

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ANDRÉIA RODRIGUES DE ASSUNÇÃO
SCHNEIDER

**Animação de Humanos Virtuais Aplicada
para Língua Brasileira de Sinais**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof^ª Dr^ª. Luciana Porcher Nedel
Orientador

Porto Alegre, abril de 2008

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Schneider, Andréia Rodrigues de Assunção

Animação de Humanos Virtuais Aplicada para Língua Brasileira de Sinais / Andréia Rodrigues de Assunção Schneider.
– Porto Alegre: PPGC da UFRGS, 2008.

108 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2008. Orientador: Luciana Porcher Nedel.

1. Animação de Humanos Virtuais. 2. Língua de sinais.
I. Nedel, Luciana Porcher. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenadora do PPGC: Prof^a. Luciana Porcher Nedel

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“And in the naked light I saw
Ten thousand people, maybe more.
People talking without speaking,
People hearing without listening,
People writing songs that voices never share
And no one deared
Disturb the sound of silence.”*
— SIMON AND GARFUNKEL

AGRADECIMENTOS

São inúmeras as pessoas a agradecer. Primeiro aos meus pais que me ensinaram ser o conhecimento a maior riqueza que podemos possuir.

Agradeço aos professores do Grupo de Computação Gráfica de UFRGS (Carla Freitas, João Comba, Luciana Nedel e Manuel Oliveira) pelos ensinamentos transmitidos, aumentando cada vez mais meu conhecimento intelectual. Em especial à minha orientadora Luciana Nedel pelo auxílio quando necessário e pela oportunidade de amadurecimento intelectual e profissional.

Agradeço à Capes por ter financiado minha pesquisa, bem como ao *Innovation Center* UFRGS/Microsoft que financiou minha pesquisa durante os últimos meses.

Muito obrigada à “Tropa de Elite da CG” pelas dúvidas sanadas, bem como as discussões filosóficas, os churrascos, etc. Esse tempo de convívio com cada um de vocês me acrescentou mais cultura, bem como conhecimento técnico. Um agradecimento especial ao Renato Silveira pelo auxílio com a parte de Cinemática Inversa e ao Alex Ziegler, pela modelagem do humanóide e geração dos gestos.

Não posso me esquecer do Bruno Schneider, que foi companheiro, amigo, paciente e me ajudou com várias dúvidas técnicas durante o mestrado. Um agradecimento mais que especial à minha filha Mariana, por fazer parte da minha vida e muitas vezes ficar quietinha enquanto eu trabalhava. Obrigada também à Tia Suzi, que muitas vezes se dispôs em ficar com a Mariana para que eu pudesse trabalhar.

Um agradecimento mais que especial à Eliane e ao Roberto, pessoas muito importantes no final desta jornada e que sempre farão parte da minha vida e da minha família.

Obrigada também a todas as pessoas que participaram dos testes para a validação da técnica. Em especial à Daiana que ajudou na elaboração dos gestos.

Enfim, agradeço a todos aqueles que direta ou indiretamente sempre torceram pelo meu sucesso. Muito obrigada!

SUMÁRIO

LISTA DE FIGURAS	7
RESUMO	9
ABSTRACT	10
1 INTRODUÇÃO	11
2 ANIMAÇÃO DE HUMANOS VIRTUAIS	13
2.1 Modelagem do humano virtual	13
2.1.1 Fisiologia e biomecânica das mãos e membros superiores do corpo	14
2.1.2 O modelo desenvolvido	16
2.2 Geração das ações	19
2.3 Adequação de uma ação ao espaço disponível	21
2.3.1 Determinação da cadeia cinemática	22
2.3.2 Definição do alvo	22
2.3.3 O algoritmo	23
2.3.4 Colisão das juntas com a fronteira do espaço disponível	23
2.3.5 Tratamento de colisão entre as partes do corpo	26
2.4 Considerações finais	27
3 LÍNGUAS DE SINAIS	29
3.1 Fonologia das línguas de sinais	29
3.1.1 Fonemas nas línguas de sinais	30
3.1.2 Aspectos não-manuais das línguas de sinais	31
3.1.3 Restrições na formação de sinais	31
3.2 Morfologia das línguas de sinais	32
3.3 Sintaxe das línguas de sinais	34
3.4 Considerações finais	36
4 TRABALHOS RELACIONADOS	37
4.1 HST - Hand Sign Translator	37
4.1.1 A interface do HST	38
4.1.2 O processo de tradução	38
4.1.3 Animação e modelagem das mãos	40
4.2 Virtual Signing - ViSiCast	41
4.2.1 Tradução do inglês para o ESL	42
4.2.2 Animação e modelagem	43
4.3 VSigns	44

4.3.1	SignWriting	44
4.3.2	SWML	44
4.3.3	Processo de tradução	45
4.3.4	Modelagem e animação	45
4.4	Outros trabalhos	46
4.4.1	SIGNSIM	47
4.4.2	AGA-Sign	49
4.5	Considerações finais	50
5	ANIMANDO LIBRAS	52
5.1	Determinação do espaço disponível para a animação de um sinal .	53
5.2	Geração e animação de um sinal em LIBRAS	54
5.3	Implementação dos módulos	56
5.3.1	Modelador	56
5.3.2	Gerador de ações	56
5.3.3	Animador	58
5.4	Considerações finais	59
6	VALIDAÇÃO	60
7	CONSIDERAÇÕES FINAIS	63
7.1	Conclusões	63
7.2	Trabalhos futuros	64
	REFERÊNCIAS	65
	APÊNDICE A V-ART	69
A.1	A estrutura do V-ART	69
A.2	Articulações	70
A.3	Animações	71
	APÊNDICE B XML DO HUMANÓIDE	74
	APÊNDICE C XML DO GESTO OBRIGADO	88
	APÊNDICE D DTD DA CENA	96
	APÊNDICE E DTD DA ANIMAÇÃO	101
	APÊNDICE F O MÉTODO CCD	103
F.1	Definição	103
F.2	Determinação da função objetiva	103
	APÊNDICE G QUESTIONÁRIO APLICADO NOS TESTES	107

LISTA DE FIGURAS

Figura 2.1:	Passos para a animação de um gesto	13
Figura 2.2:	Classificação dos movimentos do ombro: (a) rotação externa, (b) rotação interna, (c) adução, (d) flexão, (e) extensão, (f) abdução	14
Figura 2.3:	Pronação(a) e supinação(b)	15
Figura 2.4:	Desvio ulnar(a) e radial(b)	15
Figura 2.5:	Inclinação lateral à esquerda	16
Figura 2.6:	Ossos dos dedos	16
Figura 2.7:	Juntas da parte superior do corpo	18
Figura 2.8:	Oposição do polegar - pode-se comparar ao gesto de "pinçar" algo utilizando os dedos	18
Figura 2.9:	Aspecto final do humanóide	19
Figura 2.10:	Pedaço da árvore de representação do esqueleto humano	19
Figura 2.11:	Representação gráfica dos interpoladores	20
Figura 2.12:	Gesto com uma cadeia cinemática	22
Figura 2.13:	Gesto com duas cadeias cinemáticas	23
Figura 2.14:	Determinação de novo alvo	23
Figura 2.15:	Comparação entre modelos com e sem interpenetração de malhas	26
Figura 2.16:	Gesto com interpenetração	26
Figura 2.17:	Esferas para determinação de colisão com partes do corpo	27
Figura 2.18:	Gesto da figura 2.16 sem interpenetração	28
Figura 3.1:	Algumas configurações de mão em LIBRAS	30
Figura 3.2:	Movimento nos gestos TRABALHAR (a) e EM-PÉ (b)	31
Figura 3.3:	Configuração final do sinal laranja (a) e do sinal aprender (b)	31
Figura 3.4:	Algumas orientações em LIBRAS: (a) para o corpo e (b) para baixo	32
Figura 3.5:	Formação do sinal do substantivo TELEFONE (a) através da mudança de movimento no sinal do verbo TELEFONAR (b)	34
Figura 3.6:	Sinais representando as letras A (esquerda), B (centro) e C (direita)	34
Figura 3.7:	Determinação de pontos com o referente presente	35
Figura 3.8:	Determinação de pontos com o referente ausente	36
Figura 4.1:	Dicionário de LIBRAS	38
Figura 4.2:	Primeira parte do tutorial de HST	39
Figura 4.3:	Segunda parte do tutorial de HST	39
Figura 4.4:	Representação gráfica do ESD	40
Figura 4.5:	Ossos e juntas da mão direita	41
Figura 4.6:	Representação do dedo	41
Figura 4.7:	Representação gráfica das mãos em HST	42

Figura 4.8:	Exemplo de aplicativo do ViSiCast	42
Figura 4.9:	Transcrição do sinal Going-to	43
Figura 4.10:	Representação da posição das mãos	45
Figura 4.11:	Representação de movimentos em SignWriting	45
Figura 4.12:	Representação de contato	46
Figura 4.13:	Resultado da aplicação da rotação nas juntas do avatar	46
Figura 4.14:	Módulo de escrita de sinais	47
Figura 4.15:	Módulo de escrita da língua portuguesa	48
Figura 4.16:	Animação do sinal COMEÇAR	49
Figura 4.17:	Editor 3D	49
Figura 4.18:	Visualizador 3D	49
Figura 4.19:	Arquitetura do AGA-Sign	50
Figura 5.1:	Arquitetura do sistema de animação	52
Figura 5.2:	Posição dos braços na determinação do espaço de enunciação	53
Figura 5.3:	Espaço de enunciação	54
Figura 5.4:	Configurações finais dos gestos do sinal BOM - Configuração intermediária (a) e configuração final (b)	54
Figura 5.5:	Juntas que compõem a cadeia e o <i>end effector</i> do braço direito	55
Figura 5.6:	Pose de descanso (a) e pose utilizada para o cálculo da cinemática inversa quando o espaço de enunciação é 10% maior que o padrão	55
Figura 5.7:	Trecho do XML que descreve o humanóide	56
Figura 5.8:	Interface do Action Generator	57
Figura 5.9:	Trecho do XML que descreve um dos gestos que compõem o sinal OBRIGADO	57
Figura 5.10:	Sistema de animação para Libras	58
Figura 5.11:	Interface do módulo animador (<i>Human Libras</i>)	59
Figura 5.12:	Pose de descanso	59
Figura 6.1:	Interface modificada do Animador de LIBRAS para os testes	60
Figura 6.2:	Gesto ACHAR em três configurações	61

RESUMO

Os surdos possuem a capacidade de utilizar a língua oral para se comunicar limitada e por isso tem como língua materna as línguas gestuais. Isso dificulta a utilização, de maneira satisfatória, dos serviços básicos, bem como a inserção na sociedade ouvinte, que é composta pela maioria da população. Devido ao fato desta língua ser gestual, é viável afirmar que se pode simular seus sinais através de animação de humanos virtuais, sem perder a percepção correta do significado do mesmo (que palavra o sinal representa).

O presente trabalho descreve uma técnica de animação aplicada em LIBRAS. A idéia principal é, baseado na descrição da animação de um determinado sinal, executar seu movimento de forma mais, ou menos ampla para que se consiga aproveitar o espaço disponível para a gesticulação, sem entretanto perder o significado do sinal.

A animação computacional de um sinal deve o mais próximo possível do real, ou seja, seu significado deve ser facilmente entendido e sua execução deve ser natural (suave e contínua). Para isso os sinais devem ser definidos de acordo com as limitações de movimentação das articulações humanas, bem como ao campo de visão do receptor. Além disso alguns parâmetros devem ser analisados e definidos: velocidade do movimento, tempo e amplitude dos sinais. Outro aspecto importante a ser tratado é o espaço que é disponível para a execução do sinal: dependendo do espaço, o sinal deve ser animado de forma a se adequar a ele.

A implementação da técnica resultou em um sistema de animação para LIBRAS composto por três módulos:

- um modelador do humano virtual, de forma que as articulações e DOFs deste sejam anatomicamente coerentes com a realidade;
- um gerador de gestos, o qual é responsável pela transformação dos parâmetros como velocidade, tempo de execução do gesto, configuração das juntas, em um arquivo que descreve a animação da pose. Cabe ressaltar que as palavras em LIBRAS são conhecidas como sinais. Já um sinal é composto por um ou vários gestos e estes são compostos por poses;
- um animador, o qual é responsável por gerar a animação de um sinal previamente criado, adequando (se necessário) a amplitude deste sinal ao espaço disponível para a execução do mesmo.

O sistema criado foi submetido a testes para que a técnica fosse validada. O que se buscou com os testes foi verificar se os sinais gerados eram passíveis de entendimento, ou seja, se a animação gerada representava determinada palavra.

Todos os aspectos acima mencionados são apresentados e analisados em detalhes.

Palavras-chave: Animação de Humanos Virtuais, Língua de sinais.

Virtual Human Animation Applied in Brazilian Sign Language

ABSTRACT

Deaf people have a limited capacity of using oral language to communicate. Because of this, they use gestural languages as their native language. This makes it especially difficult for them to make use of basic services in a satisfactory way and to properly integrate the hearing world, to which the majority of the population belongs. Due to the fact that this language is only gestural, it is possible to say that the signs it comprises of can be simulated with the animation of virtual humans without losing the correct perception of their inherent meanings (what words they represent).

This work describes a technique of animation for LIBRAS. The main idea is to take the movement of a sign from a description of its animation and execute it in a more or less wide manner in order to better use the available space for gesticulation without losing the meaning.

The computer animation of a sign must be as close to the real gesture as possible. Its meaning must be easily understood and its execution must be natural (smooth and continuous). For that, the signs must be defined in accordance with the movement limitations imposed by the human joints, and the field of view of the receiver. Besides that, some relevant parameters must be analyzed and defined: speed of the movement, time and amplitude of the signs. Another important aspect to be addressed is the space that is available for the execution of the sign: depending on the area, the sign must be animated in a manner that makes it properly fit in it. The implementation of the technique resulted in a animation system for LIBRAS, that consists of three modules:

- a virtual human modeler, so that the joints and DOFs are anatomically consistent with reality;
- a gesture generator, which is responsible for the processing of parameters such as speed, time of execution of the gesture, joint configuration, in a file that describes the animation of the pose. It is worth emphasizing that the words in LIBRAS are known as signs. Already a sign is composed of one or more gestures and they are composed of poses;
- an animator, which is responsible for generating the animation of a previously created sign, fitting (if necessary) the sign amplitude to the space available for its animation.

The generated system has been submitted for tests in order to validate the technique. The goal of the tests was to check whether the generated signs were understandable - if the generated animation represented a certain word.

All aspects above are presented and analyzed in detail.

Keywords: Virtual Human Animation, Signed Language.

1 INTRODUÇÃO

O tipo de comunicação mais comum entre pessoas é a língua falada. Através dela há a interação entre indivíduos, a transmissão de pensamentos e transferência de conhecimento.

