar ou trabalhar.

A Schubert Carvalho por não ter desistido em dar a sua pequena contribuição para o progresso da ciência. E por fim a Deus por ter me dado o dom da vida, pois sem ele nada do que fiz seria possível.

Vocês estão aqui, não para copiar, mas sim para propor soluções.

(Wu Shin Ting, aula de computação gráfica)

Se ainda não terminou é por que ainda não chegou ao fim.

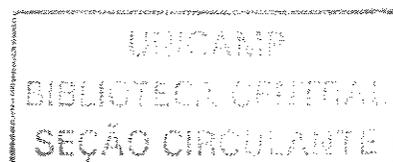
(Autor desconhecido)

Sumário

Resumo	vii
Abstract	viii
Agradecimentos	x
1 Introdução	1
1.1 Objetivo do Trabalho	3
1.2 Organização do Texto	3
2 Agentes Virtuais e Visão Sintética	5
2.1 Agentes Virtuais	5
2.2 Visão	6
2.2.1 Visão Real	6
2.2.2 Visão Sintética	7
2.2.3 Modelo de Câmera	8
2.2.4 Trabalhos Relacionados	9
2.3 Discussões e Considerações	11

3	Controle da Atenção: Vergência e Disparidade	13
3.1	Atenção	14
3.2	Vergência	15
3.2.1	O Problema da Vergência	16
3.2.2	Técnicas para o Controle da Vergência	18
3.3	Disparidade	19
3.3.1	O Problema da Disparidade	19
3.4	Trabalhos Relacionados	21
3.5	Discussões e Considerações	23
4	Arquitetura do Sistema	25
4.1	Descrição dos Módulos da Arquitetura	25
4.1.1	Modelo do Ambiente	26
4.1.2	Agente Virtual	27
4.1.3	Módulo de Percepção Visual	28
4.1.4	Módulo de Decisão	29
4.1.5	Módulo de Controle	30
4.2	Executando a Vergência	31
4.3	Considerações	32
5	Implementações Realizadas na Arquitetura	33
5.1	Plataforma de Simulação: Agente Virtual	33
5.2	Modelo da Fóvea: Imagens Multi-Resolução	36

5.3	Extração de Características	39
5.4	Controle da Atenção: Algoritmo de Vergência	40
5.5	Implementação do Algoritmo de Vergência	43
5.5.1	Aquisição da Informação de Profundidade: <i>Z-buffer</i>	43
5.5.2	Implementação dos Movimentos Verticais	45
5.5.3	Implementação dos Movimentos Horizontais	46
5.5.4	Implementação da Coordenação dos Movimentos	52
5.5.5	Política de Deslocamento	53
5.5.6	Generalizando o modelo para o deslocamento do Sistema Gráfico	56
6	Experimentos e Resultados	60
6.1	Coordenação dos olhos e pescoço - experimento 1	61
6.2	Vergindo em locais diferentes - experimento 2	64
6.3	Mudança de atenção em grupo - experimento 3	65
7	Conclusões e Perspectivas	69
7.1	Dificuldades Encontradas	70
7.2	Limitações do Algoritmo e do Sistema Gráfico	70
7.3	Objetivos e Contribuições Alcançados	71
7.4	Trabalhos Futuros	72
	Bibliografia	73



Lista de Tabelas

6.1	Tabela dos valores das coordenadas de deslocamento do experimento 1	64
6.2	Tabela dos valores das coordenadas de deslocamento do experimento 2	65

Lista de Figuras

2.1	Modelo de câmera virtual	8
3.1	Vergência	16
3.2	Reconstrução estéreo	20
4.1	Arquitetura modular do Câmbio	26
4.2	Mecânica dos olhos e do pescoço	28
4.3	Arquitetura para o controle da atenção do agente virtual	31
5.1	Interface gráfica	34
5.2	Espaço de manipulação do agente virtual	35
5.3	Ilustração do algoritmo da fóvea	37
5.4	Imagens em Multi-resolução com o mesmo tamanho	38
5.5	Imagens em Multi-resolução escalonadas	38
5.6	Código, em OpenGL, para obtenção da informação de profundidade	44
5.7	Dinâmica dos movimentos verticais	46
5.8	Situação final de vergência horizontal.	47
5.9	Dinâmica dos movimentos horizontais dos pescoço	48

5.10	Ajuste do ângulo de vergência inicial	50
5.11	Área de interseção da vergência horizontal dos olhos	51
5.12	Modelo Matemático de deslocamento	55
5.13	Deslocamento do sistema gráfico	57
5.14	Modelo Matemático de deslocamento vetorial	58
6.1	Situação inicial do sistema.	61
6.2	Movimentos de vergência horizontais e verticais e verticais, horizontais	63
6.3	Movimentos de vergência variados	66
6.4	Ambiente virtual em grupo	68

Capítulo 1

Introdução

O primeiro relato encontrado na literatura sobre a inserção de um agente gráfico no computador, com o objetivo de representar uma pessoa, foi em 1965 [12] pelo pesquisador Ivan Sutherland e o primeiro experimento só foi realizado em 1968, onde o agente gráfico foi construído utilizando um modelo computacional em arame "wireframe". Desde então, o crescente avanço tecnológico permitiu a realização de pesquisas em computação gráfica e o surgimento dos ambientes virtuais [49], tendo sido desenvolvidas muitas aplicações envolvendo agentes gráficos, principalmente aqueles que se parecem com o ser humano (os humanos virtuais).

As aplicações envolvendo humanos virtuais cobrem um leque repleto de possibilidades, abrangendo áreas de pesquisa referentes a entretenimento (jogos de computador) [12], filmes e sistemas de realidade virtual (para aumentar a sensação de presença [44, 49] ou simplesmente para populacionar os ambientes virtuais criados por estes sistemas [14, 48])

Atualmente, existe um grande interesse tanto acadêmico quanto industrial para o desenvolvimento de humanos virtuais inteligentes, que possam, por exemplo, navegar autonomamente pelo ambiente ou estabelecer uma comunicação mais realística com outros agentes inseridos no mesmo ambiente virtual, utilizando principalmente mecanismos baseados em visão [43, 41, 15, 28, 17, 12, 11]. Outro interesse é o desenvolvimento de mecanismos mais realistas para representar os gestos dos agentes e em particular o controle dos movimentos dos olhos tem chamado mais atenção [14, 51, 50, 52, 32, 12].

A visualização do foco de atenção (direção dos olhos do agente virtual) de outros agentes virtuais é importante para que se saiba aonde o agente está olhando, apenas com a sua localização atencional [51]. Este entendimento pode ser utilizado em tarefas que necessitem de uma comunicação não verbal, favorecendo um processo de interação mais amigável entre os participantes de um ambiente virtual [12]. Outro motivo para o controle da atenção é fazer com que o agente possa desempenhar tarefas de reconhecimento, colocando o objeto desejado na fóvea (centro da retina) e para isto a fixação dos olhos no objeto torna-se uma tarefa necessária para o processo de extração de características.

O controle da atenção pode ser obtido através do controle dos ângulos de vergência, formados entre as retas normais a cada olho e o ponto de visão do agente. Existem diversas formas de se controlar estes ângulos, para que os olhos do agente sejam fixados no ponto de interesse desejado, entre estas técnicas a mais utilizada faz uso da disparidade entre as imagens obtidas pelas duas câmeras. Porém, o grande problema desta técnica são os cálculos de correlação, geralmente necessários para se obter medidas da disparidade entre as imagens [38, 24].

Os algoritmos de vergência utilizados atualmente para controlar a atenção de agentes virtuais [18, 19] são baseados em técnicas de visão computacional. Geralmente, estes algoritmos são desenvolvidos para agentes ou plataformas robóticas reais. Por este motivo, o principal problema enfrentado por estes algoritmos é a quantidade de cálculos necessários, nas imagens capturadas, para transformar a resposta de uma função em um valor que será utilizado para gerar um deslocamento angular das câmeras.

Entretanto, como o agente virtual está inserido em um ambiente simulado, a visão deste agente pode ser considerada uma visão sintética [43, 41, 15, 28, 17, 12, 11]. Portanto, a informação fornecida pelo ambiente virtual, por exemplo a profundidade que será utilizada para o controle dos olhos do agente com o objetivo de fixá-los no local desejado [51], pode ser extraída com maior facilidade sem a necessidade de maciços cálculos de correlação.

Tendo como base a visão sintética do humano virtual e os problemas enfrentados atualmente pelos algoritmos de vergência empregados para controlar a atenção de agentes virtuais, o presente trabalho propõe um algoritmo de vergência simples baseado na informação de pro-

fundidade do ambiente *Z-buffer*, onde os únicos cálculos necessários para o controle da atenção do agente são os cálculos geométricos de distância que são usados para gerar os deslocamentos direcionais. Os testes de simulação usados para validar o algoritmo mostraram que os olhos do humano virtual são fixados em qualquer local desejado do ambiente. Para tarefas que necessitam de mais realismo a coordenação dos movimentos da cabeça do agente direcionam os olhos no local desejado.

1.1 Objetivo do Trabalho

O objetivo principal deste trabalho é estudar e propor um algoritmo de vergência, baseado em técnicas de visão sintética, para o controle da atenção de humanos virtuais. O funcionamento deste algoritmo não deve depender de cálculos de correlação, pois este é o principal problema enfrentado hoje para a execução de tarefas em tempo real.

Além disso, propõe-se um modelo computacional para a representação da fóvea, baseado em imagens multi-resolução, para acelerar o processo de extração de características, caso isto venha acontecer futuramente. Uma plataforma de simulação foi desenvolvida para testar e validar computacionalmente nosso algoritmo e também propõe-se uma arquitetura mais geral, além da arquitetura específica para o controle da atenção, para a integração de outras funcionalidades perceptuais ao nosso agente, tais como: memória e reconhecimento.

Durante o desenvolvimento desta dissertação foram publicados artigos em conferências [9, 50, 51, 10] e em uma revista nacional [52].

1.2 Organização do Texto

O capítulo 2 aborda assuntos referentes a humanos virtuais, aos diferentes tipos de visão que podem ser usadas pelo agente, apresenta uma breve introdução sobre o modelo de câmera virtual e apresenta alguns dos principais trabalhos encontrados na literatura sobre visão sintética. O capítulo 3 apresenta uma introdução sobre os diferentes tipos de atenção que podem

ser inseridas no agente, apresenta os problemas associados à vergência e à disparidade e também apresenta os principais trabalhos referentes ao controle da atenção de humanos virtuais que utilizaram algum tipo de algoritmo de vergência. No capítulo 4 é feita uma descrição dos módulos da arquitetura proposta para o controle da atenção. O capítulo 5 trata dos aspectos de implementação do sistema, fazendo uma descrição das partes que foram implementadas e do algoritmo de vergência proposto nesta dissertação. Os demais capítulos abordam os experimentos e resultados (capítulo 6) e as conclusões e perspectivas (capítulo 7).

Capítulo 2

Agentes Virtuais e Visão Sintética

Nos últimos anos, a comunidade acadêmica e industrial passou a mostrar um crescente interesse em pesquisas envolvendo a adição de sentidos perceptuais à criaturas sintéticas. As entidades gráficas mais pesquisadas atualmente são aquelas que se parecem com o ser humano (humanos virtuais). Entre os sentidos mais estudados, a visão é sem dúvida o mais necessário e poderoso. Entretanto, dependendo da tarefa que se deseje realizar é preciso saber, inicialmente, qual o tipo de visão que melhor se encaixa para o cumprimento da mesma. Nesse sentido, o objetivo deste capítulo é mostrar a importância da visão para humanos virtuais e como o tópico visão pode ser entendido e abordado. Para isso, faz-se necessário entender como é o processo de representação de informações e sua utilização por agentes virtuais em ambientes simulados (seção 2.1), bem como distinguir entre visão real e visão sintética (seção 2.2).

2.1 Agentes Virtuais

A sub-área de “Humanos virtuais” pode ser considerada um tópico específico de pesquisa científica na área de realidade virtual que trata da simulação de entidades gráficas com forma humana no computador [12]. No presente trabalho, os termos agentes virtuais, agentes sintéticos ou avatares serão utilizados indiferentemente para sua representação e os termos olho e câmera possuem o mesmo significado.

Como citado no Capítulo 1, o interesse em pesquisas relacionadas a humanos virtuais é grande devido as suas inúmeras aplicações. Um dos tópicos de pesquisa muito abordado é a adição da capacidade sensorial de **visão** em agentes virtuais. Um dos grandes desafios na adição da percepção visual é justamente o controle da atenção (*gaze control*) do agente [32]. Controlar a atenção resume-se basicamente a realizar o controle da coordenação dos movimentos da cabeça (olhos e pescoço) do agente, para fixar os olhos num ponto de interesse ou objeto do ambiente virtual. Para isso, a utilização de técnicas de visão sintética é uma das perspectivas encontradas hoje para o controle da atenção destes agentes [41, 43, 28, 32, 7, 2].

A utilização de técnicas baseadas em visão sintética favorece a que todos estes movimentos possam ser realizados com maior rapidez [12, 11], pois todos os cálculos matemáticos necessários para o controle da atenção, que geralmente são utilizados por plataformas robóticas reais, podem ser ignorados no ambiente simulado.

2.2 Visão

A visão é sem dúvida o mais importante e poderoso sentido sensorial existente em organismos biológicos e também o mais complexo [20, 17]. Através dela podemos inferir informações contidas no ambiente, podemos nos locomover por ambientes estáticos ou dinâmicos, evitando a colisão com objetos. Neste trabalho, diferenciamos a visão em duas abordagens (ou tipos), em relação as complexidades computacionais associadas as mesmas: 1) visão real e 2) visão sintética.

2.2.1 Visão Real

Em relação aos problemas computacionais enfrentados por plataformas robóticas reais, pode-se dizer que a visão real está relacionada com o tratamento de imagens obtidas do mundo real, objetivando extrair alguma informação útil para o cumprimento de uma tarefa, ou seja, o principal problema é a imensa quantidade de cálculos que devem ser realizados nas imagens

capturadas, para a obtenção de informação útil. Estes cálculos, geralmente, são considerados uma barreira para as tarefas que precisam ser executadas em tempo real [20].

2.2.2 Visão Sintética

Embora o objetivo de fornecer a capacidade sensorial visual para um agente virtual seja, basicamente, o mesmo que para um agente real, a visão utilizada por ambos é diferente [12]. Isto por que a visão do agente virtual é uma visão sintética, ou seja, é uma visão simulada [12, 41, 40, 51, 28, 7, 43, 2, 8].

O uso da visão sintética traz alguns benefícios, em relação a quantidade de cálculos necessários para a execução de uma tarefa, pois algoritmos e estruturas de dados provenientes da computação gráfica [13] (por exemplo, *Z-buffer* e *octree*) podem ser utilizados pelos mecanismos visuais do agente. Outra propriedade da visão sintética é que o ambiente pode ser modelado com características específicas (padrões de cor) para favorecer a extração de informação visual para o cumprimento de uma tarefa [28, 7, 8].

O *Z-buffer* pode ser usado para inferir a informação de profundidade do ambiente [40, 22, 51, 43]. Este conhecimento pode ser utilizado para fornecer a distância entre objetos [40, 22, 43] ou como um dos parâmetros para o controle dos movimentos dos olhos do agente virtual [51].

A *octree* pode ser utilizada como uma estrutura de memória de longa duração [41] para construir, por exemplo, um mapa do ambiente. Este mapa pode ser usado pelo agente como um guia para a navegação autônoma no ambiente, evitando a colisão com os objetos inseridos no mesmo [41, 40].

Técnicas baseadas na visão sintética de agentes virtuais, inseridos em um ambiente simulado, podem usufruir das técnicas da computação gráfica para obter mais realismo em seus ambientes e fazer com que algumas tarefas possam ser executadas com maior rapidez [12].

2.2.3 Modelo de Câmera

Um agente virtual utiliza câmeras sintéticas para visualizar o ambiente. Em outras palavras, uma cena 3D pode ser vista a partir de qualquer posição do espaço tri-dimensional, quando uma câmera sintética é posicionada e orientada neste espaço de coordenadas 3D. Esta câmera possui a funcionalidade de descrever a visão sintética do agente e, de acordo com a posição e a direção dos olhos do mesmo, a câmera mostra o conteúdo da cena (através de imagens) que está sendo visualizada. Um modelo de câmera virtual é mostrado na Figura 2.1.

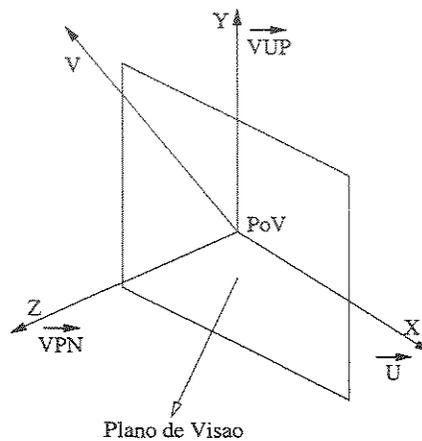


Figura 2.1: Modelo de câmera virtual.

A área de visão é definida pelo plano de visão. A origem deste plano (O_x, O_y, O_z), a qual define a posição do olho ou da câmera do observador é chamada de ponto de visão (PoV) e o vetor unitário neste plano é chamado de vetor normal ao plano (\vec{VPN}). Outro vetor unitário perpendicular a este plano é o "View Up Vector" (\vec{VUP}), o qual define a direção do ponto de visão, caso $\vec{VUP} = (0, 1, 0)$ a direção é para cima (isto significa que a imagem aparece normal), caso $\vec{VUP} = (0, -1, 0)$ a imagem aparece invertida.

A câmera virtual possui o seu próprio sistema de coordenadas, o qual é definido pelos vetores unitários \vec{VPN} , \vec{VUP} e \vec{U} ($\vec{U} = \vec{VPN} \times \vec{VUP}$, onde \times denota produto vetorial). Estes vetores são ortogonais entre si. O ponto de interesse ($P_i x, P_i y, P_i z$), que diz para onde o observador deve olhar, é colocado no referencial da câmera de acordo com as transformações

mostradas pela equação 2.1:

$$V = T.R_x.R_y.R_z.F \quad (2.1)$$

Para que o ponto de interesse seja colocado no centro do referencial de câmera, é necessário que este ponto seja transladado para a origem deste referencial (matriz T) e depois é necessário alinhar o referencial da câmera com sistema de referencial global. Para isso, um conjunto de transformações de rotação (R_x, R_y, R_z) torna-se necessário e, no final do processo de transformações, é necessário mudar o sinal da coordenada Z (matriz F) para que o eixo Z da câmera fique no sentido contrário do eixo Z do sistema global (mais informações sobre modelo de câmera podem ser encontradas na literatura a respeito [13]).

A equação 2.1 também pode ser escrita na forma matricial abaixo:

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \begin{bmatrix} R_{x1} & R_{y1} & R_{z1} & 0 \\ R_{x2} & R_{y2} & R_{z2} & 0 \\ R_{x3} & R_{y3} & R_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

com, $T_x, T_y, T_z = (P_x - O_x, P_y - O_y, P_z - O_z)$. $R_z = \frac{V\vec{PN}}{\|V\vec{PN}\|}$, $R_y = \frac{V\vec{UN}}{\|V\vec{UN}\|}$ e $R_x = R_y \times R_z$.

Ao final destas transformações, a direção do olho do observador estará na mesma direção do ponto de interesse desejado.

2.2.4 Trabalhos Relacionados

Alguns trabalhos de pesquisa utilizam o conceito de visão sintética para fornecer técnicas de percepção visual para os seus agentes virtuais. Renault et al [43] propôs um sistema baseado em visão sintética para animar o seu agente. O principal objetivo do sistema de visão era permitir que o agente se locomovesse autonomamente por um corredor, evitando a colisão com outros agentes sintéticos. A cena é renderizada a partir do ponto de visão do agente e a saída do

sistema de visão é armazenada em um vetor 2D. Cada elemento do vetor contém a cor do pixel, no qual o campo de visão do agente é projetado. Cada pixel representa a distância, a partir do olho do agente, para o ponto de observação e um identificador do objeto (para qualquer objeto que esteja naquela posição). A informação de distância (profundidade) é obtida pelo *Z-buffer*. Com este conhecimento visual do ambiente, o agente é capaz de se locomover sem colidir com outros agentes sintéticos ou objetos inseridos no ambiente.

Estudando o problema de navegação autônoma de agentes virtuais, Norser et al [41] estendeu o seu trabalho inicial [40], adicionando mecanismos de memória e aprendizado para a animação dos seus agentes. Inicialmente, foi considerado que o problema de navegação é dividido em duas partes: navegação global e navegação local. O algoritmo para navegação global guarda um modelo do ambiente, baseado em um mapa de aprendizado simplificado, para realizar um planejamento do caminho de navegação. Este mapa não reflete mudanças ocorridas recentemente. Para tratar estas mudanças, o algoritmo para navegação local recebe a entrada diretamente do ambiente para alcançar objetivos e sub-objetivos fornecido pela navegação global e para evitar a colisão com obstáculos inesperados. A representação interna do ambiente (mapa de aprendizado), para a navegação global, é construída com uma "octree". E assim como em Renault et al [43], a navegação local é baseada no *Z-buffer*.

