

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” - UNIVEM
PROGRAMA DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

FÁBIO ALEXANDRE CARAVIERI MODESTO

IMPLEMENTAÇÃO DE UMA FERRAMENTA PARA GERAÇÃO DE
MOVIMENTOS PARAMETRIZADOS PARA HUMANOS VIRTUAIS

Marília

2005

Fábio Alexandre Caravieri Modesto

IMPLEMENTAÇÃO DE UMA FERRAMENTA PARA GERAÇÃO DE
MOVIMENTOS PARAMETRIZADOS PARA HUMANOS VIRTUAIS

Dissertação apresentada ao Programa de
Mestrado do Centro Universitário
Eurípides de Marília, mantido pela
Fundação de Ensino Eurípides Soares da
Rocha, para obtenção do Título de
Mestre em Ciência da Computação.

Orientador:
Prof. Dr. José Remo Ferreira Brega

Marília
2005

FÁBIO ALEXANDRE CARAVIERI MODESTO

IMPLEMENTAÇÃO DE UMA FERRAMENTA PARA GERAÇÃO DE
MOVIMENTOS PARAMETRIZADO PARA HUMANOS VIRTUAIS

Banca examinadora da dissertação apresentada ao Programa de Mestrado da UNIVEM,/F.E.E.S.R., para obtenção do Título de Mestre em Ciência da Computação.

Resultado: _____

ORIENTADOR: Prof. Dr. José Remo Ferreira Brega

1º EXAMINADOR: _____

2º EXAMINADOR: _____

Marília, ____ de _____ de 2005.

*Dedico este trabalho aos
meus Avós, que com
simplicidade me mostraram
que apesar dos obstáculos
que a vida nos impõe
nunca devemos desistir*

AGRADECIMENTOS

A realização deste trabalho não poderia ter acontecido de forma satisfatória se não fosse o empenho e suporte irrestrito prestado pelo professor Dr. José Remo Ferreira Brega, que sempre se mostrou atencioso no decorrer deste trabalho, apontando sempre uma direção a seguir. A ele, meu muito obrigado.

Agradeço aos professores que de forma simples em direta além de transmitir conhecimentos dos livros, passaram a essência de ser um pesquisador.

Agradeço a minha família que nem sempre compreendeu a horas dedicada a pesquisa, mas sempre estiveram me apoiando.

Agradeço ao amigo César Augusto Cusin, um irmão, que nos momentos fáceis e difíceis sempre esteve próximo.

Agradeço pela ajuda dos amigos Thiago Luiz Parrillo Rizzo e Larissa Pavarni, meu sincero agradecimento.

Agradeço aos membros da banca por disporem de seu precioso tempo na leitura deste trabalho.

A imaginação é algo infinito, só cabe o homem saber utilizá-la.

Julio Verne

MODESTO, Fábio Alexandre Caravieri. **Implementação de uma Ferramenta para Geração de Movimentos Parametrizados para Humanos Virtuais**. 2005. 90 f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

RESUMO

A sensação de imersão é uma característica fundamental na interface entre usuários e Ambientes Virtuais. E isso não é exceção para aplicação em Humanos Virtuais. Para que isso aconteça é desejado que os movimentos gerados se aproximem a movimentos reais. Este trabalho apresenta conceitos de Realidade Virtual, mostrando sua tecnologia e enfatizando a utilização de HV e criação de movimentos utilizando a técnica de *keyframing* de criação de movimento de forma parametrizada. E também é apresentada uma proposta de implementação de uma ferramenta que utiliza o *keyframing*, interpolação de quadro, para criação de movimentos.

Palavras-chave: Realidade Virtual, HV, Agentes Virtuais, Avatar, Parametrização, interpolação, keyframing

MODESTO, Fábio Alexandre Caravieri. **Implementação de uma Ferramenta para Geração de Movimentos Parametrizados para Humanos Virtuais**. 2005. 90 f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

ABSTRACT

The immersion sensation is a fundamental characteristic in the interface between users and Virtual Environment. And that is not exception for application of Virtual Humans. For that to happen to movements generated that approach to movements Real they are wanted. This work presents concepts of Virtual Reality, showing your technology and emphasizing the virtual humans' use and creation of movements using the technique of keyframing of creation of movement of form parametrized. It is it is also presented a study proposal and implement of a tool that it uses the keyframing, picture interpolation, for creation of movements.

Keywords: Virtual Reality, Virtual Human, Virtual Agent, Avatar, Parametrized, interpolation, keyframing

LISTA DE FIGURAS

Figura 1.1: Mostra um Humano Virtual usado para testes ergonômicos de projetos de veículos baseados por computador.....	21
Figura 1.2: Mostra interação de HV em um AV (BINDIGANAVALE, 1999)	21
Figura 1.3: Mostra um Humano Virtual em um ambiente de treinamento (RICKEL; JOHNSON 1998).....	22
Figura 1.4: Visão geral superior-esquerda de F-22 baia de armas (BADLER et al 2002).....	22
Figura 1.5: Mostra um Humano Virtual como um guia turístico (BADLER <i>et. al</i> 2003)	23
Figura 1.6: Aplicações de HV, Tecnologia e Ciência (BADLER 1997).....	25
Figura 2.1: Modelo de Humano Poliedral	29
Figura 2.2: Deformação do dorso (Azuola <i>et al.</i> , 1994).....	29
Figura 2.3: “Corpo Liso” do Jack como o ocupante virtual de um modelo CAD de um helicóptero apache (BADLER, 1997)	31
Figura 2.4: Comandos de ações médicas (da esquerda para direita, de cima para baixo) tais como: checando pupilas, estabilizando coluna cervical, checando passagem de ar, checando respiração, auscultando (STANSFIELD et al., 1997)	33
Figura 2.5: Segurando um martelo (LEVISON, 1996)	36
Figura 2.6: Retirando/movendo o martelo (LEVISON, 1996).....	36
Figura 2.7: Comunicação por gestos (DOUVILLE, 1996)	36
Figura 2.8: Exemplo de um algoritmo de OSR (LEVISON, 1996)	37

Figura 2.9: Exemplo de comandos OSR (LEVISON, 1996).....	38
Figura 2.10: Exemplo de comandos OSR (LEVISON, 1996).....	38
Figura 2.11: O JACK PRESENTER aponta para o slide e para o expectador (BADLER, 1997)	40
Figura 2.12: O JACK PRESENTER aponta e olha para um furacão em um mapa do tempo (BADLER, 1997)	41
Figura 2.13: STEVE puxa o medidor de óleo (RICKEL; JOHSON, 1998).....	44
Figura 2.14: Arquitetura do mundo virtual. Mostrando componentes de um agente e um participante (humano) (RICKEL; JOHSON, 1998)	44
Figura 2.15: Os três módulos principais do STEVE e as informações que eles enviam e recebem.....	46
Figura 3.1: Figura articulada com 30 graus de liberdade (MULTON et al., 1999).....	48
Figura 3.2: Fases do personagem no movimento de andar (MULTON et al., 1999).....	49
Figura 3.3: Sinopse de BOULIC para um sistema de locomoção (Multon net al., 1999).....	55
Figura 4.2: <i>Módulo Ambiente</i>	60
Figura 4.3 : Controles superiores <i>módulo ambiente</i>	61
Figura 4.4: Especificação do grafo de cena da classe Ambiente.....	62
Figura 4.5: Fragmento do Código da implementação do controle Camera1	62
Figura 4.6: Visão da Camera1	63
Figura 4.7: Seleção da Camera1	63
Figura 4.8: Codificação da Seleção das Câmeras.....	64

Figura 4.9 Especificação das coordenadas gerais da Classe Camera	64
Figura 4.10: Especificação da Hierarquia do Avatar.....	65
Figura 4.11: Diagrama de Classes da Hierarquia do Avatar.	65
Figura 4.12: <i>Módulo</i> Controle	66
Figura 4.13: Exemplo de Seleção do braço do avatar	67
Figura 4.13: Exemplo de fragmento de Código Fonte que altera a cor o membro selecionado	67
Figura 4.14: Exemplo do fragmento do Código Fonte da recuperação dos valores das posições dos eixos X, Y e Z	68
Figura 4.15: Exemplo do fragmento do código fonte responsável pela movimentação do braço esquerdo.....	68
Figura 4.16: <i>Módulo</i> KeyFramer	69
Figura 4.17: Classe KeyFrame	70
Figura 4.18: Esquema de funcionamento	71
Figura 4.19: Passagem das coordenadas X, Y e Z para o <i>módulo</i> KeyFrame.	71
Figura 4.20: Pseudocódigo do método passagemX()	72
Figura 4.21: Trecho do código fonte do método que carrega do vetor de <i>frames</i> chaves.....	72
Figura 4.22: Layout do arquivo de trabalho da aplicação.	73
Figura 5.1: Exemplo da roscopia em animação 3D (LAMBERTON 2004)	74
Figura 5.2: O avatar simulando um abraço	76
Figura 5.3: Movimento Braço Direito	77

Figura 5.4: Gráfico do Movimento do Braço Direito.....	77
Figura 5.5: Movimento Braço Esquerdo	77
Figura 5.6: Gráfico do Movimento do Braço Esquerdo	78
Figura 5.7: Movimento do Antebraço Direito.....	78
Figura 5.8: Gráfico do Movimento do Antebraço Direito.....	78
Figura 5.9: Movimento do AnteBraço Esquerdo.....	78
Figura 5.10: Gráfico do Movimento do Antebraço Esquerdo.....	79
Figura 5.7: Demonstração da seqüência de movimento para comparação de qualidade de acordo com o número de <i>frames</i>	80
Figura 5.8: Gráfico de desempenho com variação de tempo constantes.....	80
Figura 5.9: Variando <i>frame</i> por segundo.....	81
Figura 5.10: Gráfico de movimento com variação de <i>frames</i> por segundo e quadros chaves .	81
Figura 5.11: Exemplo de um arquivo <i>filmstrip</i>	82
Figura 5.12: Exemplo da geração de movimento utilizando rotoscopia "a"	83
Figura 5.13: Curva de movimento de simulação usando rotoscopia - letra "a"	84
Figura 5.14: Exemplo de geração de movimento usando a letra "i"	84
Figura 5.15: Curva de movimento de simulação usando rotoscopia - letra "i"	85
Figura 5.16: Curva de movimento da simulação sem rotoscopia.....	86
Figura 5.17: Gráfico comparativo entre as duas configurações utiizadas.	87

LISTA DE TABELAS

Tabela 1: <i>Softwares</i> utilizados para Desenvolvimento.....	75
Tabela 2: Recursos de <i>Hardware</i> para Desenvolvimento	75
Tabela 3: Recursos de <i>Hardware</i> utilizados para testes.....	76
Tabela 4: Dados de movimentos simulados com rotoscopia - letra "a"	83
Tabela 5 : Dados de movimentos simulados com rotoscopia - letra "i"	85
Tabela 6: Dados de movimento simulado sem rotocospia	86

LISTA DE ABREVIATURAS E SIGLAS

2D:	Bidimensional
3D:	Tridimensional
API:	Application Programming Interface (Interface Programada da Aplicação)
AV:	Ambiente Virtual
DIS:	Bistributed Interactive Simulation (Simulação Interativa Distribuída)
DOF:	Degree of Freedom (Grau de Liberdade)
FFD:	Free-Form Deformation (Deformação de Formas Livres)
GUI	Graphical User Interface (Interface Gráfica do Usuário)
HMD:	Head-Mouted Display
HTTP:	Hyper Text Transfer Protocol
HV:	Humano Virtual
ISO:	International Standards Organization (Organização Internacional de Padrões)
LBS:	Linguagem Brasileiras de Sinais
IFS:	Footstrike Esquerdo
LIBRAS:	Linguagem Brasileiras de Sinais
LOA:	Level of Articulation (Nível de Articulação)
ITO:	Takeoff esquerdo
rFS:	Footstrike direito
rTO:	Takeoff Direito
RTP:	Transport Protocol for Real-Time
RV:	Realidade Virtual

SMIL: Synchronized Multimedia Integration Language

STEVE: Training Expert for Virtual Environments

SVG: Scalable Vector Graphics

LISTA DE SIMBOLOS MATEMÁTICOS

- $\Delta\theta$: Deslocamento Angular
- ΔX : Movimento da ligação final de uma cadeia (um pé, por exemplo)
especificado pelo animador
- $J\Delta\theta$: Relação da Matriz Jacobiana com Deslocamento Angular

SUMÁRIO

INTRODUÇÃO.....	18
Proposta do Trabalho.....	18
Descrição dos capítulos	19
1. HUMANOS VIRTUAIS	21
2. AGENTES VIRTUAIS E AVATARES	27
2.1 Aspectos Importantes	27
2.2 Exemplos de Humanos Virtuais	39
3. TÉCNICAS ESTUDADAS PARA GERAÇÃO DE MOVIMENTOS PARAMETRIZADOS.....	48
3.1 Humanos caminhando	49
3.2 Animação Cinemática	51
3.3 Dinâmica.....	55
4. GERADOR DE MOVIMENTOS PARAMETRIZADOS PARA HV	57
4.1 Objetivos principais.....	57
4.2 Aspectos Gerais da Ferramenta	59
5. TESTE DO PROTÓTIPO E ANÁLISE DE RESULTADOS OBTIDOS	74
5.1 Recursos Utilizados	75
5.2 Testes realizados e resultados obtidos	76
5.3 Avaliação da qualidade do movimento utilizando roscopia.....	82
CONCLUSÕES.....	88
Aspectos gerais do trabalho.....	88
Contribuições.....	90
Trabalhos Futuros	91
REFERÊNCIAS	92

INTRODUÇÃO

A Realidade Virtual (RV), como é denominada por aqueles que a pesquisam, é uma área em pleno desenvolvimento no escopo da Ciência da Computação.

Fornecer uma poderosa interface entre “usuário”, aplicação e dados. Sua meta é representar o mundo real de forma sintética (computacional) e tridimensional, a isto se denomina Ambiente Virtual (AV). Este deve oferecer três características básicas e essenciais a seu(s) usuário(s), que são imersão, interação e envolvimento.

As aplicações de RV estão presentes nas mais diversas áreas, por exemplo: engenharia, simulação, visualização científica, medicina, educação, entretenimento, etc. Podem ser usadas na Internet, em ambientes mono-usuários e ambientes multi-usuários.

Para esses exemplos citados, existem associados mundos virtuais específicos e nestes, objetos que contemplam entidades correspondentes do mundo real, como: casas, veículos, ferramentas, máquinas e principalmente pessoas.

Quando se representam pessoas no mundo virtual, é comumente usado o termo Humano Virtual (BADLER, 1997). Eles interagem em tempo real com os componentes do mundo virtual, bem como com os humanos reais, implementando atributos de uma pessoa real, como: gestos, movimentos, meios de se comunicar, aparência, etc.

Proposta do Trabalho

O uso de Humanos Virtuais (HV) é indiscutivelmente variado, desde estudantes de história visitando a Grécia antiga, a pacientes que sofrem de fobias possam ensaiar situações que causem estas fobias, na busca de provável cura, porém em um ambiente seguro. Sua utilidade está na educação, entretenimento, terapia e treinamento (GRATCH *et al.*, 2002).

O treinamento com o uso de recursos de RV visa uma sensação de envolvimento entre o usuário e o Humano Virtual com o qual está interagindo. Que pode estar passando instruções de como uma máquina opera, manutenção de um sistema mecânico complexo ou procedimentos de exercícios físicos em uma sessão de fisioterapia (RICKEL; JOHSON, 1998) (BALDER *et al.*, 2002) (SABARISH *et al.*, 2005).

Os propósitos desta dissertação são:

1. Levantar os principais requisitos e tecnologias para o desenvolvimento de ferramentas para geração movimentos para HV, utilizando a técnica conhecida como *keyframing*;
2. Implementar um protótipo para implementação de uma ferramenta utilizando esta tecnologia de geração de movimentos parametrizados para HV.

Descrição dos capítulos

Além da Introdução, esta dissertação está organizada em mais oito capítulos, descritos a seguir:

- CAPÍTULO 1: serão discutidas as características, funcionalidades e utilização de HV de forma geral;
- CAPÍTULO 2: serão definidos dentro do mundo virtual, as categorias existentes de HV, definindo a diferença entre Agentes Virtuais e Avatares, bem como, quais características são consideradas importantes para a implementação de ferramenta de geração de movimentos e apresentação de exemplos de HV existente: JACK e STEVE;
- CAPÍTULO 3: serão apresentadas algumas técnicas existentes para a geração de movimentos parametrizados;

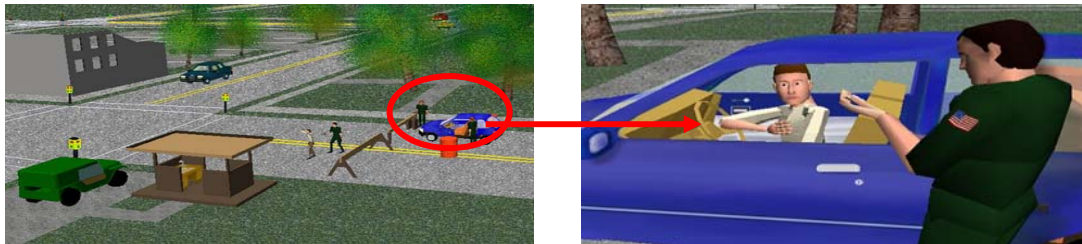
- CAPÍTULO 4: apresenta uma visão geral da arquitetura da ferramenta proposta, bem como as funcionalidades providas e as implementações, utilizando-se dos serviços desta ferramenta.
- CAPÍTULO 5: descreve-se os recursos de *hardware* e *software* utilizados, os testes realizados, bem com a análise dos resultados obtidos.
- CONCLUSÃO: conclusões e contribuições do projeto
- REFERÊNCIAS: forneceram a base teórica utilizada neste trabalho.

1. HUMANOS VIRTUAIS

Humanos Virtuais (HV) (BADLER, 1997) são modelos de pessoas que podem ser usadas como substitutos de “*peças reais*” em testes ergonômicos baseados em computador para projetos de veículos, trabalho de campo, ferramentas, linhas de produção, e para a simulação prévia da construção destes (Figura 1.1), para representações humanas do participante ou de outros em ambientes de virtuais de tempo real (Figura 1.2).



Figura 1.1: Mostra um Humano Virtual usado para testes ergonômicos de projetos de veículos baseados por computador.



(a)

(b)

Figura 1.2: Mostra interação de HV em um AV (BINDIGANAVALÉ, 2004)

Pode-se utilizar HV na área de educação ou treinamento (RICKEL; JOHNSON, 1998) (CONDE et.al 2003) (BADLER 2002), permitindo a criação de tutoriais orientados a tarefas, onde guiam e dialogam com o estudante, onde o objetivo do humano virtual é transmitir ao estudante algum desempenho físico, uma operação procedimental como

o funcionamento de um equipamento (Figura 1.3), ou até mesmo transmitir conteúdos didáticos como História, Geografia ou Física.

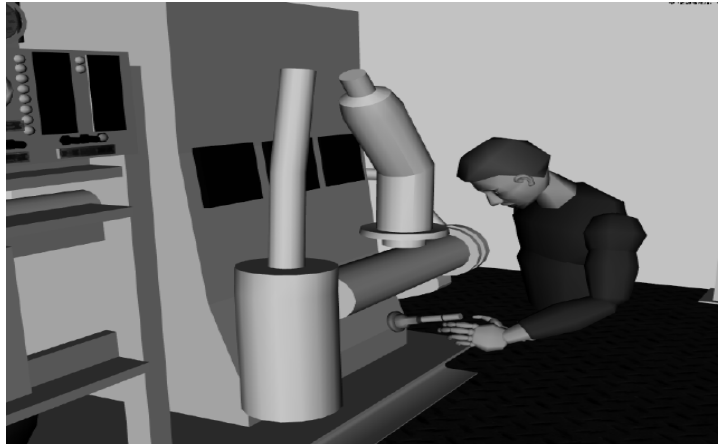


Figura 1.3: Mostra um Humano Virtual em um ambiente de treinamento (RICKEL; JOHNSON 1998)

HV usados para treinamento de instruções e procedimento, de manutenção de sistemas complexos tendem a diminuir erros, complexidades das tarefas (tempo) e atualização de instruções manuais. Por exemplo, caças militares, são muito complexos, e sua manutenção pode exceder muitas vezes o seu valor de compra (BADLER et al 2002) (Figura 1.4).