Devido a certas limitações, o surdo não consegue adquirir o conhecimento de uma língua oral, tendo assim dificuldades em atuar plenamente na sociedade, encontrando barreiras ao utilizar serviços básicos, tais como correios, bancos, escolas, etc. Assim não estão inseridos completamente na sociedade ouvinte. Devido a essa limitação no aprendizado de uma língua oral, os surdos tem como língua materna as gestuais, que possuem uma estrutura gramatical própria e são compostas pelos níveis lingüísticos fonológico, morfológico, sintático e semântico.

Em cada país essa língua é diferente sintaticamente. Tanto a LIBRAS (Língua BRAsileira de Sinais), quanto a Língua Americana de Sinais (ASL) e a Língua de Sinais Britânica, por exemplo, são mutuamente incompreensíveis. Basicamente o que diferencia as línguas de sinais das outras é a sua modalidade visual-espacial: enquanto nas línguas orais é utilizada a fala para emitir as idéias e a audição para o seu recebimento, nas gestuais há a utilização de gestos corporais para a emissão e da visão para o recebimento de idéias. O sinal analisado pelos olhos corresponde diretamente aos movimentos de articulações, enquanto ouvidos analisam apenas o efeito do som produzido. Por isso toda a construção estrutural nas línguas gestuais é dependente das limitações do espaço de visão.

Para que haja comunicação entre surdos e ouvintes, costuma-se utilizar tradutores de LIBRAS, seja pessoalmente, ou através de vídeos. Porém, esse tipo de interação tem alguns problemas, tais como o baixo número de intérpretes existentes no Brasil, fazendo com que muitas vezes não sejam encontrados no momento necessário, além do custo de gravação e armazenagem de vídeo.

É fato que a animação de movimentos humanos através de humanóides está cada vez mais presente no cotidiano: em filmes, jogos e telejornais, sendo aliada importante em diversas áreas, como na psicologia (tratamento de fobias) (EMMELKAMP, P. et al, 2002) e nos esportes (reprodução dos movimentos dos atletas) (BIDEAU, B. et al, 2004).

Por causa do uso de gestos na formação da língua de sinais, a animação computacional pode ser usada para auxiliar os surdos em tarefas do dia a dia, como pagamento de contas, visita a museus e até mesmo na alfabetização, fazendo assim com que os mesmos tenham uma inclusão cada vez maior na sociedade ouvinte que o circunda.

Uma animação voltada para língua de sinais deve produzir sinais o mais próximos possível do mundo real. O significado do gesto gerado como resultado deve ser facilmente entendido pelo usuário e sua execução deve ser suave e contínua. A técnica usada para animar o humanóide deve estar baseada nas limitações morfológicas das articulações, bem como no espaço disponível para a execução dos gestos.

Os parâmetros que definem um gesto são: a configuração das mãos (forma como estão posicionados os dedos e as mãos), o ponto de articulação (área no corpo ou no espaço de enunciação definido pelo corpo, em que ou perto do qual o sinal é articulado (KLIMA; BELLUGI, 1975)) e o movimento, além das componentes não manuais (expressões faciais). Muitas vezes uma pequena mudança em apenas um desses parâmetros modifica totalmente o significado da palavra, o que ressalta sua importância.

O objetivo deste trabalho é apresentar uma técnica de animação computacional, através de humanos virtuais, adaptada para a execução de sinais em LIBRAS. Para tal é necessário:

- Avaliar quais parâmetros são importantes para se ter uma animação passível de entendimento;
- Adequar os gestos ao espaço disponível para sua execução.

Acredita-se que esse tipo de trabalho poderá ajudar uma parcela da população brasileira que por não possuir como língua materna a língua falada, é forçada a se adaptar a uma língua a qual tem muita dificuldade de assimilação e utilização.

O restante do texto está organizado da seguinte forma:

- o capítulo 2 expõe a idéia principal deste trabalho, mostrando o que deve ser feito para se gerar animações de gestos, levando-se em consideração um espaço dinâmico para execução dos mesmos.
- o capítulo 3 apresenta um estudo sobre LIBRAS e sua estruturação fonética, sintática e semântica;
- o capítulo 4 apresenta os principais trabalhos recentes desenvolvidos nesta área, apontando as direções que outros trabalhos tem seguido;
- o capítulo 5 trata da aplicação da técnica apresentada no capítulo 2 para geração de sinais em LIBRAS, além de mostrar uma implementação da mesma;
- o capítulo 6 mostra os resultados e testes feitos para a validação da técnica;
- o capítulo 7 conclui o trabalho, apresentando trabalhos futuros a serem desenvolvidos.

2 ANIMAÇÃO DE HUMANOS VIRTUAIS

Fazer animações que se baseiam em gestos humanos é uma atividade complicada, haja visto que o corpo humano é uma estrutura complexa, geralmente representado por um objeto articulado em um programa de computador. Este objeto deve possuir várias juntas, com vários graus de liberdade (DOF), onde cada DOF normalmente representa um eixo de rotação.

Uma ação pode ser descrita com relação ao comportamento das juntas ao longo do tempo, ou seja, as posições e orientações de seus DOFs. Para tal deve-se determinar a duração do sinal, as juntas que irão se mover, bem como de que forma elas irão se mover (qual será a configuração final de seus DOFs em um determinado instante de tempo e com qual velocidade eles devem convergir para esta configuração). Ao se descrever uma ação através das amplitudes das juntas, velocidade, etc., deve-se tomar o cuidado de não gerar movimentos que o tornem pouco realista.

Uma mesma ação pode ser executada em amplitudes diferentes, sendo estas determinadas pelo espaço físico existente para sua realização (parque, elevador lotado, etc.). Para tal não é necessário uma remodelagem da mesma e sim uma adequação ao novo espaço determinado.

Para que uma ação seja animada em um espaço de realização dinâmico, alguns passos devem ser seguidos, conforme figura 2.1.

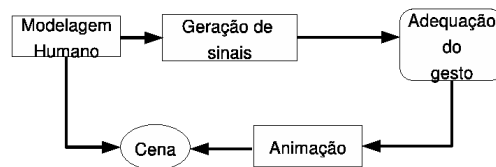


Figura 2.1: Passos para a animação de um gesto

2.1 Modelagem do humano virtual

Modelar um humano virtual com todas as articulações existentes no corpo humano é uma tarefa complicada, haja visto que existem algumas partes do esqueleto complexas, tais como o ombro que é formado por um conjunto de articulações, as quais formam uma estrutura não hierárquica, conforme Thalmann (MAUREL; THALMANN, 2000) e a do

cotovelo, que envolve deslizamento de um osso sobre outro em algumas movimentações.

Para se ter uma animação o mais próxima possível da realidade, as juntas e seus DOFs devem ser desenvolvidos de forma a manter a coerência entre nossos movimentos e os movimentos do humanóide (não se pode determinar por exemplo que a junta representante do joelho possa se dobrar para a frente do corpo). Assim é necessário um estudo sobre a fisiologia humana.

2.1.1 Fisiologia e biomecânica das mãos e membros superiores do corpo

Todos os movimentos do corpo ocorrem em torno de um eixo e de um plano e são classificados em:

- Flexão: movimento de dobra de um osso sobre o outro causando uma diminuição do ângulo da articulação. Como exemplo tem-se a flexão do ombro na figura 2.2(d);
- Extensão: movimento antagônico à flexão. Como exemplo tem-se a extensão do ombro na figura 2.2(e);
- Abdução: movimento para longe da linha média do corpo. Como exemplo tem-se a abdução do ombro na figura 2.2(f);
- Adução: movimento de aproximação da linha média do corpo. Como exemplo tem-se a adução do ombro na figura 2.2(c);
- Rotação: movimento de um osso ou parte dele em torno de seu eixo longitudinal. Como exemplo tem-se as rotações interna e externa do ombro na figura 2.2(a) e (b);

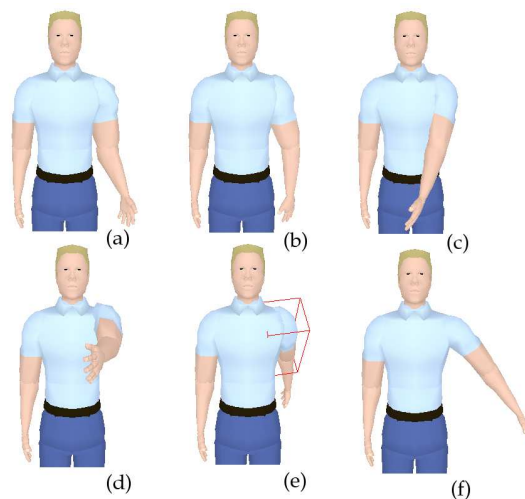


Figura 2.2: Classificação dos movimentos do ombro: (a) rotação externa, (b) rotação interna, (c) adução, (d) flexão, (e) extensão, (f) abdução

- Pronação: Rotação interna do antebraço. Na figura 2.3 (a) pode ser vista a pronação do antebraço esquerdo;
- Supinação: movimento contrário à pronação. Pode ser vista na figura 2.3 (b), no antebraço esquerdo;

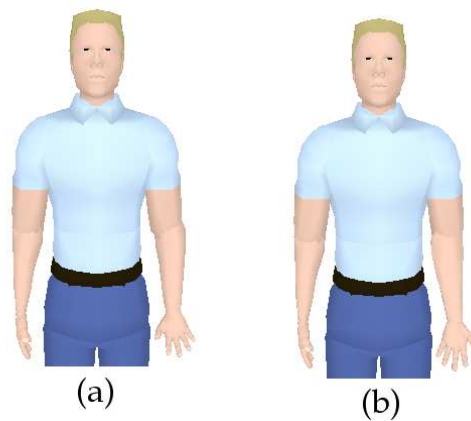


Figura 2.3: Pronação(a) e supinação(b)

- Desvio ulnar: adução do punho. O desvio ulnar do pulso esquerdo pode ser visto na figura 2.4(a);
- Desvio radial: abdução do punho, visto no pulso esquerdo, na figura 2.4(b);

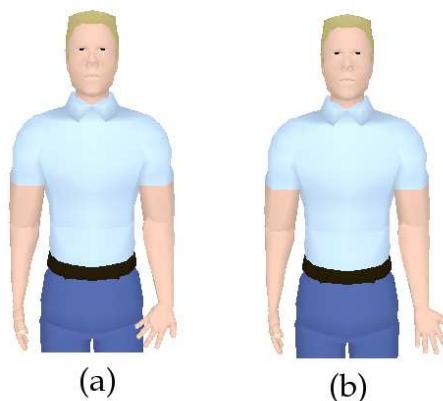


Figura 2.4: Desvio ulnar(a) e radial(b)

- Inclinação lateral: quando o tronco se move para a direita ou para a esquerda, como pode ser visto na figura 2.5.

O ombro corresponde à junção de duas clavículas, duas escápulas e o esterno. Já a mão é composta de 27 ossos e os dedos são compostos por 3 ossos (falanges proximal, média e distal), conforme figura 2.6. O polegar merece um estudo a parte, pois não possui a falange média e possui oposição aos outros dedos.

O sistema de articulação do cotovelo é composto por três ossos (úmero, ulna e rádio) e três articulações, fazendo com que o mesmo possua movimentos de flexão, supinação e pronação. A coluna vertebral é composta de 33 vértebras, das quais 24 se unem para formar uma coluna flexível. Elas são classificadas como cervicais, torácicas, lombares, sacrais e coccígeas.



Figura 2.5: Inclinação lateral à esquerda



Figura 2.6: Ossos dos dedos

As duas primeiras vértebras cervicais são estruturas altamente especializadas dedicadas à sustentação do crânio. A primeira, denominada atlas, tem duas grandes articulações que permitem a flexão e extensão da cabeça. A segunda vértebra, denominada áxis, é responsável por permitir o giro do atlas, tornando possível girar ou balançar a cabeça de um lado a outro. No resto da região cervical, na região torácica e na lombar as articulações permitem flexão, extensão e rotação.

As tabelas 2.1 e 2.2 mostram as amplitudes das articulações, em graus.

2.1.2 O modelo desenvolvido

Para que os sinais se aproximem ao máximo do realismo, as articulações devem possuir as amplitudes anteriormente especificadas na subseção 2.1.1, baseadas em (KENDALL; MCCREARY; PROVANCE, 1995) e (HOPPENFELD, 2004).

O modelo desenvolvido para essa aplicação possui 46 juntas divididas em uniaxiais (um grau de liberdade), biaxiais (dois graus de liberdade) e poliaxiais (3 graus de liberdade), imitando assim as juntas humanas (joelhos como juntas uniaxiais, cotovelos como biaxiais e ombros como poliaxiais, por exemplo) e 46 malhas triangulares, formando as partes do corpo (cabeça, braços, etc.). Os DOFs sofrem apenas rotações em torno de seus eixos, não havendo assim DOFs translacionais.

A figura 2.7 mostra as juntas da região superior do corpo do humanóide. Os dedos

Tabela 2.1: Amplitude de movimentos (medidas em graus)

Amplitude	ombro	cotovelo	punho	dedos			polegar	
				proximal	média	distal	proximal	média
Abdução	180	–	–	20	–	–	–	–
Adução	45	–	–	0	–	–	–	–
Flexão	180	135	80	90	100	90	50	90
Extensão	45	0	70	35	0	20	0	20
Rotação interna	55	–	–	–	–	–	–	–
Rotação externa	45	–	–	–	–	–	–	–
Supinação	–	90	–	–	–	–	–	–
Pronação	–	90	–	–	–	–	–	–
Desvio Ulnar	–	–	30	–	–	–	–	–
Desvio Radial	–	–	20	–	–	–	–	–

Tabela 2.2: Amplitude de movimentos (medidas em graus) da coluna

Amplitude	coluna				
	primeira vértebra	segunda vértebra	demais vértebras	tórax	lombar
Rotação	–	180	8 a 120	2 a 10	3
Flexão	90	–	5 a 100	7 a 10	12 a 20
Extensão	85	–	–	10	12 a 20

possuem juntas uniaxiais nas falanginhas e falangetas. Já nas falanges há juntas biaxiais.

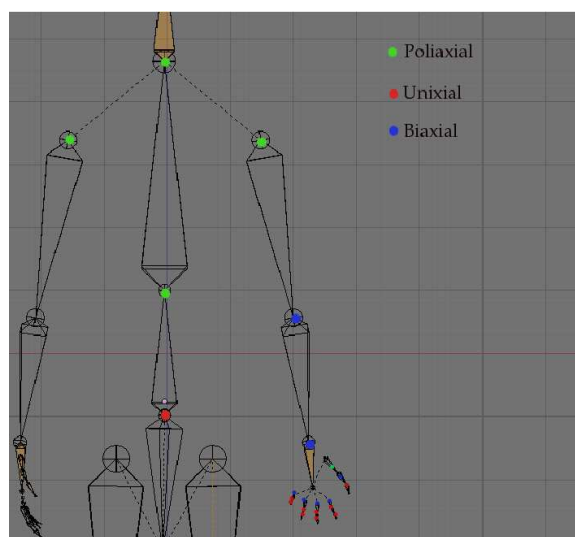


Figura 2.7: Juntas da parte superior do corpo

No caso do polegar, para que o mesmo pudesse fazer o gesto de "pinçar" (visto na figura 2.8) foi adicionada uma junta além das duas existentes. Essa junta é poliaxial e faz o papel da junta que existe na ligação entre o pulso e o polegar.