Kuffner et al [28], tendo como base as idéias de Norser et al [41], desenvolveu um sistema baseado em percepção para a navegação autônoma de agentes virtuais. A sua principal contribuição foi o desenvolvimento de um algoritmo para a simulação da memória e da percepção visual de um agente sintético. Assim como em Renault et al [43], a cena é renderizada a partir do ponto de visão do agente e uma cor é atribuída para cada objeto ou parte do objeto contido no ambiente. Estes objetos e as suas localizações são adicionadas incrementalmente ao modelo interno do ambiente que o agente possui. Um vetor contendo os objetos percebidos e as suas localizações é mantido e atualizado a medida que o agente explora uma parte desconhecida do ambiente sintético, isto provê um tipo de memória espacial para cada agente. Da mesma forma que nas abordagens anteriores, este método armazena a geometria do objeto e uma lista contendo um identificador e a posição dos objetos. Isto fornece uma representação rápida e compacta do mundo virtual para cada agente e com base nestas informações o agente é capaz

de navegar autonomamente pelo ambiente virtual.

Peter et al [8, 7], propôs um módulo de visão sintética para agentes virtuais, baseado nos modelos descritos em Noser et al [41] e Kuffner et al [28]. O modelo de Peter usa coloração falsa e *octrees* dinâmicas para representar a memória visual do agente. A cena é renderizada a partir do ponto de visão de cada agente e para cada objeto na cena uma única cor é adicionada. E novamente, a informação dos objetos é extraída e armazenada em um vetor que contém todas as informações necessárias do objeto e do ambiente para que o agente virtual possa utilizá-las para o cumprimento de uma certa tarefa.

Em Renault et al [43] é comentado brevemente que os olhos e a cabeça do humano virtual possuem movimentos de rotação. Porém, não é mostrado como estes movimentos são tratados ou controlados. Nos outros trabalhos [41, 40, 28, 7, 8], a visão dos agentes é utilizada para guiá-los em tarefas de exploração e navegação autônoma pelo ambiente. Porém, nada é comentado a respeito do controle da atenção destes agentes virtuais, ou seja, considera-se que não existem movimentos tanto dos olhos quanto da cabeça dos mesmos.

2.3 Discussões e Considerações

Pelo exposto nas seções acima, ressaltamos a importância da utilização de humanos virtuais com a capacidade sensorial visual e chamamos a atenção para os dois tipos de visão que podem ser utilizadas pelo agente virtual.

A visão sintética pode ser uma ferramenta extremamente útil ao agente virtual para que ele possa inferir informação do ambiente, útil para a execução de determinadas tarefas, tais como: navegação autônoma, evitando a colisão com objetos inseridos no ambiente; reconhecimento de objetos e a introdução de mais realismo em uma cena. De acordo com os trabalhos mostrados na seção 2.2.4 é possível perceber que os algoritmos desenvolvidos não abordam aspectos referentes ao controle da atenção destes agentes e por isso não podem ser usados em tarefas que necessitem que os olhos sejam fixados em regiões específicas do ambiente.

Por este motivo, acreditamos que o desenvolvimento de algoritmos que possuam a fi-

nalidade de controlar a atenção de agentes virtuais é de grande importância para facilitar ou agilizar tarefas relacionadas com reconhecimento e realismo dos movimentos.

Capítulo 3

Controle da Atenção: Vergência e Disparidade

O capítulo 2, mostra que a maioria dos algoritmos desenvolvidos inicialmente (envolvendo a adição do sentido visual para humanos virtuais) têm como objetivo fornecer um mapa visual do ambiente para o agente, para que ele possa navegar autonomamente sem colidir com os objetos que aparecem no seu caminho. Entretanto, estes algoritmos não abordam aspectos referentes ao controle da atenção (por exemplo, os movimentos da cabeça) dos mesmos, o que os torna muito ineficientes para tarefas que exijam a fixação dos olhos do agente em objetos ou regiões de interesse específicas do ambiente.

Neste capítulo, são abordados dois tópicos importantes (vergência e disparidade) referente ao controle da atenção de agentes virtuais. O objetivo é mostrar como os processos atencionais podem ser controlados, na maioria dos casos. Para isso, faz-se necessário entender alguns conceitos referentes aos diferentes tipos de atenção que podem ser usadas pelo agente virtual (seção 3.1), bem como o problema da vergência (seção 3.2), incluindo algumas técnicas que podem ser utilizadas para controlar os movimentos de vergência. Ainda, uma discussão acerca do problema da disparidade, visto que esta técnica é uma das mais utilizadas para o controle da vergência, é necessária (seção 3.3). Apresentamos os principais trabalhos que utilizam técnicas de vergência, para controlar a atenção de humanos virtuais (seção 3.4), discutindo-os.

3.1 Atenção

A atenção pode ser definida como a habilidade de selecionar um tópico de interesse, ou um objetivo, e fazer com que os processos atencionais "sintonizem" neste objetivo, para extrair informação útil na execução de uma tarefa [20]. No presente trabalho, são considerados apenas tópicos de interesse reais, ou seja, objetos (imagens) contidos no ambiente do agente virtual. Dentro deste contexto mais específico, um tópico de interesse passa a ser então uma região ou um ponto de interesse do ambiente.

O processo atencional está dividido, basicamente, em duas categorias: 1) a mudança do foco de atenção e 2) a fixação dos olhos em uma determinada região do ambiente. Neste trabalho, a habilidade de mudança do foco de atenção não será tratada. O problema aqui abordado é o de fornecida uma região de interesse, passada previamente para o agente virtual, o processo atencional deve fixar os olhos do agente naquela região. A atenção também pode ser classificada sob diferentes aspectos:

- **Quanto ao Sentido:** geralmente, considera-se que a atenção pode ocorrer de duas formas diferentes: *bottom-up* (baseada na imagem) ou *top-down* (baseada na tarefa) [20, 25]. A atenção *bottom-up* ocorre acasionalmente, por exemplo, um indivíduo que se encontra dirigindo numa estrada tem o seu foco de atenção mudado quando ele é surpreendido por uma placa. Já a atenção *top-down*, o indivíduo é guiado internamente em direção a um estímulo particular, dependendo da tarefa ou da interpretação da tarefa sendo executada, por exemplo, quando ele está procurando algum objeto no ambiente.
- **Quanto a Localização no Mapa Sensorial Visual:** a atenção pode ser encoberta (*covert*) ou descoberta (*overt*) [31, 20]. Na atenção descoberta, o objetivo é colocado no centro do mapa sensorial visual, para que alguma informação seja extraída, sendo geralmente realizados movimentos físicos para isso (sacádicos [3]). Na atenção encoberta, não são realizados movimentos físicos e o alvo atencional pode se situar em qualquer lugar do mapa sensorial em questão. Em se tratando de um sistema sensorial visual, a ocorrência do primeiro tipo de atenção (descoberta) seria uma situação ideal, pois naquela posição central a resolução é maior, a exemplo do que acontece na fóvea [20].

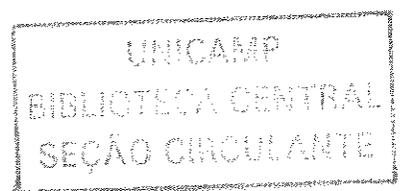
No presente trabalho, a localização visual é baseada na atenção descoberta, isto é, o objetivo é colocar o centro da imagem no centro dos olhos do agente virtual. O motivo desta escolha é que o objeto em questão pode ser visualizado com maior clareza. Com isso, mais realismo pode ser inserido em uma cena virtual ou um conjunto melhor de características pode ser extraído, caso se pretenda reconhecer algum objeto do ambiente. Entretanto, para que o ponto de interesse seja colocado no centro dos olhos, torna-se necessário controlar os ângulos de vergência de cada olho.

3.2 Vergência

O sistema visual humano é tão bom em localizar informações (por exemplo, objetos de interesse) no mundo que nos rodeia que, geralmente, não damos muita importância às restrições físicas em que ele opera [3, 4].

Os humanos possuem uma visão "foveativa" [3, 20, 4]. A fóvea (centro da retina) é a região do olho humano que possui a maior concentração de fotorreceptores (alta resolução), mais do que na periferia dos olhos [3, 4]. Em outras palavras, para ver um objeto claramente é necessário que os olhos sejam movidos na direção do objeto, de maneira que a imagem do objeto seja colocada diretamente na fóvea. Um modelo para a fóvea foi proposto [9, 10], caso exista a necessidade de extração de características. Obs: o reconhecimento de objetos não foi abordado neste trabalho, o modelo da fóvea é para trabalhos futuros.

O movimento dos olhos não são suaves [27]. Eles são compostos por saltos ligeiramente rápidos (movimentos sacádicos), acredita-se que a importância do controle destes movimentos está associada com a detecção de objetos para a extração de características [32, 19]. Os movimentos sacádicos, também, re-orientam os olhos de modo que outra cena seja projetada na fóvea. Após cada sacádico, existe um pequeno período de fixação, onde os olhos se encontram relativamente estabilizados. Normalmente, os olhos se movem em conjunto realizando deslocamentos iguais [29, 3, 46]. Para objetos próximos, é necessário que os olhos sejam rotacionados, um em direção ao outro ou na mesma direção, de modo que o objeto seja colocado na fóvea de



ambos os olhos. Este movimento de rotação realizado pelos olhos é denominado vergência [3], o qual é capaz de determinar unicamente a posição do ponto de fixação ou a região de interesse dos olhos no espaço [46]. A Figura 3.1 mostra a situação de vergência.

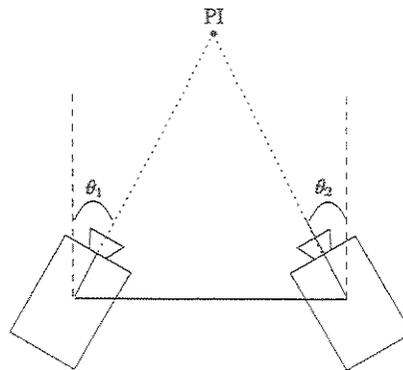


Figura 3.1: Situação de vergência. O ponto de interesse(PI) é colocado no centro óptico das duas câmeras. θ_1 e θ_2 são os ângulos de vergência.

3.2.1 O Problema da Vergência

Geometricamente falando, a função do sistema de vergência é controlar os ângulos de vergência de acordo com a distância das câmeras até o ponto de fixação, ao longo da direção específica de algum olhar fixo no ambiente (Figura 3.1). Em alguns casos, a motivação para a utilização da vergência é manter o ponto de fixação próximo de algum objeto do ambiente, com o objetivo de reconhecê-lo [27, 20]. Em outras situações, a vergência é utilizada para produzir movimentos mais realísticos nos olhos e na cabeça do agente virtual, inserido num ambiente simulado [51, 32].

Como os ângulos de vergência estão diretamente relacionados com a distância do ponto de fixação [27, 46, 1], o problema da vergência pode ser definido da seguinte maneira: a tarefa de controlar os ângulos de vergência de acordo com a distância do ponto de fixação, para manter ambos os olhos fixados em algum lugar alvo (ponto de interesse) do ambiente, de modo que o ponto de interesse coincida com o centro dos olhos. O controle da vergência é apenas um dos aspectos relacionados com o grande problema do processo atencional [27, 20].

Geralmente, a fixação é conseguida trabalhando-se em níveis, ou seja, primeiro é realizada uma fixação grossa e depois ajustes mais finos são realizados para colocar ambos os olhos no ponto de interesse correto [26]. O processo de fixação pode ser dividido, basicamente, em duas categorias, conforme o movimento das câmeras [26, 20, 5]:

- **Ponto de fixação em ambas as câmeras:** nesta situação, o ponto de interesse é passado simultaneamente para as duas câmeras e o processo de fixação pode ocorrer em paralelo. Pois, a informação do ponto de interesse é suficiente para que ocorra uma fixação simultânea. Neste caso, deve-se garantir que o mesmo ponto objeto seja passado, para que o resultado seja ambas as câmeras vergidas num único ponto no espaço.
- **Ponto de fixação em uma câmera e o cálculo da disparidade:** esta situação é similar a anterior. A única diferença é que não existe redundância de informação. Primeiro uma das câmeras é fixada no ponto de interesse e depois, através da minimização do erro da disparidade [20], a outra câmera é deslocada para o ponto de interesse. Nesta situação, existe o conceito de olho dominante, isto é, o movimento de uma das câmeras é dependente ou está subordinado ao movimento da outra câmera.

A vergência também adiciona muitas vantagens para os sistemas baseados em visão, tanto para sistemas reais (visão real) quanto para sistemas simulados (visão sintética) [27, 51, 30, 32], tais como:

- **Tarefas de reconhecimento:** fixar num objeto de interesse e fazer com que os pontos deste objeto coincidam com o eixo ótico da câmera, pode favorecer a extração de características deste objeto. Pois, os detalhes do objeto podem ser vistos com maior clareza, ao exemplo do que acontece na fóvea.
- **Facilita o casamento estéreo:** por definição, o ponto de fixação possui disparidade zero [27]. Isto torna possível a utilização de algoritmos estéreos que aceitam, apenas, um pequeno intervalo de valores de disparidades, isto agiliza o processo da reconstrução estéreo. Este pequeno intervalo de busca, também reduz a ambigüidade do casamento, por que candidatos ao casamento fora deste intervalo podem ser desconsiderados.

- **Aumento do realismo em ambientes virtuais:** é possível saber, de acordo com a direção dos olhos do agente virtual, para onde o avatar está olhando. Isto pode favorecer uma comunicação não verbal ou uma interação social mais amigável entre os participantes de um ambiente virtual.

Neste trabalho, o ponto de interesse é passado para as duas câmeras e o processo de fixação ocorre simultaneamente, isto é, os olhos do avatar possuem movimentos de vergência independentes. A independência dos movimentos agiliza o processo de fixação dos olhos, pois o processo de fixação pode ocorrer em paralelo.

3.2.2 Técnicas para o Controle da Vergência

O controle da vergência consiste em rotacionar as câmeras, de modo que o eixo ótico das mesmas se encontrem no ponto de interesse [27] (Figura 3.1). Assim, a efetivação da vergência pode ser obtida, basicamente, por três tipos diferentes de técnicas [46, 30, 20, 29]:

- **Correlação:** a motivação para o uso desta técnica vem do fato de que se a vergência em um objeto está correta, então as imagens estéreo deste objeto devem ser muito similares, pelo menos ao redor do ponto de fixação. Neste caso, considera-se que as imagens estão fundidas binocularmente. O objetivo do sistema que controla a vergência, é minimizar esta diferença. Geralmente, cálculos envolvendo operações de convolução, (por exemplo, correlação cruzada normalizada [38, 24]) são utilizadas para minimizar o erro.
- **Rastreamento:** nesta técnica, assume-se que a vergência ocorre naturalmente quando estamos acompanhando a trajetória de um objeto. Para que ocorra a vergência, é necessário manter um *template* muito similar para ambos os olhos. Isto pode ser obtido utilizando-se um olho dominante que tem a função de passar o seu *template* para o olho secundário. Este processo mantém a similaridade entre os dois *templates* e garante que os dois olhos estão olhando para o mesmo objeto e é necessário manter uma constante atualização dos *templates* até que os olhos estejam vergidos no objeto desejado.

- **Disparidade:** este último grupo trata de estimar a disparidade binocular (horizontal, vertical e/ou rotacional) entre as duas câmeras, alguns trabalhos consideram que a disparidade horizontal é a mais importante [46, 29]. A disparidade binocular está relacionada com a informação de profundidade, a qual está relacionada diretamente com os ângulos de vergência. Esta técnica usa a informação da disparidade para controlar diretamente os ângulos de vergência. Por controlar diretamente estes ângulos, esta técnica é uma das mais utilizadas para o controle da vergência.

Um grande problema em utilizar estas técnicas é que todas elas necessitam que muitos cálculos matemáticos sejam realizados, para que os olhos do agente virtual possam ser direcionados e fixados na região de interesse.

3.3 Disparidade

A definição de disparidade está relacionada com o problema da reconstrução estéreo, ou seja, é necessário estabelecer primeiro quais os pixels das imagens das câmeras esquerda e direita que correspondem ao mesmo ponto no espaço [46, 20].

3.3.1 O Problema da Disparidade

Obter a disparidade entre as imagens de duas câmeras é um processo relativamente simples, porém custoso: é necessário estabelecer, para todos os pontos, quais pontos na imagem da câmera direita que correspondem aos pontos da imagem da câmera esquerda, os quais mapeiam nos mesmos pontos no espaço (Figura 3.2). Infelizmente, esta não é uma tarefa fácil e para se obter uma solução, muitas vezes é necessário resolver o problema da reconstrução estéreo de forma completa.

Na técnica tradicional de reconstrução estéreo, um modelo computacional da cena é construído, através de uma série de cálculos realizados sobre duas ou mais imagens, tomadas

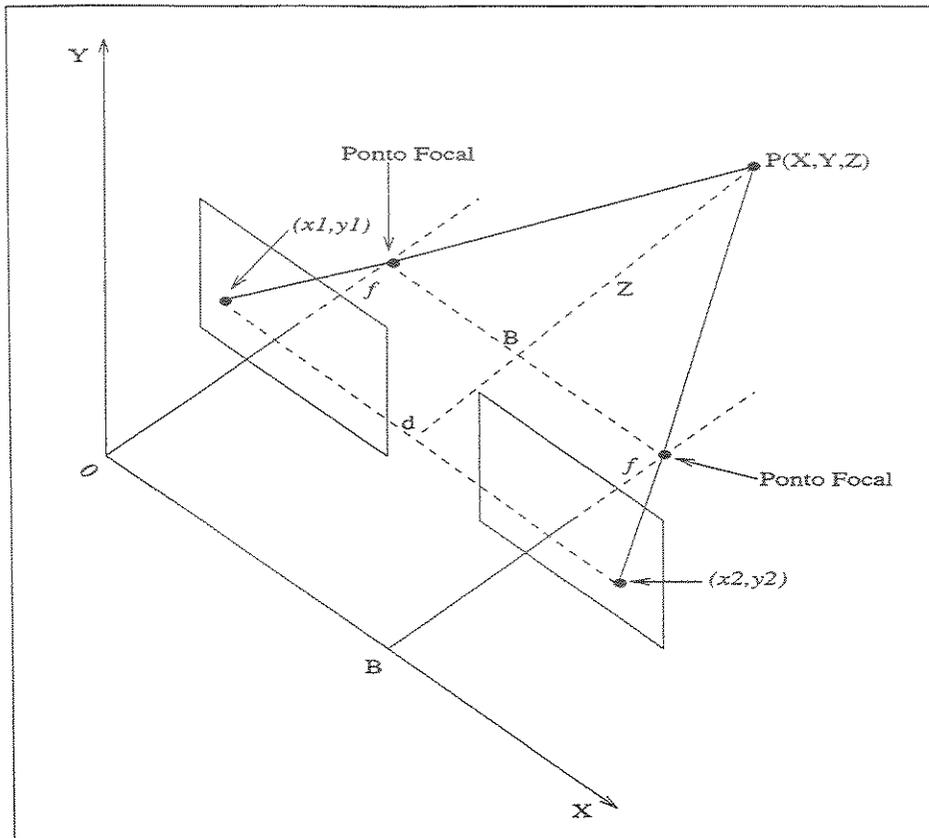


Figura 3.2: Modelo simples para a obtenção da reconstrução estéreo. Um ponto típico $P(X, Y, Z)$ é projetado em dois planos imagens (esquerdo = 1 e direito = 2). Por conveniência de representação, os planos imagens estão rotacionados de um valor angular de π *radianos* em torno dos pontos focais.

de pontos de vista diferentes de uma mesma cena [21, 24, 6]. A Figura 3.2 descreve um modelo simples para a obtenção da reconstrução estéreo.

O parâmetro B , denominado linha base (ou simplesmente base) é a distância entre os centros focais ou o centro de perspectiva das câmeras. O parâmetro f (distância focal), cujo valor é o mesmo para as duas câmeras, é a distância entre cada plano imagem e os pontos focais. Os sistemas de coordenadas da imagem esquerdo e direito são (x_1, y_1) e (x_2, y_2) e suas origens estão nas coordenadas (em sistema global) $(0, 0, f)$ e $(B, 0, f)$ respectivamente. Usando similaridade de triângulos as seguintes equações de projeção perspectiva são obtidas:

$$x_1 = \frac{fX}{Z}, x_2 = \frac{f(X+B)}{Z}, y_1 = y_2 = \frac{fY}{Z} \quad (3.1)$$

Definindo a disparidade como $d = x_2 - x_1$ (devido ao modelo ideal da Figura 3.2, esta só é definida para o eixo X), a partir das Equações 3.1 as seguintes equações de inversão de perspectiva podem ser obtidas:

$$X = \frac{Bx_1}{d}, Y = \frac{By_1}{d}, Z = \frac{Bf}{d} \quad (3.2)$$

se tivermos a disparidade d determinada para todos os pontos da imagem, a profundidade em cada ponto pode ser calculada por simples similaridade entre triângulos.

Geralmente, o controle da vergência é baseado em correção de erros e o que se considera é a disparidade d para cada píxel das duas imagens, ou seja, as equações 3.1 e 3.2 podem ser descartadas e apenas a equação da disparidade $d = x_2 - x_1$ é considerada. Entretanto, a obtenção da disparidade (correspondência entre os píxels das duas imagens) envolve cálculos massivos de correlação. Por este motivo, o processo estéreo é geralmente considerado o gargalo para qualquer sistema de visão ativa.