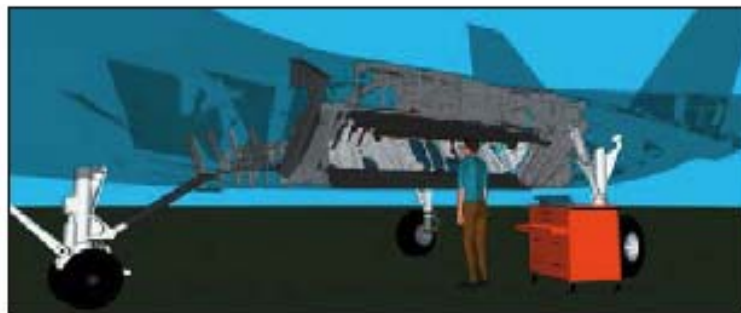


Figura 1.4: Visão geral superior-esquerda de F-22 baia de armas (BADLER et al 2002)

Devido aos avanços tecnológicos recentes permitiu-se passar um grau de veracidade aos olhos do usuário, através das aplicações em tempo real. Esse grau de veracidade pode ser dividido em três conceitos que traduzem a própria RV, que são: imersão, interação e envolvimento possível entre os HV e os usuários que estão utilizando suas aplicações. Isso já

foi explorado por outras formas de mídia, por exemplo, o cinema, especialmente considerando-se a animação, onde os animadores tentam passar a idéia de um personagem real. Para ilustrar isso, os animadores da Disney, que deram um grande passo neste sentido. A Disney tem uma clássica referência a seu trabalho, que diz: “A animação da Disney faz com que os expectadores acreditem nos personagens, suas aventuras, suas tristezas fazendo as pessoas rirem e às vezes chorarem. Existe um ingrediente especial em nosso tipo de animação que produzem desenhos que parecem pensar, tomar decisões e atuar neste ambiente; isso é que cria a ilusão de vida” (THOMAS; JOHSON, 1981).

Porém HV tem uma característica diferente dos personagens animados ou de jogos, ao propor representar os seres humanos virtualmente, não só representando seu exterior como: movimento, mas apresentar reação, emoção, interação e tomada de decisão que o aproxima do “natural” (BADLER et. al 2003) (BATES 1994) (BATES *et al.*, 1992) (GRATCH 2002) (Figura 1.5).

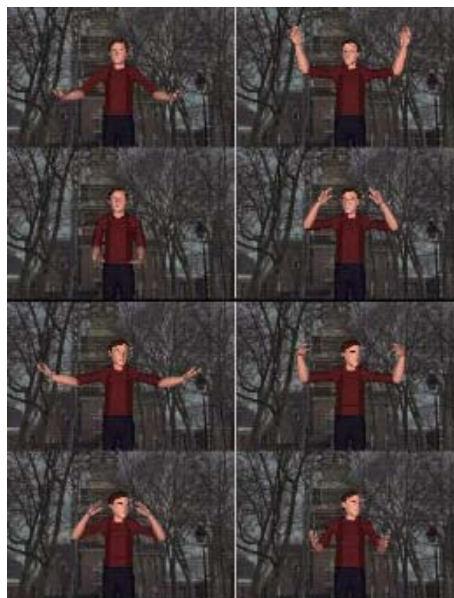


Figura 1.5: Mostra um Humano Virtual como um guia turístico (BADLER *et. al* 2003)

Há muitas razões para que se projete modelos humanos especialistas que individualmente otimizem seu caráter, desempenho, inteligência e assim por diante (BATES *et al.*, 1992).

Segundo Gratch (2002) as pessoas infundem em seus comportamentos verbais e não verbais, emoções e personalidade, e modelar tal comportamento é essencial para se construir HV realísticos.

Assim, pesquisadores chegaram a uma abordagem computacional que pode ser dividida em comunicação-dirigida e simulação. A primeira, quando o HV toma uma expressão emocional de acordo com a ação do usuário. Por exemplo, fazendo um pedido com uma feição triste, pode provocar que o ouvinte se compadeça e se motive a dar uma resposta afirmativa. Aplicações tutoriais seguem a abordagem de comunicação dirigida, onde as expressões emocionais são usadas intencionalmente com o objetivo de motivar os estudantes e aumentar seu nível de aprendizagem. A segunda abordagem, visa uma simulação da emoção "verdadeira" (ao invés de ser simplesmente carregada). Na simulação são utilizados teorias de avaliação de emoção; elas vêem as emoções como reações de validade para eventos e, objetivos que guiam as metas dos agentes, padrões e atitudes. Por exemplo um agente que assiste a jogada campeã de um jogo, ele deve reagir de formas distintas de acordo com time de sua preferência (GRATCH *et al.*, 2002) (ANDRÉ *et al.*, 2000).

Quando se pensa na construção de um HV, deve-se ter claro os diversos domínios de utilização e seu relacionamento com o ser humano (mundo real), como a seguir (BATES *et al.*, 1992) (CASSELL *et al.*, 1994).

- **Análise de Fatores Humanos:** altura, habilidades, comportamento, habilidade em desenvolver certa atividade/trabalho e uso designado no ambiente.
- **Agentes e Avatares de Tempo-Real:** nenhum ser humano é igual a outro. Pessoas trazem consigo um conjunto de características que não são inatas, que criam, preservam ou aprimoram através da comunicação com outros indivíduos, e isto é uma variante em cada indivíduo. Também não pode-se negligenciar que cada um

vem de uma origem étnica diferente, tais fatores podem refletir nos HV desde a aparência, até a ação/reação e escolhas.

- ❑ **Geração e Compreensão de Instruções:** o ser humano se comunica com outros em contexto rico de linguagem, sensações e experiências vivenciadas que precisam ser transportadas ao avatar e/ou agente virtual gerados.
- ❑ **Simulação Bio-Médica:** o corpo humano é uma estrutura física complexa, bem como suas funções. Para entender o comportamento humano, respostas psicológicas e fraquezas; deve-se representar um bio-modelo.
- ❑ **Análise de Forma e Movimento:** ter o entendimento que o que se vê e sente no mundo real deve ser transportado até o modelo do mundo físico, bem como a forma geométrica e deformações do objeto.

Pode-se verificar a relação entre HV e os domínios de utilizações na Figura 1.6.

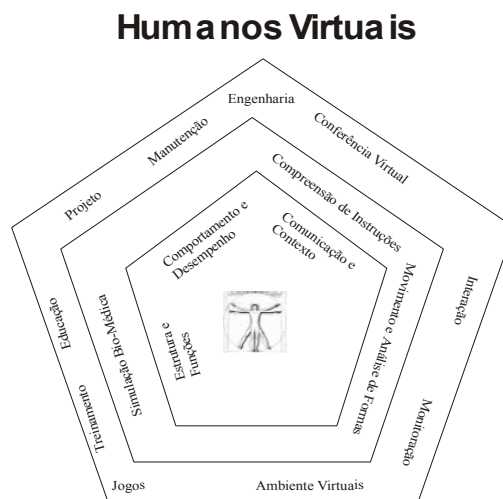


Figura 1.6: Aplicações de HV, Tecnologia e Ciência (BADLER 1997)

Quando se propõe a construir modelos de HV, existem vários critérios de fidelidade virtual que devem ser observados. Precisa-se ter o entendimento, que este tipo de aplicação, é dependente, por exemplo, da altura da pessoa/humano, habilidades, limite de força e articulações .

Também se deve contemplar que o comportamento de tempo real é essencial a jogos, treinamentos, simulações genéricas e militares.

Diferentes aplicações necessitam de diferentes tipos de fidelidade virtual. Ao se pensar em uma aplicação envolvendo HV, alguns pontos devem ser considerados. Estas considerações devem verificar o que se espera do Humano Virtual, o que se espera visualizar nele e quais as características são importantes para seu sucesso.

O projeto de HV, quanto a modelagem, deve obedecer alguns critérios, que podem ser divididos em:

- **Aparência:** que pode ir desde uma caricatura até um modelo físico;
- **Funcionalidade:** ações estilizadas até limitações humanas;
- **Tempo:** geração *off-line* à produção em tempo real;
- **Autonomia:** animação direcionada à ações inteligentes e;
- **Individualidade:** pessoa específica várias personalidades.

Hoje existe uma necessidade crescente de incorporar atributos físicos e biomecânicos aos modelos de certas aplicações, isso é facilmente obtido com os Humanos 3D. Pode-se criar HV com limitações funcionais que fogem dos modelos tradicionais, que são meras caricaturas, inspiradas em seres humanos reais (BADLER 1997) (HODGINS et al 1995) (LE et al 2003)

A movimentação dos HV pode ser obtida tanto por captura de movimentos quanto por síntese computacional (BADLER 1997) (BAERLOCHER; BOULIC 2001).

Os HV têm começado a exibir um estágio prematuro de autonomia e inteligência, em sua reação e tomada de decisão em “cenas específicas” (mudança de ambientes onde antes eram forçados a ter movimentos fixos) (ROUSSEAU; HAYES-ROTH, 1996) (MAES et al., 1995).

2. AGENTES VIRTUAIS E AVATARES

Qualquer que seja o ambiente virtual (AV) que se contemple, este é povoado por objetos. Os quais podem ser fixos ou móveis. Nestes últimos, encontram-se os HV.

Pode-se classificar os HVs em duas categorias distintas: Agentes Virtuais e Avatares.

Segundo Wilcox (1998), avatar é toda a representação de personagem controlada por “alguém” do mundo real.

Segundo Thalmann *et. al.* (1999), avatar é uma representação gráfica de uma pessoa real no AV, junto com seu comportamento e agente virtual é um processo de software que tem comportamento autônomo, não necessariamente representado por um entidade gráfica.

Badler (1997) considera agente virtual como um humano virtual que é criado e controlado por software. E avatar, o humano virtual controlado por um participante “vivo”.

2.1 Aspectos Importantes

Com o objetivo de propor uma biblioteca de movimentos para HV, serão considerados cinco aspectos importantes para o seu projeto.

São eles: aparência, movimento, interatividade, comunicação e gestos. Cada um deles é especificado nas seções a seguir.

2.1.1 Aparência

Pode-se representar um avatar por um ícone 2D, por desenhos (como personagens de desenho animado), por vídeo composto, objetos 3D ou por corpos completos. O interesse desta pesquisa justamente está na última representação.

O processo de criação de modelos de HV introduz uma série de pontos a serem analisados, tais como a escala do corpo, cálculo da segmentação do mesmo, centro de juntas, e a aspiração de se conseguir uma aparência real ao humano virtual.

Antropometria é um ramo da antropologia física, que estuda as medidas do corpo humano. Cada medida é indispensável para se criar uma interface homem/máquina. O conhecimento do tamanho do corpo que varia de acordo com um projeto populacional particular é crítico. Se, por exemplo, se considerarem itens como roupas, equipamentos de segurança e locais de trabalhos são projetados para acomodar seus usuários destinatários. Assim, há a preocupação em criar um humano virtual antropometricamente correto (AZUOLA *et al.*, 1994).

Pode-se representar HV por um conjunto de segmentos geométricos conectados a juntas. Nem todas as juntas do corpo humano são representadas por uma junta no modelo. Juntas que podem ser representadas nos modelos, são juntas sinoviais (elemento que permite o movimento das articulações no corpo humano) com um a três ângulos de liberdade (*DOF – Degree of Freedom*) e limites especificados. Módulos especiais cuidam de modelos mais complicados (juntas complexas) como ombro e espinha. (AZUOLA *et al.*, 1994).

HV podem ser concebidos também utilizando modelos poliedrais, neste caso eles são compostos por uma geometria simples totalizando por volta de dois mil poliedros de fácil manipulação e escalabilidade (Figura 2.1) (AZUOLA *et al.*, 1994)



Figura 2.1: Modelo de Humano Poliedral

Outro método de se criar modelos virtuais de humanos são os modelos realistas. Utiliza a técnica de animação de ponto de vista. São modelos com mais de vinte mil polígonos cada e assim necessitam equipamentos rápidos e com grande poder de processamento para sua manipulação (Figura 2.2) (AZUOLA et al., 1994). Algumas técnicas de escalonamento linear podem ser usadas para gerar modelos de diferentes tamanhos (AZUOLA et al., 1994).



Figura 2.2: Deformação do dorso (Azuola et al., 1994)

2.1.2 Movimento

O interesse maior é a representação do movimento tal como acontece com os humanos do mundo real, para que naturalmente se obtenha estruturas de juntas e superfícies mais realistas possíveis.

Talvez isso seja preferível para projetar movimentos para modelos com um grande número de juntas, diminuir o detalhamento de modelagem e o detalhe da junta, de acordo com a exigência da aplicação (GRANIERI *et al.*, 1995).

Pode-se citar o agente Jack (BADLER 1997), que é formado por um modelo poligonal com segmentos rígidos e juntas móveis, possuindo limites bastantes precisos para utilização ergonômicas (BADLER 1997).

Por exemplo, o modelo de um soldado com 110 polígonos, com seu desenho simplificado, cor e textura mapeada pode ser aceito como uma representação de um soldado.

Porém, o modelo de um ocupante de um veículo precisa caracterizar visualmente movimentos adequados, tanto visualmente quanto na geometria de juntas. O modelo dessa representação tem que ser agradável e passar a ilusão da forma mais aceitável possível. Este modelo deve representar quem está no controle daquele veículo ao expectador. Da mesma forma quando este vai ao cinema e vê seres e veículos oriundos dos caros efeitos especiais e sai convencido da veracidade destes.

Para ilustrar isso pode-se citar como exemplo, novamente, o agente Jack (AZUOLA *et al.*, 1994), composto de um corpo liso e utiliza formas de deformação livre (*FFD – Free Form Deformation*), o que permite aumentar o nível de mobilidade do humano virtual (Figura 2.3).

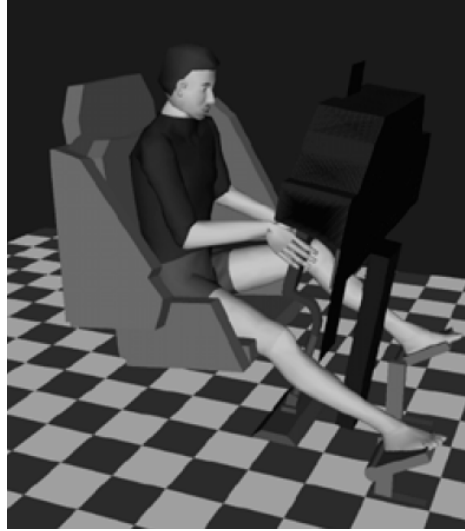


Figura 2.3: “Corpo Liso” do Jack como o ocupante virtual de um modelo CAD de um helicóptero apache (BADLER, 1997)

Várias são as técnicas para se gerar o movimento do agente. Segundo Badler (1997) os movimentos sintetizados em relações aos outros tipos, permitem a geração de movimentos para mais de uma junta ao mesmo tempo. Por exemplo, pode-se armazenar uma série de mudanças dos ângulos das juntas em uma espécie de base de dados ou biblioteca assim um movimento específico pode ser recuperado e reutilizado em tempo real quando necessário (GRANIERI et al., 1995). Não é permitida nenhuma divergência das transformações pré-armazenadas, mesmo o corpo podendo ser reorientado e ser reutilizado com velocidade variada.

As vantagens dos movimentos pré-armazenados estão na velocidade, armazenamento e segurança algorítmica (reduzindo a computação). Destacando-se o tamanho reduzido do conjunto de parâmetros (menos informação a ser obtida e comunicada) e conseqüentemente, generalizando o controle de movimento: caminhar, procurar, olhar, etc.

2.1.3 Interatividade

Para se controlar a interatividade do HV, seja ele um avatar ou agente virtual, existem diferentes técnicas para geração de movimento. Qualquer que seja, ela deve proporcionar um modo de ativar movimento ou comportamento desejado do humano virtual.

Pode se especificar movimentos de forma bem simples através de rastreamento por meio de sensores, rastreamento de resultado final (por exemplo, cinemática inversa), invocação externa (como menus), voz, botões de ação (sintetiza ou interpreta dados pré armazenados). Pode-se também realizar o movimento através de dispositivos de entrada de dados como mouses, teclados, luvas, entre outros. Enquanto há progresso considerável em técnicas de visualização computadorizada utilizando captura de movimentos humanos, unindo a mobilidade do usuário com movimentos genéricos (AZUOLA et al., 1994) (ESSA; PENTLAND, 1995) (DECARLO; METAXAS, 1996) (KAKADIARIS; METAXAS, 1996) (GRATCH et al, 2002).

Outra forma de interação é a utilização da fala, como meio de interação, criando um banco lingüístico que acione ações argumentadas e elaboradas através de síntese de movimento de baixo nível. Por exemplo, esta técnica pode ser usada com grande vantagem em aplicações de ambientes virtuais imersivos como o *Simulated Casuaties and Medics for Emergency Training* (STANSFIELD et al., 1997).

Neste caso a arquitetura do Humano Virtual deve compreender um interpretador semântico de informações e até mesmo um reativador de baixo nível no gerador de movimentos que permita movimentos genéricos e um ambiente sensível a contexto (Figura 2.4).

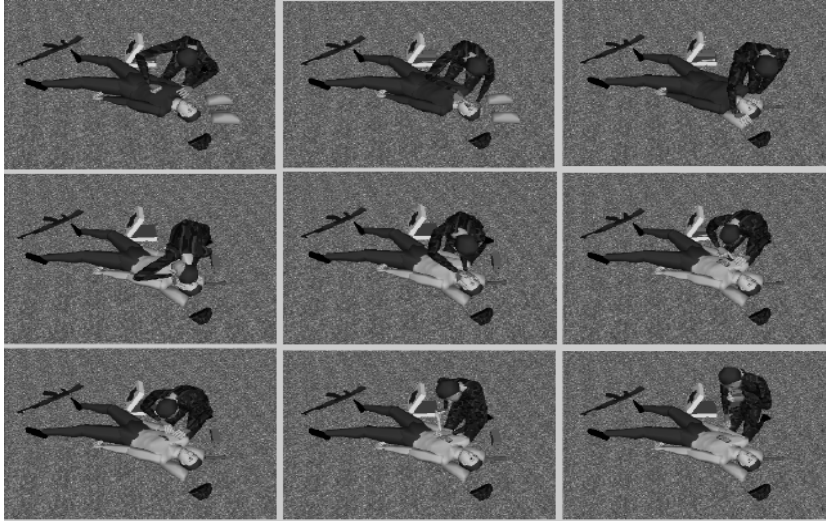


Figura 2.4: Comandos de ações médicas (da esquerda para direita, de cima para baixo) tais como: checando pupilas, estabilizando coluna cervical, checando passagem de ar, checando respiração, auscultando (STANSFIELD et al., 1997)

2.1.4 Comunicação

Com o aparecimento dos extraordinários efeitos especiais e jogos interativos, a criação de um agente que incorpore características autônomas em tempo real seria uma das últimas fronteiras da computação gráfica (AZUOLA et al., 1994).

Por exemplo, alguns seres retratados, sejam qual for: personagens de animação, personagens de ficção, ou personagens conhecidos (*cartoons*), técnicas de animação computadorizada empregadas tem um grande apelo visual. Qualquer que seja a finalidade, utiliza-se técnicas de interpolação quadros chave através de parâmetros ou movimentos capturados do real. E tais técnicas são interativas, *off-line* e dependente das habilidades do animador.

É esperado que agentes que tenham um comportamento autônomo, sejam entidades que respondam ao estímulo humano em tempo real. Com comportamentos percebidos como apropriados a interação, e necessários aos participantes. Esse conjunto de comportamentos do agente pode ser criado com antecedência e pode ser recuperado e modificado em tempo real de acordo com a necessidade. Por exemplo, alguns jogos restringem os movimentos que o

personagem pode realizar sobre o comando do usuário, obrigando este a interagir em um ambiente de finitas escolhas.

Porém como um paradoxo aos jogos, a interação entre pessoas reais está imersa em um ambiente de infinitas possibilidades. Suas ações são realmente em tempo real, não são pré-determinadas, um largo conteúdo de possibilidades, alto teor de contextualidade, comportamento que pode se dizer que é representado de forma sutil, comunicação, etc.