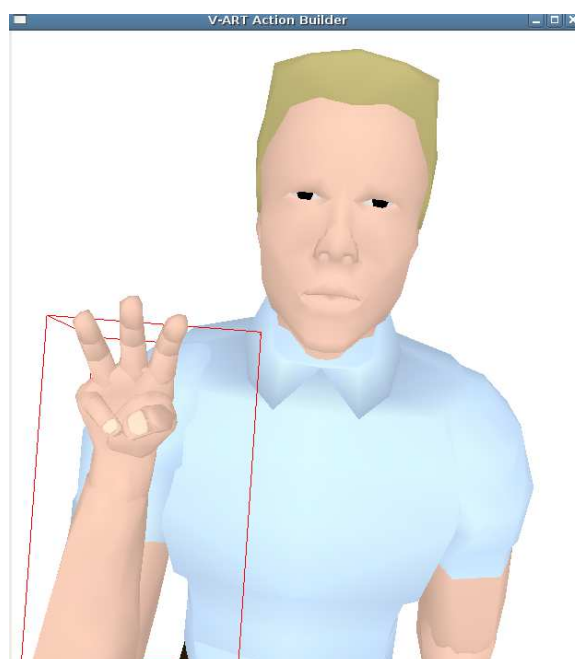


Figura 2.8: Oposição do polegar - pode-se comparar ao gesto de "pinçar" algo utilizando os dedos

Os ossos do modelo devem também seguir o padrão humano, com proporções entre os membros próximas da realidade de um esqueleto humano. Por exemplo, a maioria das pessoas possui braços com tamanhos muito próximos aos antebraços. Assim não é desejável modelar um humanoide que tenha o antebraço medindo a metade do braço.

Uma imagem do modelo pode ser vista na figura 2.9.

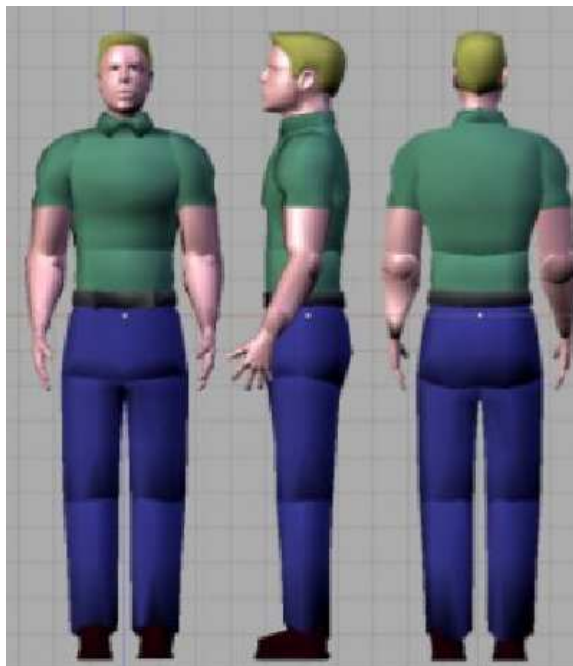


Figura 2.9: Aspecto final do humanóide

A representação computacional do humanóide pode ser vista como uma árvore aonde os nós são as juntas, com atributos definindo seu tipo (DOFs), seguidas por objetos gráficos (malhas triangulares no caso). Na figura 2.10 tem-se uma parte da árvore que representa uma parte do modelo de um esqueleto humano onde tem-se um nó raiz sendo uniaxial (apenas um DOF) tendo como nós filhos uma malha (objeto gráfico) e uma nova junta poliaxial (três DOFs) e uma malha como nó filho.

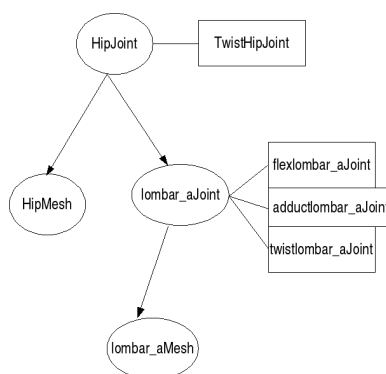


Figura 2.10: Pedaco da árvore de representação do esqueleto humano

2.2 Geração das ações

Ações são usualmente criadas através de captura de movimento, ou através de aplicativos especialmente desenvolvidos para gerar ações a partir de um modelo definido.

A maior desvantagem da captura de movimentos é a calibragem dos dispositivos de captura para que se obtenham movimentos aceitáveis para a composição da ação. Outra desvantagem é na determinação da quantidade de sensores e possíveis localizações para

se capturar todos os movimentos necessários. Já a vantagem é uma maior aproximação do realismo, haja visto que todos os movimentos do corpo são capturados.

Os aplicativos definidores de ações tem a vantagem de não necessitar de dispositivos de captura, não tendo assim o problema da calibração. Porém geralmente o tempo gasto para a obtenção de uma ação realista é maior.

Foi desenvolvido um aplicativo, para a obtenção de ações denominado *Action Generator*, cujos detalhes podem ser vistos na seção 5.3.2. Nele são especificados os parâmetros necessários para a animação: o nome da ação (no caso o gesto) a ser definida, a velocidade de execução do mesmo, se o gesto se repete ou não (se é cíclico), sua duração, o tipo de interpolação entre as configurações inicial e final, além da posição final, do tempo inicial da movimentação e do tempo final para cada DOF que irá se mover.

Alguns parâmetros mencionados anteriormente merecem um olhar mais minucioso. A duração de um gesto é determinada em segundos e indica o tempo de execução do mesmo. Os tempos inicial e final, de cada DOF a se mover, são normalizados (valores de zero a um) e indicam, baseados na duração total do gesto, quando a configuração deste DOF será modificada. Suponha que um gesto tenha duração de 4 segundos e um determinado DOF possui tempo inicial e final 0.5 e 1, respectivamente. Isto significa que após 2 segundos do início do gesto o DOF em questão começa a se mover e esta mudança será executada até o fim da animação (4 segundos).

O que se tem determinado são posições (ou configurações) finais dos DOFs que irão se mover no gesto. Na criação do modelo tem-se os limites dos DOFs determinados em graus. Estes limites são normalizados (0 para o mínimo e 1 para o máximo) e a posição final de um DOF no gesto será um valor entre 0 e 1, inclusive.

Uma ação não pode passar de uma configuração qualquer para a final repentinamente. Essa mudança deve ser suave, sem quebra de continuidade. Assim ela deve ser regida por uma função do tipo $C_{DOF} = f(t)$, ou seja, a configuração do DOF deve ser uma função dependente do tempo.

Dentro das funções de interpolação vale a pena destacar duas: a linear e a *ease_in_ease_out*. A linear é definida por uma função linear do tipo $X(t) = n + Yt$ e produz resultados retilíneos, ou seja, o coeficiente de modificação do DOF para intervalos de tempos iguais é o mesmo. Já a *ease_in_ease_out* é definida por uma função senoidal $X(t) = \text{sen}(t) + n$ e fornece resultados não retilíneos. Nesta, a sensação que se tem é que há uma aceleração e desaceleração do movimento em determinados momentos. Uma representação gráfica das duas pode ser visto na figura 2.11.

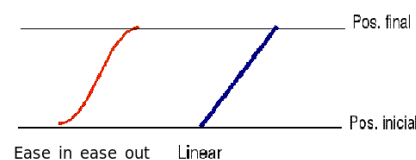


Figura 2.11: Representação gráfica dos interpoladores

Outra coisa a ser tratada na animação é como combinar duas ou mais ações de forma harmoniosa. Isso é feito da seguinte forma: quando uma movimentação atinge a config-

uração desejada ela é desativada e se tiver alguma outra animação a ser realizada, ela é executada imediatamente, sem quebra de continuidade.

2.3 Adequação de uma ação ao espaço disponível

Uma ação pode ser rearranjada, dependendo do espaço disponível para sua execução. Para isso deve-se fazer uma análise das novas posições das juntas para que toda a ação esteja contida no novo espaço. Este problema pode ser pensado da seguinte forma: dadas a posição e orientação desejadas de um determinado DOF (posição e orientação alvo), como as juntas vão se mover para que essa configuração dada desse DOF seja alcançada, sendo respeitadas as restrições dos limites das juntas e do espaço.

Existem algumas abordagens para resolver este problema, entre elas o uso de roadmaps (UGHINI; NEDEL, 2007) e cinemática inversa. A técnica escolhida para a resolução desse problema foi a cinemática inversa, pois a determinação do caminho exato a ser seguido para a execução do gesto é algo mais complexo e custoso que a abordagem escolhida.

Formalizando o conceito, cinemática inversa é o processo de determinação do comportamento das juntas de uma cadeia (conjunto de juntas) para que uma determinada posição e/ou orientação sejam alcançadas.

Antes da apresentação do método utilizado, alguns conceitos devem ser mostrados:

- *end effector*: posição e orientação da última junta de uma cadeia. Geralmente é ele que deve se aproximar, ou encontrar o alvo;
- *goal* ou alvo: objetivo a ser alcançado. Geralmente é uma posição ou orientação no espaço;
- cadeia: conjunto de juntas, ligadas por ossos, as quais irão se mover para que o *end effector* alcance o alvo.

São vários os métodos existentes para o cálculo da cinemática inversa, dentre eles o CCD (*Cyclic-coordinate descent*) que é uma abordagem iterativa baseada em heurística, que procura minimizar os erros de posição e orientação, variando uma junta de cada vez. Em cada iteração há o percorrido da cadeia cinemática, da junta a ser movida até seu último descendente. Cada junta é movida de forma a minimizar uma determinada função objetiva. Esta minimização de cada junta é simples o bastante para que uma solução analítica seja formulada, fazendo com que cada iteração seja rápida. Após a definição do novo estado da junta, essa modificação é repassada para as outras juntas posteriores e para o *end-effector*. Isto acontece até que o alvo (posição final desejada para o *end-effector*) seja encontrado. Maiores informações sobre este método podem ser vistas no anexo A.3.

A técnica de cinemática inversa adotada tem a desvantagem de gerar movimentos nem sempre naturais, devido ao fato de ser um método iterativo (a junta se move um pouco de cada vez, podendo gerar movimentos não desejados, como os de ir (para encontrar o alvo) e vir (se tiver passado pelo alvo)).

Para que isso não aconteça o passo da cinemática inversa é feito em *background*, ou seja, a animação não é mostrada. Após a determinação das novas configurações das juntas esses valores são então incorporados na descrição do gesto padrão através da inclusão de uma variável que indique o quando deve ser acrescentado ou retirado da posição para que o movimento saia de acordo com o padrão. Assim temos que a nova posição será determinada pela equação $p_f = p_d + \delta$ onde $\delta = q_i - p_d$, onde p_f é a posição final do

DOF, p_d é a posição inicial do DOF na descrição do gesto), δ é a variável de acréscimo ou decréscimo da posição, e q_i é a posição do DOF encontrada na cinemática inversa.

2.3.1 Determinação da cadeia cinemática

Cadeia cinemática é o conjunto de juntas que farão parte do movimento de encontro ao alvo. No caso de LIBRAS essas juntas são as correspondentes ao pulso, cotovelo e ombro, haja visto que tanto a configuração dos dedos como das outras partes superiores do corpo permanecem conforme as do gesto padrão.

O movimento pode possuir uma ou duas cadeias cinemáticas (braço esquerdo e/ou braço direito) conforme as juntas que se movem na ação. Por exemplo uma ação que representa o sinal BOM em LIBRAS (figura 2.12) possui apenas uma cadeia cinemática, já a que representa DIA (figura 2.13) possui duas cadeias. Como a parte esquerda do modelo é apenas um espelhamento da parte direita, por motivos de otimização se o lado direito se movimentar conforme o esquerdo, há a determinação dos valores das juntas de uma cadeia e estes mesmos valores são repassados às juntas correspondentes à outra cadeia.

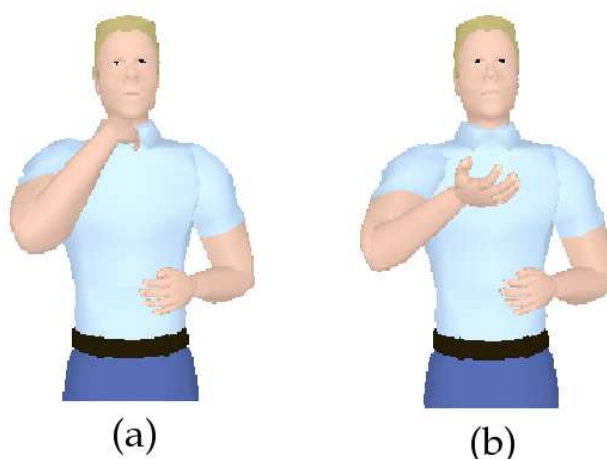


Figura 2.12: Gesto com uma cadeia cinemática

2.3.2 Definição do alvo

Através do gesto já criado é possível a determinação do alvo. Isso é feito através da ferramenta de geração de ações da seguinte forma. Quando uma pose é determinada, calcula-se a posição e orientação finais do *end-effector*. Esses valores são guardados e utilizados posteriormente da seguinte forma: os valores dos alvos são lidos e armazenados em tempo de execução. Ao se escolher a palavra a ser animada o valor correspondente do *goal* para o mesmo é então utilizado.

O alvo pode ficar fora do espaço disponível, com a diminuição deste. Quando isto ocorre o alvo deve ser recalculado. Este cálculo é feito da seguinte forma: move-se o braço nas dimensões que estão fora dos limites da esfera até que o ponto se encontre dentro da mesma. A figura 2.14 mostra que com a diminuição do espaço parte do braço do humanóide fica do lado de fora do mesmo. O alvo é então recalculado, fazendo com que toda a região que estava fora do espaço anteriormente fique do lado de dentro.