3.4 Trabalhos Relacionados

A quantidade de trabalhos envolvendo o controle da atenção de humanos virtuais, usando técnicas de vergência, não são muitos [18, 19]. E não foi encontrada nenhuma referência a respeito de trabalhos que utilizaram alguma técnica de visão sintética, para controlar a atenção de agentes virtuais.

Pesquisando o problema de visão animada para criaturas virtuais, Terzopoulos et al [16] propôs um algoritmo para animar a visão de peixes artificiais inseridos em ambientes virtuais. Algumas das funcionalidades deste algoritmo foram modificadas (por exemplo, foram inseridos mais graus de liberdade para controlar os movimentos independentes, horizontal e vertical, dos olhos e do pescoço do humano virtual) posteriormente por Rabie et al [18, 19], para controlar a

atenção de humanos virtuais. O controle da atenção começa quando uma região de interesse é detectada na periferia do campo visual, isto é feito por uma técnica baseada na intersecção do histograma. Logo em seguida, o ponto de fixação é então localizado, utilizando-se o método *histogram backprojection*. O objetivo deste método é informar onde estão as cores dos pixels, na imagem, que correspondem ao objeto que está sendo observado. Depois que o ponto de interesse é detectado, os olhos realizam movimentos sacádicos, isto é, os ângulos iniciais de cada olho vão sendo incrementados gradativamente com base na informação dos pixels obtida pelo método *histogram backprojection*, para trazer o ponto de interesse para a fóvea. Durante o processo de foveação, cada olho é controlado independentemente, quer dizer, até este ponto os olhos ainda não estão "vergados" no ponto de interesse. A vergência é conseguida através do cálculo da disparidade das imagens que foram colocadas na fóvea, dos olhos esquerdo e direito. Primeiro o olho direito (olho dominante) é fixado no objeto e através da correção de erros da disparidade, utilizando-se cálculos de correlação, o olho esquerdo é deslocado até o ponto de fixação (durante o processo final de vergência, quando são realizados os cálculos de correlação, os olhos possuem movimentos dependentes). Neste momento, o processo de vergência é finalizado e os olhos são fixados no ponto de interesse.

O algoritmo proposto por Rabie et al é muito interessante do ponto de vista de realismo no controle dos movimentos atencionais do humano virtual. Pois, são tratados os movimentos independentes, horizontal e vertical, dos olhos e da cabeça do agente. Entretanto, não está claro em nenhum dos seus trabalhos como é feita a coordenação desses movimentos, ou seja, quem deve se movimentar primeiro (olho ou pescoço) e em que condições ou restrições estes movimentos devem ser executados. Embora o objetivo do algoritmo proposto seja para ser utilizado em humanos virtuais munidos de visão sintética, nenhuma propriedade gráfica do ambiente é utilizada para facilitar ou agilizar o controle da atenção do agente virtual. Para a correta execução dos movimentos de vergência, assim como em sistemas reais de visão, é necessário a utilização de cálculos de correlação.

O algoritmo de vergência proposto nesta dissertação, para controlar a atenção de agentes virtuais, também considera movimentos independentes (horizontal e vertical) tanto dos olhos quanto do pescoço do agente. E tendo como base algumas restrições físicas de deslocamento,

o algoritmo é capaz de coordenar os movimentos da cabeça do agente, isto é, dependendo da localização do ponto de interesse três situações podem ocorrer durante o processo de fixação: 1) apenas o pescoço se movimentada; 2) o pescoço é movimentado primeiro e em seguida os olhos são movimentados para correta fixação no ponto de interesse e 3) apenas os olhos são movimentados.

O controle da vergência está relacionado diretamente com a percepção de profundidade do ambiente [46]. Portanto, decidimos utilizar a informação proveniente do *Z-buffer*, para a correta coordenação dos movimentos da cabeça do agente. A informação do *Z-buffer* é suficiente para que os olhos sejam fixados no ponto de interesse desejado, isto é, cálculos de correlação não são necessários.

3.5 Discussões e Considerações

Neste capítulo foram abordados assuntos referentes ao controle da atenção de agentes virtuais. A técnica mais utilizada para a obtenção deste controle é o algoritmo de vergência. Geralmente, técnicas baseadas em medidas de correlação, rastreamento e disparidade são utilizadas para a obtenção dos parâmetros de entrada do algoritmo. Dentre estas três técnicas, a disparidade é a mais utilizada

O principal motivo para a uso da disparidade é por que ela está relacionada com a informação de profundidade, a qual está relacionada diretamente com os ângulos de vergência. A grande dificuldade para a utilização desta técnica está associada com o problema da correspondência estéreo, onde muitas vezes cálculos maciços de correlação devem ser usados para a obtenção dos parâmetros da disparidade.

Por outro lado, quando estamos executando tarefas em ambientes simulados que necessitam do controle da atenção de humanos virtuais, todos os cálculos associados para a obtenção da disparidade podem ser evitados. Pois, estes parâmetros podem ser estimados com a utilização de técnicas de visão sintética, como por exemplo o *Z-buffer*.

O *Z-buffer* pode ser utilizado para a obtenção do parâmetro de profundidade, o qual pode

ser empregado diretamente para controlar os ângulos de vergência. Até o presente momento, não existem trabalhos baseados nesta técnica, ou seja, que utilizam a informação do *Z-buffer* para controlar a atenção de humanos virtuais.

Capítulo 4

Arquitetura do Sistema

A descrição funcional da arquitetura proposta, utilizada para o controle da atenção de agentes virtuais, é mostrada neste capítulo. As funcionalidades da arquitetura (Figura 4.1) são bem mais gerais, ou seja, aspectos referentes a aprendizado, reconhecimento e mudança de atenção também são inseridos nesta arquitetura. Porém, o escopo deste trabalho engloba, apenas as partes da arquitetura responsáveis pelo algoritmo de vergência, o qual é o responsável pela coordenação dos movimentos da cabeça do humano virtual. E também é feita uma descrição mais detalhada das funcionalidades da arquitetura do sub-módulo de vergência (Figura 4.3), pois este é o principal assunto desta dissertação.

4.1 Descrição dos Módulos da Arquitetura

Denominamos Câmbio o sistema resultante para simulação do ambiente virtual, humanóide e sistema de visão. Ele foi desenvolvido baseado em uma arquitetura modular, mostrada na Figura 4.1. Outra característica utilizada, é a habilidade da troca de um módulo por outro contendo uma implementação diferente, sem a necessidade de modificar outras partes da arquitetura. Atualmente, a arquitetura do Câmbio está dividida em cinco módulos principais: módulo do ambiente, módulo de percepção visual, agente virtual, módulo de decisão e módulo de controle.

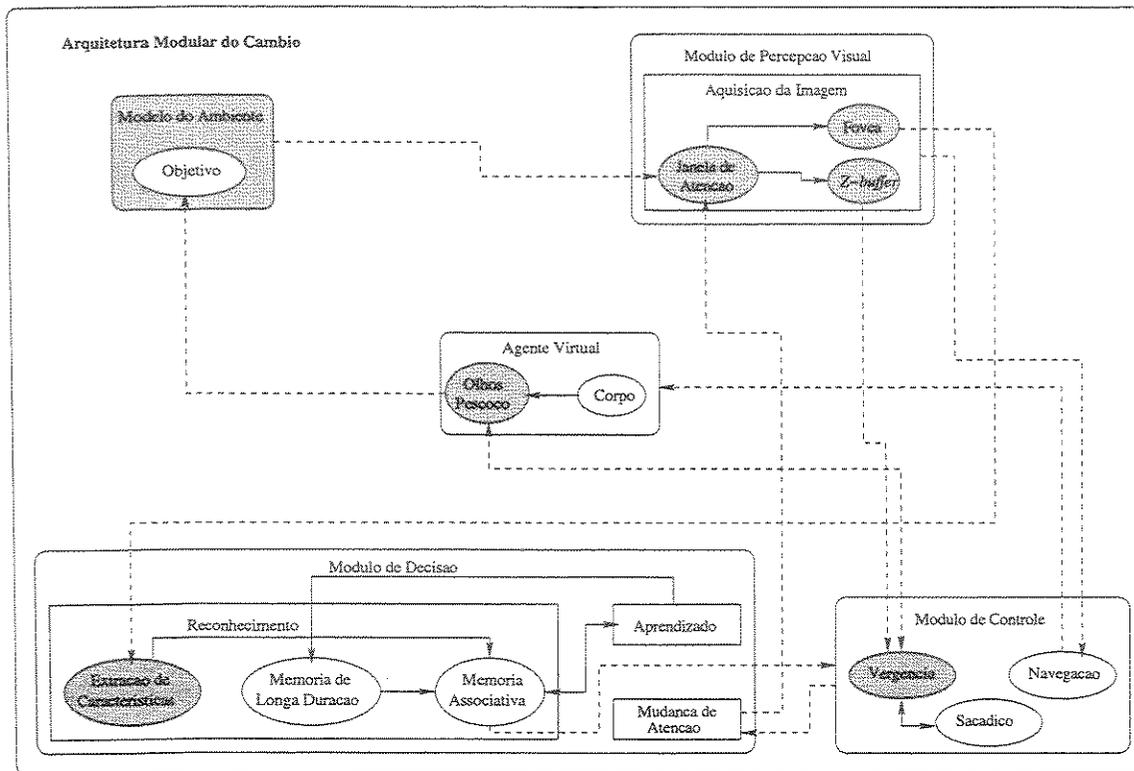


Figura 4.1: Arquitetura modular do Câmbio. As linhas tracejadas representam a comunicação externa entre os módulos e as linhas sólidas a comunicação interna. Os módulos em cinza são os que foram implementados até o momento.

4.1.1 Modelo do Ambiente

Na arquitetura do Câmbio, é utilizado um modelo geométrico simplificado para a criação do ambiente virtual. Objetos simples (esferas, cubos, etc.) são utilizados para o cumprimento de tarefas, por exemplo, servindo como alvo de interesse para o direcionamento e posterior fixação dos olhos do agente virtual.

Outra função deste módulo é fornecer informação sensorial (gerando e renderizando uma cena em OpenGL) a medida que esta é requisitada pelo módulo de percepção visual. Em outras palavras, o ambiente é modelado de acordo com as tarefas que necessitam ser desenvolvidas.

Em resumo, este módulo fornece o modelo do mundo, onde o agente virtual está inserido, com os parâmetros necessários para que a informação proveniente do ambiente seja verificada e uma resposta sensorial possa ser gerada (de acordo com a tarefa realizada).

4.1.2 Agente Virtual

O agente virtual é considerado um módulo a parte. Isto permite que diferentes agentes, com diferentes estruturas corporais, possam ser construídos e inseridos no ambiente virtual. Outra funcionalidade deste módulo é enviar a posição dos olhos e do pescoço do agente para o submódulo de vergência.

Para os testes de validação no nosso algoritmo foi necessário construir apenas um agente simplificado, onde as funcionalidades do algoritmo pudessem ser testadas. Por isso, nosso agente virtual é composto pelas seguintes estruturas (Figura 4.2):

- **Olhos:** são responsáveis por representar a aquisição da informação sensorial visual. Possuem dois graus de liberdade, podendo executar movimentos horizontais independentes e verticais dependentes.
- **Pescoço:** é responsável por representar os movimentos da cabeça do agente virtual. Também possui dois graus de liberdade, podendo executar movimentos de rotação na horizontal e na vertical.

Apesar do nosso agente ser simplificado, o nosso algoritmo é capaz de controlar a cabeça de qualquer humano virtual que possua as mesmas estruturas do nosso agente, olhos e pescoço, os quais representam a cabeça do agente.

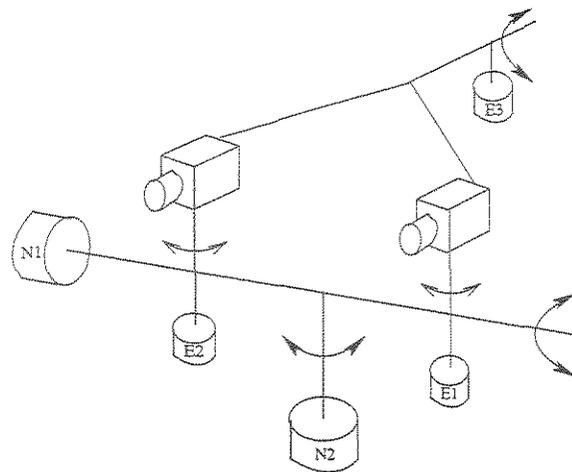


Figura 4.2: Mecânica dos movimentos dos olhos e do pescoço. (E1, E2) e (E3) são os motores responsáveis pelos movimentos horizontal e vertical dos olhos. N1 e N2 são os motores responsáveis pelos movimentos vertical e horizontal do pescoço. O movimento horizontal dos olhos é independente e o movimento vertical é dependente, ou seja, na vertical os olhos se movem juntos.

4.1.3 Módulo de Percepção Visual

Este é o módulo de visão do agente sintético, o qual faz a aquisição das imagens do ambiente virtual. A funcionalidade deste módulo é fornecer serviços, baseados em informação visual, principalmente para os módulos de decisão e controle. É utilizada uma janela de atenção, com tamanho 256x256 pixels, para fazer a aquisição das imagens para que o agente virtual possa desempenhar suas tarefas. As informações que este módulo oferece são: as imagens em multi-resolução da fóvea e a informação de profundidade (fornecida pelo *Z-buffer*).

A informação de profundidade é passada para o sub-módulo de vergência (contido no módulo de controle), para que se dê início ao processo de controle da atenção do agente virtual.

As imagens em multi-resolução são compostas de quatro imagens, com tamanhos de 32x32 pixels, geradas a partir de uma imagem de 256x256 pixels (janela de atenção). Estas quatro imagens são passadas para o sub-módulo de reconhecimento (contido no módulo de

decisão) para que seja dado início ao processo de extração de características.

4.1.4 Módulo de Decisão

Este módulo engloba os algoritmos de mais alto nível, tais como: reconhecimento, aprendizado e mudança de atenção. Geralmente, estes algoritmos são utilizados em tarefas que envolvem algum tipo de inteligência.

No sub-módulo de reconhecimento, o processo de extração de características é realizado com a utilização de operadores gaussianos, nas imagens fornecidas pela fôvea. O uso dos mesmos é provado ser útil para a extração de características visuais voltadas tanto para propósitos atencionais [33, 34, 35] quanto para reconhecimento de padrões [33, 36, 45, 39]. Estas características são utilizadas por algoritmos de mais alto nível, para a realização de alguma tarefa.

O Sub-módulo de reconhecimento também é responsável pelo próprio reconhecimento de objetos, inseridos no ambiente do agente virtual. O reconhecimento de objetos requer que a informação proveniente da etapa de extração de características seja associada a uma mesma representação em memória. O endereço desta associação pode ser fornecido por uma memória associativa [33, 34].

A representação que contiver as propriedades mais similares àquelas do padrão de entrada corrente, torna-se a mais ativa. Se o valor desta ativação estiver acima de um limiar, considera-se que o padrão corrente foi reconhecido. Caso contrário, mais informação deve ser adquirida. Se após um certo número de tentativas visando melhorar a informação de entrada, o limiar acima não tiver sido alcançado, considera-se que um objeto desconhecido foi descoberto e novas informações relativas a ele devem ser armazenadas na memória de longa duração.

Um supervisor de aprendizado é automaticamente acionado para "aprender" algumas coisas sobre a nova representação. Aqui, "aprender" significa inserir as novas características, como por exemplo um rótulo ou um índice, na memória de longa duração e treinar novamente a memória associativa, já com o novo padrão incorporado. A memória associativa específica, a

partir da representação do objeto correntemente em foco, o endereço do padrão mais próximo a ele contido na memória de longa duração. Representações do objeto, estas representações não são visuais e sim uma representação invariante, podem ser armazenadas através da forma [6, 53] ou através da textura dos mesmos [42, 23, 37, 6].

Nesta dissertação, o processo de extração de características foi implementado para a utilização em trabalhos futuros. Pois, como citado anteriormente, os processos de mais alto nível (que analisam estas características para gerar uma ação) não são o escopo desta dissertação, sendo implementados geralmente em aplicações de inteligência artificial, a serem feitas a título de trabalhos futuros.

4.1.5 Módulo de Controle

A função do módulo de controle é receber ações em alto nível, enviadas pelo módulo de decisão, ou ações de baixo nível, enviadas pelos módulos de percepção visual e agente virtual, e transformá-las em diretivas de controle de baixo nível para efetivamente controlar os atuadores do agente virtual (no caso, os motores visuais).

Este módulo é composto pelo sub-módulo de navegação, o qual é responsável pelo deslocamento do agente virtual pelo ambiente; pelo sub-módulo sacádico, responsável pelos movimentos sacádicos da cabeça do agente; e pelo sub-módulo de vergênica, o qual é responsável pelo controle da atenção do agente virtual.

As informações enviadas pelo módulo de decisão são utilizadas para mudar o foco de atenção do agente. E a informação de profundidade é transformada em deslocamentos atencionais para controlar os movimentos dos olhos e do pescoço do agente. Em outras palavras, a informação de profundidade diz para onde o agente virtual deve olhar e o Sub-módulo de Vergência é responsável pela coordenação dos movimentos da cabeça do agente, fazendo com que os olhos sejam fixados neste ponto de interesse, e o sacádico controla a velocidade destes movimentos.

4.2 Executando a Vergência

Como descrito anteriormente, o sub-módulo de vergência é responsável pela fixação dos olhos do agente virtual no local desejado. Ele está dividido de acordo com os módulos mostrados na arquitetura da Figura 4.3.

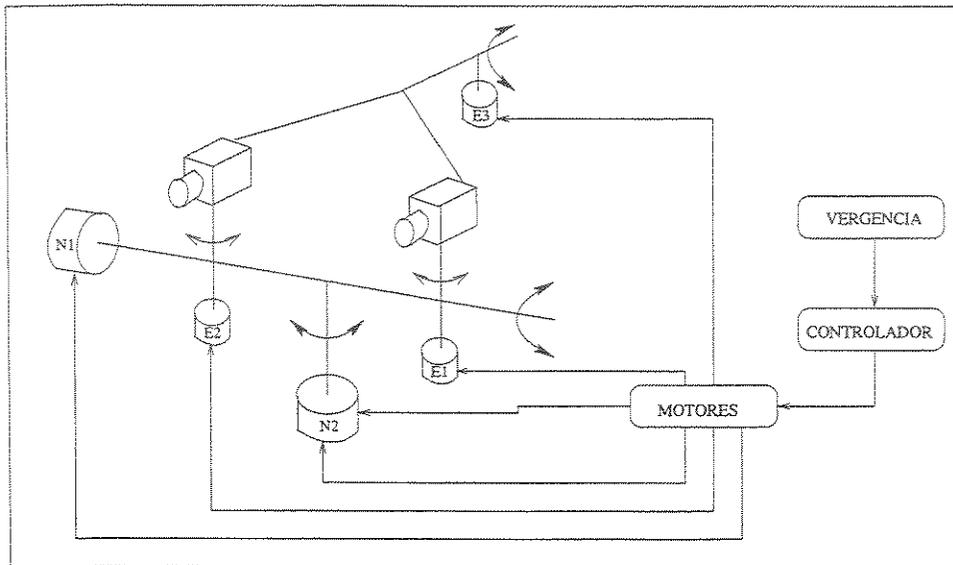


Figura 4.3: Arquitetura para o controle da atenção do agente virtual.

O módulo de vergência recebe como parâmetros de entrada a informação de profundidade do *Z-buffer* e os valores angulares da direção dos olhos e do pescoço do agente virtual e as passa para o módulo controlador.

O módulo controlador por sua vez, é responsável apenas pela coordenação dos movimentos direcionais da cabeça do agente virtual fornecendo os parâmetros de deslocamento para os motores, para que os olhos sejam fixados no ponto de interesse desejado. Este módulo decide quem vai se deslocar primeiro e quando se dará o deslocamento, de acordo com as restrições físicas dos olhos e do pescoço e da posição do novo ponto de interesse.

As informações destas restrições são transformadas em movimentos direcionais, que são passados e executados pelos motores (o movimento de cada motor é independente um do

outro). A função de cada motor é apenas movimentar os olhos e o pescoço do agente, onde a quantidade deste deslocamento é fornecida pelo controlador. O Módulo Controlador fica em um ciclo (*while*) interno até que o ponto de interesse seja colocado na fóvea de cada olho.

4.3 Considerações

Neste capítulo descrevemos as funcionalidades, tanto da arquitetura geral do sistema quanto da arquitetura específica para o controle da atenção do agente virtual. Algumas das funcionalidades (de alguns dos módulos da arquitetura) foram comentadas, mas não foram implementadas, pois estes módulos abordam problemas referentes a inteligência do agente virtual e da navegação dele pelo ambiente, os quais estão fora do escopo desta dissertação, e ficam a título de trabalhos futuros.

O agente virtual aqui descrito representa um modelo simplificado de um humano virtual, composto apenas pelos olhos e pescoço do mesmo (estas partes representam a cabeça do humano virtual). Este modelo simplificado foi suficiente para testar e validar as funcionalidades do nosso algoritmo de vergência.

Capítulo 5

Implementações Realizadas na Arquitetura

Para validar e testar computacionalmente o funcionamento do nosso algoritmo de vergência, o modelo simplificado do agente virtual intitulado Câmbio, apresentado no Capítulo 4, composto pelas partes essenciais da cabeça de um humano virtual (olhos e pescoço) foi construído.