A comunicação entre as pessoas, realizada das mais variadas formas, apresentando um meio de comunicação multimodal e isso pode ser codificado em vários níveis de comunicação (fala, gestos, sinais, etc.).

Esses níveis podem ser incorporados aos HV, assim podendo-se se comunicar por fala, movimentos labiais, movimentos dos olhos, movimentos de mãos, postura corpórea, um movimento de um braço (ou outro membro) e até sinais de mãos. Durante uma interação interpessoal, muitas pessoas se desvencilham de ações corpóreas ou gestos enquanto estão falando, por exemplo. E de contra partida outra pessoa pode utilizar uma linguagem de sinais não verbalizada para se comunicar.

Para a maioria das pessoas, a geração de todas as ações do corpo associadas com a comunicação são de origem natural e subconsciente. Nos caso dos atores, em algumas citações deve-se, ou tem-se a necessidade de criar comportamentos que não são “naturais” e isto requer uma certa habilidade, treinamento e disciplina. Até mesmo para outros tipos de pessoas, contudo, em suas interações cotidianas podem exigir uma “atuação” consciente: isto é, a administração do comportamento manifestado encobre as emoções verdadeiras, exceto a face, dramatizando ou simplesmente mentindo.

Assim, o conteúdo comunicativo retratado por um HV é função de todos os meios de comunicação disponíveis do corpo e conseqüentemente a percepção do observador do estado interior que o HV se encontra e depende das inter-relações destes meios de

comunicação, bem como o seu conteúdo. Talvez para um “ator”, no caso do rastreamento ou animador experiente, isso pode ser óbvio, mas para alguém sem experiência, isso pode causar o fracasso no controle dos gestos ou tentar corrigir o comportamento deste personagem.

No caso do ator, pode acontecer dele não interiorizar as metas e motivações de seu personagem, o bastante, para usar a motricidade do seu próprio corpo para manifestar este direcionamento interno como comportamento apropriado. Já no caso dos animadores, eles têm que ter o conhecimento que todos os aspectos de seus personagens tem que ser consistentes com o estado mental desejado desde voz, forma e movimentos a serem controlados para um produto final. Mas não pode-se sondar a mente ou verificar o estado psicológico de um personagem animado estabelecendo um diálogo.

Quando se usa representação de pessoas em tempo real, isso deixa de ser utopia para ser algo possível. Pessoas reais apresentam múltiplos comportamentos (honestidade, sinceridade, papéis sociais, relacionamentos, poder, intenções, etc.); existe uma estreita e importante relação entre estes comportamentos e o que pode ser chamado de “veracidade”. Um personagem tem veracidade se ele passa emoção e um estado mental pela observação de seu comportamento. Uma vez que cada comportamento utiliza-se um meio ou canal para apresentar como o estado interno do HV se encontra, eles não devem estar em conflito e devem apresentar uma visão consistente.

2.1.5 Gestos

Existem duas ações, pelo menos, de primordial importância, realizadas por braços humanos, quando se fala da representação de HV. São elas: a capacidade buscar e segurar objetos (DOUVILLE et al., 1996) (Figura 2.5 e 2.6) e comunicação, como uma extensão ou

complemento da faculdade da fala através de sinais, gestos, etc. (Figura 2.7) (CASSEL et al., 1997).

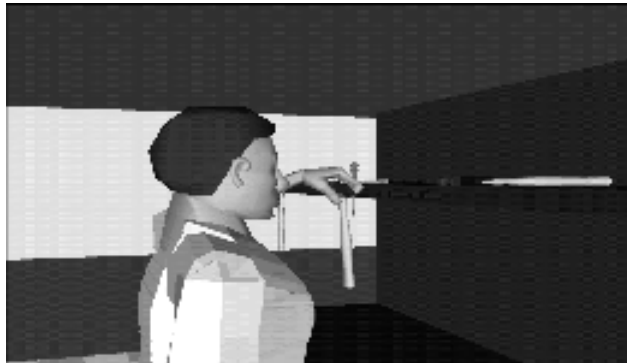


Figura 2.5: Segurando um martelo (LEVISON, 1996)



Figura 2.6: Retirando/movendo o martelo (LEVISON, 1996)

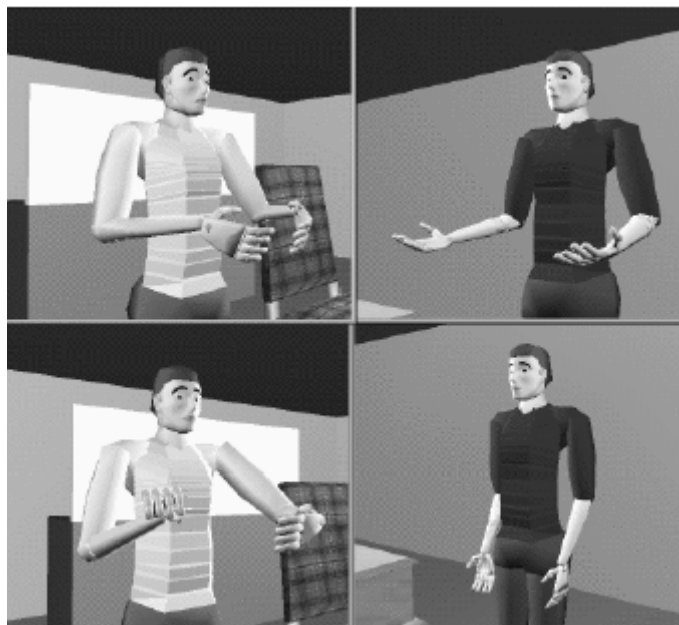


Figura 2.7: Comunicação por gestos (DOUVILLE, 1996)

A primeira função é uma consequência da experiência humana (vivência). Para transferir isso para os HV, Badler (1997) utiliza uma Tabela Relacional de Objetos Específicos (OSR) (LEVISON, 1996). A qual possui campos que demarcam o local a ser agarrado objeto, orientado o formato específico da mão do HV, direção que a mão deve ter para pegar o objeto e outras funções necessárias para esta ação.

O OSR é definido manualmente e o agente deve buscar um dado local para segurar o objeto. Para isso, utiliza-se um vetor de aproximação como guia para cinemática inversa para mover o braço e sabe qual formato da mão apropriado será resultante da colocação dos dedos no objeto. E a própria mão se fecha através de informação da geometria local e detecção de colisão.

Na Figura 2.8 há um exemplo de algoritmo usado pelo OSR .

```

Select action outline
Expand action outline to OSRactions
For the set of OSRactions, do:
    Add instance information
    Calculate range of values for parameters
    Use purpose to select parameter settings
    Test feasibility of OSRaction
If all OSRactions are feasible, replace with Behaviors
Report feasibility judgement/discrepancy list
  
```

Figura 2.8: Exemplo de um algoritmo de OSR (LEVISON, 1996)

Um exemplo de sua utilização é, por exemplo, o uso de ferramentas: martelo, serrote, chave de fenda e parafusos. Pode-se considerar que o martelo e serrote são segurados da mesma maneira enquanto os outros não. Assim o OSR para este exemplo poderia ser (Figuras 2.9 e 2.10):

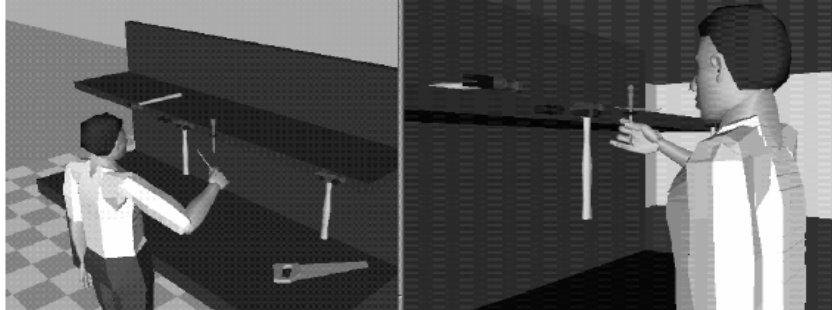


Figura 2.9: Exemplo de comandos OSR (LEVISON, 1996)

```
(TAGrasp jack hammer (TAUse))
(TAGrasp jack socket-wrench (TAUse))
(TAGrasp jack screwdriver (TAUse))
```

Figura 2.10: Exemplo de comandos OSR (LEVISON, 1996)

Um agente segurando um martelo tem modo adequado de segurar esse objeto e se ele segurasse uma chave de fenda seria outra forma.

A segunda função de gestos é a comunicação não verbal, pode-se considerar que gestos podem representar objetos do ambiente, apontando indicações de localizações aos participantes do ambiente, bem como indicar a estes localizações de outros que também estão no AV. E usar os gestos também aqui, para intensificar a comunicação entre os inter-locutores (CASSEL *et. al.*, 1994). Existem pesquisas de inserção de uma cultura específica e até mesmo de personalidade individual (BADLER, 1997).

2.2 Exemplos de Humanos Virtuais

2.2.1 JACK

O centro da tecnologia que envolve o JACK (BADLER 1997) é a interação aliada a uma plataforma gráfica programável. O foco de sua pesquisa se encontra em duas áreas: a simulação do comportamento humano e engenharia de fatores humanos.

A simulação do comportamento humano simula comportamentos físicos tais como: andar, balançar, segurar, escalar, etc. A engenharia de fatores humanos permite a criação de ferramentas para suportar testes de produtos que necessitem das habilidades humanas, como por exemplo, projetar o comportamento de um motorista, uma vez que ele está concentrado na condução do veículo levando em conta a visibilidade e o uso de volantes, marchas, etc. (AZULA et al., 1994).

O AV JACK é construído por qualquer quantidade de objetos 3D. Estes podem representar um objeto ou agente distinto e são decompostos em segmentos geométricos, segmentos são unidos por juntas. O JACK, por padrão, representa um segmento como sendo um poliedro 3D. Ele também disponibiliza, mecanismos para integrar outras representações de segmentos, como descrição de parâmetros.

O modelo humano do JACK apresenta em tempo real cinemática inversa, esforço e funções do comportamento humano. Dada uma posição desejada para o movimento de uma junta, a cinemática inversa determina ângulos apropriados para as juntas intermediárias.

Devido ao tempo real a cinemática inversa faz com que o resultado final seja “o movimento de um braço” ou “dobrar o dorso”. Também há funções de comportamento humano, como ver um objeto, no sentido de visualização do agente, balanço, para um ajuste automático de postura baseado no centro de massa do agente e detecção de colisão, o que torna mais realista o modelo humano.

Tsukasa Noma (Sociedade de Pesquisa Ultramarítima do Ministério de Educação, Ciências, Esportes e Cultura do Japão) durante sua permanência na Universidade de Pennsylvania, desenvolveu um humano virtual para apresentações. Uma extensão baseada no projeto do JACK, conhecida como JACK PRESENTER.

Sua interface com o usuário era feita na forma de texto/fala ou por comandos pré-definidos, e linguagem corporal. Como o texto é processado, o JACK realiza a apresentação com fala e atos correspondentes para exibir a apresentação no quadro-branco com textura mapeada ou imagem. Se voltando para ele e ao mesmo tempo para quem estiver visualizando a apresentação (Figuras 2.11 e 2.12).



Figura 2.11: O JACK PRESENTER aponta para o slide e para o expectador (BADLER, 1997)

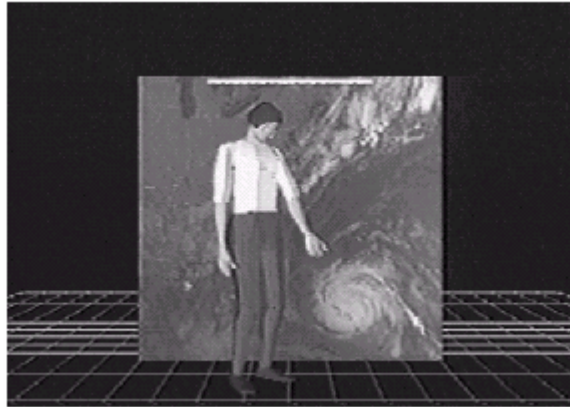


Figura 2.12: O JACK PRESENTER aponta e olha para um furacão em um mapa do tempo (BADLER, 1997)

Os componentes importantes do JACK PRESENTER são:

- Entradas próprias para representações dos cenários para apresentação;
- Movimento natural com recursos para apresentação;
- Geração de movimento em tempo real sincronizado com recurso de voz.

As entradas podem ser realizadas por “fala” utilizando um sintetizador de voz ou por comandos, ambos alterando de alguma forma a apresentação. O texto pode ser criado com antecedência ou anotado manualmente através de comandos que carregam a imagem no quadro, aponta para algum lugar da imagem e gesticula tanto para o quadro quanto para audiência. Uma interface possibilita a geração do texto e comandos on-line, oriundos de outros programas. Um programa de controle sofisticado passa as instruções ao apresentador em tempo real.

Os movimentos do JACK PRESENTER tentam simular o mesmo que de um professor, ou seja, ele usa o quadro como elemento visual auxiliar, nela pode ser exibido texto ou textura. Como um passo de pré-processamento da apresentação, são associados pontos estratégicos a locais específicos na imagem, como por exemplo na Figura 3.11 e 3.12, a posição do texto e o furacão respectivamente. Como o JACK PRESENTER utiliza a fala, ele deve agir de acordo com alguns princípios, que são:

Para onde apontar e com que mão;

- Que gesticular e com que mão;
- Onde olhar (quadro ou expectadores);
- Onde se colocar para aumentar a visibilidade da imagem e o seu local apontado.

Smith *et. al.* (1997) prototiparam um sistema chamado JACKMOO, que é um híbrido do JACK e do LAMB DAMOO (sistema interativo, multi-usuário, com recursos de rede e programável) (CURTIS, 1997) utilizado em conferências baseadas em texto, treinamento (especialista/educacional) e outras aplicações colaborativas.

JACKMOO permite a utilização de uma informação semântica mais rica, necessárias para implementar ações humanas que um avatar deve reproduzir, representar essas ações em forma de orações naturais e imperativas. Ele permite testar o controle de movimentos de uma avatar através de uma linguagem.

A principal importância do JACKMOO é a associação de verbos que descrevem ações humanas com possíveis e variadas ações realizadas pelo JACK. Ações como andar passo a passo, olhar em direção de, andar ao redor, são implementadas no interior do ambiente JACK em forma de programas, fornecendo uma interface para o controle do humano virtual avatar.

3.2.2 STEVE

O STEVE (*Soar Training Expert for Virtual Environments*) integra três áreas: sistemas tutores inteligentes, computação gráfica e arquitetura de agentes. A combinação resultante é um conjunto único de capacidades. O STEVE apresenta algumas habilidades pedagógicas no mundo dos tutores inteligentes. (RICKEL; JOHSON, 1998)

O STEVE pode interagir com outras pessoas e outros agentes no AV. O objetivo de seus desenvolvedores é suportar um grupo de treinamento, onde um grupo de pessoas, de diferentes localidades, possam coexistir num AV qualquer e aprender como um grupo. Agentes como STEVE podem ter duas linhas de ação, primeiro servir como tutores para estudantes individuais ou para o treinamento de um grupo.

Para ilustrar as habilidades do STEVE, aqui é apresentado um pequeno exemplo de treinamento. Suponha que o STEVE esteja demonstrando um compressor a ar de alta pressão de um navio. O *head-mounted display* (HMD) do estudante propicia uma visão 3D do ambiente do navio que inclui o compressor a sua frente e o STEVE a seu lado. Quando o estudante se move ou sua cabeça, ele percebe a mudanças correspondentes na visão do HMD e ele está equipado com um microfone para que o estudante se comunique com o STEVE.

Depois da introdução da tarefa, STEVE começa a demonstração. “Eu irei agora checar o óleo”, diz STEVE, e ele se move para o painel do compressor. STEVE olha para o painel, aponta-o e se volta ao estudante e diz: “vamos tirar o medidor de óleo”, ele puxa-o para fora.

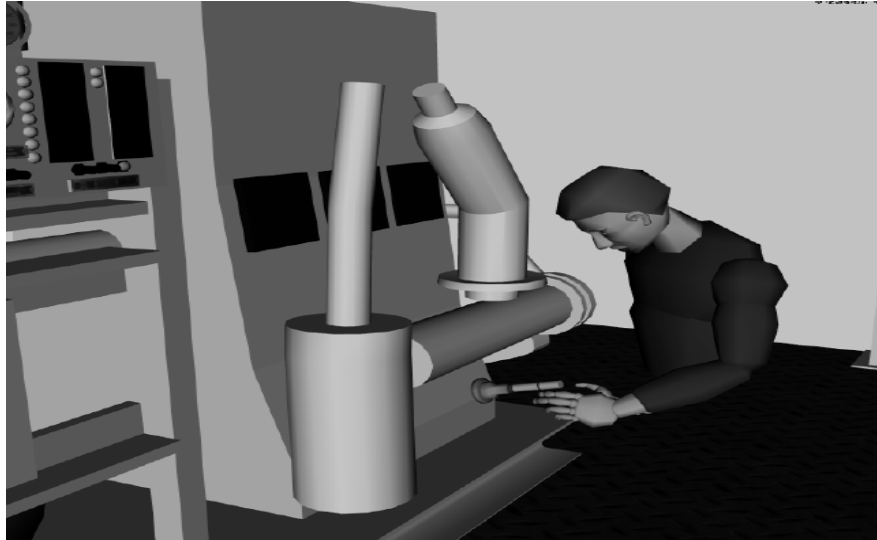


Figura 2.13: STEVE puxa o medidor de óleo (RICKEL; JOHSON, 1998)

A seguir, aponta par ao indicador de nível, Steve diz "agora nós vamos checar o nível de óleo do medidor. Como você pode ver, o nível do óleo está normal" . Para terminar a subtarefa, Steve diz “próxima, inserir o medidor de óleo” e ele o faz.

Arquitetura do mundo virtual, mostrando o agente (no caso STEVE) e o participante humano (Figura 2.14):

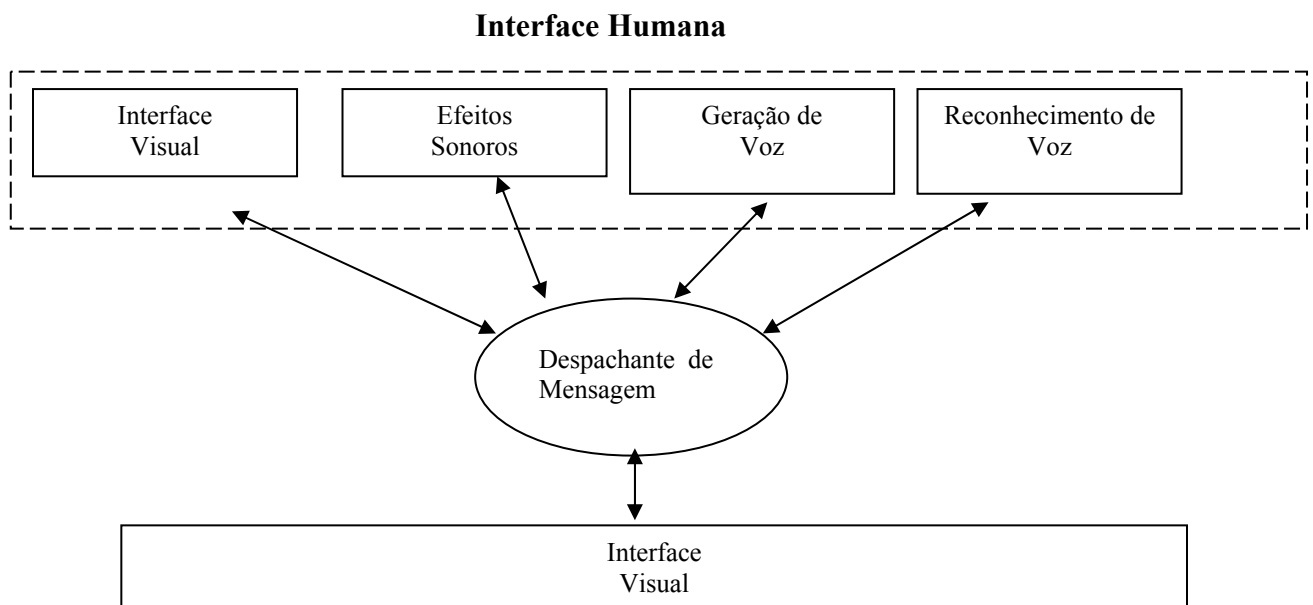


Figura 2.14: Arquitetura do mundo virtual. Mostrando componentes de um agente e um participante (humano) (RICKEL; JOHSON, 1998)

Arquitetura do STEVE possui os seguintes componentes:

- **SIMULADOR:** o comportamento do mundo virtual é controlado pelo simulador. O STEVE utiliza a simulação VIVIDS, desenvolvida pelo *USC Behavioral Technologies Laboratory* (RICKEL; JOHSON, 1998).
- **INTERFACE VISUAL:** cada participante humano possui uma interface visual que permite ver os outros participantes e manipular o mundo virtual. O participante é conectado ao mundo através de vários dispositivos: visualiza o mundo através de um HMD (*Head--Mmounted Display*), seus movimentos são rastreados através de sensores conectados a cabeça e mãos e interage com o mundo através de toque utilizando uma luva de dados. A interface visual tem dois objetivos: receber mensagens de outros componentes (principalmente do simulador) que informam mudanças no mundo e saídas para representações 3D através do HMD dos participantes. E informar a outros componentes quando uma pessoa interage com algum objeto (RICKEL; JOHSON, 1998).
- **ÁUDIO:** cada participante possui um componente de áudio, recebe mensagens do simulador de mensagens a localização e o rádio audível de vários sons, e a difusão adequada destes através de fones de ouvidos existentes nos HMD (RICKEL; JOHSON, 1998).
- **GERADOR DE VOZ:** cada participante disponibiliza de um componente gerador de voz que recebe mensagens de texto de outros componentes (primariamente dos agentes), convertem o texto em voz e por difusão a voz chega até os fones de ouvidos dos HMD's (RICKEL; JOHSON, 1998).
- **RECONHECIMENTO DE VOZ:** cada participante possui um componente de reconhecimento de voz que captura a voz através de um microfone, a reconhece como

uma gramática, e emite uma representação semântica da voz (fala) para outros componentes (RICKEL; JOHSON, 1998).