Figura 2.13: Gestos com duas cadeias cinemáticas

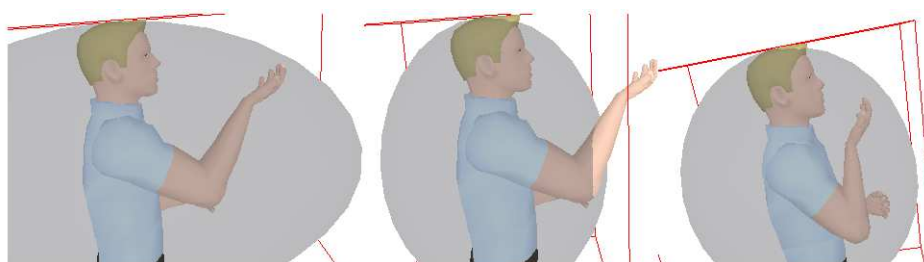


Figura 2.14: Determinação de novo alvo

2.3.3 O algoritmo

Determinados então a cadeia, o alvo e o *end-effector*, estes são usados para a determinação da configuração final das juntas da cadeia. Os algoritmos 1 e 2 mostram a técnica de adequação do gesto ao espaço de enunciação.

Algumas vezes o *end effector* não consegue convergir para o alvo, porém consegue alcançar uma configuração muito próxima à desejada, que não acarreta prejuízos ao entendimento do gesto. Para que aconteça isso no algoritmo de cinemática inversa, a cada iteração é verificado se a distância entre o alvo e o *end effector* são menores que esta tolerância.

2.3.4 Colisão das juntas com a fronteira do espaço disponível

A cada movimentação de DOF, deve-se testar também se as juntas permanecem dentro do espaço determinado. Se não, este DOF deve se mover até que todas as juntas estejam contidas no espaço. Para isso é utilizado detecção de colisão entre as juntas da cadeia e a forma geométrica que descreve o espaço disponível.

É preciso verificar se o ponto que descreve a posição da junta em questão está dentro do espaço geométrico. Isto é feito através da substituição, na equação que define a geometria do espaço, do ponto que representa a junta. Através do resultado obtido sabe-se se a junta está ou não dentro do espaço definido.

Algorithm 1 SolverCCD (lista de DOFs da cadeia, posição e orientação do end effector, posição e orientação do alvo, n. iterações, bounding box, conjunto de esferas, mapa de distâncias, ação, tolerância, tolerância máxima da orientação, tolerância mínima da orientação)

```

phi, phiAux:real;
vetorFinal: vetor com as configurações finais dos DOFs;
it: inteiro;
for (cada DOF da cadeia cinemática) do
  phi = CalculaPhi (posição e orientação do end effector, posição e orientação do alvo);
  {O método CalculaPhi é visto no algoritmo 2}
  phiAux = phi/k; {k no caso vale 100}
  {Colisão é a função que verifica se as juntas da cadeia estão contidas no espaço
  disponível (vide subseção 2.3.4). Interpenetração é a função que verifica se houve
  colisão entre as partes do corpo (vide subseção 2.3.5).}
  while (phiAux <= phi e Not Colidiu(lista de DOFs da cadeia, bounding box) e Not
  Interpenetração(vetores de esferas, mapa de distâncias)) do
    Mover o DOF até a configuração phiAux;
  end while
  posFinal do DOF = Posição atual do DOF+phiAux;
  if (it<=n. de iterações) then
    aplicam-se as transformações da cadeia ao end effector e aos filhos da junta.
    lengAux = end effector atual-posição do alvo;
    tOr= produto escalar entre a orientação do end effector e a do alvo;
    if (lengAux.Length() > tolerancia ou (tolerância máxima da orien-
    tação<tOr<tolerância mínima da orientação)) then
      SolverCCDAux(lista de DOFs da cadeia sem o acaba de ser usado, posição e
      orientação do end effector, posição e orientação do alvo, n. iterações,bounding
      box, conjunto de esferas, mapa de distâncias, ação, tolerância, tolerância máx-
      ima da orientação, tolerância mínima da orientação))
      it = it+1;
    else
      for (todos os DOFS da cadeia) do
        mover o DOF para a posição anterior
        modificar o valor do erro na ação
      end for
    end if
  end if
end for

```

Algorithm 2 CalculaPhi(posição e orientação do end effector, posição e orientação do alvo)

```

phi, phi1, phi2, phi3:real; {phi é rotação calculada, phi1, phi2 e phi3 são candidatos ao
valor da rotação}
alpha, ro, wp, wo = real;
pic, pid, ujc, ujd: vetores;
{Os significados das variáveis acima podem ser vistos no anexo A.3}
it: inteiro;
{***** Cálculos vistos no anexo A.3 *****}
pic = PIC(posição do DOF, posição do end effector);
pid = PID(posição do DOF, posição do alvo);
ujc = UJC(orientação do DOF, orientação do end effector);
ujd = PID(orientação do DOF, orientação do alvo);
alpha = Alpha();
ro = Ro(pid, pic);
wo = Wo();
wp = Wp();
k1 = K1(wp, pid, pic, ujc, orientação do DOF);
k2 = K2(wp, pid, pic, ujc);
k3 = K1(wp, pid, pic, ujc, orientação do DOF);
valor = k3/(k2-k1);
phi = 500; {Valor hipotético. Sabe-se que phi nunca será esse valor.}
phi1 = atan(valor); phi2 = phi1+3.14159; phi3 = phi1-3.14159;
if ((k1-k2)*cos(phi1)-k3*sin(phi1) < 0.0) then
    {a função acima é a segunda derivada da função objetiva apresentada no anexo A.3}
    phi = phi1;
end if
if (phi2 > -3.14159 e phi2 < 3.14159) then
    if ((k1-k2)*cos(phi2)-k3*sin(phi2) < 0.0) then
        if (phi ≠ 500) then
            phi = min(phi, phi2);
        else
            phi = phi2;
        end if
    end if
end if
if (phi3 > -3.14159 e phi3 < 3.14159) then
    if ((k1-k2)*cos(phi3)-k3*sin(phi3) < 0.0) then
        if (phi ≠ 500) then
            phi = min(phi, phi3);
        else
            phi = phi3;
        end if
    end if
end if

```

2.3.5 Tratamento de colisão entre as partes do corpo

Para que não haja interpenetração de duas partes do corpo é necessário tratar a colisão entre elas. Várias são as técnicas existentes para resolver este problema, dentre elas detecção de colisão por *bounding boxes*, detecção de colisão entre malhas, etc.

O modelo do humanóide foi construído, sem levar em conta a existência de deformação de malha. Dessa forma, se as partes do corpo não tiverem se sobrepondo na sua ligação e se essa sobreposição não for arredondada, têm-se problemas ao se movimentar as juntas, podendo aparecer falhas que fazem com que o modelo não seja realista.

A figura 2.15 ilustra uma situação como a descrita anteriormente. Percebe-se que no primeiro modelo (figura 2.15 (a)) há uma sobreposição suave do pescoço com o tronco, fazendo com que a movimentação da cabeça para o lado retorne um resultado realista. Já o segundo modelo (figura 2.15 (b)) mostra a mesma situação com um modelo aonde não há sobreposição de malhas. Nota-se que o resultado da movimentação faz com que o pescoço se separe do tronco.

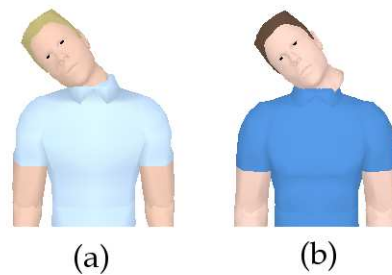


Figura 2.15: Comparação entre modelos com e sem interpenetração de malhas

Devido à sobreposição de partes do corpo, há sempre interpenetrações e consequentemente colisões que devem ser desconsideradas. É necessário então desenvolver uma maneira de informar quais colisões são importantes para que partes do corpo não entrem em outras, conforme o que acontece na figura 2.16.



Figura 2.16: Gesto com interpenetração

Para otimizar os cálculos de detecção de colisão, foram adicionadas esferas dentro

do tronco, braços e antebraços (vide figura 2.17). Estas esferas são colocadas apenas nessas regiões devido ao fato da animação dos sinais ser realizada pela parte superior do corpo. Como a configuração das mãos é determinada na descrição do gesto e não muda, independente do espaço de enunciação existente, é verdadeira a afirmação de que não há interpenetração entre a mão e outra parte do corpo quando o gesto é executado. Isto faz com que não haja necessidade de existirem esferas nas mãos e dedos.

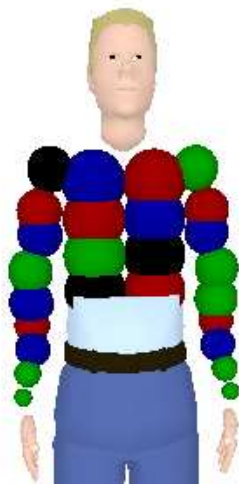


Figura 2.17: Esferas para determinação de colisão com partes do corpo

Foram criadas 26 esferas no total, sendo 8 no tronco, 5 em cada antebraço e 4 em cada braço. As esferas possuem tamanhos variados, de maneira que consigam estar totalmente contidas dentro do volume que representa a parte do corpo. Além disso foi criada uma grade de tamanho 26×26 , aonde cada célula da mesma recebe como valor a distância entre os centros de duas esferas distintas. Se o valor for zero significa que a colisão entre as mesmas deve ser descartada.

Assim, no momento do cálculo da cinemática inversa há também a verificação se a distância das esferas que estão inseridas na cadeia cinemática é menor que a distância guardada na grade anterior (quando o valor é diferente de zero). Se sim houve interpenetração e o DOF que se moveu sofre uma regressão até que a colisão deixe de existir. Com esta verificação, o resultado do movimento visto na figura 2.16 fica conforme a figura 2.18.

As interpenetrações a serem analisadas são 6: entre os dois braços, entre o braço de um lado e o antebraço do outro lado e entre braços e antebraços com o tronco.

2.4 Considerações finais

Foi apresentada uma técnica de animação que leva em consideração a dinamicidade do espaço disponível para a execução do movimento. Para isso foram desenvolvidas a elaboração de um modelo com os limites de movimentação das juntas anatomicamente corretos, a geração geométrica do espaço, a criação dos gestos e sua adequação ao espaço existente.

A idéia principal da técnica é, através da descrição (obtida na modelagem) de uma ação, animá-la com amplitudes diferentes. Para tal propósito foi usada a técnica de cinemática inversa conhecida por CCD, com verificação de colisão entre os DOFs e o volume



Figura 2.18: Gesto da figura 2.16 sem interpenetração

geométrico que representa o espaço disponível, além da verificação de interpenetração entre as partes superiores do corpo.

Algumas vezes o *end effector* não consegue convergir para o alvo, porém consegue alcançar uma configuração muito próxima à desejada, o que não acarreta prejuízos ao entendimento do gesto. Para que aconteça isso basta, no algoritmo de cinemática inversa, verificar a cada iteração se a distância entre o alvo e o *end effector* são menores que uma tolerância admitida.

3 LÍNGUAS DE SINAIS

Conceituando formalmente língua de sinais, pode-se dizer que este tipo de linguagem se refere a estruturas linguísticas utilizadas por surdos na expressão e elaboração do pensamento e na comunicação (PONTES; ORTH, 1999).

Essas línguas são formadas pela associação de movimentos das partes do corpo acima do quadril e se diferem das línguas faladas devido à maneira como são produzidas e recebidas. Isto não ocorre através do canal oral-auditivo, mas através do canal gestual-visual.

Segundo Karnopp (KARNOPP, 2000) enquanto o emissor constrói uma sentença a partir de vários elementos (configurações de mãos, braços, ombro, expressão facial, etc.), o receptor utiliza a visão ao invés da audição para entender o que está sendo emitido. Por causa disso, os sinais são construídos baseando-se nos limites articulares do corpo, bem como nos limites do sistema visual humano.

Um dos grandes diferenciadores entre as línguas de sinais e as orais é a seqüencialidade temporal, sendo as línguas orais lineares (fonemas se sucedem seqüencialmente) e as de sinais paralelas (os sinais podem ser emitidos utilizando-se diversas partes do corpo concorrentemente).

As línguas de sinais são tão complexas e expressivas quanto as línguas orais e também são definidas através de sistemas fonológico (configurações e movimentos dos elementos que estão envolvidos nos sinais), morfológico (formação de sinais), sintático e semântico (MARCATO; ROCHA; LIMA, 1990). Nelas os elementos comunicativos são chamados palavras e os léxicos sinais. Todos esses sistemas serão vistos com mais detalhes neste capítulo.

3.1 Fonologia das línguas de sinais

A fonologia estuda os fonemas de uma língua. Entende-se por fonemas unidades mínimas que estabelecem distinção entre os vocábulos de uma língua. Os objetivos desta área são a determinação das unidades que formam os sinais e dos possíveis padrões de combinações entre essas unidades tanto quanto as variações permitidas e possíveis.

Nas línguas de sinais são usados os membros superiores para a produção do sinal. Um sinal é produzido através de expressões manuais e não-manuais. Tem-se por expressões manuais todo o movimento produzido pelas mãos, braços e antebraços, e não-manuais movimento da face, dos olhos, da cabeça e do tronco.

3.1.1 Fonemas nas línguas de sinais

De acordo com Ferreira Brito (BRITO, 1974), citado por Macarto (MARCATO; ROCHA; LIMA, 1990), os três parâmetros primários que se combinam nas línguas de sinais são: a configuração das mãos, o ponto de articulação e o movimento.

Configuração das mãos é como estão posicionados os dedos e as mãos. Em LIBRAS tem-se 46 configurações diferentes de mão. Algumas delas podem ser vistas na figura 3.1.



Figura 3.1: Algumas configurações de mão em LIBRAS

Espaço de articulação (ou espaço de enunciação) é o local no espaço que contém todos os pontos dentro do raio de alcance das mãos em que os sinais são articulados (QUADROS; KARNOPP, 2004). Este espaço é dinâmico, podendo mudar de tamanho ou orientação conforme haja necessidade (obstáculos físicos que impedem a realização do gesto padrão, grau de conhecimento de LIBRAS por parte do receptor, estado emocional do emissor, etc.).

Entende-se por movimento a variação de posição espacial de um objeto no decorrer do tempo, ou seja, movimento é a execução de um sinal. Para que ele exista é necessário existir um objeto (sinal a ser realizado) e um espaço de enunciação. De acordo com Ferreira Brito (BRITO, 1974) o movimento em LIBRAS se localiza nas mãos, pulsos e antebraço, podendo ter repetições no mesmo e também características como tensão e velocidade. Os sinais podem ter ou não movimento. O sinal EM-PÉ não têm movimento (repousa-se uma mão na configuração adequada, sobre a outra conforme figura 3.2 (b)), diferente do sinal TRABALHAR (tem-se o movimento para frente e para trás de ambos os braços, conforme figura 3.2 (a)), por exemplo.

O ponto de articulação, ou locação, é o local no espaço de enunciação em que ou perto do qual o sinal é produzido (KLIMA; BELLUGI, 1975). Existem palavras que diferem entre si apenas pela diferença do ponto de articulação, como laranja (figura 3.3 (a)) e aprender (figura 3.3 (b)).