Convém ressaltar que, apesar da plataforma de simulação ter sido desenvolvida para testar o nosso algoritmo, ela pode ser utilizada com propósitos mais gerais, como por exemplo, para testar outros algoritmos de visão computacional que tenham a finalidade de serem usados em uma cabeça estéreo. Ainda, ela pode ser usada para o desenvolvimento de modelos matemáticos de visão. Em resumo, nossa plataforma pode ser utilizada como uma ferramenta de simulação dando apoio ao desenvolvimento e ao teste de algoritmos de visão, onde não se tenha uma plataforma real para isso.

5.1 Plataforma de Simulação: Agente Virtual

Para avaliar computacionalmente nosso algoritmo de vergência, desenvolvemos uma plataforma de simulação (agente virtual), onde o algoritmo foi implementado e testado. O agente

o sintético Câmbio, cuja interface gráfica é mostrada pela Figura 5.1, possui cinco controladores (Figura 4.3) que são utilizados para controlar os movimentos do pescoço e dos dois olhos. A modelagem gráfica da plataforma foi desenvolvida em OpenGL e o controle físico dos movimentos do Câmbio foram desenvolvidos na linguagem C.

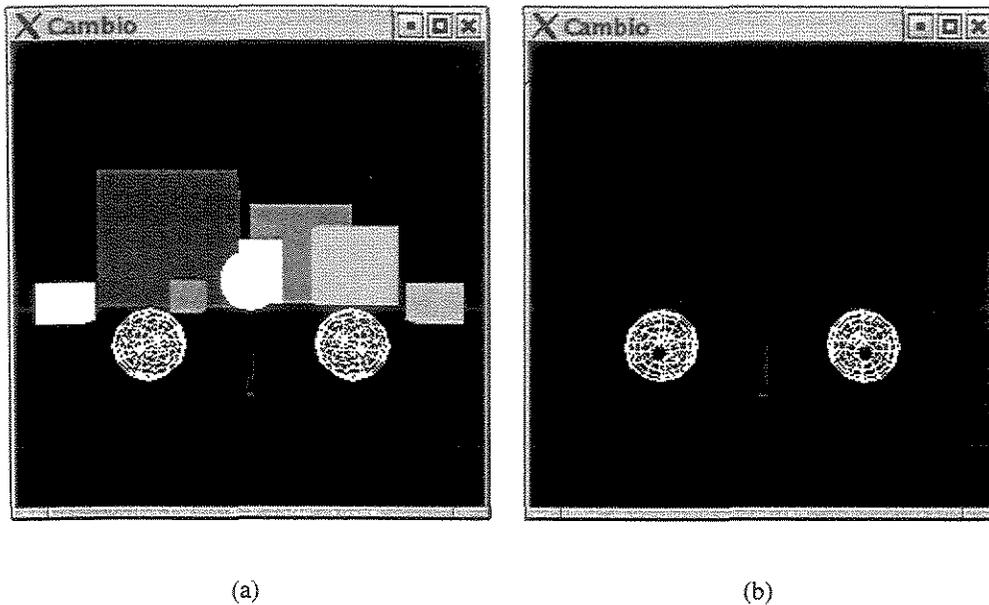


Figura 5.1: Interface gráfica. (a): mostra uma visão de trás para frente do Câmbio e alguns objetos inseridos no ambiente virtual. (b): mostra uma visão de frente para trás do nosso agente virtual.

O sistema possui três referenciais de câmeras, um em sistema global (coordenadas de mundo) e os outros dois são os referenciais dos olhos esquerdo e direito do agente. A cabeça do Câmbio possui quatro graus de liberdade, sendo estes relativos aos movimentos de pan e tilt e aos movimentos de vergência de cada olho (direito e esquerdo). O espaço de manipulação dos movimentos do "Câmbio" engloba uma circunferência completa podendo cobrir todo o ambiente (caso não sejam consideradas restrições de movimentos inseridas pelo software). Como pode ser visto na Figura 5.2, por motivos de ilustração, os movimentos de pan e tilt do pescoço estão restritos em $[-115, +115]$ pixels (os valores em pixels são transformados em deslocamentos angulares), mas o Câmbio pode mover o pescoço meia circunferência para cada lado e para baixo ou para cima (na implementação foi considerada uma restrição de 127 pixels). Movimentos para

a esquerda ou para baixo correspondem a valores negativos entre 0 e -127 e movimentos para a direita ou para cima correspondem a valores positivos entre 0 e +127. Isto permite que o agente possa olhar para qualquer lugar do ambiente.

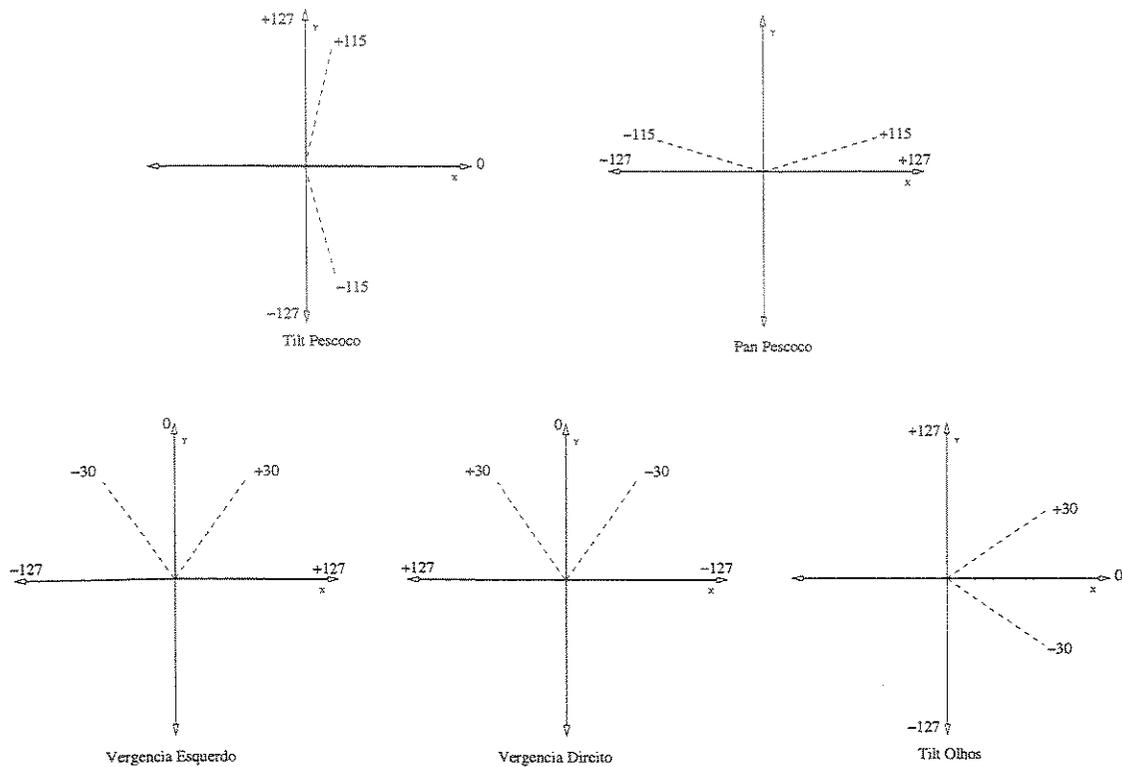


Figura 5.2: São mostrados os espaços de manipulações de cada grau de liberdade dos olhos e do pescoço do "Câmbio" e também os limites para estes determinados pela plataforma (linhas sólidas) e pelo software (linhas pontilhadas).

Os movimentos de vergência também podem cobrir meia circunferência. O agente pode movimentar os seus olhos em -127 referente a vergência interna ou mínima (olhos convergindo uma para o outro) e +127 referente a vergência externa ou máxima (olhos divergindo um do outro). Além destes limites estabelecidos pela plataforma, limites de movimento também podem ser determinados pelo software, para simular as restrições físicas dos movimentos dos olhos e do pescoço de seres humanos. É importante ressaltar que os limites estabelecidos para

os movimentos dos olhos e pescoço podem ser modificados, até que se chegue a um resultado satisfatório, na prática.

Na implementação, o agente está posicionado inicialmente nas coordenadas (0,0,0) (pescoço) e (-10,0,0) e (10,0,0) (olho esquerdo e olho direito). A linha base ou distância entre os olhos possui tamanho de 20 pixels.

5.2 Modelo da Fóvea: Imagens Multi-Resolução

Na prática, para que um operador visual de baixo nível seja aplicado em uma imagem que foi capturada, geralmente é necessário que vários cálculos de convolução sejam realizados. Muitas vezes, cada operação de convolução é realizada entre uma imagem de tamanho $n \times n$ e um kernel de tamanho p (n e p são números inteiros de pixels), isto requer $n^2 p^2$ operações de multiplicação e soma. Estas operações fornecem resultados muito demorados, especialmente quando é necessário realizar várias operações de convolução em seqüência, em uma determinada imagem. Muitas vezes, é isto que acontece com tarefas que envolvem visão de baixo nível.

O uso de imagens em multi-resolução (ou multi-escala) é uma técnica que tenta diminuir este custo de processamento. Esta técnica favorece a utilização de várias imagens com tamanhos reduzidos (ou não), as quais são imagens sub-amostradas (ou não) da original. Estas imagens são obtidas de tal forma que elas contenham em conjunto a mesma (ou um pouco menos) quantidade de energia (informação visual) contida na imagem original.

Um modelo biológico para a geração de imagens em multi-escala pode se basear no uso de derivadas gaussianas com kernels de diferentes tamanhos [47]. Neste caso, pode-se calcular as derivadas diretamente das imagens originais e mostrar os resultados em diferentes resoluções. O problema desta representação é que, dependendo do tamanho da imagem, os cálculos para a extração de características podem ser demorados.

Neste trabalho, utilizamos uma abordagem simples para a geração de imagens em multi-resolução, nosso modelo é similar ao do olho humano (também conhecido como imagens da retina), ou seja, ao exemplo do que acontece na fóvea (nesta região, tem-se imagens com alta

resolução). Para transpor esta idéia para o domínio computacional, capturamos uma imagem (janela de atenção do agente virtual, Figura 4.1) e geramos quatro novas imagens, através da redução do tamanho e da resolução da imagem original.

A obtenção destas imagens ocorre da seguinte maneira: a janela de atenção captura uma imagem de 256x256 pixels; a primeira imagem (a de menor resolução) é obtida pela convolução da imagem original com um filtro média de kernel 8x8 (para maiores detalhes sobre este tipo de filtro, ver [6]) e em seguida faz-se uma reamostragem a cada 8 pixels na imagem convoluída; para a obtenção da segunda imagem, pega-se uma área de 128x128 pixels mais central da imagem original e faz-se a convolução desta nova imagem com um filtro média de kernel 4x4 e em seguida faz-se uma reamostragem a cada 4 pixels na imagem convoluída; na terceira imagem, pega-se uma área de 64x64 pixels mais central da imagem original e faz-se a convolução desta nova imagem com um filtro média de kernel 2x2 e em seguida faz-se uma reamostragem a cada 2 pixels na imagem convoluída; e a última imagem (a de maior resolução), pega-se a área de 32x32 pixels mais central da imagem original e faz-se a convolução desta nova imagem com um filtro média de kernel 1x1. Na verdade, neste último nível, uma simples transferência de superfície de origem é executada, sem que se faça uma convolução propriamente dita. A Figura 5.3 mostra os passos seguidos pelo algoritmo.

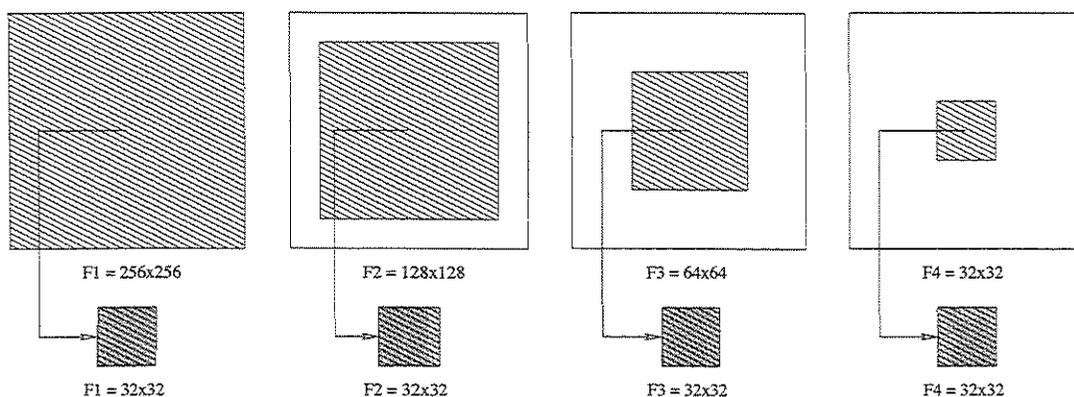


Figura 5.3: Ilustração do algoritmo para a obtenção das imagens da fóvea. Os passos 1 a 4 são representados pelas imagens da esquerda para a direita.

O processo de execução deste algoritmo termina com quatro imagens com diferentes

níveis de resolução, representando diferentes porções de informação visual contidas na imagem original. Uma aplicação deste algoritmo é mostrada pela Figura 5.4, onde a parte mais central (imagem mais a direita) aparece em alta resolução.

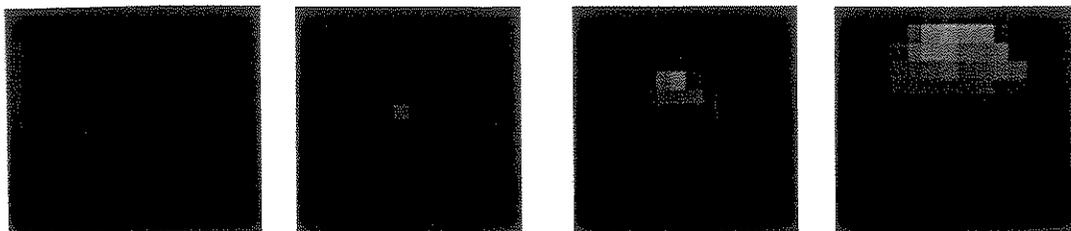


Figura 5.4: Imagem em Multi-Resolução de uma esfera. Todas as imagens possuem o mesmo tamanho, neste caso, $32 \times 32 = 1024$ pixels. O nível de resolução aumenta da esquerda para a direita. A medida que a imagem vai sendo colocada na parte mais central do olho (fóvea).

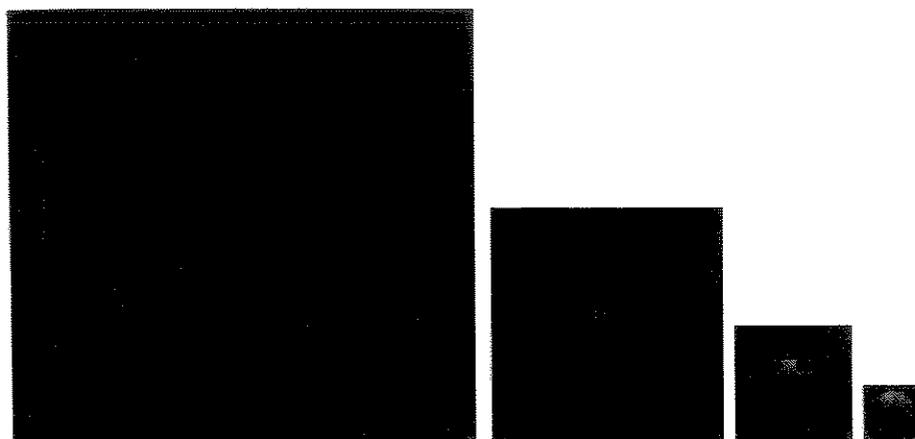


Figura 5.5: Imagens em Multi-resolução de uma esfera. Nesta figura as imagens foram re-escaladas, com o objetivo de mostrar quais as regiões que cada imagem cobre da imagem original. Quando se passa de uma representação para outra à sua direita, a área coberta diminui 2 vezes.

A representação da quantidade de informação extraída da imagem original pode ser vista na Figura 5.5, onde as imagens em multi-resolução mostradas pela Figura 5.4 são re-escaladas para mostrar o tamanho das regiões que são cobertas da imagem original. Este processo de

obtenção das imagens multi-resolução está relacionado diretamente com o esquema da fóvea, onde a imagem final (com a maior resolução) mostra uma pequena área da imagem original, e cada área maior da imagem original é representada com resolução menor.

Uma vez de posse destas imagens menores, podemos aplicar operações que podem ser realizadas com um custo mais reduzido. Em alguns casos, se precisarmos de uma imagem com alta resolução, por exemplo, para tarefas de reconhecimento de padrões (extração de características), podemos querer que o agente virtual coloque a região de interesse na fóvea dos dois olhos e então podemos executar as tarefas de reconhecimento, apenas na imagem de alta resolução, visto que ela contém a região de interesse para o comprimento da tarefa.

O algoritmo para a obtenção das imagens em multi-resolução opera diretamente na imagem que é obtida do *frameBuffer* (um tipo de memória) do OpenGL. A leitura da imagem original e a escrita das novas imagens são realizadas com os próprios métodos do OpenGL, por isso sendo muito rápido (tempo irrisório, desconsiderado no processamento).

5.3 Extração de Características

Na etapa de extração de características visuais, utilizamos operadores de convolução bidimensional, usando como máscaras funções definidas por derivadas parciais (de ordem zero, um e dois) do operador gaussiano, com o valor de $\sigma = 1.7$. Este valor foi definido em função de análises realizadas em trabalhos anteriores [20]. A equação 5.1 representa a fórmula básica para a distribuição gaussiana. A equação 5.2 representa a equação de convolução gaussiana e seus kernels são dados pelas equações 5.3, 5.4 e 5.5, cada um em suas direções ortogonais (X e Y). É possível notar que modificamos os kernels das derivadas, sem perda de generalização. Esta modificação torna mais fácil a implementação usando valores inteiros, que depois são normalizados para evitar problemas numéricos computacionais como estouro de valor.

$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \quad (5.1)$$

$$G_{d=x,y}^{(k=0,1,2)} = g_d^{(k)} * I_t \quad (5.2)$$

$$\left. \begin{aligned} g_x^{(0)}(x, y) &= \lambda e^{ax^2} \\ g_y^{(0)}(x, y) &= \lambda e^{ay^2} \end{aligned} \right\} \quad (5.3)$$

$$\left. \begin{aligned} g_x^{(1)}(x, y) &= 2a\lambda e^{a(x^2+y^2)}x \\ g_y^{(1)}(x, y) &= 2a\lambda e^{a(x^2+y^2)}y \end{aligned} \right\} \quad (5.4)$$

$$\left. \begin{aligned} g_x^{(2)}(x, y) &= 2a\lambda e^{a(x^2+y^2)}(2ax^2 + 1) \\ g_y^{(2)}(x, y) &= 2a\lambda e^{a(x^2+y^2)}(2ay^2 + 1) \end{aligned} \right\} \quad (5.5)$$

$\forall(x, y) \in ([-s, +s], [-s, +s])$; onde $a = \frac{-1}{2\sigma^2}$, $\lambda = \frac{1}{\sigma\sqrt{2\pi}}$, $\sigma = 1.7$, and $s = 3$.

A aplicação das gaussianas é uma fase de pré-processamento antes da extração final das características da imagem. Para gerar a parte gaussiana, cada imagem gerada pela fóvea (uma de cada olho) (seção 5.2) é convoluída com os seis kernels. No total são geradas 48 imagens de características (4 da fóvea do olho esquerdo + 4 da fóvea do olho direito) x 6 kernels. Geralmente, estas matrizes de características (por serem muito grandes) não são utilizadas diretamente como parâmetros do vetor de características [6].

A etapa final do processo de extração de características se dá pela aplicação de medidas estatísticas (média e variância) [37, 6] nas matrizes obtidas no processo de convolução anterior, para diminuir a dimensionalidade do vetor de características.

5.4 Controle da Atenção: Algoritmo de Vergência

Os objetivos principais do algoritmo de vergência são: manter sempre ambos os olhos centrados em algum lugar do ambiente, lugar este que é estabelecido de acordo com a tarefa que está sendo realizada, e colocar o ponto de interesse no centro ótico das câmeras. A importância deste segundo objetivo foi discutida na seção 3.2, no capítulo 3. O algoritmo proposto divide

um movimento de rotação 3D (R_x, R_y, R_z) em dois movimentos, um em R_x (vertical) e outro em R_y (horizontal).

Notamos que o algoritmo será utilizado para controlar a atenção de humanos virtuais, então o sistema visual adotado é baseado na fisiologia humana. Sendo assim, para que seja possível executar e manter a vergência em um determinado ponto do espaço, de forma que os movimentos da cabeça do agente virtual sejam mais realísticos, consideramos quatro restrições físicas de movimento (limites máximos para a vergência). As duas primeiras restrições tratam da quantidade ou valor em graus que cada olho pode se movimentar nas direções vertical e/ou horizontal, sem que seja necessário um movimento do pescoço (para movimentos verticais os olhos são deslocados juntos na mesma direção). E, as outras restrições tratam da quantidade ou valor em graus que o pescoço pode se movimentar nas direções vertical e/ou horizontal, quando não for possível alcançar o objeto apenas com os movimentos dos olhos.