- **AGENTE:** cada agente STEVE executa como um componente separado (RICKEL; JOHSON, 1998).

O STEVE possui três módulos principais, que são: percepção, cognição e controle motor (Figura 2.15):

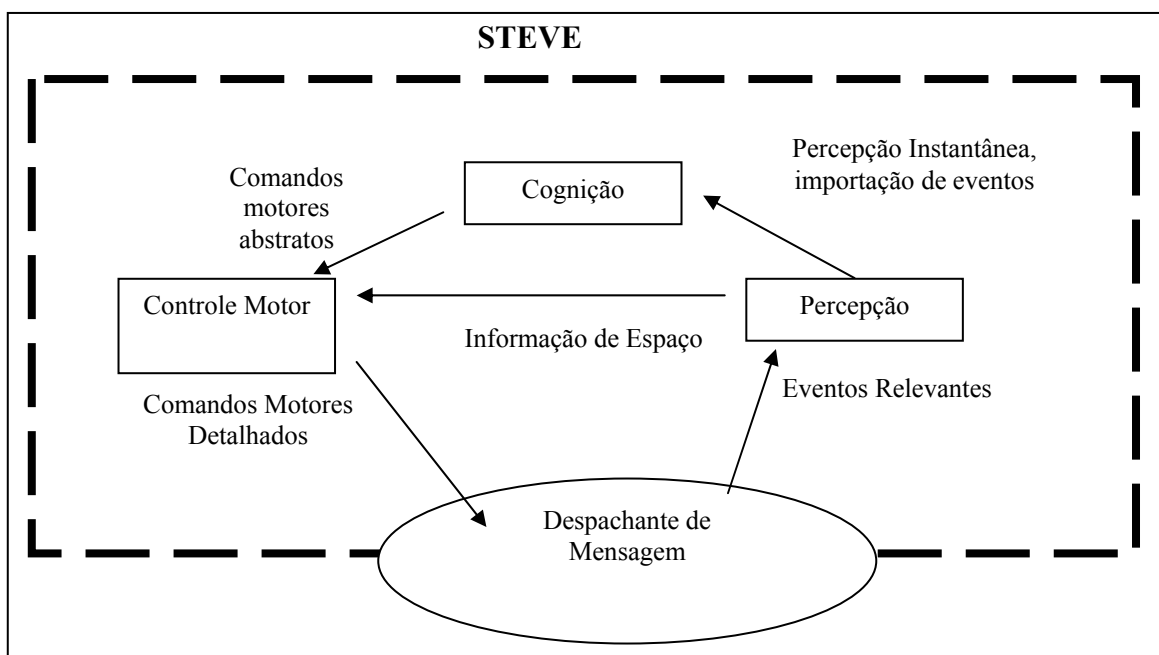


Figura 2.15: Os três módulos principais do STEVE e as informações que eles enviam e recebem.

- Percepção envia mensagem a mensagem e identifica eventos que são relevantes para o STEVE, como ações tomadas por pessoas ou agentes e mudança no estado do mundo virtual.
- Cognição interpreta as entradas recebidas pelo módulo de percepção, busca metas apropriadas, constrói e executa planos para alcançar essas metas e envia comandos motores para controlar o corpo do agente.

- Controle Motor decompõe esses comandos em uma seqüência de instruções de baixo nível que é enviada aos outros componentes pelo despachante de mensagens.

3. TÉCNICAS ESTUDADAS PARA GERAÇÃO DE MOVIMENTOS PARAMETRIZADOS

A animação de modelos complexos como HV são geralmente criada através de uma representação simples de um modelo (um esqueleto), que é uma figura articulada a ligações rígidas a dobradiças (juntas).

O movimento é primeiro computado para esta estrutura (esqueleto), e pode ser exibido interativamente. O animador satisfeito com a seqüência global de movimentos, pode criar representações de alto nível de movimento para o personagem, cobrindo o “esqueleto” com superfícies deformáveis, onde se pode modelar a pele.

Uma exemplo de uma figura articulada está apresentada na Figura 3.1.

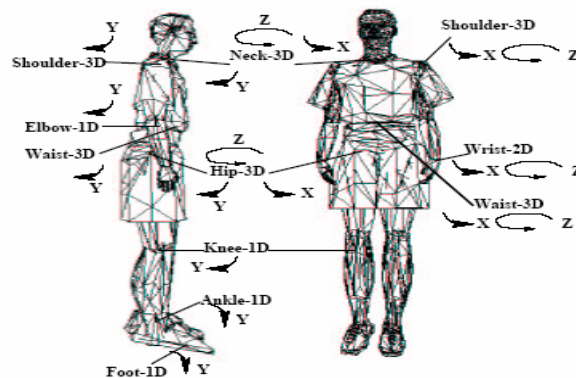


Figura 3.1: Figura articulada com 30 graus de liberdade (MULTON et al., 1999)

Matematicamente, o “esqueleto” é uma seqüência de *frames* dispostos de formas hierárquicas, caracterizadas por posição e orientação ao *frame* mestre. O conjunto de parâmetros $\theta = (\theta_1, \dots, \theta_N)$ que correspondem aos graus de liberdade é chamado “vetor de estado” ou “coordenadas generalizadas” da Figura articulada (Figura 3.1). O movimento

sintetizado do esqueleto depende da definição de como o vetor de estado se altera com o passar do tempo.

3.1 Humanos caminhando

A biomecânica caracteriza-se em estudar como o ser humano se movimenta, como é realizada a ação de caminhar, com seus movimentos, juntas e atributos necessários.

Ela caracteriza o caminhar como uma sucessão de fases separadas, primeiro *footstrike-fs* (quando o pé entra em contato com o solo) e posteriormente, *takeoff-TO* (quando ele abandona o solo) (Figura 3.2).

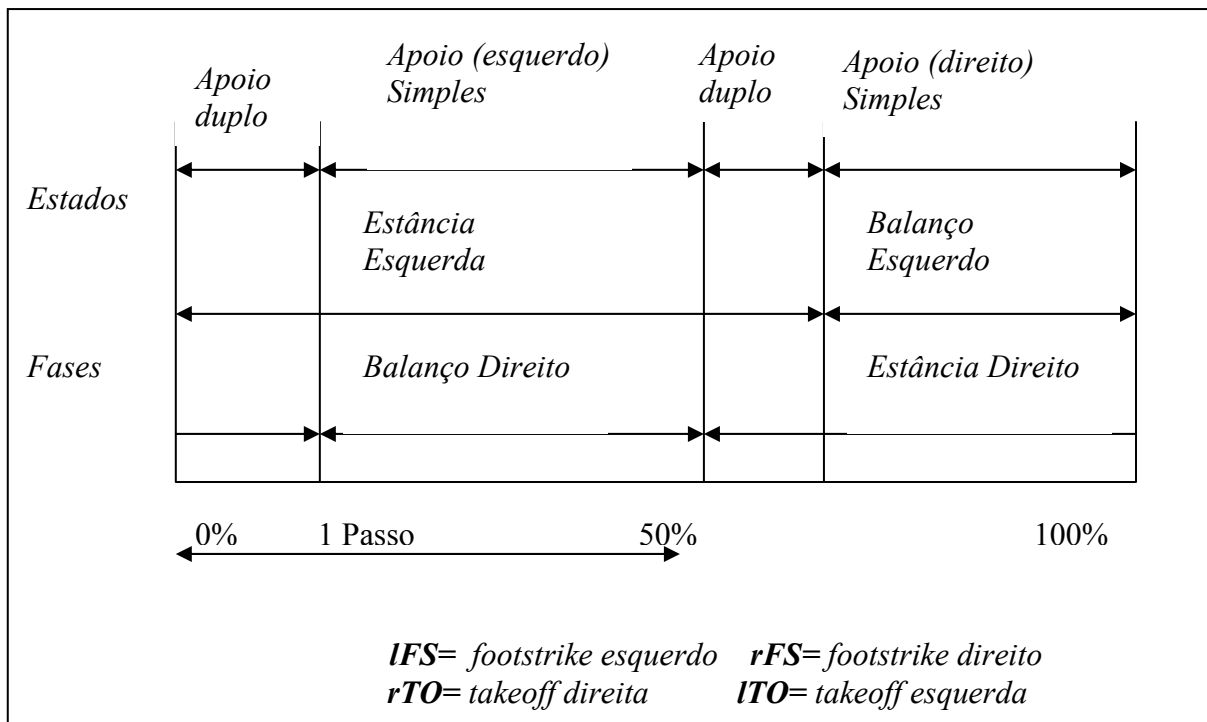


Figura 3.2: Fases do personagem no movimento de andar (MULTON et al., 1999)

Multuon *et. al.* (1998), definem como "passo largo" a saída de um pé esquerdo do chão até este pé deixar novamente o chão, enquanto passo seria apenas parte desse ciclo.

Quatro *footstrike* e eventos partida acontecem durante um “passo largo”: partida esquerda (*lTO*), *footstrike* esquerdo (*IFS*), *takeoff* direita (*rTO*) e *footstrike* direito (*rFS*). No ciclo de fases do movimento de andar, ocorrem dois períodos distintos, que são:

Período de apoio é chamado de fase de instância;

Período sem apoio é chamado de fase de balanço.

Isso caracteriza o movimento de cada perna (Figura 3.2).

Uma definição mais detalhada é dada por Nilson *et. al.* (1985). Eles usam extensões e passos flexíveis para locomoção dos sub-ciclos associados com cada junta. Cada extensão/passo flexível tem sua característica pelo seu lugar dentro do contexto do movimento de “andar”, sua duração e associação dos valores de seus ângulos que podem ir de um ângulo baixo (próximo do solo, por exemplo 20°) até um alto (ângulo mais aberto, por exemplo 60°).

A pesquisa de Nilson *et. al.*(1985), se concentra em como ocorre a alteração dos valores destes ângulos relacionados com a velocidade do andar. Quando a velocidade aumenta, a duração do estado de apoio duplo decrementa regressivamente até zero, de modo que o pedestre alterou seu passo de caminhar para correr.

Verificou-se que, se tenta sintetizar, realistamente, o movimento de andar está no contato dos pés com o solo, ou seja, a penetração dos pés no solo. Já que isso é sustentado pelo fato que os pés não deslizam, o esforço desigual para a especificação de como os pés podem ficar sobre o solo pode se converter em equivalente esforço durante a fase de apoio.

A modelagem do apoio duplo é um problema que os algoritmos de animação computadorizada, que facilmente competem com correntes abertas, tem dificuldade de

controlar *loops* fechados. Esta é uma razão que o movimento de andar é mais difícil de representar que o de correr, que não precisa de nenhum apoio duplo.

Em se tratando do movimento de andar, o uso de uma simulação física para ajudar a representação sintética na representação de movimentos realísticos ainda está envolto em dúvidas. Primeiro, segundo HODGINS, WOOTEN, BROGAN e O'BRIEN (1995) a ação da dinâmica é menos importante para o movimento de um personagem que outros movimentos humanos como: correr, saltar e mergulhar. Segundo, o modo como alguém anda diz muito mais da personalidade, humor, ou seja, a essência desta pessoa, cujas características são muito difíceis de modelar e disponibilizar. (MULTON et al., 1999).

Existem modelos que utilizam dinâmica. Um bom simulador provê uma modelagem precisa de contatos/fricções com o solo requerido. Problema principal é encontrar controles parametrizados que gerem ações musculares com o decorrer do tempo. Estas ações devem fazer o modelo caminhar enquanto seu equilíbrio é mantido. Pode-se lançar mão de técnicas utilizadas na robótica para controlar bípedes para lançar “luz” a esta questão. Se for escolhido o método de animação que utiliza cinemática, o modelo será principalmente confiável na descrição nos movimentos de caminhada tais como descritas na biomecânica (MULTON et al., 1999).

3.2 Animação Cinemática

O primeiro conjunto de ferramentas desenvolvido para a animação computadorizada foram as cinemáticas diretas e inversas.

3.2.1 Cinemática Direta e Inversa

A cinemática direta consiste na especificação de um vetor de estados por figura articulada de acordo com o tempo decorrido. Esta especificação é normalmente usada para um conjunto pequeno de quadros chaves (*key frames*), enquanto são usadas técnicas de interpolação para gerar os quadros intermediários, referentes a cada quadro chave. Um dos principais problemas é a elaboração conveniente do projeto de quadros chaves de animação, e escolha adequada de técnicas de interpolações. O problema posterior, e em particular o caminho orientado que pode ser representado e interpolado tem que ser extensamente estudado (WATT, WATT 1992). Posições chave projetadas permanecem sobre o controle da mão do animador e o resultado do movimento depende de suas habilidades. Muitas vezes o conhecimento físico e biomecânico pode ajudar o animador a construir quadros chave.

O uso exclusivo da cinemática direta torna-a difícil, adicionando esforço ao movimento, tal como, a especificação que os pés não devem penetrar no solo durante a fase de apoio.

Isso pode ser resolvido com a cinemática inversa. O movimento ΔX da ligação final de uma cadeia (um pé, por exemplo) é especificado pelo animador em termos de variáveis globais. É computada a variação do vetor de estados (por exemplo, a variação entre ligações intermediárias) que irá de encontro ao esforço.

A relação entre a tarefa “principal“ ΔX e os deslocamentos angulares $\Delta\theta$ tomam a forma dada pela equação 3.1

$$\Delta X = J\Delta\theta$$

Onde J é a matriz Jacobiana de um sistema. J não é diretamente inversa, devido as diferentes dimensões de X e θ (por exemplo, existem infinitas posições angulares das juntas que um pé pode assumir numa mesma posição cartesiana). Assim a solução freqüentemente mais usada é:

$$\Delta\theta = J^+\Delta X + \alpha(IJ) \Delta z$$

Onde J^+ é o pseudo-inverso da matriz Jacobiana, α é uma constante de penalidade, I é a identidade da matriz, e Δz é uma constante para minimizar, chamada de tarefa secundária. Esta tarefa secundária é obrigada no espaço nulo da tarefa principal. Assim o segundo termo não afeta a tarefa principal. Pode-se dizer que a tarefa secundária é z . Geralmente z é usado para responder por limites angulares em comum ou minimizar algum critério energético.

3.2.2 A animação cinemática aplicada ao andar humano

Toda tentativa em sintetizar a locomoção humana usando a cinemática, está baseada no conhecimento biomecânico e combina as cinemáticas direta e inversa para geração dos movimentos.

Foram desenvolvidos métodos que geram grupos de quadros chaves (*keyframes*) a partir de informações biomecânicas. Provendo assim, uma fácil sincronia do resultado final. Isto é feito através de uma máquina de estados finitos controlada por parâmetros de alto nível, como comprimento do passo ou freqüência do passo.

Zeltzer (1986) define isto. Coloca-se o conhecimento de locomoção biomecânica dentro da hierarquia de uma máquina de estados simultânea que controla o “andar” de um esqueleto sintético. A chave (animação) que refere-se a postura é referenciada a cada estado na máquina. Então estas posturas são interpoladas entre si para produzir o movimento linear entre os valores angulares.

O problema em se usar a cinemática direta, sempre foi o pé transpassar o solo durante o movimento. O primeiro método que solucionou isto e que utiliza a cinemática direta, consistia em mudar o membro raiz do esqueleto, enquanto o pé de apoio se movimenta: a cada *footstrike*, um novo pé de apoio se torna a raiz, e sua posição é fixada nas coordenadas globais. Bruderlin *et. al.* (1989) utilizou esse método para simular um pendulo invertido para a computação do movimento realístico para a posição de uma perna. Em sua pesquisa apenas uma junta de cada perna de apoio se move em um dado momento, o que pode conduzir ligeiramente a movimentos artificiais. Bruderlin e Calver (BRUDERLIN e CALVER 1989) otimizaram esta técnica pela simulação simultânea de todos os graus de liberdade, conduzindo a um andar liso e parametrizado. Eles utilizaram o mesmo tipo de solução para produzir movimentos de corrida (BRUDERLIN. e CALVERT 1996).

Outro modo de solucionar a questão relacionada à posição dos pés é utilizar cinemática inversa (GIRARD, MACIEJEWSKI 1995) (MACIEJEWSKI 1995) (BOULIC *et al.*, 1990) (BOULIC *et al.*, 1992) . Por exemplo, Boulic *et al.*, (1990) utilizou primeiro cinemática direta, criando posições chave do movimento que serão interpolados. Um método é usado para corrigir a posição da perna entre as posições definidas: se o pé está transpassando o solo, toma-se da cinemática inversa para alterar sua posição, assim modela-se seu contato com o solo (ver Figura 3.3). A segunda maneira é utilizar cinemática inversa para manter o balanço do HV (BOULIC , THALMANN *et al.*, 1992).

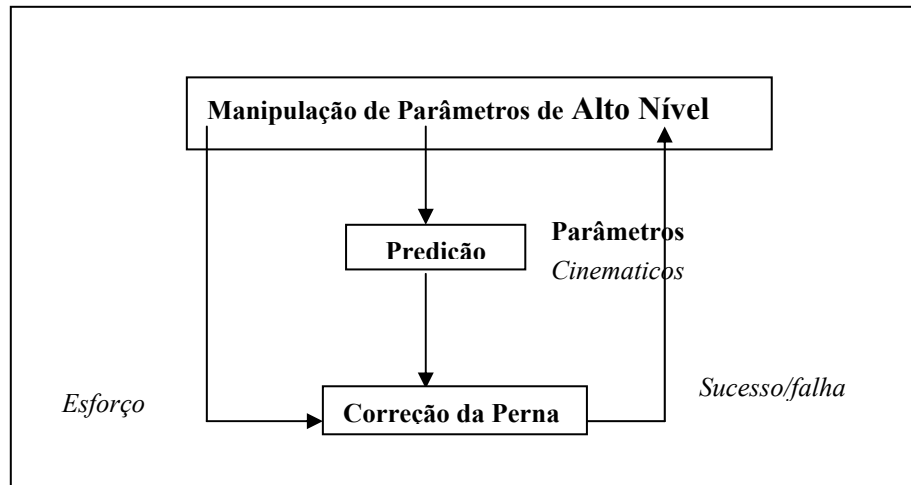


Figura 3.3: Sinopse de BOULIC para um sistema de locomoção (Multon net al., 1999)

3.3 Dinâmica

O grau de realismo, quando se sintetiza movimentos característicos de seres humanos, como andar, correr, movimentar braços ou pernas está intrinsecamente relacionado a leis físicas. Por exemplo quando se anda e o pé entra em contato com o solo, entram em ação princípios conhecidos pelas “leis de Newton”. Neste caso pode-se utilizar a terceira lei de Newton, que diz que toda ação tem uma reação em igual e contrária, e o atrito, que define que quando se exerce uma força contra o solo (considerando-o aproximadamente plano), o mesmo exercerá uma força na mesma intensidade sentido contrario a primeira, devido a isto que quando se anda não se desliza pelo solo.