As locações são divididas em quatro regiões principais, estas subdivididas em regiões menores: cabeça (topo, testa, rosto, parte superior do rosto, parte inferior do rosto, orelha, olhos, nariz, boca, bochechas e queixo), mão (palma, costas das mãos, lado do indicador, lado do dedo mínimo e dedos), tronco (pescoço, ombro, busto, estômago, cintura, braços, antebraço, cotovelo e pulso) e espaço neutro.

Orientação é a direção para a qual a palma da mão aponta na produção do sinal. Ferreira Brito (BRITO, 1974) enumera seis tipos de orientação da palma da mão, algumas

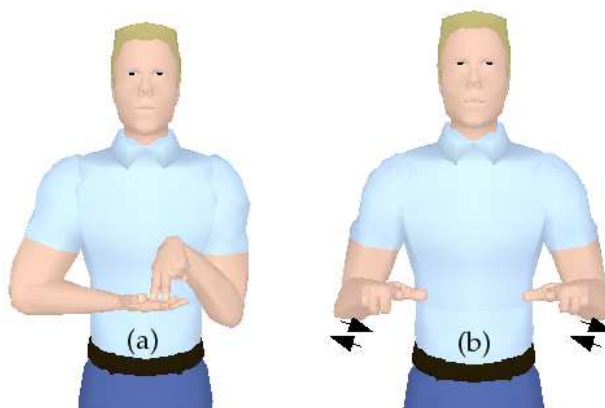


Figura 3.2: Movimento nos gestos TRABALHAR (a) e EM-PÉ (b)

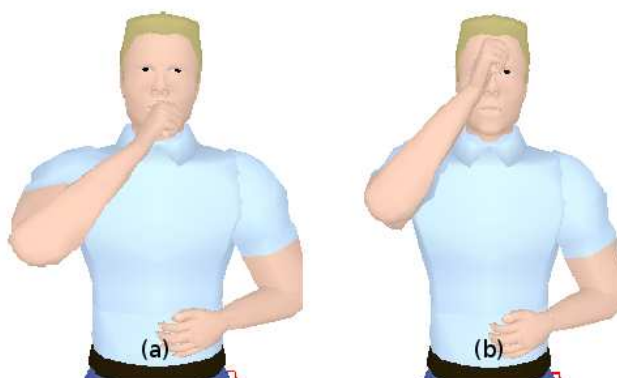


Figura 3.3: Configuração final do sinal laranja (a) e do sinal aprender (b)

das quais podem ser vistas na figura 3.4: para cima, para baixo, para o corpo, para a frente, para esquerda e para a direita.

3.1.2 Aspectos não-manuais das línguas de sinais

As expressões não-manuais desempenham dois papéis nas línguas de sinais: marcação de construções sintáticas e diferenciação dos itens léxicos. Estas expressões determinam, por exemplo, a negação e a interrogação das sentenças. Elas possuem um papel muito importante no entendimento das sentenças, podendo ser agrupadas em movimento de face, movimento de olhos, cabeça ou tronco, conforme tabela 3.1 (BRITO, 1974).

3.1.3 Restrições na formação de sinais

Algumas restrições na formação dos sinais são impostas pelo sistema visual e pela fisiologia das mãos. A área da face é a região onde a acuidade visual é maior, onde o interlocutor fixa o olhar. Assim, é mais fácil determinar nela pequenas diferenças das unidades na formação dos sinais.

Na classificação proposta por Battiston (BATTISON, 2000), há duas restrições na

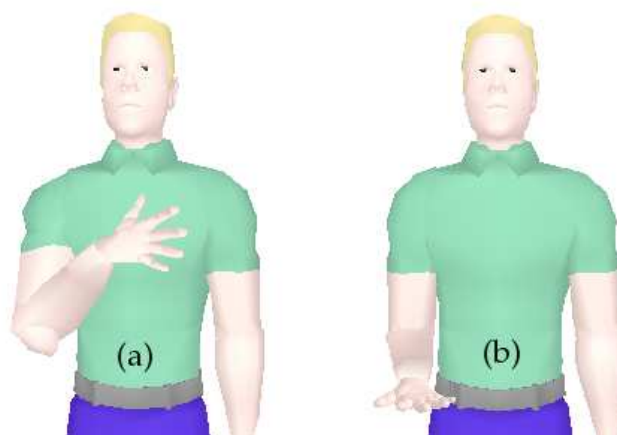


Figura 3.4: Algumas orientações em LIBRAS: (a) para o corpo e (b) para baixo

produção de sinais envolvendo as duas mãos: a condição de simetria e a condição de dominância.

1. Condição de simetria: caso as duas mãos se movam na produção do sinal, a configuração das mesmas deve ser igual, a locação deve ser a mesma ou simétrica e o movimento deve ser simultâneo ou alternado.
2. Condição de dominância: se as mãos apresentam configurações distintas, então a mão ativa produz o movimento e a passiva serve de apoio.

As restrições na formação de sinais restringem a complexidade dos sinais para que eles sejam mais facilmente executados e entendidos. O resultado disso é uma previsibilidade maior na formação dos sinais.

Hulst (HULST, 1989) apresentou a idéia de dependência entre as mãos na formação dos sinais. De acordo com o mesmo, a mão não-dominante é representada como dependente da dominante. Assim, em sinais feitos com as duas mãos, a mão não-dominante é uma cópia da dominante, ou funciona como ponto de articulação (ponto de referência para a execução do sinal).

3.2 Morfologia das línguas de sinais

Morfologia é o estudo da estrutura dos sinais (léxico), de como os sinais são formados. Um exemplo disso é a formação nominal, que é feita através da mudança de movimento no sinal do verbo. No caso da figura 3.5 observa-se a representação do sinal TELEFONE (figura 3.5 (a)) e TELEFONAR (figura 3.5 (b)). A locação, configuração e orientação da mão são iguais tanto para o verbo quanto para o nome derivado, porém no substantivo (TELEFONE) tem-se o movimento representado por flechas, indicando um “ir e voltar” do cotovelo. Já no verbo (TELEFONAR) tem-se por representação do movimento uma curva direcional, indicando que o movimento será curvilíneo e unidirecional.

A estrutura dos sinais de LIBRAS é bastante complexa e apresenta alguns aspectos que não são encontrados nas línguas orais, tais como a forma como é adicionada ao sinal a negação e a flexão, entre outros.

Algumas palavras em LIBRAS são gesticuladas através de soletração manual, que é uma representação manual da ortografia do português, envolvendo uma sequência de

Tabela 3.1: Expressões não-manuais em LIBRAS

<i>rosto</i>
sobrancelhas franzidas olhos arregalados sobrancelhas levantadas bochechas infladas bochechas contraídas lábios contraídos e projetados contração do lábio superior franzir do nariz
<i>cabeça</i>
balanceamento para frente e para trás balanceamento para os lados inclinação para frente inclinação para o lado inclinação para trás
<i>rosto e cabeça</i>
cabeça projetada para a frente, olhos levemente cerrados, sobrancelhas franzidas cabeça projetada para trás e olhos arregalados
<i>tronco</i>
para frente para trás balanceamento alternado dos ombros balanceamento simultâneo dos ombros balanceamento de um único ombro

configurações de mão que tem correspondência direta com a sequência de letras do português escrito. As palavras estrangeiras são incorporadas através desta soletração via um sistema que representa a ortografia dessa língua estrangeira ao invés do português. Os sinais correspondentes às letras A, B e C do alfabeto podem ser vistas na figura 3.6.

Existem algumas palavras que utilizam parcialmente este alfabeto, como por exemplo a palavra azul, que utiliza a soletração do a e do l, com um movimento entre elas.

Um aspecto específico das línguas gestuais é o sistema combinatório de palavras para a geração de novas. Para as línguas orais essa combinação é muitas vezes feita através da adição de um sufixo ou um prefixo à uma raiz. Nas línguas de sinais essas formas são resultantes de processos não concatenativos em que uma raiz é enriquecida com vários movimentos e contornos no espaço de sinalização.

A respeito da negação, há três formas desta ser incorporada em LIBRAS: através do movimento de negação com a cabeça juntamente com o sinal a ser negado, através do movimento de negação do dedo indicador acompanhado do sinal a ser negado ou através de expressão facial.

A flexão em LIBRAS é feitas através de:

- Pessoa: mudam as referências pessoais no verbo. Isso é feito através de apontamento propriamente dito (apontar com o dedo). Os referentes são introduzidos no espaço à frente do sinalizador, através do apontamento em diferentes locais (QUADROS; KARNOPP, 2004);

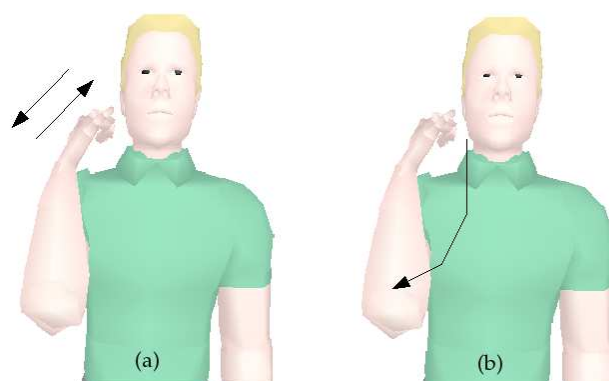


Figura 3.5: Formação do sinal do substantivo TELEFONE (a) através da mudança de movimento no sinal do verbo TELEFONAR (b)



Figura 3.6: Sinais representando as letras A (esquerda), B (centro) e C (direita)

- Grau: apresenta distinções para menor, maior, mais próximo, muito, entre outros. Alguns desses possuem sinal próprio, tal como grande e pequeno;
- Gênero: em LIBRAS há uma ausência de marcação de gênero. O sinal referente a homem ou mulher é executado após o sinal a ser flexionado. Existem algumas palavras que possuem sinais diferentes quanto à flexão, como é o caso das palavras pai e mãe;
- Foco temporal: indica temporalidade. O tempo em LIBRAS é representado por aspectos espaciais no qual passado é indicado com um movimento para trás da mão e futuro com movimento de mão para frente. Palavras como ontem, amanhã e hoje possuem sinais próprios em LIBRAS;

3.3 Sintaxe das línguas de sinais

Entende-se por sintaxe espacial o estudo dos elementos sintáticos (sujeito, verbo e objeto) posicionados no espaço de enunciação. As referências usadas no discurso podem ser (QUADROS; KARNOPP, 2004):

- Fazer o sinal em um local particular do espaço de enunciação;
- Direcionar a cabeça e os olhos em direção a uma localização particular simultaneamente com o sinal de um substantivo ou com o apontamento para o mesmo;

- Usar um apontamento ostensivo (pronome) antes do sinal de um referente específico;
- Usar uma apontamento ostensivo em uma localização particular quando a referência for óbvia;
- Usar um verbo com concordância incorporando os referentes previamente introduzidos no espaço.

Em LIBRAS, quem está executando o sinal indica a quem se refere através de localizações no espaço, sendo que estes referentes podem estar ausentes ou não. Quando presentes, os pontos de referência são determinados por suas posições físicas. Quando ausentes, são estabelecidos pontos abstratos no espaço.

A figura 3.7 ilustra a determinação da localização de um referente presente (no caso o receptor). O emissor utiliza o indicador apontado para si para se referir a ele mesmo e o indicador apontado para o receptor para referenciá-lo. No caso em questão, o receptor está na frente do emissor. No caso da figura 3.8, o emissor aponta para um ponto no espaço, indicando alguém citado na conversa e que não está presente. Isso é feito através da sinalização através do indicador a um ponto que não seja a referência de alguém que esteja presente na conversação.

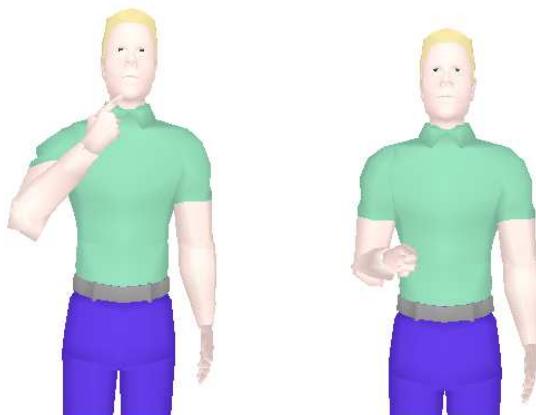


Figura 3.7: Determinação de pontos com o referente presente

As frases em LIBRAS podem apresentar várias ordens. Ferreira-Brito (BRITO, 1974) afirma parecer existir uma ordenação mais básica que as demais: a ordem Sujeito-Verbo-Objeto (SVO). Algumas características com relação à ordem são importantes, tais como:

1. Todas as frases com a ordem SVO são gramaticais.
2. As ordens OSV e SOV ocorrem somente quando há alguma coisa a mais na sentença, com marcas não-manuais.
3. Apesar de ocorrerem construções SOV e OSV associadas a marcas não-manuais, se houver uma estrutura complexa na posição de objeto, não será possível mudá-lo de ordem. Por exemplo, a ordem SOV não pode existir quando o objeto é uma oração subordinada, tal como “Eu quero que Maria trabalhe mais”.



Figura 3.8: Determinação de pontos com o referente ausente

4. Os advérbios temporais e de frequência não podem interromper uma relação entre o verbo e o objeto. Por exemplo, não vai existir a forma “JOÃO COMPRAR ONTEM CARRO” para a frase “Ontem João comprou um carro”.

3.4 Considerações finais

Assim como as línguas faladas, as línguas gestuais possuem uma formação própria, com aspectos morfológicos, sintáticos e semânticos.

Todos os sinais são formados através da combinação de configurações dos membros superiores do corpo (mãos, dedos, braços, tronco, cabeça e expressões faciais) e de movimentos específicos (por exemplo mover o braço para frente e para trás). Além disso para que um sinal seja formado da maneira correta é necessário considerar outros fatores tais como o ponto de articulação e o espaço de enunciação, além de tensão e velocidade na execução do mesmo.

As frases em LIBRAS não possuem a mesma estrutura sintática das frases em português (não há, por exemplo, a existência de artigo na frase e os verbos são sempre no infinitivo, etc.), o que torna o mapeamento de uma língua para a outra algo complexo.

4 TRABALHOS RELACIONADOS

Quando se fala em desenvolvimento de software para pessoas com necessidades especiais, deve-se levar tais necessidades em consideração, o que faz com que ferramentas de auxílio a essas pessoas sejam escassas, haja visto que os softwares mais comuns são feitos com foco em pessoas que não possuem tais necessidades.

Com relação aos surdos, o que se espera é a possibilidade de uma independência comparável à do ouvinte ao usar serviços básicos (correios, bancos, etc.), não precisando portanto contar com a boa vontade de um ouvinte. Isto pode ser possível através do uso de novas tecnologias, como por exemplo, a animação de humanos virtuais servindo como um intérprete entre eles e pessoas ouvintes.