Estabelecidos os valores máximos para os deslocamentos direcionais, para os olhos e pescoço, a segunda etapa do algoritmo trata da coordenação dos movimentos da cabeça do agente, ou seja, o algoritmo deve informar quando movimentar apenas os olhos ou apenas pescoço, ou ambos, em cada direção. Uma maneira simples de coordenar todos estes movimentos é considerar que os movimentos horizontais são subordinados aos verticais, ou seja, primeiro o sistema se movimenta na vertical e depois na horizontal. Considerar esta subordinação, facilitou o processo de coordenação. Pois, foi possível dividir uma única tarefa considerada complexa, em quatro outras sub-tarefas menos complexas: duas sub-tarefas coordenam os movimentos verticais e as outras duas os movimentos horizontais.

No caso das duas sub-tarefas para os movimentos verticais, a primeira checa se o ponto de interesse está dentro dos limites verticais máximos dos olhos, se isto for verdade apenas os olhos são movimentados (juntos) nesta direção, senão é por que o ponto de interesse está fora deste limite. Então, a segunda sub-tarefa faz com que o ponto de interesse seja colocado na direção do centro do pescoço (realizando um deslocamento vertical do pescoço naquela direção) e em seguida a primeira sub-tarefa faz uma nova checagem, para atualizar a posição dos olhos. Estas sub-tarefas direcionam os olhos do agente no ponto de interesse, em relação ao valor da coordenada "Y".

O funcionamento das outras duas sub-tarefas é semelhante com as anteriores. A primeira (considerada mais complexa), checa se o ponto de interesse está dentro dos limites horizontais máximos dos olhos, se isto for verdade apenas os olhos são movimentados nesta direção (o limite máximo está relacionado com a formação do triângulo durante o processo de vergência, a seção 5.5.3 aborda maiores detalhes), senão é por que o ponto de interesse está fora deste limite. Então, a segunda sub-tarefa faz com que o ponto de interesse seja colocado na direção horizontal do centro do pescoço (realizando um deslocamento horizontal do pescoço nesta direção) e em seguida a primeira sub-tarefa faz uma nova checagem, para mais uma vez atualizar a posição dos olhos (neste caso os ângulos horizontais formados, por cada um dos olhos, serão iguais). Estas sub-tarefas direcionam os olhos do agente no ponto de interesse, em relação ao valor da coordenada "X".

O algoritmo fica em um ciclo até que o sistema tenha alcançado o seu objetivo, ou seja, a cada término de execução de cada uma das quatro sub-tarefas é feita uma checagem para saber se os objetivos vertical e horizontal, que garantem que os olhos sejam fixados corretamente no ponto de interesse, foram alcançados. Caso não esteja, o sistema é posto para ser executado novamente até que todos os objetivos sejam alcançados. Um tratamento especial, em relação ao número de tentativas, é dado para evitar que o sistema fique em "loop" infinito.

A seguir são mostrados os passos realizados para o controle da atenção (algoritmo de vergência):

1. *Os olhos do agente virtual começam alinhados em paralelo e perpendiculares ao eixo X.*
2. *O ponto de interesse é passado pelo usuário, para o agente. A recuperação da informação 3D (X,Y,Z), do ponto em coordenadas de tela que foi passado previamente pelo usuário, é feita com o auxílio do "Z-buffer", o qual é passado para ambos os olhos do agente.*
3. *O processo atencional recebe este objetivo e em seguida dá início a execução da coordenação dos movimentos.*
4. *Movimentos verticais: é checado se o ponto de interesse, em relação a Y, está entre o limite máximo de deslocamento dos olhos. Se for verdade apenas os olhos se movimen-*

tam. Senão, o pescoço é deslocado até Y e em seguida os olhos são colocados na direção da coordenada Y do pescoço.

5. *Movimentos horizontais: é checado se o ponto de interesse, em relação a X , está entre o limite máximo de deslocamento dos olhos. Se for verdade apenas os olhos se movimentam. Senão, o pescoço é deslocado e em seguida os olhos são colocados na direção da coordenada X .*
6. *O algoritmo fica em loop, verificando se os objetivos, de cada uma das quatro sub-tarefas, foram alcançados. E com isso, tenta garantir que na situação final os olhos estejam fixados corretamente no ponto de interesse desejado.*

De modo diferente das abordagens vistas no capítulo 3, o controle da atenção do nosso agente é feito utilizando valores em coordenadas de mundo e os cálculos realizados para os deslocamentos são baseados em simples cálculos geométricos com as coordenadas X e Y .

5.5 Implementação do Algoritmo de Vergência

O algoritmo foi implementado considerando que os movimentos vertical e horizontal dos olhos e do pescoço são independentes. De certa forma, a implementação do controle da execução de todos os movimentos é simples, pois apenas a informação de distância entre dois pontos é utilizada, ou seja, nenhum cálculo matemático mais complexo (como cálculos de correlação) é necessário para o correto funcionamento do algoritmo.

5.5.1 Aquisição da Informação de Profundidade: *Z-buffer*

A informação de profundidade é um dos parâmetros que é utilizado para controlar a atenção do agente virtual. Este parâmetro está relacionado diretamente com o ponto de interesse, o qual o agente deve colocar no centro ótico das duas câmeras (ou as fôveas dos olhos).

No nosso modelo atual de implementação o ponto de fixação (seção 3.2.1) é passado pelo usuário para as duas câmeras, ou seja, o agente sabe para onde deve olhar. Pois, como mencionado no capítulo 2, a habilidade de mudança do foco de atenção não será abordada neste trabalho, o principal interesse aqui é o controle da atenção (algoritmo de vergência). Desta forma, o ponto de interesse é passado com o auxílio do *mouse*.

A recuperação do valor da coordenada de mundo ocorre da seguinte maneira: quando o usuário pressiona o botão direito do *mouse*, no ambiente virtual do Câmbio, as coordenadas de tela (x, y) do ponto de interesse são transformadas em coordenadas de mundo (X, Y, Z) , as quais são passadas imediatamente para o algoritmo de vergência, fazendo com que os olhos do agente sejam fixados naquela direção. A descrição do código, que realiza a recuperação do ponto de interesse em coordenadas de mundo, é mostrado na Figura 5.6.

```
1. void viewPoint(int x, int y){
2.     GLfloat z;
3.     H = (matrizViewport[3] - (GLint)y - 1);
4.     glReadPixels(x, H, 1, 1, GL_DEPTH_COMPONENT, GL_FLOAT, &z);
5.     gluUnProject ((GLdouble)x, (GLdouble)H, z, matrizModelview,
6.                 matrizProjecao, matrizViewport, &wx, &wy, &wz);
7. }
```

Figura 5.6: Código, em OpenGL, para obtenção do ponto de interesse, em coordenadas de mundo (X, Y, Z) , a partir do *Z-buffer*. Na linha 4 é lido o valor da coordenada Z , de acordo com a posição do ponto em coordenadas de tela (x, y) e na linha 5 é retornado o valor $(\&wx, \&wy, \&wz)$, em coordenadas de mundo, do ponto de interesse.

Apesar do ponto de interesse ser passado pelo *mouse*, o funcionamento do nosso algoritmo de vergência é independente da intervenção do usuário, isto é, a única função do usuário é dizer para onde o agente virtual deve olhar. Caso o agente estivesse navegando pelo ambiente, apenas um mecanismo um pouco mais sofisticado para ler a informação de profundidade diretamente do *Z-buffer* deveria ser desenvolvido, sem a necessidade de se fazer quaisquer mo-

dificações no algoritmo proposto.

5.5.2 Implementação dos Movimentos Verticais

Para a execução dos movimentos verticais dos olhos e do pescoço foram implementadas duas funções que representam os motores (N1 e E3) da arquitetura mostrada na Figura 4.3. Estes motores executam movimentos independentes. As áreas de deslocamento cobertas por cada um dependem das restrições físicas estabelecidas previamente pelo usuário. Na direção vertical, o pescoço pode se deslocar 127 pixels e os olhos 10 pixels, para cima ou para baixo. Lembrando que os valores em pixel são transformados deslocamentos rotacionais (seção 5.5.5), na efetivação dos movimentos.

Para que os motores N1 (pescoço) e E3 (olhos) comecem seus movimentos é necessário passar apenas o valor da coordenada DY , ou valor do ponto de interesse em Y , lido do *Z-buffer*. Com base nos valores da coordenada DY e da direção atual do pescoço ($NeckVerDir$), o código interno do motor N1 faz uma checagem para inferir a distância e a direção do deslocamento que deverá ser realizado, isto é, se $(NeckVerDir \geq 0 \text{ and } (DY - NeckVerDir) > 0)$ or $((NeckVerDir < 0 \text{ and } (DY - NeckVerDir) > 0)$, o movimento é para cima ($MoveNeckY(UP)$); senão, o movimento é para baixo ($MoveNeckY(DOWN)$).

O código interno do motor E3 é semelhante ao do N1, a única diferença está na checagem das condições para o deslocamento. Como na direção vertical os olhos se movimentam juntos, a direção atual ($EyeVerDir(LEFT)$) de um dos olhos (no caso o esquerdo $LEFT$) é utilizada no lugar de ($NeckVerDir$). Assim, se $(EyeVerDir(LEFT) \geq 0 \text{ e } (DY - EyeVerDir(LEFT)) > 0)$ ou $((EyeVerDir(LEFT) < 0 \text{ e } (DY - EyeVerDir(LEFT)) > 0)$, os olhos são movimentados para cima ($MoveEyeVer(LEFT, UP)$, $MoveEyeVer(RIGHT, UP)$), senão são movidos para baixo ($MoveEyeVer(LEFT, DOWN)$, $MoveEyeVer(RIGHT, DOWN)$). A Figura 5.7 mostra uma ilustração referente a checagem dos movimentos, realizadas por estes dois motores.

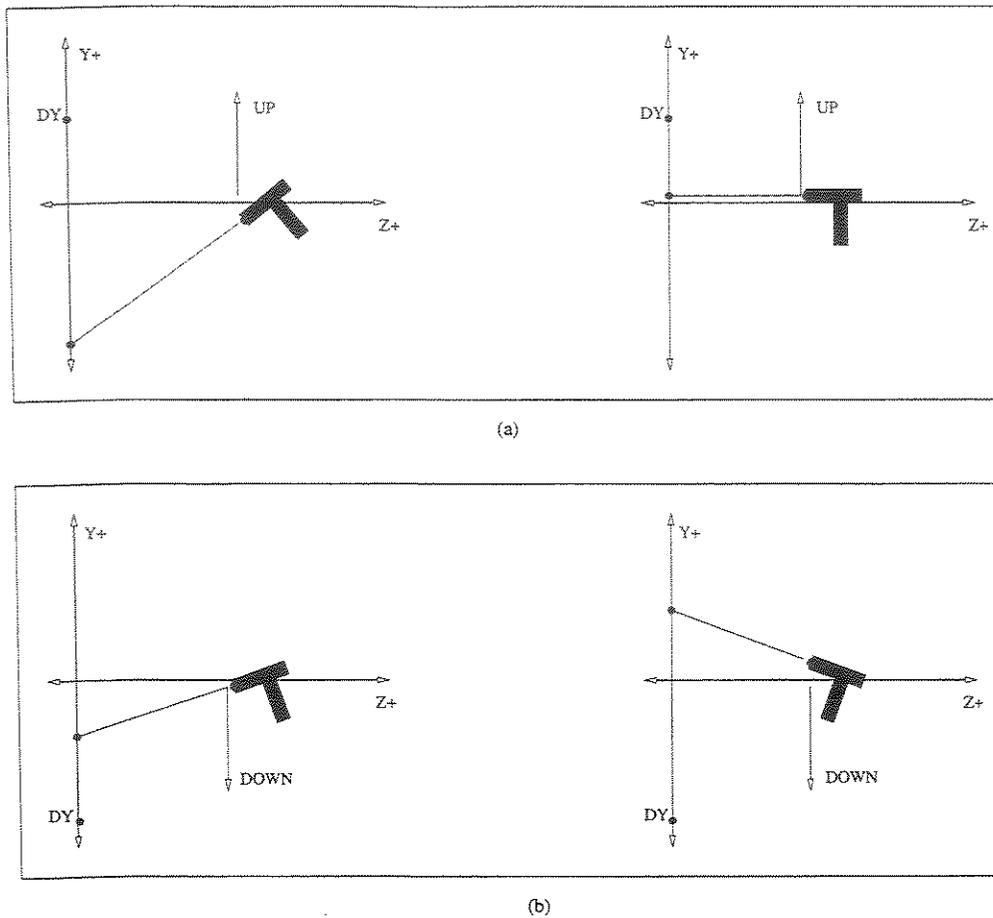


Figura 5.7: Dinâmica dos movimentos verticais dos olhos e do pescoço. DY é o valor do ponto de interesse em Y , lido do Z -buffer. O sistema está posicionado em $(0, 0, 0)$. (a) situações em que o movimento é realizado para cima. (b) situações em que o movimento é realizado para baixo. As setas indicam a direção do movimento (UP é para cima e $DOWN$ é para baixo).

5.5.3 Implementação dos Movimentos Horizontais

Para a execução dos movimentos horizontais dos olhos e do pescoço, foram implementadas duas funções que representam os motores (E1, E2 e N2) da arquitetura mostrada na Figura 4.3. Apesar das funcionalidades dos motores E1 e E2 terem sido implementadas por uma única função, eles possuem movimentos independentes. A primeira razão para a independência destes

movimentos é por que, geralmente, a posição final de vergência requer ângulos diferentes entre a normal de cada olho (Figura 5.8) e com isso, cada motor fica responsável pelo movimento de um dos olhos. Outro motivo é que o processo pode ser executado com maior velocidade, admitindo que os movimentos ocorrem em paralelo.

As áreas de deslocamento cobertas por cada motor, também dependem das restrições físicas estabelecidas previamente pelo usuário. Foi estabelecido que pescoço pode se deslocar 127 píxels e cada olho 15 píxels, para cada lado.

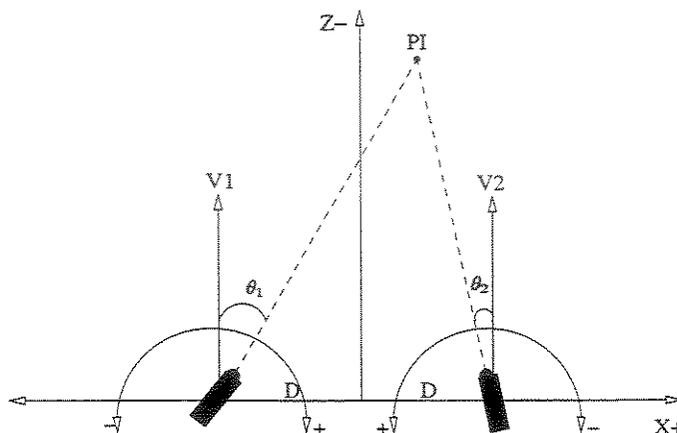


Figura 5.8: Situação final de vergência horizontal. O ponto de interesse(PI) está entre os limites de vergência dos olhos. V1 e V2 são as retas normais de cada olho. θ_1 e θ_2 são os ângulos de vergência horizontal formados entre as retas normais dos olhos esquerdo e direito. 2D é o tamanho da linha base.

Para que o motor N2 (pescoço) comece seus movimentos, é necessário passar apenas o valor da coordenada DX , ou valor do ponto de interesse em X , lido do *Z-buffer*. Com base nos valores da coordenada DX e da direção atual do pescoço (*NeckHorDir*), o código interno do motor N1 faz uma checagem para inferir a distância e a direção do deslocamento que deverá ser realizado, isto é, se ($NeckHorDir \geq 0$ e $(DX - NeckHorDir) > 0$) ou ($NeckHorDir < 0$ e $(DX - NeckHorDir) > 0$), o movimento é para direita (*MoveNeckX(RIGHT)*); senão, o movimento é para esquerda (*MoveNeckX(LEFT)*). A Figura 5.9 mostra uma ilustração referente a checagem destes movimentos. Os motores E1 e E2 são responsáveis por colocar o ponto de

fixação, em relação (DX), no centro de cada olho. De modo que na situação final, os olhos estejam convergidos um em relação ao outro (Figura 5.8).

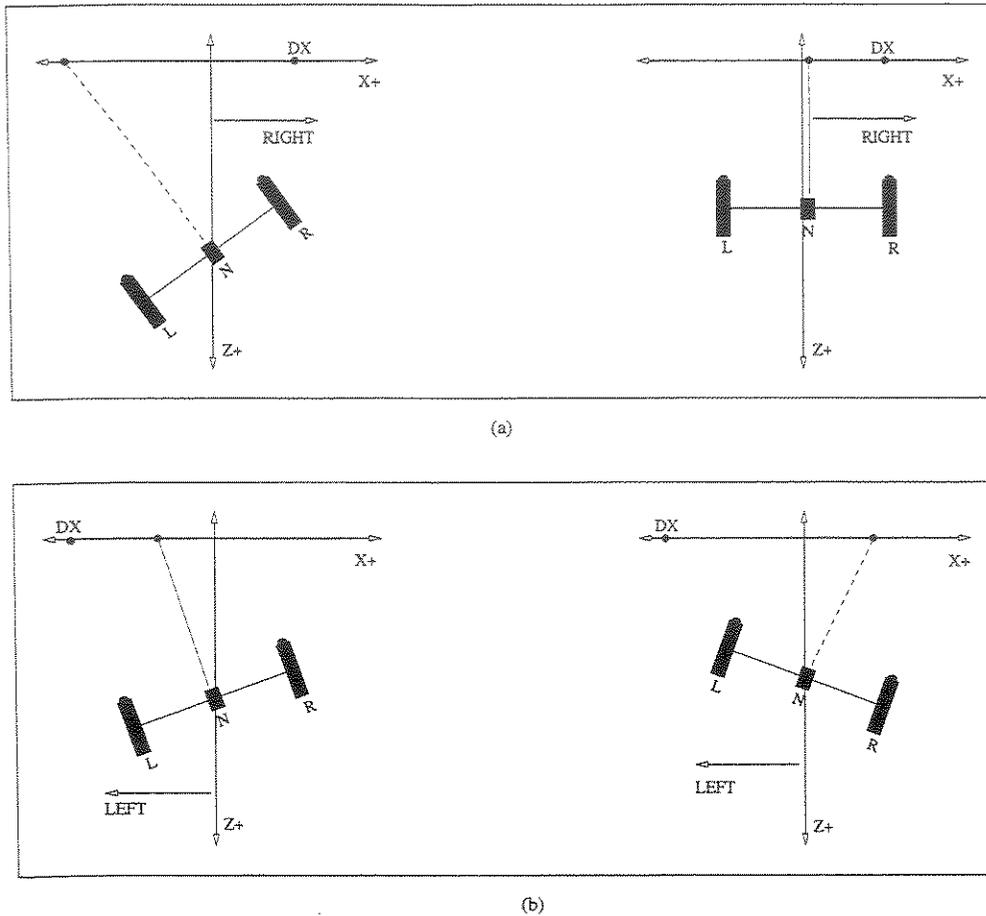


Figura 5.9: Dinâmica dos movimentos horizontais do pescoço. (a) situações em que o movimento é realizado para o lado direito. (b) situações em que o movimento é realizado para o lado esquerdo. As setas indicam a direção do movimento (RIGHT é para direita e LEFT é para esquerda). L e R representam os olhos esquerdo e direito e N o pescoço. O sistema está posicionado em $(0,0,0)$.

Na situação inicial, os olhos do Câmbio estão alinhados em paralelo. Os movimentos iniciais dos motores E1 e E2 ajustam os ângulos de vergência do Câmbio, com base na direção horizontal do pescoço ($NeckHorDir$), no limite máximo de vergência dos olhos ($MaxHorDeg$), na distância dos olhos ao pescoço D , em DX e em XPI (ponto de interesse atual, em relação

ao eixo X). Se DX está do lado esquerdo ou direito e entre a área de interseção do campo de vergência dos olhos (Figura 5.11), isto é, se $\|DX\| - |NeckHorDir| \leq (MaxHorDeg - D)$ e então $MoveEyeHor(LEFT, RIGHT)$ (move olho esquerdo para a direita) e $MoveEyeHor(RIGHT, LEFT)$ (move olho direito para a esquerda); senão, se DX está do lado esquerdo ou direito, fora da área de interseção de vergência (ambos os olhos são deslocados, somente até o limite máximo de vergência esquerdo ou direito (Figura 5.10(b)), isto é, se $\|DX\| - |NeckHorDir| > (MaxHorDeg - D)$, então $MoveEyeHor(LEFT, RIGHT)$ e $MoveEyeHor(RIGHT, LEFT)$. A Figura 5.10 mostra uma ilustração referente à checagem destes movimentos.

Depois que o processo anterior ajustou os ângulos iniciais de vergência, a próxima etapa é tratar dos deslocamentos horizontais quando os olhos já se encontram vergidos. Nesta situação os olhos se movem sempre na mesma direção (a seção 5.5.5 mostra maiores detalhes). Estes novos movimentos dependem das direções horizontais dos olhos direito ($EyeHorDir$) e esquerdo ($EyeHorDir$), $NeckHorDir$, DX e XPI . Para minimizar a tarefa do controlador principal (seção 5.5.4) foi implementado um sub-controlador que tem a funcionalidade de manter ambos os olhos vergidos no ponto de fixação, desde que este ponto esteja dentro da interseção dos limites físicos de vergência dos dois olhos (Figura 5.11).