Então pode-se usar qualquer aproximação fornecida pela Dinâmica para se conseguir um determinado realismo pré-definido, ou para obter um movimento (caminhar) sintetizado. E também para resolver questões relevantes ao movimento que a cinemática não consegue solucionar.

Como a Cinemática a Dinâmica se divide em Direta, onde utilizando as leis de Newton o calculo do movimento produz força e a Inversa, o calculo da força gera movimento.

A Dinâmica permite também a determinação de esforço, sem ter que determinar forças que gerem o movimento, não prejudicando o realismo.

4. GERADOR DE MOVIMENTOS PARAMETRIZADOS PARA HV

As aplicações que usam HV têm tomado grande destaque dentro da RV. Isto ocorre porque podem ser usadas para os mais diversos fins, e sobressaindo-se na área da educação seja ela acadêmica ou de treinamento procedimental de equipamentos.

Isso pode proporcionar aos que interagem com esse tipo de aplicação, uma educação ou treinamento prazeroso e com qualidade, devido as ferramentas de RV que podem ser utilizadas. Dando como retorno a quem promove este tipo de treinamento um maior aproveitamento do mesmo. Como foi visto no Capítulo 2 o uso do HV denominado de *Jack* (BADLER 1997) (BADLER 2002), pode ser usado para os mais diversos fins.

Para isso muitas ferramentas tem sido criadas. Este capítulo apresenta a proposta de uma ferramenta para geração de movimentos parametrizados HV utilizando JAVA3D e interpolação por cinemática inversa.

4.1 Objetivos principais

O Gerador de Movimentos Parametrizado para HV (MP3D) tem como objetivo a criação de movimentos de membros específicos do corpo humano: braço direito, braço

esquerdo, antebraço direito, antebraço esquerdo e dedos da mão direito de forma parametrizada. Após da especificação destes movimentos será disponibilizado de uma animação em tempo real do movimento determinado.

A escolha destes membros não foi feita de forma isolada, e sim visando uma eventual utilização do MP3D aliado a Linguagem Brasileira de Sinais (LIBRAS ou LBS) . O que permite a geração de uma biblioteca de sinais virtuais, já que é possível armazenar as informações criadas pelo MP3D em arquivo e recuperá-las.

A geração animada dos movimentos e reprodução são realizadas em um ambiente 3D através de um HV modelado em Java3D.

Use-se como forma de funcionamento do aplicativo, o mesmo paradigma encontrado na maioria dos softwares de animação 3D, como o *3D Studio Max* (AUTODESK 2005) e *Maya* (ALIAS 2005), onde o animador determina *frames* chaves para a animação e os *frames* existentes entre estes determinados como chaves são calculados através de cinemática inversa.

O MP3D é composto de três módulos:

- **Ambiente:** o ambiente 3D, onde se encontra o modelo humanóide 3D;
- **Comandos:** onde pode-se selecionar qual membro do corpo do humanóide se quer movimentar: braço direito, braço esquerdo, antebraço direito, antebraço esquerdo e mão direita; controles que movimentam esses tridimensionalmente nos eixo X, Y e Z;
- **Animador:** controle de animação, seleção de *frame* chaves, interpolação destes por cinemática inversa, opções de exportar a animação finalizada para arquivo e importá-la.

Sendo esses módulos implementados utilizando Java3D; e Java para gerenciar a aplicação. Na Figura 4.1 é apresentado a arquitetura geral do MP3D.

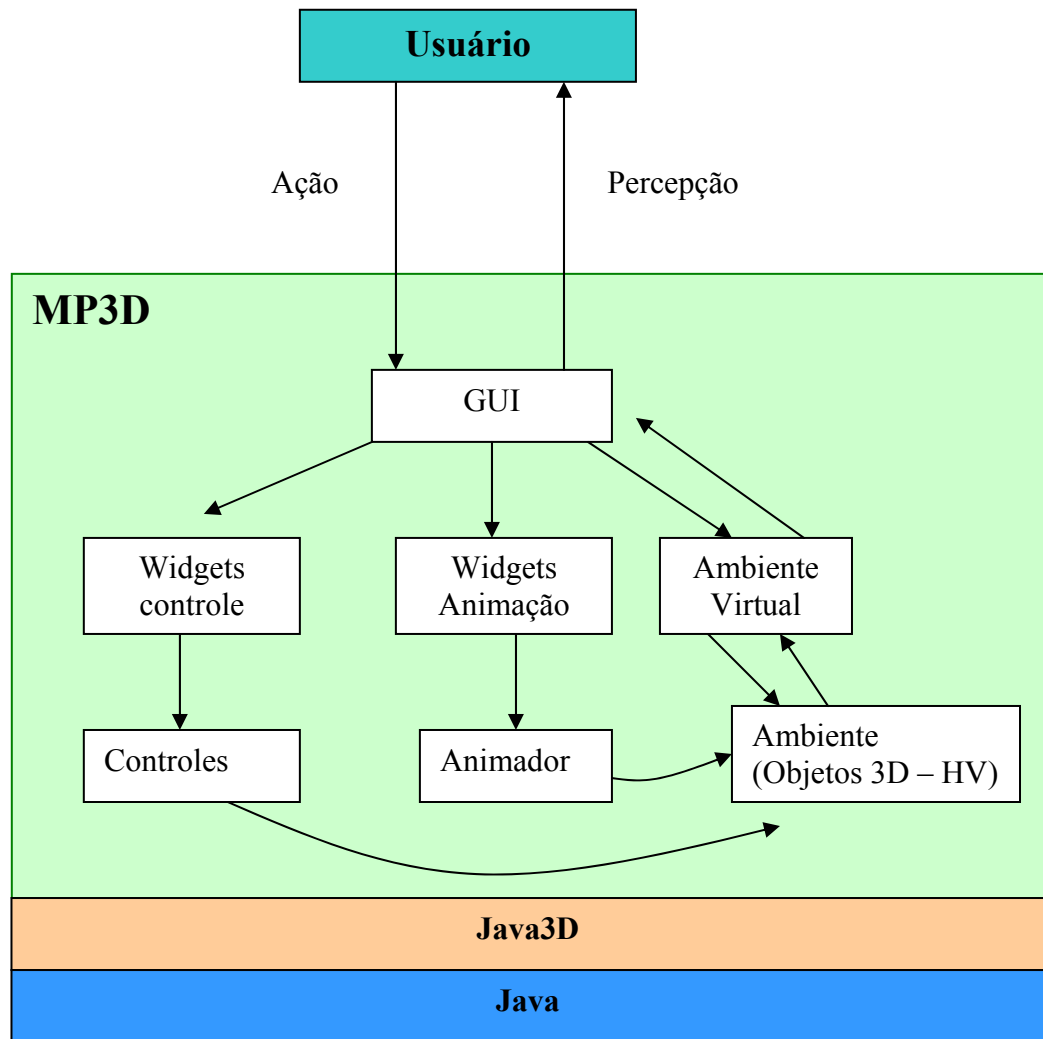


Figura 4.1: Diagrama da arquitetura do MP3D

4.2 Aspectos Gerais da Ferramenta

Os componentes da aplicação é composta por uma GUI (Graphical User Interface – Interface Gráfica com o usuário) e os controles são feitos através de *widgets* (elementos de interface na forma de botões). Suas funcionalidades são explanados a seguir:

a) Ambiente

Função de carregar um AV contendo um HV, texturas e iluminação. Nele são visualizados movimentos selecionados para o HV bem como a visualização das animações criadas.

Disponibilizou-se *widigets* na parte superior e direita do ambiente que se destinão a manipulação visual deste e do HV. A Figura 4.2 ilustra o *módulo* ambiente.



Figura 4.2: *Módulo Ambiente*

Estes controles se destinam a manipulação visual do HV e reinicializarão do mesmo no ambiente.

- **Zoom-In:** aproxima câmera do humano virtual.
- **Zoon-Out:** afasta a câmera do humano virtual.
- **Câmera1:** visualiza o humano virtual de perfil esquerdo.
- **Camera2:** visualiza o humano virtual de cima para baixo.
- **Camera3:** visualiza o humano virtual de perfil direito.
- **Reset:** HV volta a posição inicial.
- **Reiniciar:** este controle em particular, além de manipular o *módulo* ambiente, fazendo que o humano virtual volte sua posição inicial, também interage e manipula os outros módulos, fazendo-os reiniciarem suas funções

A Figura 4.3. mostra de forma ilustrativa cada um dos controles do *módulo*.

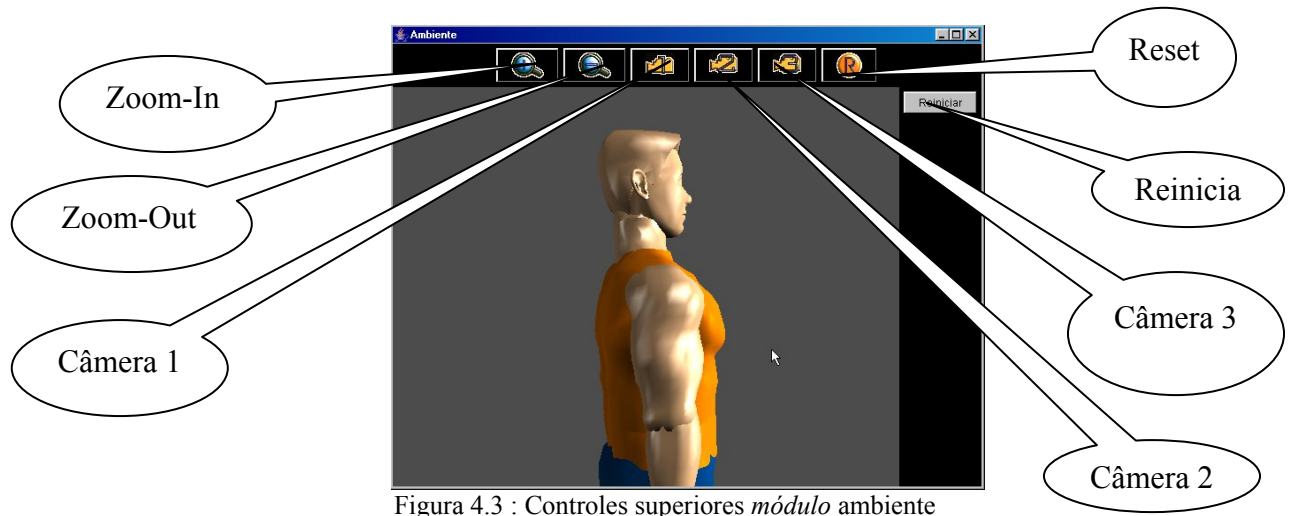


Figura 4.3 : Controles superiores *módulo ambiente*

O módulo ambiente é construído por duas classes: a Classe Ambiente e a Classe Avatar.

A classe Ambiente constitui com um receptáculo para todo *módulo ambiente*. Nela é armazenada o grafo de cena de todo ambiente, bem como a implementação dos controles que a manipulam. Na Figura 4.4 pode ser ver a implementação do grafo de cena, da classe ambiente.

```
private BranchGroup criaCena()
{
    raiz = new BranchGroup();
    regioao = new BoundingSphere(new Point3d(0,0,0),100);
    bckg = new Background(new Color3f(0.3f, 0.3f, 0.3f));
    bckg.setApplicationBounds(regiao);
    raiz.addChild(bckg);
    Color3f lColor1 = new Color3f(0.75f, 0.75f, 0.75f);
    Vector3f lDir1 = new Vector3f(-1.0f, -1.0f, -1.0f);
    Vector3f lDir2 = new Vector3f(-0.75f, 0, 0);
    Vector3f lDir3 = new Vector3f(0.75f, 0, 0);
    Color3f alColor = new Color3f(0.4f, 0.4f, 0.4f);
    luz = new AmbientLight(alColor);
    luz.setInfluencingBounds(regiao);
    luzd = new DirectionalLight(lColor1, lDir1);
    luzd.setInfluencingBounds(regiao);
    luzd1 = new DirectionalLight(lColor1, lDir2);
    luzd1.setInfluencingBounds(regiao);
}
```

```

        luzd2 = new DirectionalLight(alColor, lDir3);
        luzd2.setInfluencingBounds(regiao);
        raiz.addChild(luz);
        raiz.addChild(luzd);
        raiz.addChild(luzd1);
        raiz.addChild(luzd2);
        grupo = new TransformGroup();
        grupo.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        grupo.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
        raiz.addChild(grupo);
        a = new Avatar(new Appearance(), grupo, regiao);
        a.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        a.setCapability(TransformGroup.ALLOW_TRANSFORM_READ);
        Transform3D tAvatarEscala = new Transform3D();
        tAvatarEscala.setScale(2.30d);
        Transform3D tAvatarTranslacao = new Transform3D();
        tAvatarTranslacao.setTranslation(new Vector3f(0.0f, -0.34f,
0.0f));

        tAvatarEscala.mul(tAvatarTranslacao);
        grupo.setTransform(tAvatarEscala);
        mr = new MouseRotate(grupo);
        mr.setSchedulingBounds(regiao);
        grupo.addChild(mr);
        camera = new Camera(grupo);
        camera.setSchedulingBounds(regiao);
        raiz.addChild(camera);
        interpolador = new Interpolador(a);
        interpolador.setSchedulingBounds(regiao);
        grupo.addChild(interpolador);
        grupo.addChild(a);

        raiz.compile();

        return raiz;
    }

```

Figura 4.4: Especificação do grafo de cena da classe Ambiente

Na Figura 4.5 mostra o fragmento de código de um dos controles pertencente à classe Ambiente.

```

if (e.getSource() == camera1)
    {
        camera.camera1();
    }

```

Figura 4.5: Fragmento do Código da implementação do controle Camera1

Aqui se utiliza o método *camera1()*, pertencente a classe câmera, responsável pelo comportamento do estado da visão escolhida. Na figura 4.6 selecionou-se a visão da câmera1.

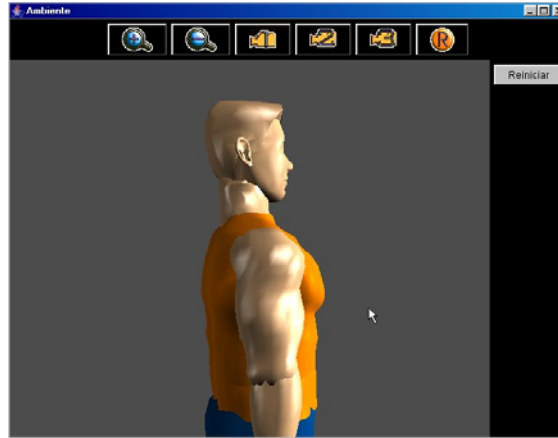


Figura 4.6: Visão da Camera1

A Figura 4.7 demonstra a codificação da `camera1`, onde dois atributos são armazenados, *estado*, que indica de o efeito de câmera será zoom (in ou out) ou rotação de câmera, *câmera*, no caso específico, qual câmera selecionada (1, 2, ou 3). Caso a seleção for a `camera1`, `camera2` ou `camera3`, o estado terá valor 3, como demonstrado na Figura 4.7.

```
public void camera1()
{
    estado = 3;
    camera = 1;
}
```

Figura 4.7: Seleção da Camera1

O *estado* determina o tipo de movimento de câmera. Caso valor seja igual a 1, produzira-se o movimento de -0.2 no eixo Z sobre seu valor atual, assim provendo a impressão do zoom out. Caso for igual a 2, se terá um movimento de +0.2 no eixo Z sobre seu valor atual, fornecendo a impressão do zoom in. E caso for igual a 3, usa-se o atributo *câmera*, para se determinar qual câmera está selecionada. Cada um tem sua coordenada de rotação já preestabelecida e quando selecionada produz a rotação respectiva. Na Figura 5.8 é mostrado a seleção de câmera.

```

else if (estado == 3)
{
    if (camera == 0)
    {
        t = t0;
        posicao.z = 0;
        t.setTranslation(posicao);
    }
    if (camera == 1)
    {
        t = t1;
    }
    if (camera == 2)
    {
        t = t2;
    }
    if (camera == 3)
    {
        t = t3;
    }
    grupo.setTransform(t);
}

```

Figura 4.8: Codificação da Seleção das Câmeras

A Figura 5.9 mostra a determinação das coordenadas das câmeras.

```

grupo = cena;
grupo.getTransform(t);
grupo.getTransform(t0);
grupo.getTransform(t1);
grupo.getTransform(t2);
grupo.getTransform(t3);
t0.setTranslation(new Vector3d(0.0f, 0.350f, 0.0f));
rot.rotY(1.8f);
t1.mul(rot);
rot.rotX(0.6f);
t2.mul(rot);
rot.rotY(-1.8);
t3.mul(rot);

```

Figura 4.9 Especificação das coordenadas gerais da Classe Camera

A outra classe do *módulo* Ambiente é a classe Avatar. Onde é modelado o avatar propriamente dito, contendo a toda geometria do modelo, bem como atributos, como texturas e iluminação.

Na Figura 4.10 é apresentado um trecho de código relacionado com a hierarquia do avatar.


```

BracoDireito bracadireito;
BracoEsquerdo braco esquerdo;
private TransformGroup corpo = new TransformGroup();
public static TransformGroup tronco = new TransformGroup();
private TransformGroup cabeca = new TransformGroup();
private TransformGroup coxa_d = new TransformGroup();
private TransformGroup coxa_e = new TransformGroup();
private TransformGroup perna_d = new TransformGroup();
private TransformGroup perna_e = new TransformGroup();
private TransformGroup pe_d = new TransformGroup();
private TransformGroup pe_e = new TransformGroup();

...

```

Figura 4.10: Especificação da Hierarquia do Avatar

O grafo de cena do avatar, que é carregado na classe ambiente, é construído por um conjunto inter-relacionado de classes. A Figura 4.11 representa o diagrama de classes da estrutura do avatar. Nela são representado apenas os atributos participantes do movimento.