Dentro dessa vertente já existem alguns sistemas baseados em animações de línguas de sinais, sendo um dos pioneiros o sistema de diálogo para sintetizar e compreender a língua gestual Japonesa (KAMATA, K. et al, 1989). Através de palavras e frases pre-definidas, seus gestos são capturados e animados quando preciso. Após este, outros sistemas de animação foram criados, tais como HST (*Hand Sign Translator*), VSigns, ViSiCast, entre outros, todos apresentados adiante.

O uso de vídeos para o ensino de línguas gestuais também é difundido, mas tem algumas limitações como, por exemplo:

- O ponto de vista fixo da câmera, fazendo com que um gesto não possa ser visto de ângulos diferentes, o que dificulta o aprendizado;
- O consumo maior de memória se comparado às animações 3D, já que os dados que descrevem os polígonos do humanoíde 3D são guardados apenas uma vez para qualquer animação e para que os gestos sejam animados é necessário apenas o conhecimento de alguns parâmetros (amplitudes, velocidade, tipo de interpolação) que são guardados em arquivos de tamanho menor que os arquivos de vídeos.

Dentro dessa linha de aplicativos que fazem uso de vídeos há o Dicionário de LIBRAS (FENEIS, 2000) visto na figura 4.1, o qual tem suas desvantagens, tais como o pequeno tamanho da janela para o vídeo, fazendo com que alguns gestos sejam de difícil entendimento, interface não muito clara, etc.

4.1 HST - Hand Sign Translator

O sistema HST (*Hand Sign Translator*) (HOLDEN; ROY, 1992) é um tradutor da língua inglesa oral para a língua de sinais chamada *signed english*, a qual foi criada por educadores australianos. Diferente da língua de sinais australiana (Auslan), ela é uma representação manual do inglês através do uso de sinais e soletramento de palavras através



Figura 4.1: Dicionário de LIBRAS

dos dedos. Os sinais usados em *signed english* sempre representam uma letra ou uma palavra e a tradução do inglês é feita por mapeamento direto, pois a sintática e a semântica não se alteram de uma para outra língua.

O HST foi criado para ser uma ferramenta educacional e funciona da seguinte forma: o usuário fornece à aplicação uma entrada textual em inglês e recebe como saída uma animação das mãos em *signed english* representando o texto inicial. A maior desvantagem do HST é a utilização de uma língua gestual que não é a oficial do país, além de utilizar apenas as mãos para a gesticulação das palavras. A animação se torna não natural à medida que não há modificação na velocidade de execução dos quadros da animação em um determinado gesto.

4.1.1 A interface do HST

O HST é dividido em duas partes. Uma delas é composta por um intérprete, onde é esperado que o sistema traduza uma sentença em inglês digitada pelo usuário através do teclado e faça animação desta sentença através das mãos virtuais, conforme visto na figura 4.2 (HOLDEN; ROY, 1992). Na segunda parte uma animação é exibida ao usuário, juntamente com uma lista de frases ou palavras simples escritas em inglês e o mesmo deve assinalar a que considera ser a tradução do gesto para o inglês (figura 4.3 (HOLDEN; ROY, 1992)).

4.1.2 O processo de tradução

Para que as animações em *signed english* sejam geradas é preciso traduzir a sentença do inglês para o *signed english* e transferir os gestos para a mão virtual. A tradução do texto em inglês para língua de sinais é feita através da animação sequencial dos signos correspondentes. As representações gráficas dos sinais são armazenadas em um dicionário, como um banco de dados, chamado *English Sign Dictionary* (ESD).

O HST analisa a entrada textual em ordem para achar o sinal correspondente no ESD, o qual guarda esses sinais em conjuntos de dados referentes às imagens com as duas mãos. Esses dados são então transformados em movimentos.

Uma imagem de duas mãos é representada por um conjunto de dados (POSE) que consiste nos ângulos das juntas dos dedos, posição do pulso e melhor ângulo de visão.

Instrumentos como um sensor de captura de movimento podem ser introduzidos nas

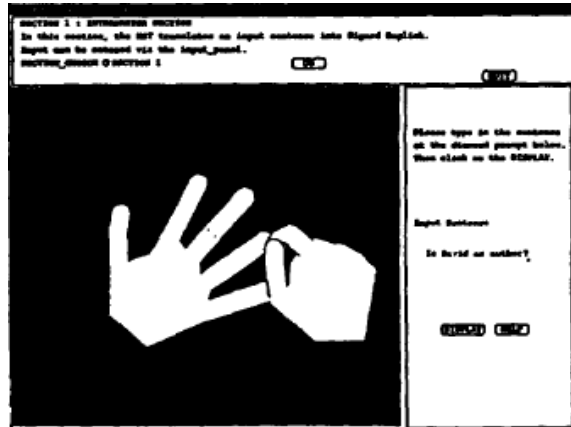


Figura 4.2: Primeira parte do tutorial de HST

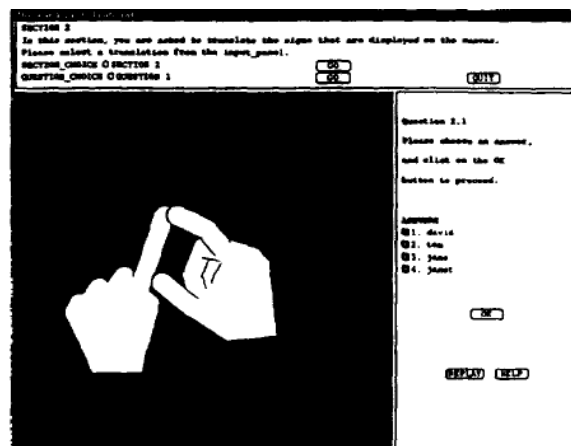


Figura 4.3: Segunda parte do tutorial de HST

juntas de uma pessoa em movimento e alimentar os ângulos das juntas diretamente em um computador (LIU; ZHUANG; PAN, 1999). Porém, esses instrumentos não são capazes de derivar precisamente a descrição cinemática durante a captura, e a especificação manual é difícil e muitas vezes ineficiente. Desta forma, como um método alternativo, fora criado um editor gráfico interativo (ESD editor) onde as mãos virtuais são facilmente manipuladas graficamente e os aspectos cinemáticos necessários para a criação do POSE são calculados.

O ESD armazena letras e palavras com seus sinais correspondentes. Um sinal pode ser uma imagem estática de duas mãos ou uma série de imagens que representam o movimento. Por exemplo, a palavra *achieve* tem seu sinal guardado em ESD com o uso de três imagens estáticas que representam quadros-chave do movimento.

Para facilitar o acesso aos dados do ESD, um arquivo de índices é usado. Ele contém as unidades de sinais existentes no ESD, com ponteiros que indicam onde os POSES correspondentes podem ser encontrados. A figura 4.4 (HOLDEN; ROY, 1992) ilustra melhor isso.

Com relação à análise do texto, pode-se dizer que o mesmo, ao ser percorrido, é dividido em uma lista de palavras separadas por espaços ou pontuações. Cada palavra é então analisada segundo as regras abaixo:

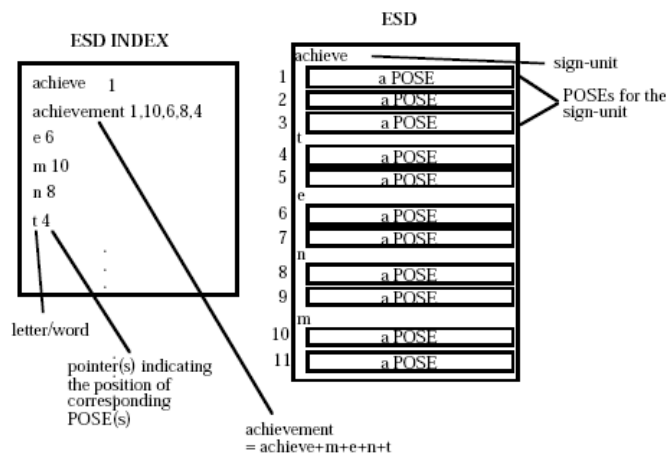


Figura 4.4: Representação gráfica do ESD

- Sinais devem ser usados apenas para a palavra designada;
- Se um sinal não existir para uma palavra, a palavra é quebrada em letras;
- Prefixos e sufixos devem ser soletrados e precedidos ou sucedidos pelo sinal da palavra.

Encontrar uma palavra composta em um texto é complicado e pode requerer análise de semântica. Para corrigir este problema, as palavras de sinais combinados são guardadas e armazenadas no ESD.

4.1.3 Animação e modelagem das mãos

A animação das mãos envolve três passos:

1. Os dados dos POSEs são usados para calcular a posição 3D no espaço cartesiano das duas mãos e o intervalo entre as poses é calculado;
2. Cada POSE é então transformada em formas de mão;
3. As imagens são renderizadas e animadas de forma suave.

As posições intermediárias dos gestos são calculadas pela interpolação linear dos ângulos das juntas de duas posições. A diferença entre cada ângulo de uma mesma junta é dividida pelo número de intervalos a serem mostrados para obter o incremento do ângulo, que é usado para determinar as poses intermediárias. Esse tipo de abordagem acaba gerando gestos pouco naturais, haja visto que não há variação de velocidade durante toda a execução do gesto.

Cada mão é modelada como um conjunto de quinze ossos ligados por juntas. Na figura 4.5 tem-se as linhas representando os ossos e os círculos representando as juntas. A junta que representa o pulso tem dois graus de liberdade, sendo eles rotação no eixo X e rotação no eixo Z. As juntas que representam as falanges proximais dos dedos possuem dois graus de liberdade, a saber rotações em Z e em Y e as outras juntas possuem apenas um grau de liberdade, sendo ele rotação em X.

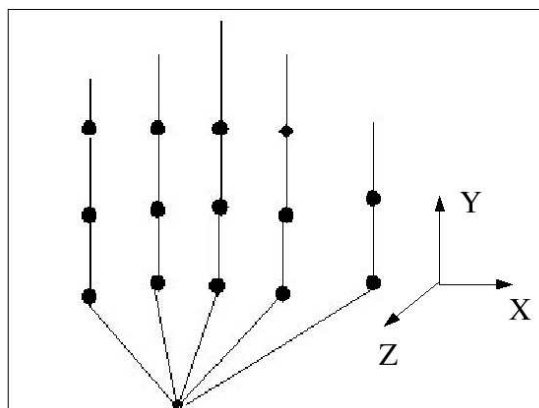


Figura 4.5: Ossos e juntas da mão direita

De acordo com Holden (HOLDEN; ROY, 1992), alguns sinais podem ser reconhecidos mais facilmente com mudanças na velocidade do movimento. Porém no HST a interpolação entre as posições guardadas no ESD é feita pela diferença entre dois ângulos das juntas. Essa diferença é então dividida pelo número de intervalos a serem mostrados, obtendo assim o incremento do ângulo. É possível determinar a velocidade com que os gestos devem ser executados, porém o aplicativo atualmente oferece uma única velocidade padrão para todos os movimentos.

A modelagem da mão é feita através de esferas e cilindros para representarem os dedos (ossos são cilindros e juntas são esferas) e a palma da mão é dividida em três malhas. Um esquema dessa modelagem pode ser visto nas figuras 4.6 e 4.7 (HOLDEN; ROY, 1992).

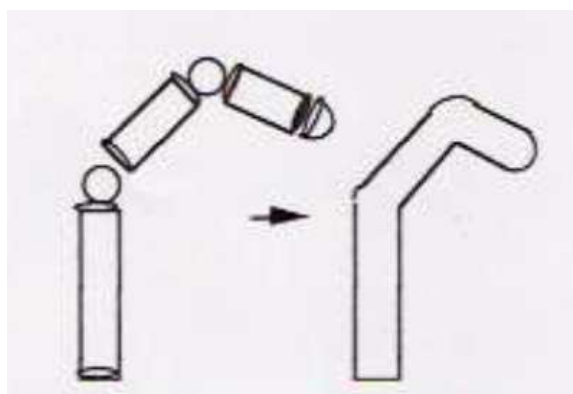


Figura 4.6: Representação do dedo

4.2 Virtual Signing - ViSiCast

ViSiCast é um projeto que procura criar meios de acesso às informações e serviços em língua de sinais (BANGHAM, J. A. et al, 2000). Para isso há o desenvolvimento de sistemas computacionais que utilizam humanos virtuais para animar gestos da ESL (European sign language), como por exemplo Simon, um humano virtual utilizado na tradução de textos impressos e capturas televisivas em língua gestual, visto na figura 4.8 (BANGHAM, J. A. et al, 2000).

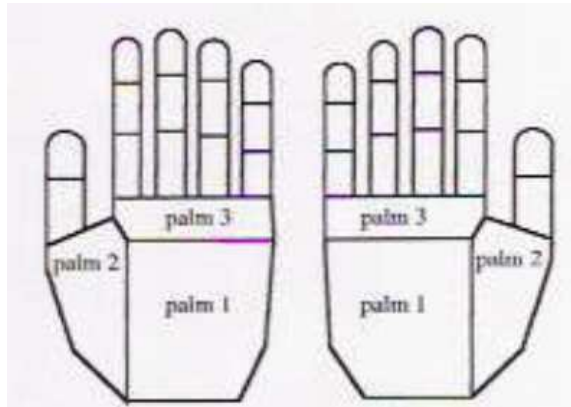


Figura 4.7: Representação gráfica das mãos em HST

Este projeto pode ser dividido em três áreas de concentração: aplicações para a Web, aplicações para transação “face a face”, como por exemplo, utilização de correio, hospitais, etc. e aplicações voltadas para a televisão. Por ser um projeto que engloba vários estágios da comunicação (televisão, “face a face”, etc.) é o mais completo atualmente, abrangendo a maior parte dos parâmetros a serem considerados, tal como a naturalidade do movimento (velocidade, aspectos não-manuais, etc.). Porém, não é levada em consideração a dinamicidade do espaço de enunciação, fazendo com que o humanoíde não possa ser colocado em qualquer cenário.



Figura 4.8: Exemplo de aplicativo do ViSiCast

Pode-se dividir em 4 os passos principais para se obter um resultado final nos aplicativos do ViSiCast: tradução do inglês para ESL, definição de uma GML (Gesture Mark-up Language), modelagem e animação do modelo gerado.

4.2.1 Tradução do inglês para o ESL

A conversão do inglês para ESL não pode ser feita através de mapeamento direto e tem suas peculiaridades, assim como qualquer tradução de linguagens naturais. Nas línguas gestuais os fonemas são relacionados com a forma das mãos, posição, orientação e movimento. A postura corporal e expressões faciais também definem um gesto e todos estes parâmetros devem ser analisados.