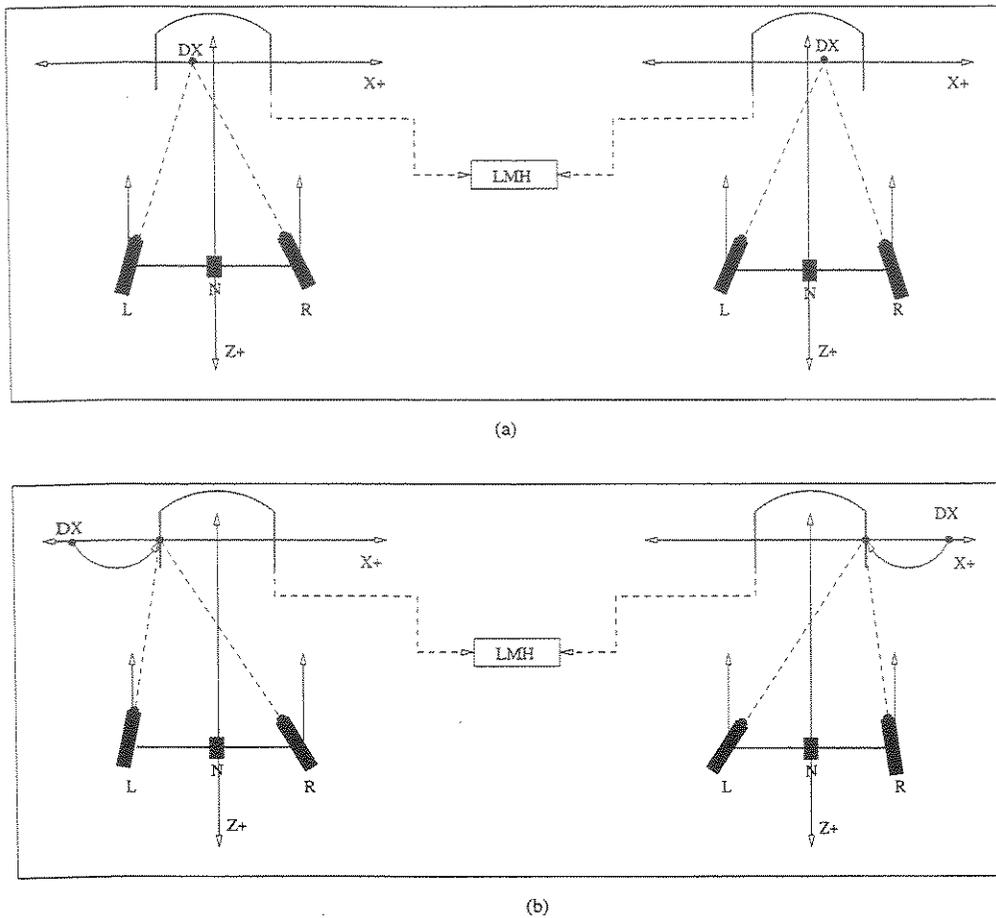


Figura 5.10: Ajuste do ângulo de vergência inicial. LMH é o limite máximo de vergência (esquerdo e direito), que é dependente das restrições físicas dos olhos. Esta restrição é utilizada para evitar que os olhos percam a vergência. (a) Dx está entre LMH , os olhos se movem normalmente até Dx . (b) Dx está além de LMH , os olhos são movimentados, somente, até LMH . L e R representam os olhos esquerdo e direito e N o pescoço. O sistema está posicionado em $(0,0,0)$.

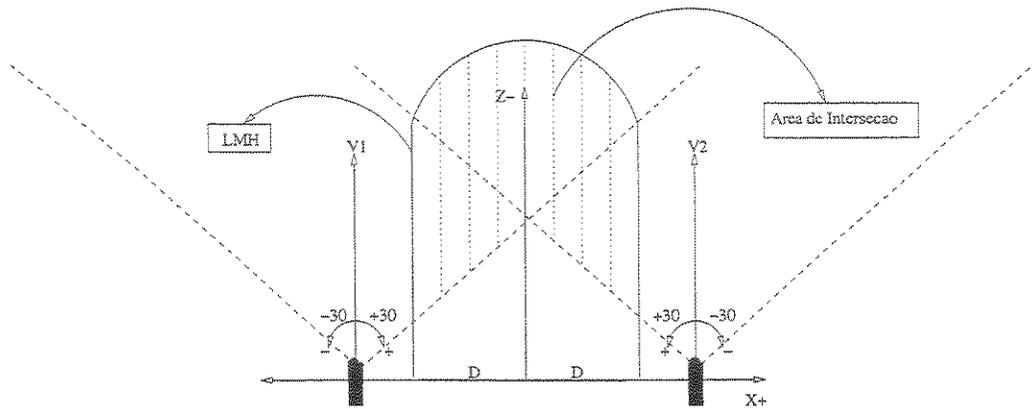


Figura 5.11: Área de interseção da vergência horizontal dos olhos. A área de interseção da vergência é definida como a região do espaço, onde os olhos podem se mover sem que ocorra a perda da vergência, quando são considerados apenas os movimentos horizontais dos mesmos. Dependendo dos valores atribuídos aos limites físicos dos olhos, a área de interseção pode aumentar ou diminuir. LMH é o limite máximo de vergência (esquerdo e direito), que é dependente das restrições físicas. L e R representam os olhos esquerdo e direito e No pescoço. O sistema está posicinado em $(0,0,0)$.

O sub-controlador é baseado na escolha de um olho dominante (a função do olho dominante é informar o lado do ponto de interesse, em relação a posição do pescoço), esta informação é usada para evitar que os olhos saiam da área de interseção da vergência. Pois, dependendo da posição do ponto de interesse, um dos olhos pode ter chegado ao seu limite máximo de deslocamento, enquanto que o outro ainda não. Portanto, esta informação é utilizada para evitar que o olho que ainda não chegou no seu limite máximo, mova-se mais do que o olho que já atingiu este limite. Caso esta regra não seja obedecida, a vergência entre os olhos será perdida. Então, se $(DX - NeckHorDir \leq 0)$, DX está do lado esquerdo e este olho é escolhido como dominante(*Dom*) e o olho direito é o não dominante(*NotDom*), senão tem-se o inverso.

Escolhido o olho dominante, os movimentos dos olhos são executados da seguinte maneira: se, $(DX - XPI) \geq 0$ e de acordo com a informação do sub-controlador, *MoveEyeHor(LEFT, RIGHT)* (move olho esquerdo para o lado direito) e *MoveEyeHor(RIGHT, RIGHT)* (move olho direito para o lado direito); senão, se $(DX - XPI) < 0$ e de acordo com a informação do

sub-controlador, $MoveEyeHor(LEFT, LEFT)$ (move olho esquerdo para o lado esquerdo) e $MoveEyeHor(RIGHT, LEFT)$ (move olho direito para o lado direito).

O funcionamento do sub-controlador depende da informação de Dom , $NotDom$, $NeckHorDir$, $MaxHorDir$ e D . Caso o deslocamento seja para o lado direito, tem-se: se $(\|DX - |NeckHorDir|\| \geq (MaxHorDeg - D))$, então $(Max = (MaxHorDeg - D))$, senão $(Max = \|DX - |NeckHorDir|\|)$ e se $(\|NotDom - |Dom|\| > 2 * Max)$ então, executa os movimentos para a direita. Caso o deslocamento seja para o lado esquerdo, tem-se: se $(\|DX - |NeckHorDir|\| \geq (MaxHorDeg - D))$, então $(Max = (MaxHorDeg - D))$, senão $(Max = \|DX - |NeckHorDir|\|)$ e se $(\|NotDom - |Dom|\| < 2 * Max)$ então, executa os movimentos para a esquerda. Max é o deslocamento máximo que os olhos podem realizar, em relação ao pescoço, dentro da área de interseção da vergência, ou seja, a condição $\|DX - |NeckHorDir|\|$ é para saber se DX está dentro ou fora dos limites máximos de interseção da vergência (Figura5.11), se for verdade os olhos Dom e $NotDom$ se movimentam até que este limite máximo seja alcançado senão eles se movimentam até LMH (Figura5.11), através do incremento ou decremento de Dom e $NotDom$ $(\|NotDom - |Dom|\| < ou > 2 * Max)$ (dependendo da direção do movimento).

5.5.4 Implementação da Coordenação dos Movimentos

O módulo controlador (Figura 4.3) é responsável pela coordenação dos movimentos dos olhos e do pescoço do agente virtual, fazendo com que os olhos sejam fixados no ponto de interesse desejado. Em termos de implementação, ele coordena a execução dos movimentos de todos os motores. O controlador primeiro coordena os movimentos verticais (primeiro é checado o deslocamento dos olhos e depois do pescoço) e depois os horizontais (primeiro é checado o deslocamento do pescoço e depois dos olhos).

De acordo com a posição de DY , a coordenação dos movimentos verticais analisa se a posição do ponto de interesse DY , em relação a direção vertical do pescoço, está dentro do limite vertical máximo de deslocamento dos olhos $(|DY - NeckVerDir| \leq MaxVerDeg)$. Caso esta condição seja verdadeira, apenas os olhos se movimentam até DY ($MotorEyeVer(DY)$,

desloca os olhos até DY), senão o pescoço se movimenta até DY e em seguida a direção vertical dos olhos é corrigida e colocada em DY ($MotorNeckVer(DY)$ e $MotorEyeVer(Dy)$).

De acordo com a posição de DX , a coordenação dos movimentos horizontais começa verificando se a posição de DX , em relação a direção do pescoço, está na área de interseção da vergência (Figura 5.11) ($|DX - NeckHorDir| \leq MaxHorDeg - D$). Caso esta condição seja verdadeira, apenas os olhos se movimentam ($MotorEyeHor(DX)$, deslocamento horizontal dos olhos). Senão o pescoço se movimenta até DX ($MotorNeckHor(DX)$) e em seguida os olhos são fixados em DX ($MotorEyeHor(DX)$).

Considerar que os movimentos da cabeça são subordinados, facilitou o controle da coordenação. A coordenação dos movimentos é muito simples, pois apenas com o conhecimento da informação da posição do ponto de interesse, em relação as coordenadas (X, Y) , e das direções vertical e horizontal do pescoço, é possível fixar os olhos do agente virtual em quaisquer pontos do ambiente.

5.5.5 Política de Deslocamento

Nosso agente virtual é composto por dois conjuntos: o sistema gráfico (olhos e pescoço, os quais compõem a cabeça do agente) e o sistema visual que é composto pelas duas câmeras virtuais, como mostrado na seção 2.2.3. Foi considerado que quando a cabeça se movimenta, as câmeras acompanham este movimento, ou seja, na posição final, os olhos e as câmeras estão na direção do ponto de interesse (com o ponto de interesse focalizado no centro ótico das duas câmeras).

Foi considerado que o sistema gráfico controla o movimento das duas câmeras, ou seja, durante o processo de deslocamento, a cabeça e as câmeras formam um conjunto único. Os referenciais para a realização dos deslocamentos não são os mesmos, o sistema gráfico se desloca em coordenadas de mundo e o visual nos referenciais de câmera. Todavia, a informação utilizada para dar início aos processos de deslocamentos é a mesma, isto é, o ponto em coordenadas de mundo obtido do *Z-buffer*.

Sendo assim, foi considerado que os movimentos de rotação da cabeça do agente, em torno dos eixos X e Y , são executados em valores discretos de pixels. Isto significa dizer que uma rotação em torno de Y de 15° é equivalente a um deslocamento horizontal de 15 pixels no eixo X , por exemplo, indo da coordenada $X_1 = 0$ até a coordenada $X_2 = 15$, (o ambiente virtual é construído utilizando esta unidade de medida). E as câmeras se movimentam do ponto inicial(P_i) até o ponto final(P_f).

Esta solução foi adotada por que o modelo de câmera do OpenGL (*gluLookAt()*) é baseado na posição (X_p, Y_p, Z_p) e na direção do ponto de visão (X_v, Y_v, Z_v) do observador e no vetor direção (VUP *vector*) (no nosso caso ele é $(0,1,0)$, para cima). Como visto na seção 2.2.3, para que a câmera seja colocada no ponto de interesse desejado (ponto de visão) é necessário passar apenas as coordenadas (X_f, Y_f, Z_f) deste ponto, que a câmera é colocada nesta direção.

Os deslocamentos de rotação do sistema gráfico são realizados com o auxílio da função (*glRotatef()*). Os parâmetros desta função são o ângulo de deslocamento e o eixo em que será realizada a rotação. Portanto, deslocamentos em X e em Y controlam os movimentos do sistema visual e gráfico nestas direções.

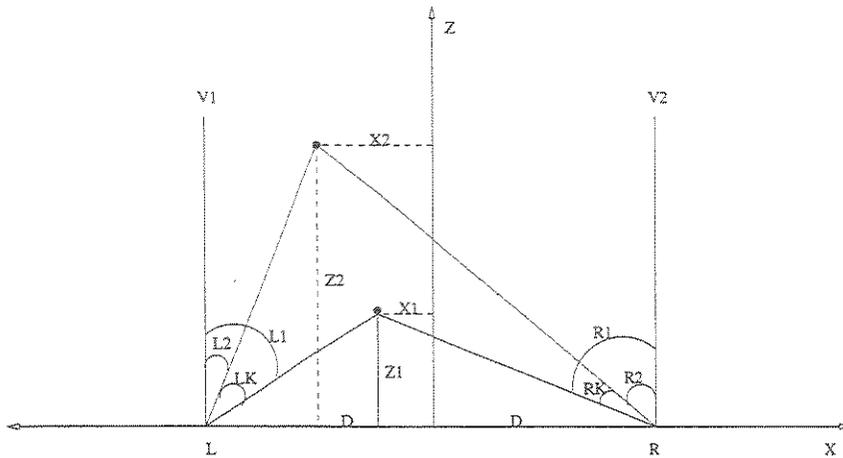


Figura 5.12: Modelo matemático de deslocamento. Nesta Figura é considerado o deslocamento horizontal. Na posição inicial os olhos esquerdo(L) e direito(R) encontram-se perpendiculares ao eixo X. Quando o ponto de interesse inicial (coordenada X_1) é passado para o agente, o conjunto é deslocado até este ponto. Logo depois, outro ponto (coordenada X_2) é passado para o agente, fazendo com que os olhos sejam colocados em X_2 .

A Figura 5.12 mostra o modelo matemático adotado para a realização dos deslocamentos horizontais e verticais dos dois sistemas. De acordo com este modelo, quatro equações de deslocamento podem ser obtidas. Para o lado direito do sistema, tem-se:

$$\tan(R_1) = \frac{D + X_1}{Z_1}, \tan(R_2) = \frac{D + X_2}{Z_2} \quad (5.6)$$

e como, $R_k = R_1 - R_2$, tem-se:

$$\tan(R_k) = \frac{D(Z_2 - Z_1) + X_1Z_2 - X_2Z_1}{Z_2Z_1} \quad (5.7)$$

E para o lado esquerdo, tem-se:

$$\tan(L_1) = \frac{D - X_1}{Z_1}, \tan(L_2) = \frac{D - X_2}{Z_2} \quad (5.8)$$

e como, $L_k = L_1 - L_2$, tem-se:

$$\tan(L_k) = \frac{D(Z_2 - Z_1) + X_2Z_1 - X_1Z_2}{Z_2Z_1} \quad (5.9)$$

Estas equações foram utilizadas para a execução dos movimentos horizontais e verticais dos dois sistemas (no caso dos movimentos verticais basta trocar X por Y). Quando o sistema sai de sua posição inicial para uma situação final de vergência, os deslocamentos horizontais são realizados pelas equações 5.6 e 5.8. Note que, dependendo do lado da coordenada X , estas equações podem ser invertidas. Depois que o sistema está em uma situação inicial de vergência (coordenada X_1 , da Figura 5.12), os próximos movimentos (coordenada X_2) são realizados com o mesmo valor de deslocamento. Isto pode ser mostrado pelas equações 5.7 e 5.9, onde a seguinte igualdade é válida: $|\tan(R_k)| = |\tan(L_k)|$.

Como os deslocamentos horizontais do sistema gráfico são realizados no eixo X , foi considerado que eles independem do valor da coordenada Z , pois estamos indo de uma coordenada inicial a outra final. Desta forma, adotou-se que os valores das coordenadas Z_s são iguais à unidade ($Z_1 = Z_2 = 1$). Portanto, o deslocamento D_x é reduzido para a seguinte equação: $D_x = |X_2 - X_1|$. No caso dos deslocamentos verticais, tem-se: $D_y = |Y_2 - Y_1|$.

Por outro lado, o deslocamento horizontal do sistema visual depende dos valores das coordenadas Z_s . Neste caso, foi considerado que quando as câmeras são deslocadas de uma posição X_1 para outra X_2 , as câmeras recebem os valores incrementais da coordenada X_1 até X_2 e o valor final Z_2 da coordenada final. No caso do deslocamento vertical, $(X_1, X_2) = (Y_1, Y_2)$. Por exemplo, digamos que o sistema vá de uma posição inicial (X_1, Y_1, Z_1) a outra final (X_{10}, Y_{10}, Z_{10}) , no sistema de coordenadas (X, Z) . O sistema gráfico se desloca de (X_1, \dots, X_{10}) e o sistema visual vai de $((X_1, Z_{10}), \dots, (X_{10}, Z_{10}))$.

5.5.6 Generalizando o modelo para o deslocamento do Sistema Gráfico

A solução proposta para o deslocamento do sistema gráfico, possui uma pequena limitação em relação aos deslocamentos realizados com os mesmos valores de coordenada X_s e diferentes valores de coordenada Z_s , ou seja, o ângulo de vergência do sistema gráfico não é modificado

quando esta situação ocorre. Isto porque usamos o *Z-buffer* para resolver o problema de determinar a profundidade, porém a escolha do ponto objetivo é feita numa imagem plana (portanto mesma coordenada X pode ocorrer para valores diferentes de z). Este caso (z diferentes) poderia ocorrer, por exemplo, em situações em que os objetos do ambiente podem ser transparentes. Neste caso, poder-se-ia querer focalizar em dois objetos sobre a mesma posição no plano X, Y , no qual o ponto objetivo é indicado pelo clique do mouse, mas com profundidades diferentes. Nota-se que ao usar o *Z-buffer*, não usamos transparência, por isso o algoritmo funciona em nossas aplicações. Ainda, a restrição colocada não inviabiliza o algoritmo por que a parte principal do agente é o sistema visual e este não possui esta limitação de deslocamento, uma vez que os valores das coordenadas Z_s são passados como parâmetros de entrada para os movimentos das câmeras. A Figura 5.13 mostra o que acontece quando esta situação ocorre.

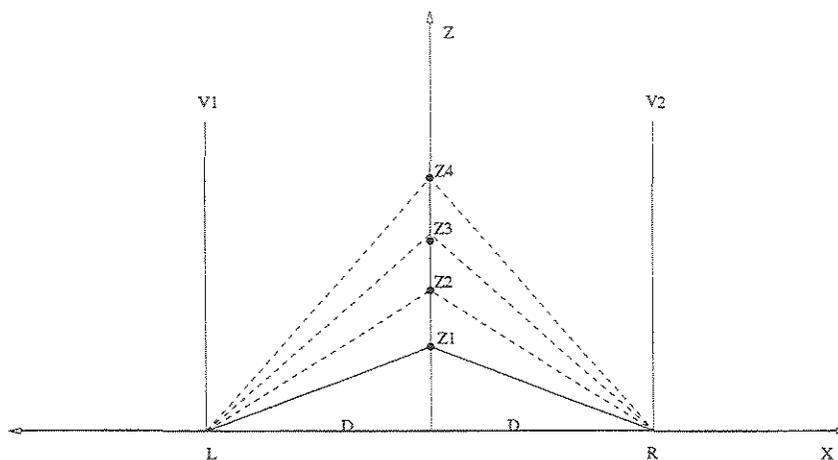


Figura 5.13: Deslocamento do sistema gráfico e visual, para diferentes valores de coordenada Z e valores iguais de coordenada X . A linha sólida representa o deslocamento do sistema gráfico e as pontilhadas o sistema visual.

Assim, para tornar o algoritmo proposto mais geral, caso outro modelo seja escolhido para determinar o ponto objetivo, devemos derivar um modelo mais genérico para o deslocamento, baseado em um sistema de mundo (global). Considere o sistema global em coordenadas de mundo mostrado na Figura 5.14. Neste sistema de mundo, dados quatro pontos, dois para as posições atuais dos olhos, o terceiro sendo o ponto atual onde o agente está vergido (focado) e

o quarto sendo o ponto objetivo para onde o agente deve mudar o seu foco de atenção. Pode-se calcular dois ângulos de deslocamento 3D de vergência, um para cada olho, os quais podem ser desdobrados em dois ângulos de rotação, um em torno de X e o outro em torno de Y . Neste caso, usando ângulos de Euler, no final obtém-se quatro ângulos, sendo dois para cada um dos olhos.