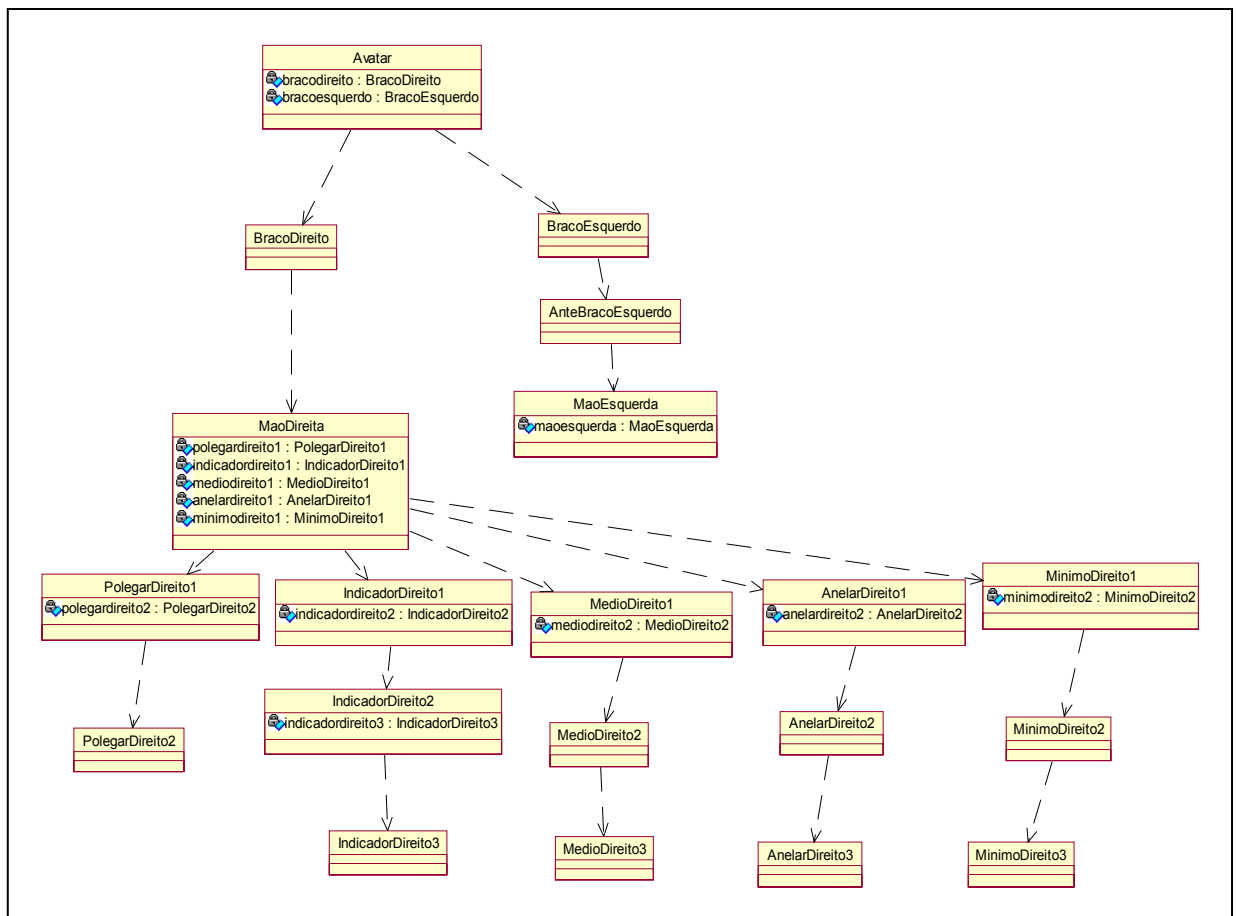


Figura 4.11: Diagrama de Classes da Hierarquia do Avatar.

b) Controles

Função de propiciar a interação entre o usuário e a aplicação. Realiza a ação parametrizada. Este módulo é composto por dois tipos de controles distintos: seleção do que se quer movimentar e em que coordenada espacial se dará esse movimento (Figura 4.12).

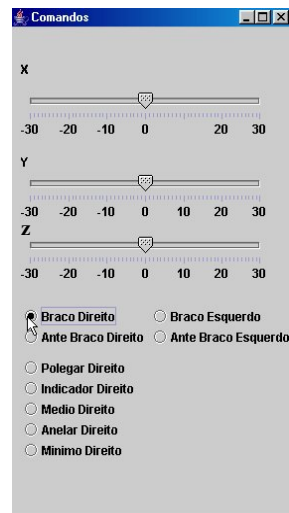


Figura 4.12: Módulo Controle

- **Caixa de Seleção:** uma caixa de seleção, onde escolhe-se que parte do HV se quer movimentar. Partes estas definidas por: braço direito, braço esquerdo, antebraço direito, antebraço esquerdo e mão direita. Foram escolhidos especificamente estes membros justificando que um aplicativo com esse fim poderia ser usado convenientemente na geração de sinais LBS, já que existe uma gama muito grande de combinações de sinais e a RV poderia ser usada amplamente para a educação deste tipo de linguagem (CUSIN 2004) (FUSCO 2004). Outra funcionalidade da seleção é a possibilidade do membro selecionado mudar de cor, para explicitar ao usuário realmente qual foi selecionado, como exemplo da Figura 4.13

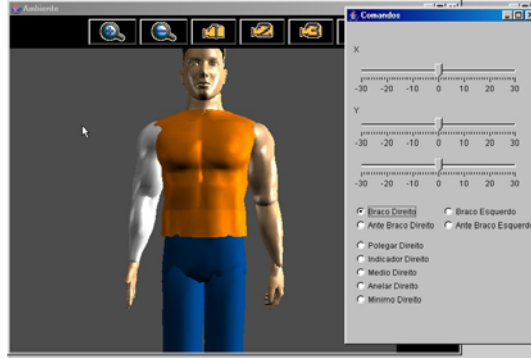


Figura 4.13: Exemplo de Seleção do braço do avatar

A Figura 4.13 apresenta um exemplo de código fonte que implementa a função de seleção.

```

if (e.getSource() == btnAnteBracoDireito)
{
    eixoX.setValue(valorantebracodireitoX);
    eixoY.setValue(valorantebracodireitoY);
    eixoZ.setValue(valorantebracodireitoZ);

    //=====Alteração de cor=====
    mBracoDireito.setDiffuseColor(dColor);
    mAnteBracoDireito.setDiffuseColor(white);
    mBracoEsquerdo.setDiffuseColor(dColor);
    mAnteBracoEsquerdo.setDiffuseColor(dColor);
}

```

Figura 4.13: Exemplo de fragmento de Código Fonte que altera a cor o membro selecionado

O atributo `dColor`, representa a textura da pele do avatar, foi criado utilizando o tipo `Color3f` (Java3D), e o `white` também. Utiliza-se o método `setDiffuseColor()` para atualizar a cor do avatar.

- **Coordenadas Espaciais:** depois de selecionado a parte do corpo do avatar que se quer movimentar, usam-se os controles de coordenadas espaciais, que movimentam a seleção nos eixos X, Y e Z .

Enquanto os controles espaciais são utilizados, parâmetros são passados para *módulo ambiente* e o HV realiza o movimento especificado. Preocupou-se em guardar e se recuperar os valores dos eixos das coordenadas, pois por exemplo, poderia-se selecionar braço direito e deslocar seu eixo X até determinado valor e quando se selecionasse o antebraço direito o eixo

X estaria deslocado neste valor. Então em cada seleção são armazenadas as devidas posições dos eixos X, Y e Z, e quando a seleção se repetir se recupere o valor anterior a esta.

O armazenamento é feito através de três variáveis auxiliares que representam os eixos X, Y e Z. Inicialmente elas recebem o valor zero. Sempre que quaisquer dos eixos são manipulados, o valor de cada um é capturado, e atualizado nela.

Quando uma nova seleção é feita, recupera-se conteúdo existente nestas variáveis auxiliares e então se atualiza a barra de controles dos eixos de coordenadas. Na Figura 4.14 é demonstrado a recuperação das informações do eixos X, Y e Z e na Figura 4.15 como é realizado o armazenamento.

```

if (e.getSource() == btnBracoDireito)
{
    //===Atualização do valor atual do eixo X, Y e Z=====
    eixoX.setValue(valorbracodireitoX);
    eixoY.setValue(valorbracodireitoY);
    eixoZ.setValue(valorbracodireitoZ);

    //=====Alteração de cor=====

    mBracoDireito.setDiffuseColor(white);
    mAntebracoDireito.setDiffuseColor(white);
    mBracoEsquerdo.setDiffuseColor(dColor);
    mAntebracoEsquerdo.setDiffuseColor(dColor);
}

```

Figura 4.14: Exemplo do fragmento do Código Fonte da recuperação dos valores das posições dos eixos X, Y e Z

```

if(eixoX.getValueIsAdjusting()==true)
{
    auxY=0.0f;
    auxZ=0.0f;
    auxX =(float) eixoX.getValue();

    valorantebraquesquerdoX=eixoX.getValue();//armazenamento do valor
    atual do eixo X

    Ambiente.interpolador.rotAntebracoE(0,auxX);//gera movimento X
    Ambiente.interpolador.rotAntebracoE(1,auxY);// gera movimento Y
    Ambiente.interpolador.rotAntebracoE(2,auxZ); gera movimento Z

    //=====passagem de parâmetro X, Y, Z para módulo animador=====
    keyFramer.passagemX(auxX);
    keyFramer.passagemY(auxY);
    keyFramer.passagemZ(auxZ);
}

```

Figura 4.15: Exemplo do fragmento do código fonte responsável pela movimentação do braço esquerdo

Finalizando o módulo Controle, utiliza-se métodos de rotação herdados da classe interpolador para produzir o movimentos e atualizar no *módulo* ambiente e usa-se os métodos *passagemX()*, *passagemY()* e *passagemZ()* (serão detalhados na página 68) para transmitir os valores respectivos dos eixos X, Y e Z para o *módulo* de *keyFramer* com objetivo de animação. A Figura 5.18 mostra o exemplo dos ajustes para um movimento.

c) **KeyFramer**

Função de gerenciar as animações geradas pelo aplicativo. Determinação de quadros chave para animação, e utilizado para a criação da animação cinemática inversa e interpolação de *frames* chave. Calculo da interpolação, ou seja, a criação de *frames* intermediários entre um intervalo de *frames* determinados. A Figura 4.16 ilustra o *módulo* *KeyFramer*.

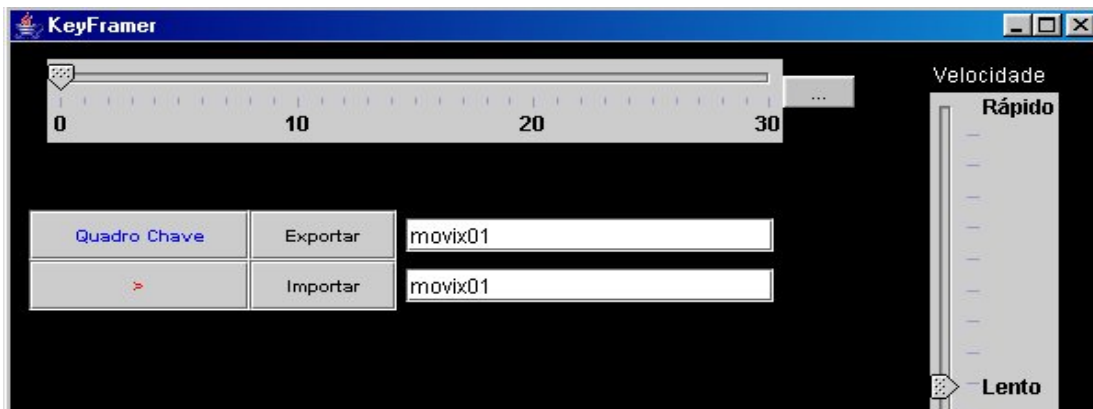


Figura 4.16: Módulo *KeyFramer*

Para armazenar os *frames* chaves, bem como, a animação total, escolheu-se parar o desenvolvimento do protótipo utilizar uma estrutura de dados do tipo vetor, e determinou-se a quantidade máxima de *frames* igual a 90, a título de testes. Então foi criado uma classe que determina um novo tipo de dado, denominado de *keyframe*, que possui a hierarquia de dados do avatar, como está na Figura 4.17.

```

class KeyFrame {
    public float[] ombroD = new float[3];
    public float[] cotoveloD = new float[3];
    public float[] polegarD = new float[3];
    public float[] indicadorD = new float[3];
    public float[] medioD = new float[3];
    public float[] anelarD = new float[3];
    public float[] minimoD = new float[3];
    public float[] ombroE = new float[3];
    public float[] cotoveloE = new float[3];
    public float[] polegarE = new float[3];
    public float[] indicadorE = new float[3];
    public float[] medioE = new float[3];
    public float[] anelarE = new float[3];
    public float[] minimoE = new float[3];
}

```

Figura 4.17: Classe *KeyFrame*

Cada atributo da classe *KeyFrame* representa as partes móveis do corpo (braço, antebraços e mãos) do avatar, eles são representados por vetores do tipo *float* de três posições.

Cada posição representa o ângulo de rotação dos eixos X, Y e Z.

O *módulo KeyFrame* é composto das seguintes funcionalidades:

- **Barra de *Frames*:** barra indicadora que informa a quantidade de *frames* e determina os *frames* chaves da animação.
- **Barra de Controle de Velocidade:** determina a velocidade do movimento para cada intervalo de *frames*.
- **Botão Quadro Chave:** depois que o *frame* foi escolhido na barra de *frames* e o movimento com avatar foi especificado, através do *módulo* controle, o botão quadro chave captura dados deste movimento, relativo a parte do corpo que foi movimentada, número do *frame* e ângulos de rotação X, Y e Z (armazenando estes, em um vetor de *frames* chave). As coordenadas do movimento são capturadas, do *módulo* controle para o *módulo* *keyframe*, através de métodos chamados *passagemX()*, *passagemY()* e *passagemZ()*. A Figura 4.18 mostra esquematicamente a passagem de parâmetros de coordenadas usando estes métodos.

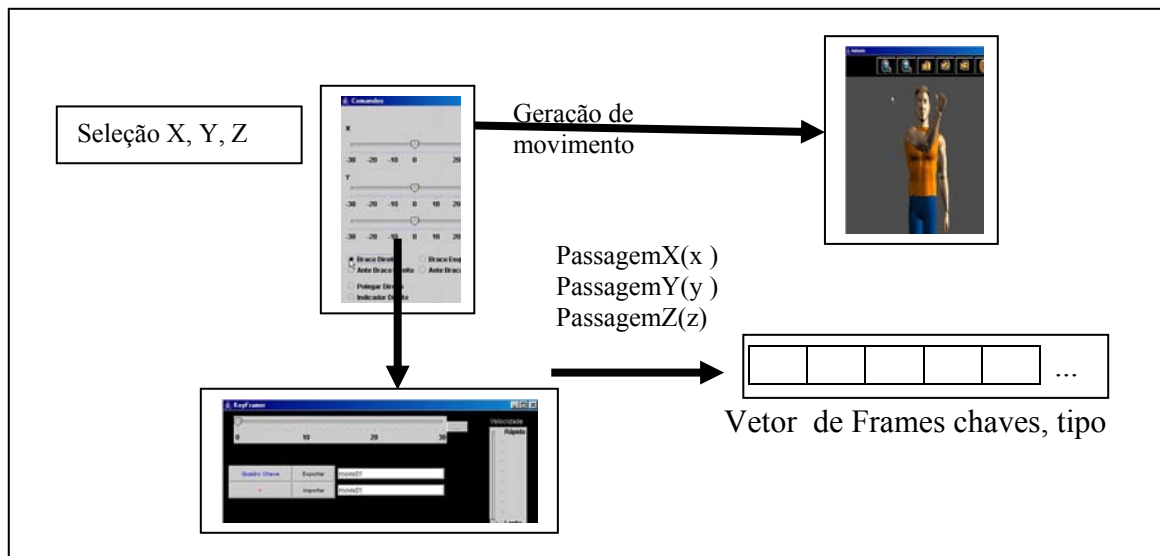


Figura 4.18: Esquema de funcionamento

A Figura 4.19 apresenta parte do código fonte do módulo Controle, onde é implementado a passagem das coordenadas X, Y e Z para o módulo *KeyFrame*.

```

if(eixoX.getValueIsAdjusting()==true)
{
    auxY=0.0f;
    auxZ=0.0f;
    auxX =(float) eixoX.getValue();
    valorbracodireitoX=eixoX.getValue();
    Ambiente.interpolador.rotBracoD(0,auxX);
    Ambiente.interpolador.rotBracoD(1,auxY);
    Ambiente.interpolador.rotBracoD(2,auxZ);

    //==Passagem de parametro dos exios X, Y e Z para o módulo KeyFrame==
    keyFramer.passagemX(auxX);
    keyFramer.passagemY(auxY);
    keyFramer.passagemZ(auxZ);
}

```

Figura 4.19: Passagem das coordenadas X, Y e Z para o *módulo KeyFrame*.

A Figura 4.20 mostra o trecho de programa relacionado o gerenciamento e armazenamento de dados recebidos pelo métodos *passagemX()* pelo *módulo KeyFramer*. Os métodos *passagemY()* e *passagemZ()* seguem a mesma lógica.

```

metodo passagemX(x: float)
{
    se(Controles.btnBracoDireito está selecionado) então
        cX=x;

    se(Controles.btnBracoEsquerdo está selecionado) então
        cX=x;

    se(Controles.btnAnteBracoDireito está selecionado) então
        cX=x;

    se(Controles.btnAnteBracoEsquerdo está selecionado) então
        cX=x;

    se(Controles.btnPolegarDireito está selecionado) então
        cX=x;

    se(Controles.btnIndicadorDireito está selecionado) então
        cX=x;

    se(Controles.btnMedioDireito está selecionado) então
        cX=x;

    se(Controles.btnAnelarDireito está selecionado) então
        cX=x;

    se(Controles.btnMinimoDireitoestá selecionado) então
        cX=x;
} //fim do metodo

```

Figura 4.20: Pseudocódigo do método passagemX()

Tendo os valores das coordenadas X, Y e Z armazenadas nas variáveis cX, cY e cZ, e tendo qual parte do corpo foi selecionado, *módulo KeyFrame* armazena esses dados no vetor de *frames* chaves (Figura 4.21)

```

if(Controles.btnBracoDireito.isSelected()==true)
{
    frameCHAVE[indiceFrame].ombroD[0]=angulo;
}

```

Figura 4.21: Trecho do código fonte do método que carrega do vetor de *frames* chaves

Por padrão o *frame* inicial é considerado 0, os valores de rotação também é 0. Quando se determina um *frame*, e um movimento, ao se ativar o botão *QuadroChave* é calculado a interpolação entre este e posição anterior (0, no caso) e assim por diante. A interpolação criará os dados relevantes aos eixos do movimento entre o intervalo.

- **Botão Play:** executa a animação definida, que pode ser visualizada no *módulo* ambiente.
- **Botão Exportar:** exporta a animação gerada para arquivo. Na Figura 4.22 é apresentado o layout do arquivo que possui a extensão **.mpr(MP3D)**.

<i>frame</i>	Velocidade	Braço direito	Braço esquerdo	Ante braço direito	Ante braço esquerdo	Polegar direito	Indicador direito	Médio direito	Anelar direito	Mínimo direito	Polegar esquerdo	Indicador esquerdo	Médio esquerdo	Anelar esquerdo	Mínimo esquerdo
No.	valor	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z	x,y;z

Figura 4.22: Layout do arquivo de trabalho da aplicação.

- **Botão Importar:** importa a animação armazenada em arquivo, segundo o layout da Figura 4.22. Depois do arquivo importado, utiliza-se o botão play para executar a animação que pode ser visualizada no *módulo* ambiente.

5. TESTE DO PROTÓTIPO E ANÁLISE DE RESULTADOS OBTIDOS

Neste capítulo são apresentados os testes realizados com o protótipo, afim de analisar a viabilidade e a qualidade desta aplicação para geração de movimentos parametrizados para HV.

Para fins de teste e comparação, foi utilizada a técnica de animação conhecida por rotoscopia, técnica criada por Max Fleischer em 1914, para os estúdios Disney, que consiste no animador analisar cada fotograma de uma ação ao vivo para, depois, aplicar desenhos sobre eles; ou seja; com esse mecanismo é possível projetar e traçar um livre movimento humano, transpondo-o para um retrato animado um *frame* de cada vez.

Esta comparação foi realizada se confrontando movimentos com os mesmos movimentos gerados por interpolação de quadros e por rotoscopia. Na Figura 5.1 demonstra-se o uso da rotoscopia em animação 3D.

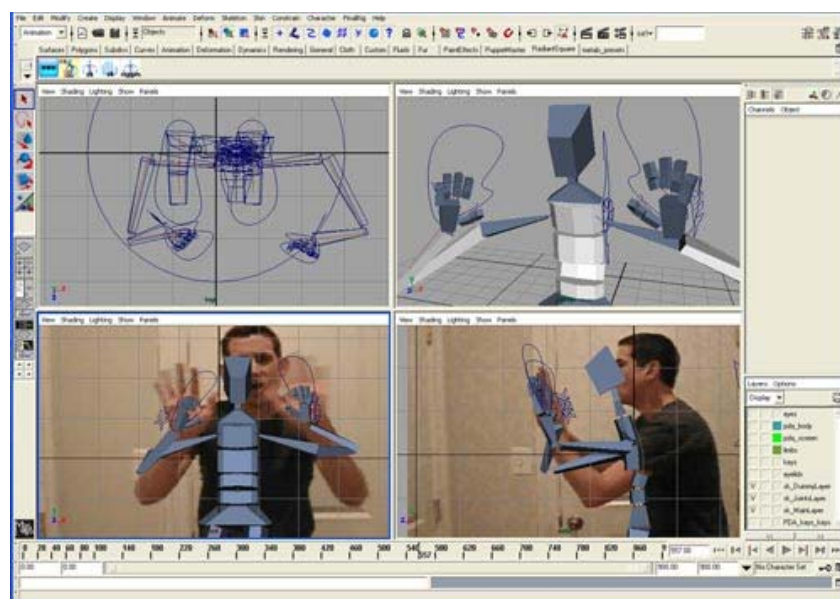


Figura 5.1: Exemplo da rotoscopia em animação 3D (LAMBERTON 2004)

5.1 Recursos Utilizados

Para o desenvolvimento do protótipo, utilizou-se como recursos de *softwares* o demonstrando na Tabela 1.