A arquitetura do sistema de tradução é definido por um pipeline de quatro estágios de tradução (MARSHALL; SAFAR, 2003):

- Parsing sintático do inglês;
- Geração da DRS (Discourse Representation Structure);
- Transferência semântica;
- Geração da descrição fonética em HamNoSys (SLEATOR; TEMPERLEY, 1991).

O *parsing* sintático é feito através do parser criado pela Universidade Carnegie Mellon (*CMU Linking Parser*). O resultado do *parsing* caracteriza as dependências sintáticas da frase e é guardado em um arquivo (SLEATOR; TEMPERLEY, 1991).

Para cada resultado é gerada também uma DRS (estrutura semântica intermediária da frase), através de regras gramaticais específicas, para capturar a semântica do texto. As DRSs permitem o isolamento de verbos, nomes, predicados baseados em adjetivos e relações temporais. Uma DRS do inglês é transformada em uma DRS da língua de sinais. Ela é então transformada em uma estrutura semântica, que é o ponto inicial para a geração do sinal.

Esta estrutura semântica é então transformada na notação HamNoSys (Hamburg Notation System), que é um XML (W3C, 2000) que descreve a ação física requerida para produzir o sinal. A GML (Gesture Mark-up Language) provê um framework geral para a representação e transmissão de informação sobre sequências de sinais. Ela deve incorporar informações sobre fonologia, fonética e articulação (BANGHAM, J. A. et al, 2000), bem como características da velocidade do movimento e outras constantes temporais. Em resumo, a GML será a ligação entre a língua de sinais e a animação. É ela que vai nortear a definição da notação HamNoSys.

Cada sinal é então quebrado em componentes como orientação da mão e movimento, descrevendo assim como o gesto deve ser animado. Um exemplo do que é gerado pela notação HamNoSys pode ser visto na figura 4.9 (KENNAWAY, 2001).

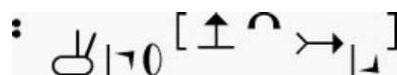


Figura 4.9: Transcrição do sinal Going-to

4.2.2 Animação e modelagem

ViSiCast tem como base dois trabalhos, o sign-anim (MARSHAL, I. et al, 1998) e Tessa (LINCOLN; COX; NAKISA, 1998), que são sistemas tradutores e animadores de texto para língua de sinais. Ambos utilizam captura de movimentos através do uso de sensores individuais para as mãos, corpo e face. Para as mãos utilizou-se cybergloves, para o pulso, braços e face, sensores magnéticos. Arquivos são então gerados para cada sinal e para cada frase (BANGHAM, J.A. et al, 2000).

Porém, essa captura possui alguns problemas, tais como a dificuldade de se calibrar os instrumentos e a gravação de uma grande quantidade de sinais, entre outros (KENNAWAY, 2001). Atualmente, com a evolução do hardware gráfico é possível gerar modelos baseados em Física sem que haja a necessidade de captura de movimento.

Alguns movimentos de sinais são representados em HamNoSys como uma sucessão de posturas estáticas. A animação dos mesmos deve ser feita considerando-se as sucessivas posturas. Com relação aos aspectos temporais, HamNoSys pode especificar o tempo de execução do sinal. Se este tempo não for especificado, um tempo padrão será utilizado.

Dada uma sequência de sinais descritas no HamNoSys e quantas vezes ela deve ser executada, pode-se gerar uma animação contínua dos sinais através da determinação dos ângulos das juntas de cada sinal a cada quadro de animação.

4.3 VSigs

VSigs é um aplicativo que gera sequências de animação a partir de uma entrada textual em SignWriting. Essas sequências são então usadas para animar um humanóide, reproduzindo assim uma tradução gestual para a entrada. Atualmente apenas os gestos das mãos estão implementados (PAPADOGIORGAKI, M. et al, 2004).

No geral a aplicação funciona da seguinte forma: é fornecida uma entrada em SignWriting e esta é convertida para SWML (Sign Writing Markup Language) (ROCHA COSTA; DIMURO, 2000), um formato XML desenvolvido para indexar e processar a notação SignWriting (GRIEVE-SMITH, 2000). Cada sinal é então convertido para uma sequência de animação corporal definida no MPEG-4 standard. Essas sequências servem para animar um humanóide modelado em VRML.

Por usar a escrita de sinais é um aplicativo muito útil aos surdos, porém gera gestos não naturais, haja visto a utilização de interpolação linear na execução dos gestos, além de gerar apenas as animações da mão de forma correta, comprometendo assim o significado do gesto. Ele também não faz uso do espaço de enunciação, além de não investigar a velocidade na execução dos gestos.

4.3.1 SignWriting

SignWriting (SUTTON, 1995) é um sistema de escrita criado em 1974 por Valerie Sutton para representar graficamente as línguas de sinais, sendo este universal, diferente das línguas gestuais. Fazendo-se uma analogia, pode-se ver que este sistema é como o alfabeto das línguas latinas: um só para todas.

SignWriting é definido por três estruturas, a saber:

- Posição de mão: indica qual a configuração da mão na execução do movimento. Ela pode estar aberta, fechada ou circular, conforme figura 4.10. Além disso, a mão pode estar paralela ou perpendicular ao chão.
- Movimentos: descreve como deve ser o movimento. Pode ser dividido em movimento de dedos e de mãos. Um exemplo pode ser visto na figura 4.11
- Contato: indica o contato dos elementos que compõem o sinal (mãos, cabeça, braços, etc.). Existem seis formas de representá-lo: tocar, pegar, bater, esfregar, escovar e contato entre dois elementos. Um exemplo pode ser visto na figura 4.12.

4.3.2 SWML

SWML (SignWriting Markup Language) é um formato baseado na notação XML com o propósito de representar os símbolos gráficos do SignWriting em um sistema computacional. Nesse arquivo há conjuntos de símbolos (sign-boxes) compostos por sequências







Representação no Sign Writing®	Configuração de mão
	
	
	

Figura 4.10: Representação da posição das mãos

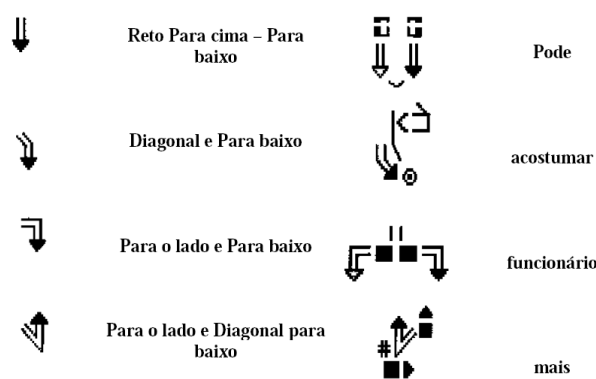


Figura 4.11: Representação de movimentos em SignWriting

de campos que contém a descrição do gesto (disposição dos dedos, configuração da mão, etc) em uma linguagem oral (PAPADOGIORGAKI, M. et al, 2004).

4.3.3 Processo de tradução

Como VSigns é feito para animar um avatar a partir da escrita de sinais, não há a necessidade de um processo de tradução como nos aplicativos anteriores, haja visto que a escrita é a representação gráfica dos gestos. No processo de tradução, portanto, é feito apenas o mapeamento dos sinais em um formato que possa ser usado para animar o avatar.

Uma lista de sinais disponíveis de acordo com o arquivo SWML é inicialmente produzida e toda a informação de cada símbolo é analisada e os símbolos convertidos para que haja a geração da animação.

4.3.4 Modelagem e animação

Nesta etapa os símbolos que estão no formato SWML são convertidos para uma sequência BAP (Body Animation Parameter). Os BAPs nada mais são que parâmetros do MPEG4 (ISO/IEC, 1997) que descrevem qualquer postura corporal. Eles denotam os ângulos de rotação das juntas (Dofs).

A conversão começa pela análise dos símbolos contidos em uma *sign box*. Se não houver símbolos descrevendo informações dinâmicas (por exemplo movimento das mãos), não haverá rotação das juntas, pois a cena deverá ser estática, sendo executada em apenas

Representação no <i>Sign Writing</i> ®	Contato	Exemplo
✱	tocar	
+	pegar	
✱	entre	

Figura 4.12: Representação de contato

um quadro.

Porém, se existe algum símbolo descrevendo informações dinâmicas haverá mais quadros, sendo que a taxa dos mesmos é especificado no arquivo BAP. O primeiro quadro da animação é gerado através da decodificação dos símbolos existentes, como na cena estática. Os quadros intermediários são gerados baseados nos campos de movimento existentes no arquivo SWML e interpolação linear é utilizada para unir os quadros e criar uma animação suave.

As sequências geradas são aplicadas em um avatar modelado através da linguagem de descrição VRML. Como os BAPs representam rotações de partes do corpo em torno de suas juntas, calcula-se as posições dessas partes após a aplicação da rotação dada e aplica-se esta transformação no humanóide. O resultado da aplicação dessas rotações pode ser visto na figura 4.13 (PAPADOGIORGAKI, M. et al, 2004).

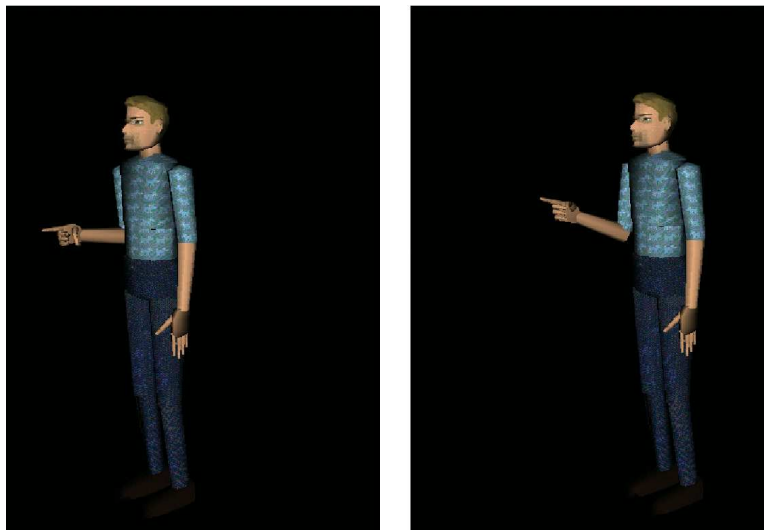


Figura 4.13: Resultado da aplicação da rotação nas juntas do avatar

4.4 Outros trabalhos

Existem outros trabalhos na literatura científica que utilizam animação de personagens para LIBRAS, sem ser este o foco principal dos mesmos. Desta forma, o resultado da animação deixa a desejar.

4.4.1 SIGNSIM

SIGNSIM (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) é uma ferramenta que serve de apoio à aprendizagem de LIBRAS. Ela é um sistema tradutor entre a escrita do português e uma das escritas de sinais chamada SignWriting.

Com relação à animação, as desvantagens desta ferramenta é não prover uma animação realista (haja visto que não é este o cerne do trabalho), tendo seu personagem como um conjunto de primitivas geométricas, não investigando assim características importantes no processo, tais como velocidade do movimento (a execução do gesto é linear), verificação do espaço de enunciação, etc.

Em resumo o sistema possui uma base de dados composta por palavras em português, seus respectivos sinais em SignWriting e as descrições de alguns quadros da animação da palavra correspondente.

Há dois módulos: o primeiro faz a tradução de SignWriting para o português. Isso é realizado através da busca na base de dados pela palavra que represente todas as características determinadas pelo usuário. Já estas características são determinadas através da escolha de alguns parâmetros de SignWriting vistos adiante. Se a palavra for encontrada, a animação associada a esta é então executada.

O segundo módulo faz o caminho contrário. Dada uma palavra digitada pelo usuário, é buscado na base de dados seu sinal e animação correspondentes. Se forem encontrados, o sinal equivalente é mostrado e a animação executada.

4.4.1.1 Módulo de escrita de sinais

Nesta etapa a palavra é escrita na língua de sinais. Para isso foi desenvolvido um editor da escrita de sinais, onde características como configuração de mãos, dedo, toque, etc. devem ser determinadas através da escolha dos mesmos na área de seleção de sinais. Na figura 4.14 (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) observa-se a interface deste módulo, bem como o editor de sinais. Nela vê-se o sinal correspondente à palavra casa, com sua tradução para o português, além da sua animação.

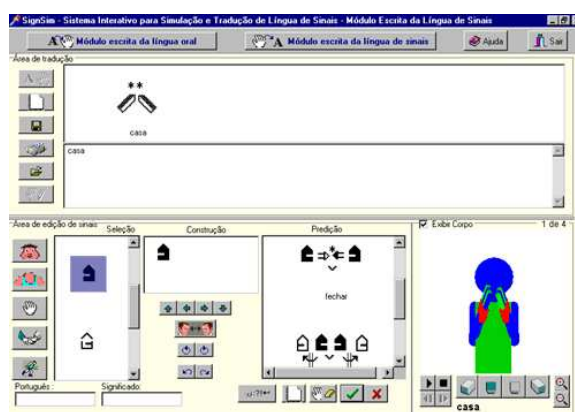


Figura 4.14: Módulo de escrita de sinais

4.4.1.2 Módulo de escrita do português

Na tradução do português para SignWriting, tem-se como entrada para o sistema uma palavra dada pelo usuário. Pode-se ter três situações:

1. Há apenas um sinal correspondente à palavra de entrada. Neste caso o processo de tradução e animação ocorre sem conflitos;
2. Há mais de um sinal correspondente. O sistema então fornece os sinais e o usuário deve escolher o que deseja;
3. Não há na base de dados informações sobre a palavra de entrada. O usuário pode então ignorar a palavra, escrevê-la através da datilologia e a animação será feita através da soletração manual; ou inserí-la na base, fornecendo para isso a palavra, o significado, suas características na escrita de sinais (configuração das mãos, movimento, etc.) e sua animação em 3D, através do editor de sinais.

Na figura 4.15 (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) vê-se a interface do módulo de escrita do português com o resultado para a frase “começar a aprender”. Pode-se observar os sinais que representam as palavras começar e aprender, bem como a animação de APRENDER.



Figura 4.15: Módulo de escrita da língua portuguesa

4.4.1.3 Animação

Nos dois módulos acima mencionados há uma parte destinada à animação 3D do sinal correspondente à palavra escrita. Conforme Campos (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) esta animação é uma sequência de estados de objetos 3D que compõem o sinal. Por exemplo, na figura 4.16 (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) percebe-se que o sinal correspondente à palavra começar tem três quadros-chave.

O modelo usado para representar um humano é construído a partir de primitivas geométricas em 3D (cilindros e paralelepípedos), formando assim a parte superior do corpo humano. De acordo com a figura 4.16 tem-se que o modelo possui aproximadamente 32 cilindros e 3 paralelepípedos.