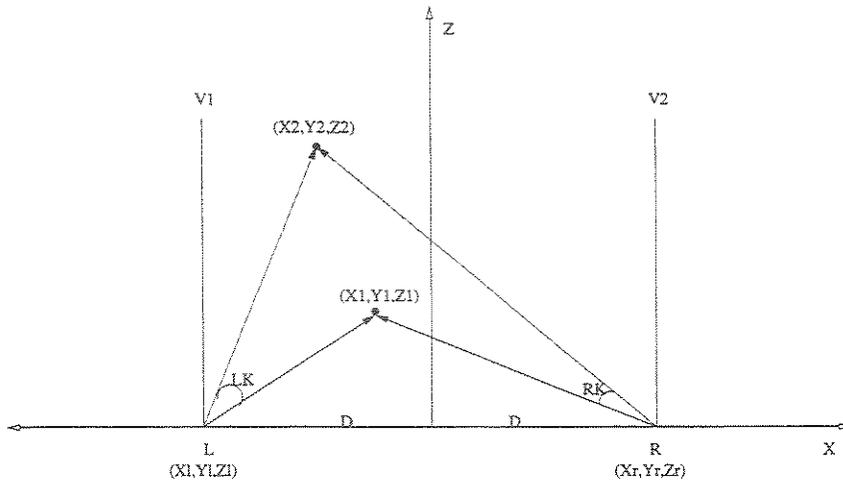


Figura 5.14: Modelo matemático de deslocamento vetorial. Esta Figura mostra como o sistema gráfico e o visual podem ser movimentados de uma situação inicial de vergência (X_1, Y_1, Z_1) para outra final (X_2, Y_2, Z_2) . Levando em consideração os ângulos de deslocamento L_k e R_k .

Sendo os pontos (X_l, Y_l, Z_l) e (X_r, Y_r, Z_r) onde os olhos estão localizados, (X_1, Y_1, Z_1) o ponto atual e (X_2, Y_2, Z_2) o ponto objetivo, pode-se formular um modelo algébrico computacional para determinar os ângulos L_{kx} , L_{ky} , R_{kx} e R_{ky} .

Sejam $d_l X_1 = (X_1 - X_l)$; $d_l Y_1 = (Y_1 - Y_l)$; $d_l Z_1 = (Z_1 - Z_l)$; $d_l X_2 = (X_2 - X_l)$; $d_l Y_2 = (Y_2 - Y_l)$; $d_l Z_2 = (Z_2 - Z_l)$; $d_r X_1 = (X_1 - X_r)$; $d_r Y_1 = (Y_1 - Y_r)$; $d_r Z_1 = (Z_1 - Z_r)$; $d_r X_2 = (X_2 - X_r)$; $d_r Y_2 = (Y_2 - Y_r)$; $d_r Z_2 = (Z_2 - Z_r)$.

Sejam os vetores $d_{l1} = (d_l X_1, d_l Y_1, d_l Z_1)$; $d_{l2} = (d_l X_2, d_l Y_2, d_l Z_2)$; $d_{r1} = (d_r X_1, d_r Y_1, d_r Z_1)$ e $d_{r2} = (d_r X_2, d_r Y_2, d_r Z_2)$.

Então, os ângulos 3D são dados por:

$$L_k = \text{acos} \frac{\langle d_{l1}, d_{l2} \rangle}{\| \langle d_{l1}, d_{l2} \rangle \|} \text{ e } R_k = \text{acos} \frac{\langle d_{r1}, d_{r2} \rangle}{\| \langle d_{r1}, d_{r2} \rangle \|}.$$

Estes são os ângulos 3D em um sistema global. Para determinar cada rotação em torno de Y e X , separadamente, podemos usar vetores com apenas duas coordenadas. Ou seja, para determinar a rotação em X , tem-se: $d_{l1} = (d_l X_1, d_l Z_1)$; $d_{l2} = (d_l X_2, d_l Z_2)$; $d_{r1} = (d_r X_1, d_r Z_1)$ e $d_{r2} = (d_r X_2, d_r Z_2)$. E os ângulos são dados por:

$$L_{kx} = \text{acos} \frac{\langle d_{l1}, d_{l2} \rangle}{\| \langle d_{l1}, d_{l2} \rangle \|}; R_{kx} = \text{acos} \frac{\langle d_{r1}, d_{r2} \rangle}{\| \langle d_{r1}, d_{r2} \rangle \|} \quad (5.10)$$

E depois para determinar a rotação em Y , tem-se: $d_{l1} = (d_l Y_1, d_l Z_1)$; $d_{l2} = (d_l Y_2, d_l Z_2)$; $d_{r1} = (d_r Y_1, d_r Z_1)$ e $d_{r2} = (d_r Y_2, d_r Z_2)$. E os ângulos são:

$$L_{ky} = \text{acos} \frac{\langle d_{l1}, d_{l2} \rangle}{\| \langle d_{l1}, d_{l2} \rangle \|}; R_{ky} = \text{acos} \frac{\langle d_{r1}, d_{r2} \rangle}{\| \langle d_{r1}, d_{r2} \rangle \|} \quad (5.11)$$

Para que estas equações 5.10 e 5.11 sejam usadas, torna-se necessário fazer algumas mudanças, quanto ao controle dos deslocamentos dos motores e do próprio controlador, para fazer com que eles se desloquem em valores de graus e não de pixels. Estas mudanças não foram implementadas em nosso modelo, devido estarmos baseados no *Z-buffer* sem transparência, e uma vez que o sistema visual consegue chegar ao objetivo sem maiores dificuldades. Esta solução poderá ser comparada com a anterior em trabalhos futuros.

Capítulo 6

Experimentos e Resultados

Este capítulo descreve os experimentos realizados para avaliar computacionalmente o algoritmo de vergência proposto e implementado nesta dissertação. Esta avaliação foi feita com o auxílio da plataforma de simulação, que também foi desenvolvida neste trabalho [9, 50, 52, 10]. O principal objetivo destes experimentos é mostrar como o controle da atenção do agente virtual se mantém (de acordo com a funcionalidade do nosso algoritmo), quando diferentes situações de vergência são necessárias para que um objetivo seja colocado na direção do ponto de visão (centro de cada olho) de um agente sintético, inserido em um ambiente virtual tri-dimensional.

O primeiro experimento realizado mostra o funcionamento do algoritmo, em relação ao controle da coordenação dos movimentos da cabeça (olhos e pescoço) do agente, levando os olhos de uma situação inicial de vergência para outra final [9, 50, 52, 10], também desejada de vergência.

O segundo experimento mostra como a atenção do agente se mantém quando objetos próximos e distantes e em alturas e posições diferentes são colocados como objetivo para o processo de fixação [9, 50, 52, 10, 51].

O terceiro experimento faz uma simulação de uma reunião virtual, onde um agente (o que está se comunicando) deve direcionar a sua cabeça e fixar os seus olhos no agente que está participando do processo de comunicação [51].

6.1 Coordenação dos olhos e pescoço - experimento 1

Foram realizados experimentos iniciais com o objetivo de avaliar o comportamento do algoritmo de vergência, em relação ao controle da coordenação dos movimentos dos olhos e do pescoço do agente, levando os olhos de uma situação inicial de vergência (ou não) para outra final de vergência e ao término do processo os olhos estão fixados no ponto de interesse desejado. A Figura 6.1 mostra o agente em uma situação inicial antes da vergência.

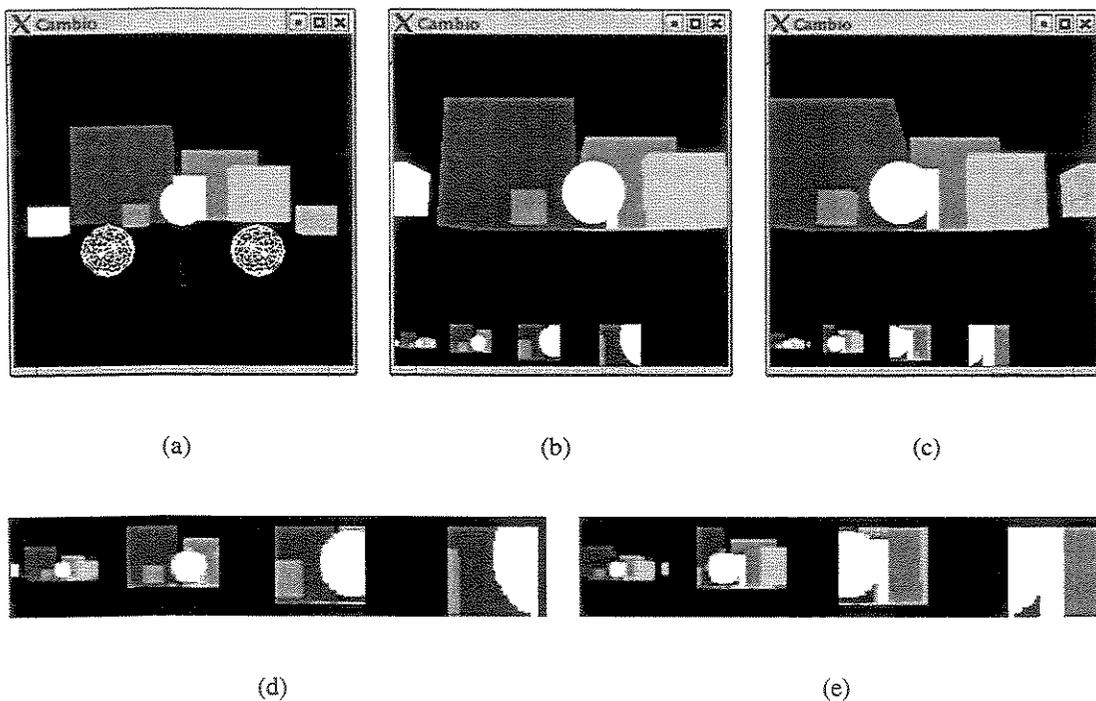


Figura 6.1: Estas figuras mostram o Câmbio em uma situação inicial, onde olhos estão paralelos um com o outro. (a) Situação inicial antes da vergência. (b) e (c) Imagens dos olhos esquerdo e direito. (d) e (f) Imagens das fóveas dos olhos esquerdo e direito.

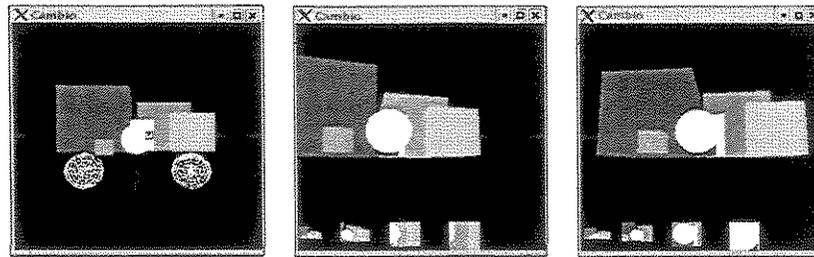
A sequência de imagens mostradas na Figura 6.2 podem ser usadas para avaliar o algoritmo, em relação a coordenação dos movimentos dos olhos e pescoço do agente, quando são necessários movimentos de vergência horizontal, vertical ou de ambos em conjunto. Os movimentos de vergência foram realizados na sequência mostrada por estas imagens, ou seja, a primeira imagem é o ponto de vergência inicial, a segunda o final da primeira e o inicial da ter-

ceira e assim por diante. De acordo com a posição do ponto de interesse (pequeno quadrado nas imagens mais à esquerda), as seguintes situações foram analisadas: 1) somente o movimento dos olhos é necessário para colocar o ponto de interesse no centro dos olhos e 2) a posição do ponto de interesse requer um movimento do pescoço, seguido de outro dos olhos.

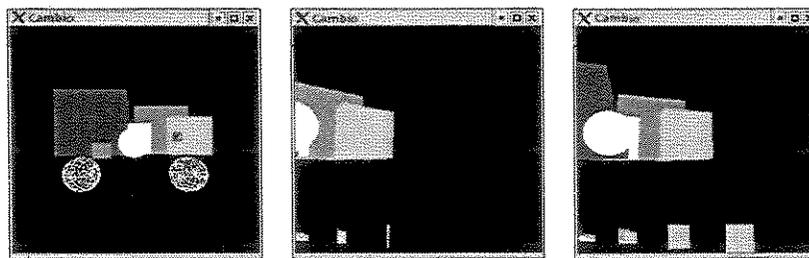
Na Figura 6.2(a), o Câmbio sai de sua posição inicial e converge na esfera central, nesta situação o agente vai até o limite máximo de vergência horizontal dos olhos. Na Figuras 6.2(b), o ponto de interesse está fora do alcance do limite de vergência horizontal esquerdo dos olhos. Nesta situação é necessário um movimento do pescoço seguido de outro dos olhos, para fixá-lo no ponto de interesse final. Na Figura 6.2(c) foram realizados, movimentos de vergência vertical e horizontal do pescoço e dos olhos. E na Figura 6.2(d), o ponto de interesse está fora do alcance do limite de vergência vertical dos olhos. Nesta situação é necessário um movimento vertical do pescoço seguido de outro dos olhos para fixá-los no ponto de interesse desejado.

As imagens originais das câmeras esquerda e direita (segunda e terceira imagens mais a direita, em cada sequência), as quais são utilizadas para a obtenção das imagens em multi-resolução (fóvea), são obtidas do *frameBuffer* do OpenGL. Como esta memória guarda apenas a informação final que será exibida na tela, algumas vezes a imagem que está no centro do olho não está no centro da tela. Com isto, a imagem em multi-resolução com maior informação visual (aquela que enquadra o objeto de interesse), algumas vezes não está na última imagem (a de maior resolução) e sim em uma das imagens anteriores, como pode ser visto na Figura 6.2(c).

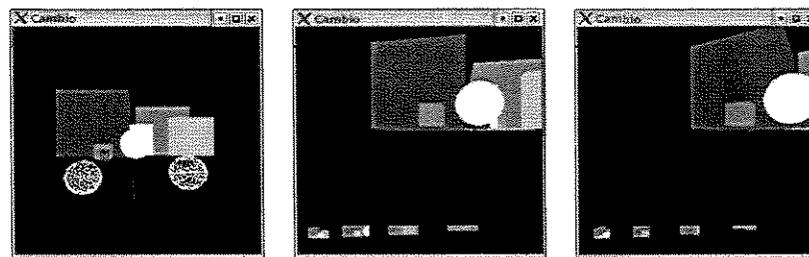
As situações de vergência impostas neste experimento mostram que o algoritmo é capaz de coordenar a atenção do agente virtual, levando os olhos de uma situação inicial de vergência a outra final. E, pode-se constatar que ao final do processo, os olhos encontram-se vergidos no objeto desejado. A Tabela 6.1 mostra a sequência destes deslocamentos de vergência em valores de coordenadas de mundo, de acordo com os objetivos (ponto de interesse) passados para o agente.



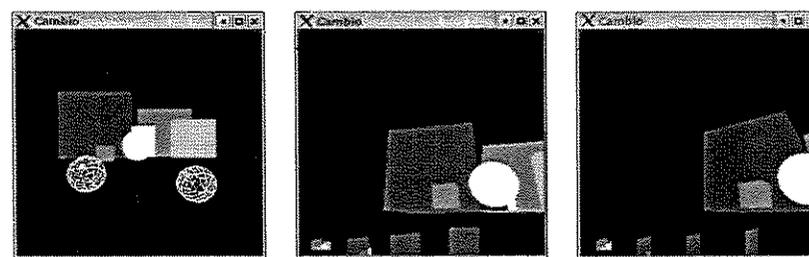
(a)



(b)



(c)



(d)

Figura 6.2: Movimentos de vergência horizontais e verticais e verticais, horizontais. O ponto de interesse está representado por um pequeno quadrado.

	VER_1	VER_2	VER_3	VER_4
Ponto de tela	(138,127)	(169,128)	(92,111)	(91,151)
Ponto de Interesse ("z-buffer")	(5,0,-122)	(28,0,-150)	(-20,-9,-124)	(-23,15,-143)
Direção inicial olho E (X,Y)	(0,0)	(15,0)	(10,0)	(10,-9)
Direção inicial olho D (X,Y)	(0,0)	(-5,0)	(-10,0)	(-10,-9)
Direção inicial Pescoço (X,Y)	(0,0)	(0,0)	(28,0)	(-20,-9)
Direção final olho E (X,Y)	(15,0)	(10,0)	(10,-9)	(10,15)
Direção final olho D (X,Y)	(-5,0)	(-10,0)	(-10,-9)	(-10,15)
Direção final pescoço (X,Y)	(0,0)	(28,0)	(-20,-9)	(-23,15)

Tabela 6.1: Tabela dos valores das coordenadas de deslocamento do experimento 1. Esta tabela mostra a sequência dos deslocamentos realizados, no espaço cartesiano (X,Y), durante as mudanças finais de vergências, de acordo com o novo ponto de interesse passado para o agente. VER = Vergência.

6.2 Vergindo em locais diferentes - experimento 2

Nesta seção, foram realizados experimentos com o objetivo de avaliar o comportamento do algoritmo de vergência, quando objetos posicionados em lugares diferentes do ambiente, são colocados como objetivo para o processo de atenção do agente virtual.

O processo de vergência dá início com os olhos do agente paralelos um ao outro Figura 6.3(a). A esfera central (Figura 6.3(b)) é passada como o primeiro objetivo para o sistema visual do agente, sendo este o ponto de interesse passado para as duas câmeras. Depois de uma checagem interna, para avaliar quais partes da cabeça (olhos ou pescoço) devem se movimentar primeiro, o controlador interno coordena o sistema gráfico e visual levando a cabeça do agente de uma situação inicial, para outra final de vergência. No final do processo os olhos do agente encontram-se fixados no ponto desejado. Esta situação pode ser constatada pelas duas imagens mais a direita da Figura 6.3(b) (imagens dos olhos esquerdo e direito, vergidos no mesmo ponto de interesse). Notamos a diferença, em relação as imagens da fóvea da Figura 6.3(a) (agente não vergido) com as imagens da fóvea da Figura 6.3(b) (agente vergido).

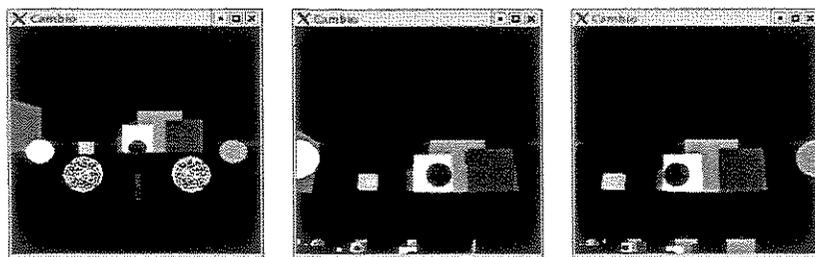
Depois que os olhos estão fixados na esfera central, o processo de vergência é requisitado mais duas vezes para posicionar a visão do agente em duas regiões distintas do ambiente. Na primeira situação, o agente é levado de uma situação inicial (Figura 6.3(b)) para outra final (Figura 6.3(c)) e depois o processo de vergência movimentava novamente o sistema visual e gráfico para outra situação final de vergência (Figura 6.3(d)). Novamente, as imagens da fóvea mostram que os olhos do agente conseguiram alcançar os seus objetivos, ou seja, a atenção do agente está centrada na esfera mais à esquerda (objetivo da vergência) e depois na esfera mais à direita (Figura 6.3(d)). A Tabela 6.2 mostra a sequência destes deslocamentos de vergência em valores de coordenadas de mundo, de acordo com os objetivos (ponto de interesse) passados para o agente.

	VER_1	VER_2	VER_3	VER_4
Ponto de tela	(0,0)	(126,113)	(27,111)	(221,111)
Ponto de Interesse ("Z-buffer")	(0,0,0)	(0,-8,-134)	(-42,-7,-94)	(40,-7,-95)
Direção inicial olho E (X,Y)	(0,0)	(0,0)	(10, -8)	(10,-7)
Direção inicial olho D (X,Y)	(0,0)	(0,0)	(-10,8)	(-10,7)
Direção inicial Pescoço (X, Y)	(0,0)	(0,0)	(0,0)	(-42,0)
Direção final olho E (X,Y)	(0,0)	(10, -8)	(10,-7)	(10,-7)
Direção final olho D (X,Y)	(0,0)	(-10,8)	(-10,7)	(-10,-7)
Direção final pescoço (X, Y)	(0,0)	(0,0)	(-42,0)	(45,0)

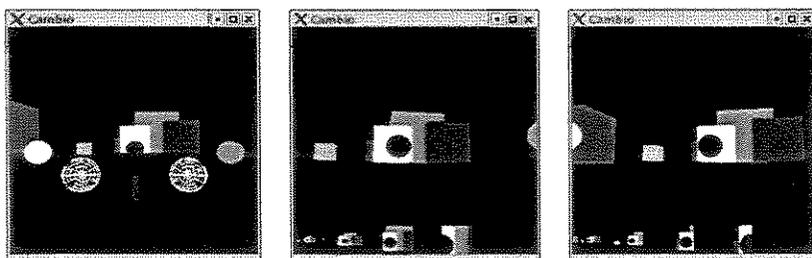
Tabela 6.2: Tabela dos valores das coordenadas de deslocamento do experimento 2. Esta tabela mostra a sequência dos deslocamentos realizados, no espaço cartesiano (X,Y), durante as mudanças finais de vergências, de acordo com o novo ponto de interesse passado para o agente. VER = Vergência.