Tabela 1: *Softwares* utilizados para Desenvolvimento

Programa	Qualificação	Desenvolvedor	Versão	Objetivo
Java3D	API Gráfica	Sun	1.3.1	Renderização de Objetos 3D
DirectX	Biblioteca Multimídia	Microsoft	8.1	Suporte aos Dispositivos de vídeo do Sistema Operacional
Java SDK 1.4.2	Gerador de Aplicações	Sun	1.4.2	Desenvolvimento de Sistemas baseados em linguagem C e C++
Windows XP	Sistema Operacional	Microsoft	XP	Sistema Operacional utilizado para desenvolvimento do protótipo

E como recursos de *Hardware* estão na Tabela 2.

Tabela 2: Recursos de *Hardware* para Desenvolvimento

Hardware	Descrição
Computador PC(<i>Personal Computer</i>)	128 <i>megabytes</i> de memória RAM (<i>Random Access Memory</i>) e processador Pentium 2.4 GHz.
Placa Aceleradora Gráfica	Modelo Sis 630, 32 MegaBytes, otimizada para DirectX
Monitor de Vídeo	AOCA, modelo Wt-58C

Para os testes além dos itens citados nas Tabela 1 e Tabela 2, optou-se pelo Windows 98 e pelos recursos de *hardware* existentes na Tabela 3, para se verificar a viabilidade do protótipo e se determinar uma configuração de *hardware* mínima.

Tabela 3: Recursos de *Hardware* utilizados para testes

Hardware	Descrição
Computador PC(<i>Personal Computer</i>)	96 <i>megabytes</i> de memória RAM (<i>Random Access Memory</i>) e processador Athlon 1.1 GHz.
Placa Aceleradora Gráfica	Modelo Sis 630, 32 MegaBytes, otimizada para DirectX (<i>onboard</i>)
Monitor de Video	AOCA, modelo WT-58C

5.2 Testes realizados e resultados obtidos

Como citado no Capítulo 4, uma das funcionalidades deste protótipo é a exportação dos dados gerados para arquivo e sua recuperação, não somente a geração do movimento. Na Figura 5.2 se tem um exemplo de um movimento simples, que utiliza os braços e antebraços direitos e esquerdo, criando a movimentação de um abraço.



Figura 5.2: O avatar simulando um abraço

As Figuras 5.3, 5.5 , 5.7 e 5.9 detalham as partes envolvidas no movimento e seus respectivos parâmetros, armazenadas no arquivo, segundo o layout apresentado na Figura 4.22, e as Figuras 5.4, 5.6, 5.8 e 5.10 mapeiam os parâmetros em forma de gráficos. Neste caso foi usando simplesmente interpolação

```

0;10;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
1;10;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
2;10;1.4444444;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
3;10;2.8888888;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
4;10;4.3333333;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
5;10;5.7777777;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
6;10;7.2222223;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
7;10;8.6666666;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
8;10;10.1111111;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
9;10;11.5555555;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
10;10;13.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0

```

Figura 5.3: Movimento Braço Direito

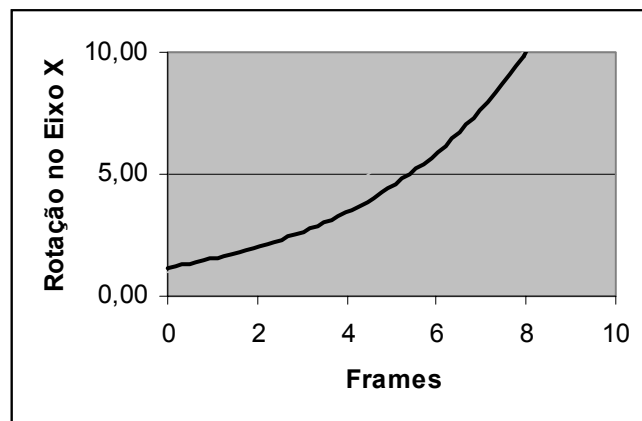


Figura 5.4: Gráfico do Movimento do Braço Direito

```

11;10;0.0;0.0;0.0;7.368421;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
12;10;0.0;0.0;0.0;8.105263;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
13;10;0.0;0.0;0.0;8.842105;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
14;10;0.0;0.0;0.0;9.578947;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
15;10;0.0;0.0;0.0;10.315789;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
16;10;0.0;0.0;0.0;11.052631;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
17;10;0.0;0.0;0.0;11.789474;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
18;10;0.0;0.0;0.0;12.526316;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
19;10;0.0;0.0;0.0;13.263158;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
20;10;0.0;0.0;0.0;14.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0

```

Figura 5.5: Movimento Braço Esquerdo

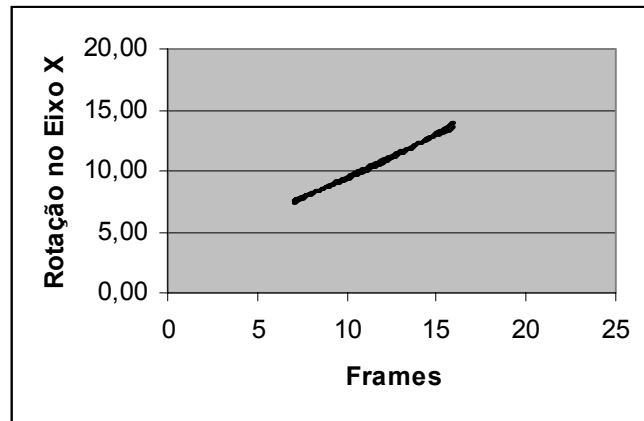


Figura 5.6: Gráfico do Movimento do Braço Esquerdo

```

21;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;8.965517;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
22;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;9.413794;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
23;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;9.862069;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
24;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;10.310345;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
25;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;10.758621;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
26;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;11.206897;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
27;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;11.655172;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
28;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;12.103448;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
29;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;12.551724;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
30;100;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;-13.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
    
```

Figura 5.7: Movimento do Antebraço Direito

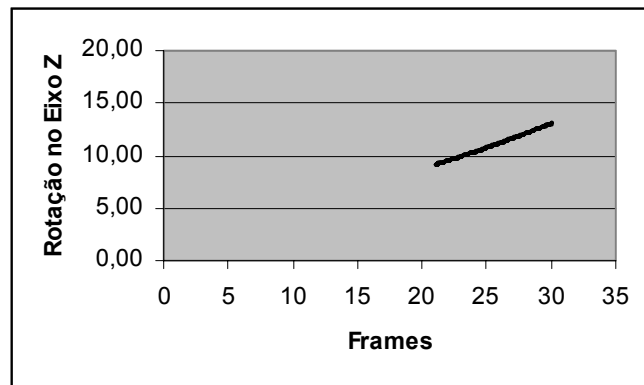


Figura 5.8: Gráfico do Movimento do Antebraço Direito

```

31;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;10.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
32;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;10.333334;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
33;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;10.666667;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
34;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;11.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
35;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;11.333334;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
36;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;11.666667;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
37;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;12.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
38;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;12.333334;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
39;44;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;12.666667;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0;0.0
    
```

Figura 5.9: Movimento do Antebraço Esquerdo

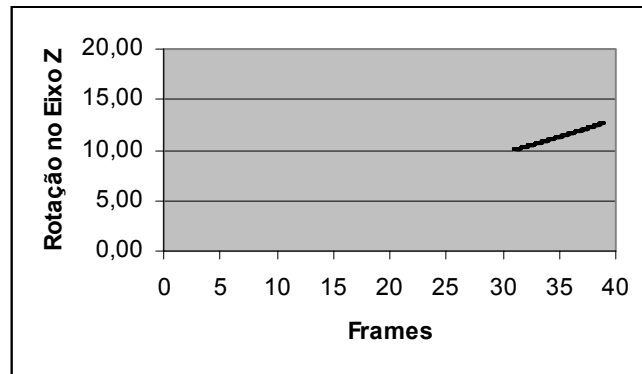


Figura 5.10: Gráfico do Movimento do Antebraço Esquerdo

Avaliou-se a influência da quantidade de quadros chaves na qualidade da animação resultante. Para esta avaliação criou-se duas animações com o mesmo movimento.

Para esta avaliação preliminar levou-se em conta atributos importantes para se ter subsídio para uma comparação, como: mesmo número de quadros totais (30 quadros, neste caso), mesma velocidade de movimento, já que o protótipo permite aplicar velocidades diferenciadas de forma global (velocidade genérica) ou velocidades diferenciadas (para cada conjunto de quadros), coordenadas espaciais de posição do avatar de movimento. A Figura 5.7 demonstra o movimento gerado.



Figura 5.7: Demonstração da seqüência de movimento para comparação de qualidade de acordo com o número de *frames*

Verificou-se que com uma velocidade constante um movimento linear, o que não deixa como um movimento natural. Gerou-se um gráfico de dispersão com estes dados (Figura 5.8).

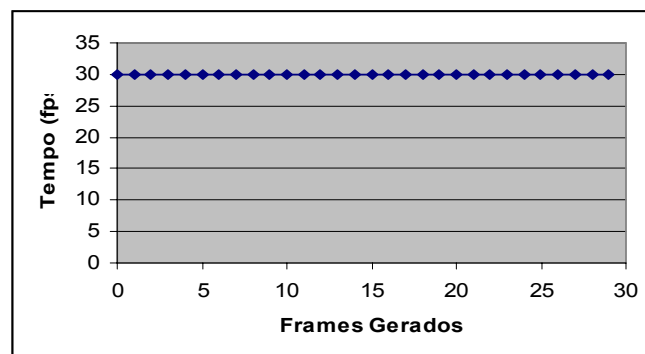


Figura 5.8: Gráfico de desempenho com variação de tempo constantes

Utilizando o mesmo exemplo, variou-se os *frames* por segundo entre os intervalos de interpolação (0-10, 11-20 e 21-30) por exemplo, 30, 80 e 50, obtendo-se um movimento não tão linear como o anterior, como o gráfico na Figura 5.9.

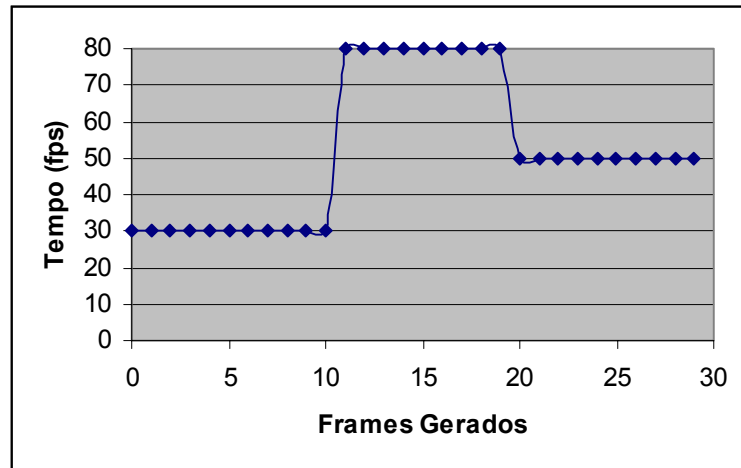


Figura 5.9: Variando *frame* por segundo

Então aumentou-se *frames* chaves (Figura 5.10), mantendo mesmo movimentos e inserindo-se mais variações entre *frames* por segundo e o resultado encontrado se aproximou do resultado final, que no caso de uma simulação é aceitável.

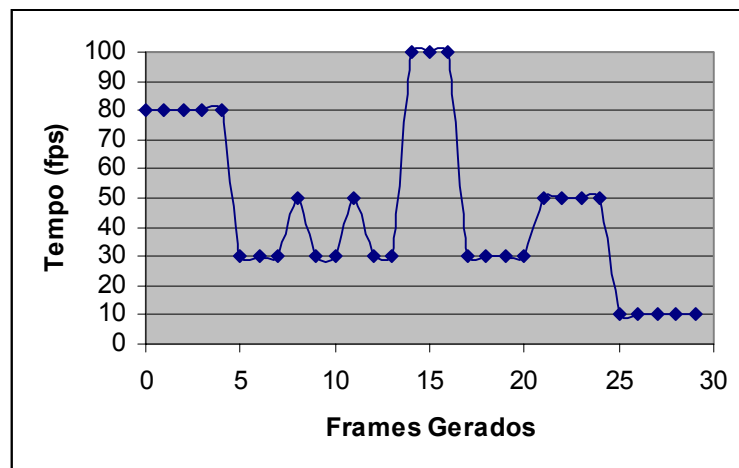


Figura 5.10: Gráfico de movimento com variação de *frames* por segundo e quadros chaves

Conclui-se que quando maior a quantidade de *frames* chaves e variação da velocidade deste, se pode obter um resultado comparável ao movimento humano.

5.3 Avaliação da qualidade do movimento utilizando roscopia

Para se confrontar o movimento gerado pelo protótipo e um movimento humano real utilizou-se a roscopia como parâmetro para a comparação.

Selecionou-se uma serie de movimentos diferentes como: um abraço, um aceno e também algumas letras em LBS. Então com ajuda de um ator foram gravados esses movimentos com uma câmera fotográfica digital Olympus D-395. Gerou-se um arquivo no formato *filmstrip* (COMPUPHASE 2004). Este arquivo cria uma seqüência de *frame a frame* do vídeo, como se fosse um filme de celulose, como se segue na Figura 5.11.



Figura 5.11: Exemplo de um arquivo *filmstrip*

Com base neste arquivo gerou-se a animação no protótipo acompanhando os movimentos do ator. O exemplo apresentado e analisado a seguir representa a seqüência de movimenta do letra "a" em LBS (Figura 5.12).

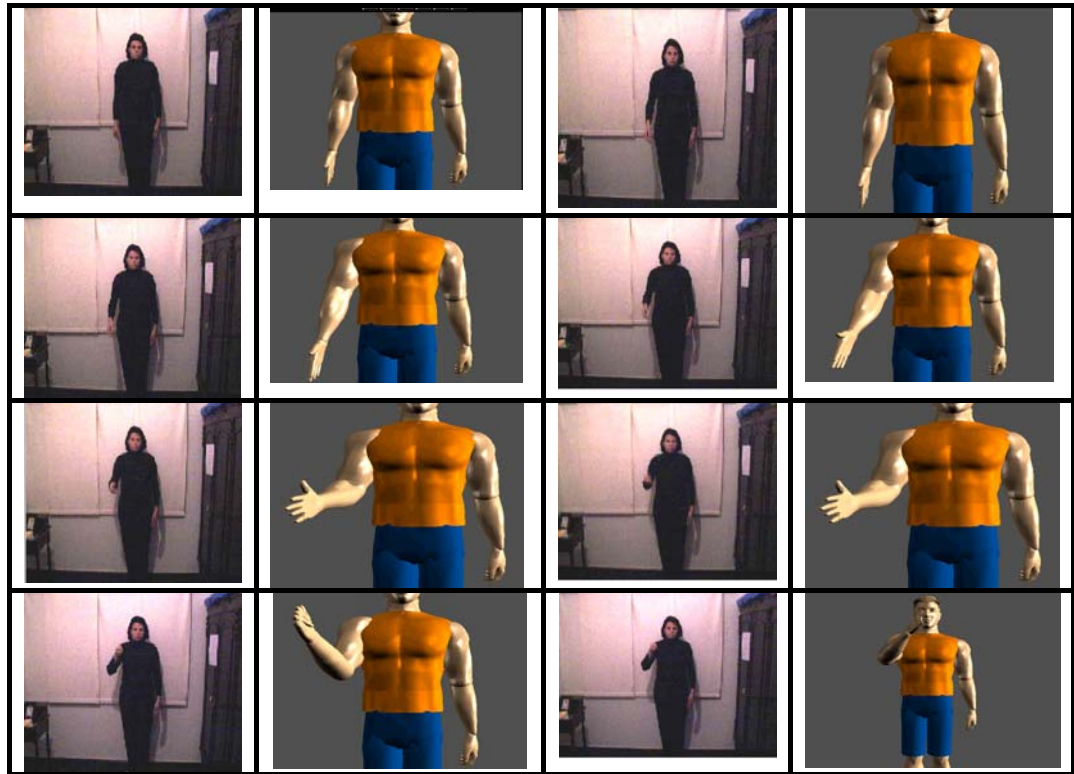


Figura 5.12: Exemplo da geração de movimento utilizando rotoescopia "a"

Os dados do movimento, *frames* chaves e *frames* por segundo, gerados utilizando o recurso de rotoescopia estão na Tabela 4.

Tabela 4: Dados de movimentos simulados com rotoescopia - letra "a"

<i>Frames</i> chaves	<i>Frames</i> por segundo
0	1
10	80
15	80
20	50
30	50
35	50
40	50
50	50
60	30
70	30
80	30
90	30

Com estes valores, gerou-se a curva de movimento, que está no gráfico de dispersão, na Figura 5.13.