6.3 Mudança de atenção em grupo - experimento 3

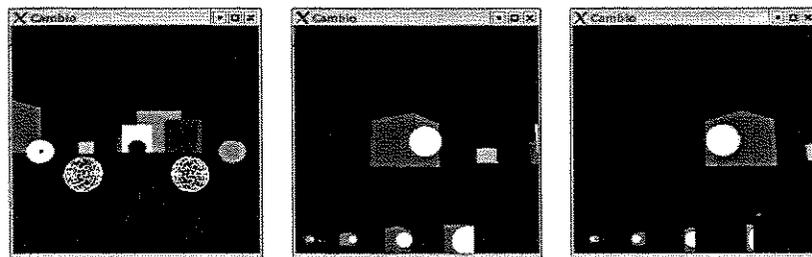
A comunicação visual é uma característica importante para ambientes que necessitam de um maior realismo entre a comunicação dos seus usuários [12, 51]. Por este motivo, o objetivo deste



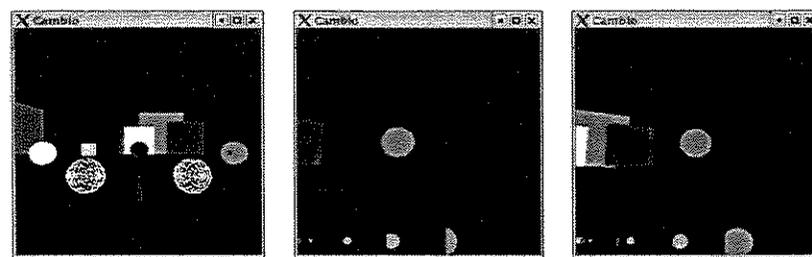
(a)



(b)



(c)



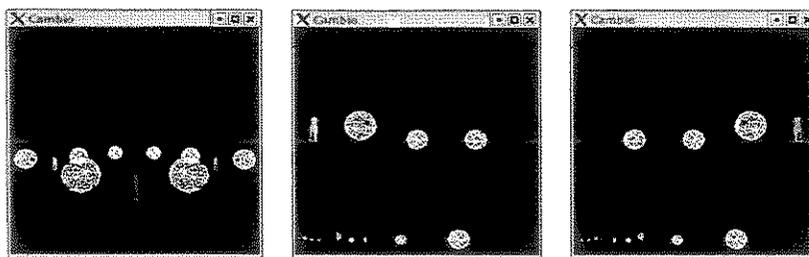
(d)

Figura 6.3: Movimentos de vergência variados. Estas imagens mostram o agente convergindo os olhos em objetos posicionados em diferentes lugar do ambiente.

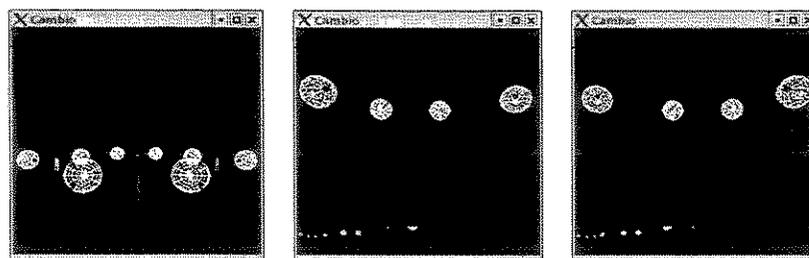
experimento é mostrar que o algoritmo de vergência pode inserir mais realismo no controle da atenção do agente, quanto ele está inserido em um ambiente virtual de grupo.

Uma situação que pode vir a acontecer em um ambiente virtual, que permita uma participação em grupo, é que um locutor (representado por um humano virtual) pode estabelecer uma comunicação visual com algum dos outros participantes. Neste caso, o algoritmo de vergência pode ser utilizado para dar mais realismo a cena, fazendo com que o direcionamento da cabeça do locutor, juntamente com o posicionamento direcional dos seus olhos, forneçam uma informação visual capaz de expressar uma situação de comunicação verbal. A Figura 6.4 mostra uma sequência de imagens que ilustram esta situação.

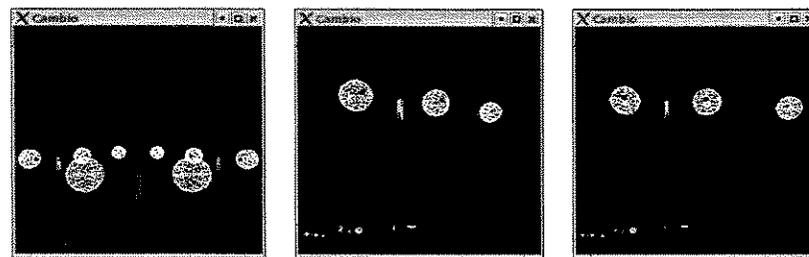
Na situação inicial do sistema (Figura 6.4(a)) os avatares não estão se comunicando. Nas Figuras 6.4(b), 6.4(c) e 6.4(d), o locutor (representado pelo avatar com o pescoço mais escuro) está estabelecendo uma comunicação visual com os participantes que se encontram a sua frente e do seu lado esquerdo e direito. As imagens dos olhos esquerdo e direito mostram esta situação.



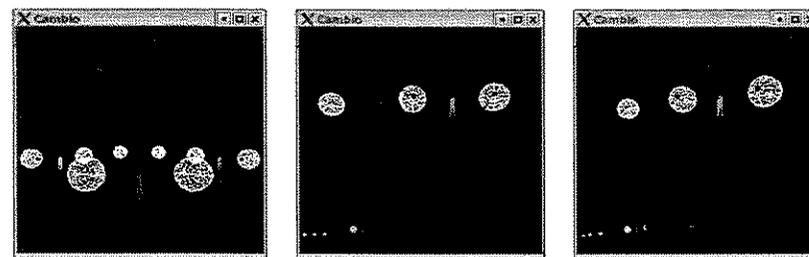
(a)



(b)



(c)



(d)

Figura 6.4: Esta figura representa um ambiente virtual que permite uma comunicação visual entre os seus participantes.

Capítulo 7

Conclusões e Perspectivas

Neste trabalho foram estudados os problemas referentes ao controle da atenção de humanos virtuais, admitindo que a visão do agente é uma visão sintética. Vimos que a atenção, em relação a localização de objetos no ambiente, pode ser dividida em dois grupos importantes. Neste trabalho, estudamos o que trata da fixação dos olhos do agente em uma determinada região do ambiente. Para alcançar este objetivo (fixação dos olhos), propomos um algoritmo de vergência que é capaz de controlar a atenção de humanos virtuais, usando técnicas de visão sintética, e ao final do processo de vergência os olhos do agente são fixados no local desejado.

Até o momento, os trabalhos que abordam o problema da vergência em humanos virtuais utilizaram algoritmos que são executados em plataformas reais, quando os principais objetivos eram a simulação realística dos movimentos dos olhos e a fixação. O grande problema enfrentado por estes algoritmos é a quantidade de cálculos de correlação necessários para o processo de deslocamento dos olhos do agente (manter a vergência).

O algoritmo de vergência proposto contorna este problema e é capaz de coordenar os movimentos de vergência vertical e horizontal dos olhos, levando-os a qualquer ponto ou região de interesse do ambiente. O controle dos deslocamentos dos sistemas gráfico e visual são baseados em simples cálculos geométricos, tendo como base a discretização dos pixels do ambiente onde o agente está inserido.

7.1 Dificuldades Encontradas

Em relação ao algoritmo de vergência proposto, a tarefa mais complexa foi a implementação da coordenação dos movimentos de vergência horizontal dos olhos (a vergência vertical também foi difícil), pois um controle mais bem elaborado teve de ser desenvolvido para não deixar o agente sair da área de vergência, ou seja, uma vez nesta área o agente não pode mais perder a vergência. Isto significa prover uma estabilização ao sistema resultante, demonstrando a convergência do algoritmo no ponto de vergência.

Ainda, o desenvolvimento da plataforma de simulação consumiu um tempo considerável neste trabalho, tendo sido necessários vários estudos para a implementação do mecanismo de vergência, baseados em movimentos humanos (movimentos sacádicos). A principal dificuldade associada a este desenvolvimento foi estabelecer uma política de controle, para os movimentos de vergência dos olhos, que fosse capaz de agrupar o sistema gráfico ao visual do agente de modo que o último estivesse subordinado ao primeiro, mas que esta subordinação não afetasse o sistema visual.

7.2 Limitações do Algoritmo e do Sistema Gráfico

O algoritmo de vergência proposto não é adequado para uma plataforma real, devido ao fato do mesmo precisar da informação do *Z-buffer*. No caso real, um algoritmo de reconstrução estéreo teria que ser desenvolvido para esta finalidade. Entretanto, o uso em plataformas reais não foi o objetivo deste trabalho, mas sim o uso em plataformas virtuais e especificamente humanos virtuais e também para favorecer maior realismo em uma comunicação não verbal.

Os movimentos de deslocamento do sistema gráfico podem ser melhorados, como visto na seção 5.5.6, para movimentos com os mesmos valores de coordenada X e com diferentes valores de profundidade Z , os ângulos de vergência não mudam. Nesta mesma seção foi proposta uma solução para tornar o algoritmo mais geral. Por outro lado, se for usada a informação da distância focal f (o OpenGL não fornece o valor deste parâmetro) a solução inicial pode

também ser corrigida, apenas com o acréscimo desta medida e o valor da coordenada Z poderia ser usado. Esta idéia será testada posteriormente.

7.3 Objetivos e Contribuições Alcançados

Consideramos que o principal objetivo desta dissertação de mestrado foi alcançado, o qual foi propor um algoritmo de vergência simples, baseado em técnicas de visão sintética, que pode ser usado e executado sem grandes dificuldades de processamento por agentes virtuais, mais especificamente humanos virtuais dotados da habilidade de visão. Pois, a grande dificuldade enfrentada pelos algoritmos atuais é o tempo de processamento para o cálculo da disparidade. Tal algoritmo foi proposto, implementado e validado por uma plataforma de simulação.

Nosso algoritmo trabalha com a informação de profundidade do *Z-buffer* e com simples cálculos geométricos, usados para controlar o sistema gráfico e visual do agente. Logo, apesar de não ter sido feito nenhum estudo comparativo de desempenho, nosso algoritmo é capaz de ser executado em tempo real, dependendo, claro, da complexidade do ambiente e do tempo para ser gerado o *Z-buffer*. Admitindo-se que a grande dificuldade para que a execução em tempo real venha a ocorrer sejam os maciços cálculos de correlação usados para a obtenção da disparidade, que no nosso caso não são necessários.

Além do algoritmo de vergência, também foram propostos: um modelo para a fóvea, uma plataforma de simulação e uma arquitetura mais geral para o acréscimo de outras funcionalidade ao agente.

O modelo da fóvea é baseado em imagens multi-resolução, ele pode ser usado para agilizar o processo de extração de características, uma vez que imagens com tamanhos menores do que a original podem ser utilizadas durante o processo de convolução.

Inicialmente, a plataforma foi desenvolvida para validar o algoritmo de vergência, mas ela também pode ser usada como uma ferramenta de simulação para o desenvolvimento e testes de algoritmos visuais [9, 50, 51, 52, 10] ou para o desenvolvimento de modelos matemáticos de visão. O uso de uma plataforma sintética é importante, por exemplo, para a validação rápida de

algoritmos caso não se tenha uma plataforma real para os testes.

Durante o levantamento bibliográfico não foram encontrados trabalhos que abordassem o mesmo assunto. Por isso, acreditamos que esta dissertação é uma contribuição original para a área de humanos virtuais dotados de visão sintética, onde o controle da atenção seja necessário.

7.4 Trabalhos Futuros

As duas primeiras extensões que podem ser realizadas são a de estudar melhor e implementar efetivamente o algoritmo mais genérico de deslocamento do sistema gráfico e o acréscimo do movimento sacádico, visto a sua importância em tarefas de reconhecimento. Neste caso, cada vez mais realístico seria o ambiente virtual.

Outras extensões podem ser realizadas em relação aos módulos da arquitetura mostrada na Figura 4.1, ou seja, estes módulos podem ser desenvolvidos e implementados para que no final, tenha-se um agente com um maior número de funcionalidades e uma plataforma de simulação mais completa, podendo ser utilizada para o desenvolvimento de humanos virtuais dotados da capacidade de navegação autônoma, utilizando para isto animação comportamental. Ainda, modelos de esqueleto ou corpo e outras funcionalidades podem ser colocadas no humano virtual para torná-lo mais realístico.

Referências Bibliográficas

- [1] Victor J.S. Bernardino A. Active vergence behavior. *VisLab - Technical Report*, October 1995.
- [2] Bordeaux C., Boulic R., and Thalmann D. An efficient and flexible perception pipeline for autonomous agents. *EUROGRAPHICS*, 18(3), 1999.
- [3] Breazed C., Edsinger A., Fitzpatrick P., and Scassellati B. Active vision for sociable robots. *IEEE Transactions on Man, Cybernetics and Systems*, XX(Y):1–12, 2000.
- [4] Breazed C., Edsinger A., Fitzpatrick P., Scassellati B., and Varchavskaia P. Social constraints on animate vision. *Humanoid Robotics. IEEE Intelligent Systems*, pages 2–7, 2000.
- [5] Capurro C., Panerai F., Grosso E., and Sandini G. Binocular active vision system using space variant sensors: Exploiting autonomous behaviours for space applications. *Proc. of International Conference on Digital Signal Processing*, 1993.
- [6] Gonzalez R. C. and Woods R. E. *Digital Image Processing*. Addison-Wesley Publishing Company, 2 edition, 1992.
- [7] Peters C. and Sullivan C. A memory model for autonomous virtual humans. *In Proceedings of Eurographics Irish Chapter Workshop*, pages 21–26, 2002.
- [8] Peters C. and Sullivan C. Synthetic vision and memory for autonomous virtual humans. *COMPUTER GRAPHICS Forum*, 21(4):743–752(10), November 2002.

- [9] Aranha C.C., Carvalho S.R., and Gonçalves L.M.G. Cambio: realistic three dimensional simulation of humanoids based on computer vision and robotics. *IEEE. XV SIBGRAPI*, pages 388–395, 2002.
- [10] Aranha C.C., Carvalho S.R., and Gonçalves L.M.G. A realistic simulated robot for vision algorithms. *The 10-th Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (Journal of WSCG'2003)*, 11(1), 2003.
- [11] Çapin T. K., Pandizic I. S., Thalmann N. M., and Thanmann D. Realistic avatars and autonomous virtual humans in vlnet networked virtual environments. *Avatars and Autonomous Virtual Humans in VLN Virtual Worlds on the Internet. J. Vince and Society. Los Alamitos*, 1998.
- [12] Çapin T. K., Pandizic I. S., Thalmann N. M., and Thanmann D. *Avatars in Networked Virtual Environments*. Jhon Willey & Sons, 1 edition, 1999.
- [13] Foley J. D., Dam A. V., Feiner S. K., and Hughes J. F. *Computer Graphics: Principles and Practice in C*. Addison-Wesley, 2 edition, 1996.
- [14] Terzopoulos D. Artificial life for computer graphics. *COMMUNICATIONS OF THE ACM*, 42(8):33–42, August 1999.
- [15] Terzopoulos D. Perceptive agents and systems in virtual reality. *Source Virtual Reality Software and Technology (VRST'2003) Proceedings of the ACM symposium on Virtual reality software and technology*, pages 1–3, october 2003.
- [16] Terzopoulos D. and Rabie F.T. Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research*, 1(1):2–19, September 2000.
- [17] Thalmann D. Animating autonomous virtual humans in virtual reality. *Proc. IFIP Conference*, 3:177–184, 1994.
- [18] Rabie F.T. *Animat Vision Active Vision in Artificial Animal*. PhD thesis, Graduate Department of Electrical and Computer Science. University of Toronto, 1999.

- [19] Rabie F.T. and Terzopoulos D. Active perception in virtual humans. *Proceedings of the thirteenth Canadian Vision Interface Conference (VI 2000)*, May 2000.
- [20] Gonçalves L. M. G. *Um Sistema de Controle Robótico para a Integração de Informação Sensorial Multimodo*. PhD thesis, Engenharia de Sistemas e Computação. Universidade Federal do Rio de Janeiro. COOPE/UFRJ., Novembro 1999.
- [21] Ballard D. H. and Brown C. M. *Computer Vision*. Prentice-Hall, 1 edition, 1982.
- [22] Noser H. and Thalmann D. Sensor based synthetic actors in a tennis game simulation. *Proc. Computer Graphics International, IEEE Computer Society*, pages 189–198, 1997.
- [23] Romeny B. M. H. *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1 edition, 1994.
- [24] Berthold K. P. HORN. *Robot Vision*. MIT Press, 1986.
- [25] Laurent Itti. *Models of Bottom-Up and Top-Down Visual Attention*. PhD thesis, California Institute of Technology, January 2000.
- [26] Batista J., Peixoto P, and Araújo H. Real-time vergence and binocular gaze control. *IROS97 - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France*, pages 7–11, September 1997.
- [27] Coombs D. J. *Real-time Gaze Holding in Binocular Robot Vision*. PhD thesis, Department of Computer Science. University of Rochester, June 1992.
- [28] Kuffner J. and Latombe J.C. Fast synthetic vision, memory, and learning models for virtual humans. *IEEE. Proc. of Computer Animation*, pages 118–127, 1999.
- [29] Piater J., Ramamritham K., and Grupen R. Learning real-time strategies for binocular vergence. *Technical Report. Computer Science Department, Lederle Graduate Research Center. University of Massachusetts*, February 1999.
- [30] Yamato J. Tracking moving object by stereo vision head with vergence for humanoid robot. Master's thesis, Department of Electrical Engineering and Computer Science. Massachusetts Institute of Technology, May 1998.

- [31] Itti L. and Koch C. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research. Elsevier Science*, 40:1489–1506, 2000.
- [32] Itti L., Dhavale N., and Pighin F. Realistic avatar eye and head animation using a neurobiological model of visual attention. *Proc. SPIE 48th Annual International Symposium on Optical Science and Technology*, 2003.
- [33] Gonçalves L.M.G. Towards a learning model for feature integration in attention control. *In Proc. of IEEE MultiSensor Fusion Conference (MFI 2001)*, August 2001.
- [34] Gonçalves L.M.G., Distanto C., Oliveira A., Wheeler D., and Grupen R.A. Neural mechanisms for learning of attention control and pattern categorization as basis for robot cognition. *In Proc. of International Intelligent Robots and Systems Conference (IROS'2000).IEEE Computer Society*, 1:70–75, October/November 2000.
- [35] Gonçalves L.M.G. and Grupen R.A. Integrating attention and categorization behaviors in robotics. *In Proc. of the 6th International Conference on Intelligent and Autonomous Systems. IEEE Computer Society (IAS-6), Venise (Italie), 25-27 Juillet 2000 Intelligent Autonomous Systems 6, IOS Press*, pages 867–874, July 2000.
- [36] Gonçalves L.M.G., Grupen R.A., Oliveira A., Wheeler D., and Fagg A. Tracing patterns and attention: Humanoid robot cognition. *The Intelligent Systems and their Applications*, 15(4):70–77, July/August 2000.
- [37] Haralick R. M. and Shapiro L. G. *Computer and Robot Vision*. Addison-Wesley, 1 edition, 1992.
- [38] D. Marr and T. Poggio. Cooperative computation of stereo disparity. In *Science*, volume 194, pages 209–236, 1976.
- [39] Rajesh M.M.H., Rao P.N., Zelinsky G.J., and Ballard D.H. Eye movements in iconic visual search. *Vision Research*, 42(11):1447–63, May 2002.
- [40] Noser N. and Thalmann D. Synthetic vision and audition for digital actors. *Proc. Eurographics*, pages 325–336, 1995.

- [41] Noser N., Renault O., Thalmann D., and Thalmann N. M. Navigation for digital actors based on synthetic vision, memory and learning. *Computer Graphics*, 19(1):7–19, 1995.
- [42] Sebe N and Lew M. S. *Principles of Visual Information Retrieval*. Springer, 2001.
- [43] Renault O., Thalmann D., and Thalmann N.M. A vision-based approach to behavioural animation. *Visualization and Computer Animation*, 1:18–21, 1990.
- [44] Kalra P, Thalmann N.M., Moccozet L., and Sannier G. Real-time animation of realistic virtual humans. *Animating Virtual Humans. IEEE*, pages 42–56, September/October 1998.
- [45] Rao P.N. and Ballard D.H. An active vision architecture based on iconic representations. *Artificial Intelligence Magazine*, 78(1-2), October 1995.
- [46] Manzotti R., Gasteratos A., Metta G., and Sandini G. Disparity estimation on log-polar images and vergence control. *Computer Vision and Image Understanding*, 89:97–117, June 2001.
- [47] Rao R.P.N. and Ballard D. An active vision architecture based on iconic vision representations. *Artificial Intelligence Journal*, 79:461–505, 1995.
- [48] Balcisoy S. and Thalmann D. Interaction between real and virtual humans in augmented reality. *In Proc. of IEEE Computer Animation*, pages 31–38, 1997.
- [49] Singhal S. and Zyda M. *Networked Virtual Environments: Design and Implementation*. Addison-Wesley, 1 edition, 1999.
- [50] Carvalho S.R., Aranha C.C., and Gonçalves L.M.G. Uma ferramenta 3d aplicada ao desenvolvimento de algoritmos perceptuais em humanóides. *V Simpósio Brasileiro de Realidade Virtual*, 2002.
- [51] Carvalho S.R. and Gonçalves L.M.G. Avatares que olham para o ponto certo. *VI Simpósio Brasileiro de Realidade Virtual*, 2003.
- [52] Carvalho S.R. and Gonçalves L.M.G. Realistic simulation of humanoids based on computer vision and robotics. *Scientia*, 13(2), 2003.

- [53] Bow S. T. *Pattern Recognition - Applications to Large Data-Set Problems*, volume 1. Electrical Engineering and Eletronics, 1 edition, 1984.