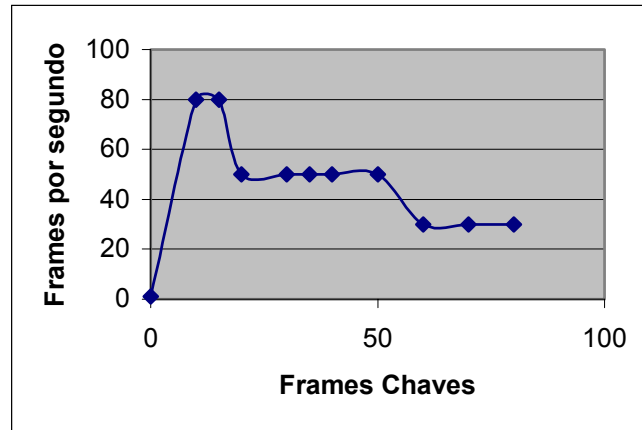


Figura 5.13: Curva de movimento de simulação usando rotoscopia - letra "a"

Outro teste utilizando rotoscopia foi a criação da letra "i" em LBS, como está na Figura 5.14

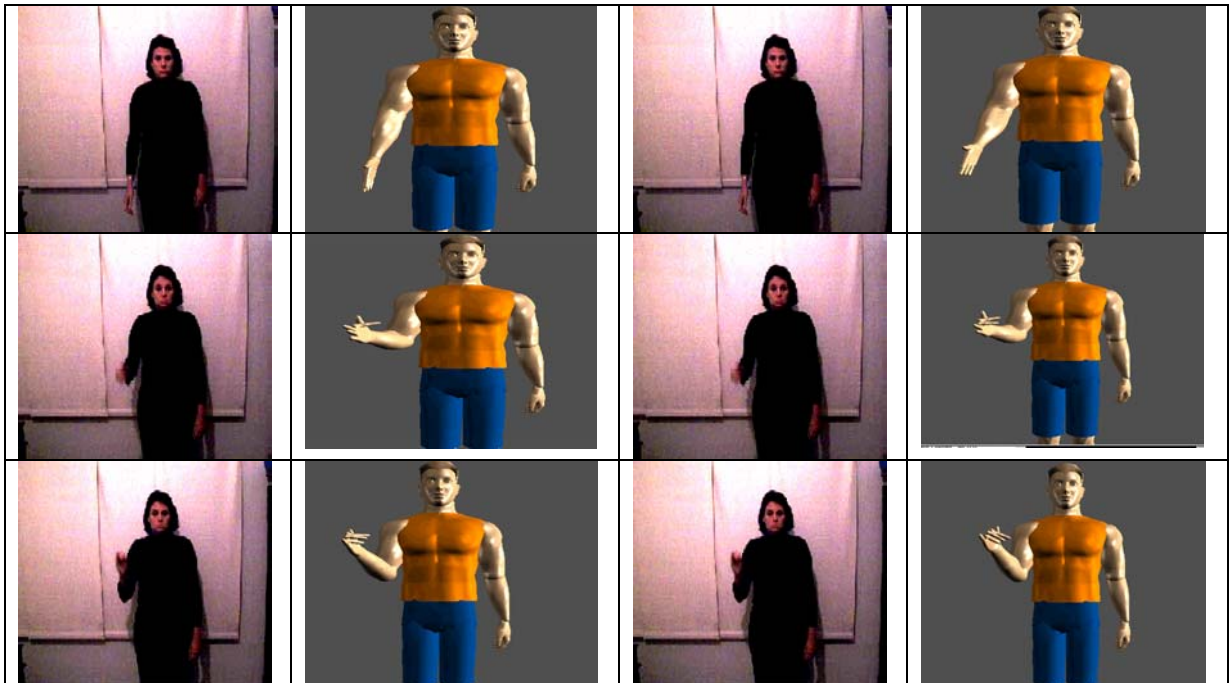


Figura 5.14: Exemplo de geração de movimento usando a letra "i"

Os dados coletados, *frames* chaves e *frames* por segundo, da geração de movimento do exemplo da Figura 5.14 constam da Tabela 5

Tabela 5 : Dados de movimentos simulados com rotoscopia - letra "i"

<i>Frames chaves</i>	<i>Frames por segundo</i>
0	80
10	80
15	50
20	30
30	50
35	60
40	60
50	60
60	80
70	60
80	40
90	40

Com estes valores gerou-se o gráfico representado pela Figura 5.15

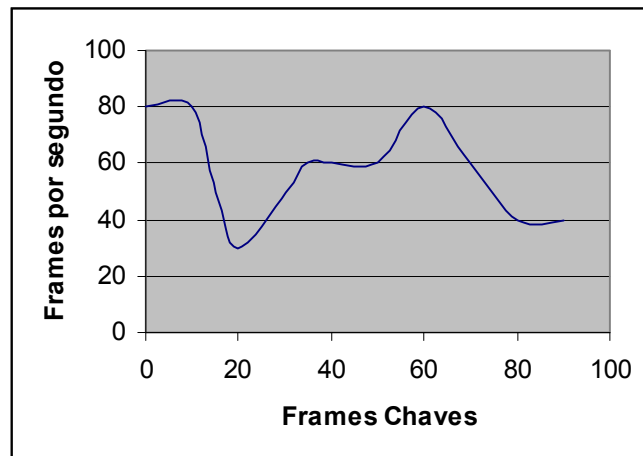


Figura 5.15: Curva de movimento de simulação usando rotoscopia - letra "i"

No teste com a rotoscopia, foram usados o mínimo de quadros possíveis, tendo um aumento da definição de quadros chaves, já que, se tinha uma referência espacial do movimento. A rotoscopia funcionou nesta seqüência de teste como um *storyboard* para o movimento final.

Então gerou-se o mesmo movimento sem rotoscopia, o que exigiu uso maior da técnica de interpolação de *frames*, já que não se tinha referência e os quadros chaves foram

reduzidos, devido a determinação algébrica dos mesmos. Os dados comparativos se encontram na Tabela 6.

Tabela 6: Dados de movimento simulado sem rotoscopia

<i>Frames Chaves</i>	<i>Frames por segundo</i>
0	1
10	80
20	80
30	50
40	30
50	30
60	30

Com base nestes dados, chegou-se ao gráfico de dispersão, que representa a curva de movimento em questão (Figura 5.16)

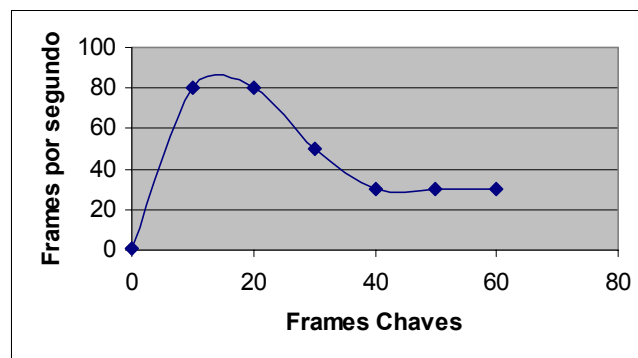


Figura 5 16: Curva de movimento da simulação sem rotoscopia

Após a experimentação do protótipo nestes dois casos apresentados (gráficos Figuras 5.13 e 5.15) e com os devidos resultados, se observa que, a animação gerada por interpolação de quadros chaves, técnica conhecida por *keyframing* (THALMANN e THALMANN 1990), obtém-se um resultado de simulação de movimento aceitável, próximo ao real.

O último teste realizado foi quando a performance de *hardware* de acordo com as Tabelas 1 e 2 já apresentadas. Este teste foi feito apenas para determinar a configuração mínima utilizada pelo protótipo. A Figura 5.17 apresenta a comparação entre as configurações de *hardware* e sistemas operacionais utilizados.

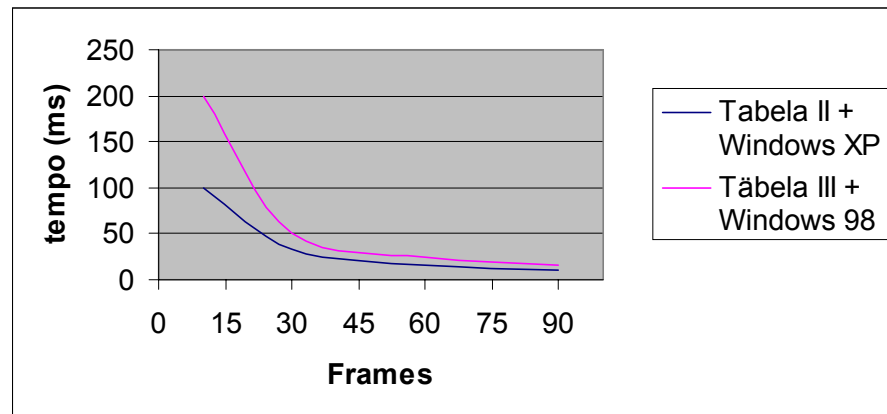


Figura 5.17: Gráfico comparativo entre as duas configurações utilizadas.

Quanto à performance, também, os testes mostraram que em relação às animações armazenadas em arquivos, a quantidade de *frames* é diretamente proporcional ao tempo que o protótipo leva para carregá-los. Porém não houve queda de desempenho na execução da animação propriamente dita.

CONCLUSÕES

O sistema de *keyframing* apresentado, proporciona ao usuário a simulação de movimentos (criação, armazenamento e recuperação) através de um avatar, e isso é feito de forma simples utilizando controles que selecionam as partes do avatar que é animado, que ao ser selecionado ter sua cor alterada explicitando a seleção, e os eixos de rotação no espaço, permitindo a captura de uma seqüência de movimentos e animação dos mesmos, podendo ser armazenado em forma de arquivo e recuperando quando for conveniente. Assim abrindo um espaço de aplicações sem precedentes, determinando um ponto importante por não exigir dos usuários nenhum equipamento ou treinamento especial, para interagir com a aplicação desenvolvida.

Este capítulo finaliza o trabalho, apresentando os aspectos gerais e as contribuições MP3D, protótipo de geração de movimento parametrizado para HV.

Aspectos gerais do trabalho

Embora existam de várias técnicas de captura de movimento atuais e comprovadas pelo uso principalmente no cinema, se escolheu a técnica de interpolação de quadros chaves

(*keyframing*) justamente pela sua simplicidade quanto aos equipamentos requeridos e resultados aceitáveis na validação de geração de movimentos parametrizados.

Para se chegar a esta conclusão os mesmos testes foram realizados, gerando-se os *frames* da animação com sua velocidade controlada pela quantidade de *frames* (30 *frames* para cada segundo de animação) e também por um controle de velocidade externo a interpolação dos quadros. E evidenciou-se que controlando a velocidade da animação através da quantidade de *frames*, consegue-se um resultado mais realístico e menos artificial.

O que não deixa técnicas como o rastreamento de movimento de fora, o que é normalmente adaptável ao protótipo, já que existe a funcionalidade de recuperar os dados de arquivos movimento gerados. Sabe-se que existe uma gama infinita de combinações de movimento que se pode pensar em realizar, o que se fez foi uma simplificação do que se podia ser feito em termos de movimento, criando uma biblioteca de movimentos simplificada, na qual os módulos do protótipo se baseiam na geração de movimentos, nas seguintes partes do avatar humanóide: braços e antebraço direito, braço e antebraço esquerdo, e mão direita.

Apesar disso acredita-se que uma gama de aplicações poderia se beneficiar das funcionalidades desse protótipo, principalmente a área do ensino, onde poderiam ser simuladas tarefas procedimentais, sinais de socorro náuticos ou sinais em LBS. Devido a tridimensionalidade que a RV proporciona, se tem uma noção de profundidade, distância e ao usuário visualizar a simulação de vários ângulos diferentes de câmera.

Contribuições

O uso de HV no processo simulações tem minimizado, principalmente, na educação dois fatores importantes: o tempo e o retorno do treinamento ou os usuários que participam daquela simulação, pois os recursos de RV possibilitam a seu usuário a presença em um AV que muitas vezes seria fisicamente impossível sua interação, por exemplo, uma visita ao planeta Marte, ou a manipulação de uma molécula.

No caso do protótipo, as simulações utilizando interpolação de quadro chave, se evidenciou, por exemplo, na criação de um movimento em particular ou uma letra em LBS, a possibilidade do usuário que visualizava aquele movimento poder ter precisamente a noção do movimento, distancias entre o corpo e as partes do corpo envolvida, tridimensionalidade espacial, aproximar ou afastar a câmera, poder movimentar o avatar em torno de um eixo de translação para ver com as partes do corpo se comportam numa visão diferente e repetir o movimento quantas vezes forem necessárias.

Em outro extremo está o usuário que gera a simulação. O MP3D possui uma interface de alto nível para gerar a simulação, o que permita pessoas que não possuam conhecimentos de programação, computação gráfica e RV, o uso da ferramenta, e também, é mais simples que os *softwares* de animação 3D existentes no mercado (como 3D Studio Max), no qual o usuário tem que ter um conhecimento profundo da ferramenta para gerar uma animação qualquer.

Também pensando na geração da simulação, se ela fosse gerada em tempo de programação, se tornaria algo prática e humanamente inviável já que seria contra-produtivo,

por exemplo, considerarmos a geração de uma biblioteca em LIBRAS que possui milhares de sinais.

Trabalhos Futuros

- Interação com rastreador;
- Seleção e movimentação das partes do avatar humanóide utilizando mouse;
- Módulo de correção de movimento e
- Módulo de geração de animação em arquivo de vídeo (avi, mov, flc, etc.)

REFERÊNCIAS

ALIAS. **Maya**. Disponível em: <http://www.alias.com/eng/index.shtml>. Acesso em: 23 Ago. 2005

ANDRÉ, E., RIST, T., VAN MULKEN, S., KLESEN, M. e BALDES, S. (2000). **The Automated Design of Believable Dialogues for Animated Presentation Teams**. In: Cassell, J., Sullivan, J., Prevost, S. and Churchill, E. (eds.): *Embodied Conversational Agents*, 220-255, Cambridge, MA: MIT Press.

AUTODESK. **3D Studio Max**. Disponível em: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5358014>. Acesso em: 23 Ago. 2005

AZUOLA, F., BADLER, N., HO, P.-H., KAKADIARIS, I., METAXAS, D., e . TING, B. **Building anthropometry-based virtual human models**. In *Proc. IMAGE VII Conf.*, 1994.

BADLER N. "**Real-time virtual humans**," Pacific Graphics 1997.

BADLER N. I. e KO H. **Animating human locomotion in real-time using inverse dynamics**. IEEE Computer Graphics & Applications, 1996

BADLER, N., ALLBECK, J., ZHAO, L., BYUN, M. "**Representing and Parameterizing Agent Behaviors**." In *Life-like Characters: Tools, Affective Functions and Applications*. Helmut Prendinger and Mitsuru Ishizuka, Ed. Springer, Germany, 2003.

BADLER, N., ERIGNAC, C. e LIU, Y. "**Virtual humans for validating maintenance procedures**." *Comm. of the ACM*, July 2002, pp. 57-63.

BAERLOCHER, P., BOULIC, R., **Parametrization and range of motion of the ball-and-socket joint**, Proc. of AVATARS'2000 Conference, Lausanne, November 2000. A shortened version has been published in "Deformable Avatars", Kluwer Academic Publishers Publ., pp.180- 190, 2001.

BARAFF D. **Linear-time dynamics using lagrange multipliers**. In SIGGRAPH 96 Conference Proceedings, Computer Graphics Proceedings, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1996. ISBN 0-201-94800-1.

BINDIGANAVALE, R. **Department of Computer and Information Science - Peen Engineering**, 1999. Disponível: <http://www.cis.upenn.edu/~rama/work.html>. Acesso em: 28 de ago. de 2004.

BARZEL, R. e BARR, A.H. **A modeling system based on dynamics**. In Proceedings of ACM SIGGRAPH. Addison Wesley, July 1988.

BARZEL, R. e BARR, A.H. **A modeling system based on dynamics**. In Proceedings of ACM SIGGRAPH. Addison Wesley, July 1988

BOULIC R. e THALMANN D.. **Combined direct and inverse kinematic control for articulated _gures motion editing**. Computer Graphics Forum, 1992.

BOULIC, R., MAGNENAT-THALMANN, N., e THALMANN, D. **A global human walking model with real-time kinematic personi_ cation**. Visual Computer, December 1990.

BRUDERLIN, A. e CALVERT, **Knowledge-driven, interactive animation of human running**. In Graphics Interface'96, May 1996.

BRUDERLIN, A. e CALVERT, T. , **Goal-directed, dynamic animation of human walking**. Computer Graphics (SIGGRAPH '89 Proceedings), Vol. 23, No. 3, pp. 233--242, July 1989. **of legged _gures**. In Proceedings of ACM SIGGRAPH. Addison Wesley, July 1985.

CASELL, J., PELACHAUD, C., BADLER, N., STEEDMAN, M., ACHORN, B., BECKET , W., DOUVILLE, B., PREVOST, S., e STONE, M.. **Animated conversation: Rule-based generation of facial expression,gesture and spoken intonation for multiple conversational agents**. In *Computer Graphics, Annual Conf. Series*, pages413–420. ACM, 1994.

COMPUPHASE. **Filmstrip File Format**. 2004. Disponível em: <http://www.compuphase.com/filmstrp.htm>. Acesso em: 28 Ago. 2005

CONDE T. , TAMBELLINI W. , e THALMANN D. **“ Behavioral Animation of Autonomous Virtual Agents Helped by Reinforcement Learning”** , 4th International Workshop on Intelligent Virtual Agents (IVA'03), Kloster Irsee, Germany, Lecture Notes in Computer Science vol. 272,175-180, Springer-Verlag: Berlin, 2003.

CUSIN, G. M. **Ambiente virtual interpretador: uma proposta de tradução de sentenças de língua portuguesa para língua brasileira de sinais.** Marília: 2004. 116p.

DECARLO, D. e METAXAS, D. **The integration of optical flow and deformable models with applications to human face shape and motion estimation.** In *Proc. CVPR*, pages 231–238. IEEE Press, 1996.

DOUVILLE, B., LEVISON, L., e BADLER, N.. **Task level object grasping for simulated agents.** *Presence*. 1996

ESSA, I. e PENTLAND, A.. **Facial expression recognition using a dynamic model and motion energy.** In *Proc. of the International Conf. on Computer Vision*, Cambridge, MA, 1995.

FAURE F.. **An energy-based method for contact force computation.** In Proceedings of Eurographics' 96, August 1996. Computer Graphics Forum, Volume 15, Number 3.

FAURE, F., DEBUNNE, G., M.P., CANI-GASCUEL, e MULTON, F. **Dynamic analysis of human walking.** In Eurographics Workshop on Computer Animation and Simulation. September 1997.

FUSCO, E. X- **Libras : um Ambiente Virtual para a língua brasileira de sinais.** Marília: 2004. 146p.

GASCUEL, Jean-Dominique e GASCUEL Marie-Paule. **Displacement constraints for interactive modeling and animation of articulated structures.** The Visual Computer. March 1994.

GRANIERI J., CRABTREE J., e BADLER N.. **Off-line production and real-time playback of human figure motion for 3D virtual environments.** In *VRAIS Conf.* IEEE Press, 1995.

GRATCH, J., RICKEL, J., ANDRE, E., BADLER, N., CASSELL, J. e PETAJAN, E. **Creating interactive virtual humans: Some assembly required.** IEEE Intelligent Systems, 2002.

HODGINS, J.K., WOOTEN, W.L., BROGAN, D.C., e O'BRIEN, J.F. **Animating human athletics.** In Proceedings of ACM SIGGRAPH, Los Angeles, California, August 1995.

HODGINS, J.K., WOOTEN, W.L., BROGAN, D.C., e O'BRIEN, J.F. **Animating human athletics.** In Proceedings of ACM SIGGRAPH, Los Angeles, California, August 1995.

ISAACS, P.M. e M.F. COHEN., **Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics**. In *Computer Graphics (SIGGRAPH '87 Proceedings, 1987*, vol. 21, pp. 215--224.

KAKADIARIS, I. e METAXAS, D.. **Model-based estimation of 3D human motion with occlusion based on active multiviewpoint selection**. In *Proc. of the Conf. on Computer Vision and Pattern Recognition*, pages 81–87. IEEE Computer Society, June 1996.

KO H. **Kinematic and Dynamic techniques for Analyzing, Predicting, and Animating Human Locomotion**. PhD thesis, University of Pennsylvania, 1994.

LAMBERTON, J. C. **Projects of Institute for Computer Graphics, George Washington University**. 2004. Disponível em <http://www.jasonlamberton.com/gwu.htm>. Acesso em 26 de jul. de 2005

LE D. D, BOULIC R., THALMANN, D. “**Intergrating Age Attributes to Virtual Human Locomotion**”, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV-5/W10, 2003.

LEVISON, L.. **Connecting planning and acting via objectspecific reasoning**. PhD thesis, CIS, University of Pennsylvania, 1996

MAES P., DARRELL T., BLUMBERG B., e PENTLAND A.. “**The ALIVE system: Full-body interaction with autonomous agents**”. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation*, pages 11–18. IEEE Computer Society Press, Los Alamitos, CA, 1995.

MULTON, F., FRANCE, L., CANI-GASCUEL, M.,P., DEBUNNE, G. **Computer Animation of Human Walking: a Survey**, 1999.

RICKEL, J., JOHSON, L.W., **Animated Agentes for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control**. *Applied Artificial Intelligence*, vol. 13, nos. 4--5, June--Aug. 1999, pp. 343--382.

ROUSSEAU, D. e HAYES-ROTH, B. “**Personality in synthetic agents**”. Technical Report KSL-96-21, Stanford Knowledge Systems Laboratory, 1996.

SABARISH, B., ZANBAKA, C., JACKSON, J., CHUNG, T., LOK, B., SHIN, M., e HODGES ,Larry. **Virtual Human Physiotherapist Framework for Personalized Training and Rehabilitation**. *Graphics Interface 2005*, Victoria, British Columbia, Canada, May 9-11, 2005.

STANSFIELD ,S., CARLSON, D., Hightower, R. e . SOBEL, A.. **A prototype VR system for training medics.** In *MMVR Proc.*, 1997.

THALMANN D. e THALMANN N. M., **Computer Animation: Theory and Practice**, Springer-Verlag. 1990

THOMAS, F. AND JOHNSON, O. “**The Illusion of Life**”,Abbeville Press, New York, 1981.

WATT A. e WATT M.. **Advanced Animation and Rendering Techniques: Theory and Practice.** ACM Press, 1992.

WITKIN ,A. e KASS, M. **Spacetime constraints.** In Proceedings of ACM SIGGRAPH, Atlanta, Georgia, August 1988. Addison Wesley.

ZELTZER, D. **Knowledge-based animation.** In ACM SIGGRAPH/SIGART, Workshop on Motion: representation and perception. 1996

ZELTZER, D. **Motor control techniques for _gure animation.** IEEE Computer Graphics and Applications, November 1982.