Para a geração dos gestos foi criado um editor. Na figura 4.17 (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) observa-se que o usuário deve determinar as posições (pontos no espaço) onde estarão as articulações no término de cada quadro de animação.

Todos os quadros, determinados pelo editor para um dado gesto são então renderizados um por vez. Para que a transição de um quadro para outro seja uniforme, o sistema

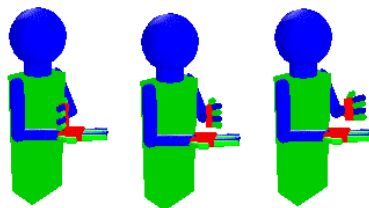


Figura 4.16: Animação do sinal COMEÇAR

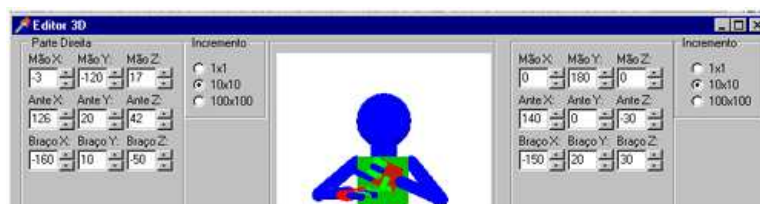


Figura 4.17: Editor 3D

calcula posições intermediárias fazendo assim com que tanto a passagem de um quadro para outro quanto a passagem de um gesto para outro sejam executadas continuamente.

Nos dois módulos do sistema há uma janela integrante que mostra a animação dos gestos, conhecida como visualizador 3D. Através dela o usuário tem ainda a opção de mudar a posição da câmera para visualizar a execução do gesto de um ponto diferente. Na figura 4.18 (BORBA CAMPOS; GIRAFFA; SANTAROSA, 2000) vê-se a interface do visualizador bem como a utilização da mudança de posição de câmera na execução do gesto referente ao sinal COMEÇAR.

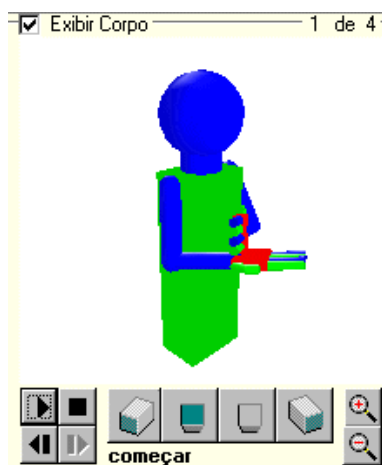


Figura 4.18: Visualizador 3D

4.4.2 AGA-Sign

AGA-Sign é um animador de gestos voltado para línguas de sinais, desenvolvido na UFRGS (DENARDI; MENEZES; COSTA, 2005). Ele é baseado no modelo AGA (animação gráfica baseada em autômatos finitos) (ACCORSI, 2000), a qual especifica a ani-

mação a partir de um conjunto de objetos e suas respectivas variações através de um autômato finito com saída, e na linguagem de descrição da animação AgaML (AGA Markup Language) que especifica a animação a partir dos objetos de animação (atores) e instanciação dos mesmos, além da especificação de uma fita de entrada. Quando os autômatos são simulados através da leitura da fita de entrada, as transições entre os estados controlam a animação dos objetos.

O foco principal deste trabalho é a determinação de como deve ser a especificação do autômato para que haja a tradução da palavra escrita em SignWriting para uma animação. A figura 4.19 (DENARDI; MENEZES; COSTA, 2005) mostra os passos a serem seguidos para se gerar uma animação. Depois da entrada em SignWriting gerada através do editor SW, há a conversão dos símbolos em texto que representa a posição, rotação, variação, preenchimento e forma dos mesmos. Feito isso há a tradução da representação definida anteriormente em uma fita de animação, de acordo com as definições AgaML. Esta fita é então usada como entrada no Aga-sign, que é responsável pela animação.

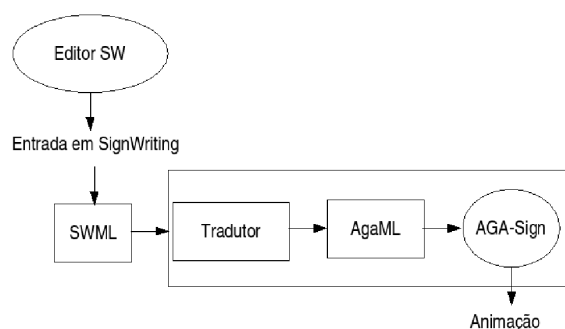


Figura 4.19: Arquitetura do AGA-Sign

O modelo usado na animação é composto por primitivas geométricas 2D e representações de sinais de SignWriting para as mãos. Como não há interpolação entre gestos, não há uma mudança suave entre eles, o que não corresponde à realidade.

4.5 Considerações finais

Das três ferramentas que tem como foco a animação observa-se que a mais completa é a ViSiCast. Nela a animação é suave, contínua e realista. Já na Vsigns e na HST não se percebe a naturalidade dos movimentos, o que é de suma importância para o entendimento do sinal.

A tabela 4.1 compara as ferramentas que possuem como foco a animação de humanos virtuais de acordo com alguns parâmetros:

- Uso de sistema simbólico de representação, seja SignWriting ou outro, o que auxilia uma maior compreensão do surdo, que muitas vezes não é alfabetizado em português;
- exploração do espaço de enunciação, mudando-o sempre que há necessidade, seja por características intrínsecas ao humano virtual (personalidade, grau de conhecimento em LIBRAS por parte do receptor) ou por barreiras físicas (obstáculos, lugares cheios, etc.);

- realismo na animação, ou seja, se a execução dos sinais é suave e contínua e se estes são passíveis de entendimento;
- uso de LIBRAS, tendo assim um sistema voltado para surdos brasileiros.

Percebe-se então que as ferramentas existentes não foram desenvolvidas para LIBRAS e não levam em conta o espaço de enunciação, velocidade e tensão dos movimentos, o que motivou a criação de uma técnica de animação que leva em conta todos esses parâmetros.

Tabela 4.1: Comparação entre as técnicas

Parâmetro	HST	VSigns	ViSiCast
Uso de sistema simbólico de representação	não	sim	não
Exploração do espaço de enunciação	não	não	não
Realismo na animação	não	não	sim
Uso de LIBRAS	não	não	não

5 ANIMANDO LIBRAS

Como LIBRAS é uma língua baseada em gestos, a técnica apresentada no Capítulo 2 pode ser usada para representar sinais, sendo que algumas etapas devem ser consideradas: modelagem do objeto, geração dos gestos e animação, de forma que o movimento ocorra sempre dentro de um espaço determinado. A implementação de um protótipo deve considerar então três módulos diferentes: um modelador, um gerador de ações e um animador, (ver figura 5.1). Estes módulos serão vistos em detalhes adiante. Em resumo gera-se o modelo desejado através do modelador. Este modelo é utilizado no gerador de ações para a criação dos gestos necessários à formação dos sinais. Os gestos criados são armazenados para posterior utilização. O modelo também é utilizado no animador juntamente com os gestos e a animação dos sinais é feita conforme necessidade.

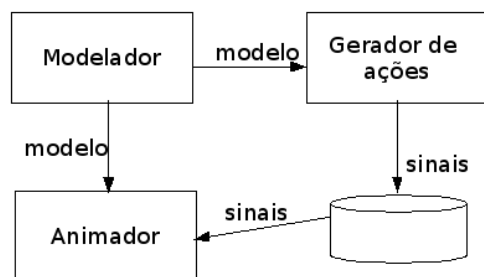


Figura 5.1: Arquitetura do sistema de animação

No caso de animação de sinais em LIBRAS deve ser previsto, além da naturalidade do sinal (o quanto é próximo da realidade), seu entendimento, ou seja, o quanto é passível de compreensão. De acordo com Marcato (MARCATO; ROCHA; LIMA, 1990), ao se gerar uma animação computacional para línguas de sinais deve-se considerar como parâmetros:

- Configuração das mãos;
- Ponto de articulação;
- Movimento;
- Aspectos não-manuais.

Tanto a configuração das mãos quanto os pontos de articulação de um determinado gesto são estipulados indiretamente na descrição de um gesto. Ao se definir os valores

das rotações das juntas das mãos para se chegar a uma posição em um tempo estipulado, a configuração das mãos é determinada. Já as posições das outras juntas em conjunto com as das mãos vão estipular os pontos de articulação. O movimento engloba todos os parâmetros envolvidos na ação do gesto, os quais já são determinados separadamente.

Com relação aos aspectos não manuais, cabe ressaltar que este trabalho não engloba os parâmetros faciais na obtenção do gesto. Apesar de sua importância na interpretação de LIBRAS, a animação facial está fora do escopo deste trabalho, merecendo um estudo a parte.

5.1 Determinação do espaço disponível para a animação de um sinal

Além dos parâmetros anteriormente citados, para que um sistema gere uma determinada animação gestual para Libras através de um humano virtual deve-se definir o espaço no qual um sinal é executado, chamado espaço de enunciação. Este espaço é definido pelo volume que engloba a região onde um sinal pode ser articulado. Sua parte principal é aquela que pode ser alcançada com os braços nem muito esticados, nem muito próximos ao corpo. A posição dos braços na determinação desse espaço pode ser comparada à posição dos mesmos quando se faz tricô. Isso pode ser visto na figura 5.2.



Figura 5.2: Posição dos braços na determinação do espaço de enunciação

A representação geométrica deste espaço é uma elipsóide cujas dimensões são calculadas a partir da envoltória convexa do humanóide. Na figura 5.3 tem-se a imagem do espaço, que é calculado da seguinte forma: tem-se B_x como a largura da envoltória, R_x é o raio que define a largura da elipsóide e seu valor é a metade da medida de B_x . A metade da altura da envoltória (B_z) define o diâmetro da elipsóide que é a altura da mesma e seu raio (R_z) será consequentemente a metade de B_z e o raio que define a profundidade do espaço (R_y) é definido como a metade da profundidade da envoltória (B_y). Esta elipsóide é definida pelos raios anteriores e um centro, o qual é calculado da seguinte forma: Dados o ponto $P(P_x, P_y, P_z)$ e os raios R_x , R_y e R_z , o centro da elipsóide E é o ponto $C(C_x, C_y, C_z)$, onde as coordenadas são calculadas por:

$$\begin{aligned} C_x &= R_x - P_x \\ C_y &= R_y - P_y \\ C_z &= R_z - P_z \end{aligned}$$

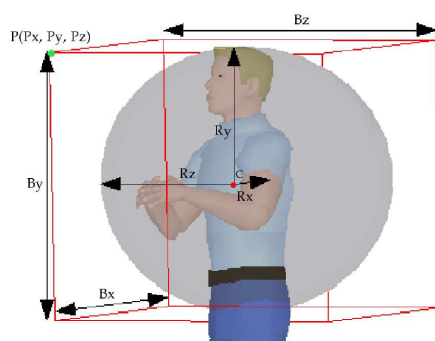


Figura 5.3: Espaço de enunciação

5.2 Geração e animação de um sinal em LIBRAS

Em LIBRAS um sinal é a representação gestual de uma palavra. Para que se consiga animar um sinal é necessário algumas vezes quebrá-los em pedaços (um sinal pode ter pontos de articulação intermediários até chegar à configuração final). Esses pedaços são conhecidos como elementos (denominados aqui gestos). Por exemplo tem-se o sinal BOM (figura 5.4). Este sinal é composto por dois gestos: o primeiro leva o braço direito de uma configuração qualquer para a configuração que pode ser vista na figura 5.4(a) e o segundo leva o braço direito para a posição final vista na figura 5.4(b). Os gestos por sua vez são definidos em função de *key-frames* ou poses.

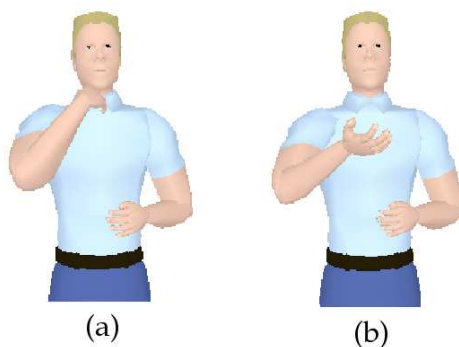


Figura 5.4: Configurações finais dos gestos do sinal BOM - Configuração intermediária (a) e configuração final (b)

Conforme dito na seção 2.3 o *end effector* é o ponto final da cadeia cinemática, ou um ponto que esteja ligado à cadeia. Ele é o ponto que deve se mover em encontro com o alvo. Como em LIBRAS a configuração das mãos e o ponto de articulação devem permanecer semelhantes ao padrão, é válido mover todas as juntas da cadeia para se chegar ao comportamento desejado. Com isso é coerente escolher como ponto final uma junta da mão, já que esta não poderá de forma alguma mudar drasticamente sua posição e orientação quando confrontada com seu valor no gesto já existente.

Os melhores pontos a serem escolhidos são os que representam as juntas das falanges dos dedos, pois são as juntas mais próximas do último ponto da cadeia. O escolhido no caso foi a falange do dedo médio, conforme figura 5.5.

No teste de colisão do algoritmo apresentado na sub-seção 2.3.3, é preciso verificar se

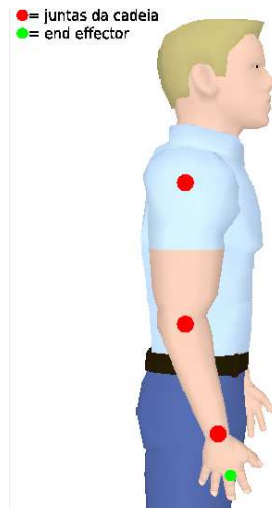


Figura 5.5: Juntas que compõem a cadeia e o *end effector* do braço direito

o ponto que descreve a posição da junta em questão está dentro da elipsóide que descreve o espaço de enunciação. Isto é feito através da substituição, na equação da elipsóide, do ponto que representa a junta. Se o resultado da equação é maior que 1 significa que a junta está fora do espaço de enunciação. A equação em questão é

$$\frac{(x_j^2)}{a^2} + \frac{(y_j^2)}{b^2} + \frac{(z_j^2)}{c^2} > 1 \quad (5.1)$$

Onde $P_j = (x_j, y_j, z_j)$ é a posição da junta e a é o raio da elipsóide no eixo x , b é o raio no eixo y e c é o raio no eixo z .

Como a cada término de execução de um sinal o humanóide volta para a pose dita padrão (posição de descanso), ao iniciar outro sinal ele usa como posição inicial a configuração final da pose de descanso. O problema que ocorre é que para um mesmo gesto, tanto o *end effector* quanto a posição inicial das juntas será igual. Assim, independente do tamanho do espaço de enunciação, as juntas vão se mover da mesma forma. Para que isso não ocorra é preciso também alterar a posição de descanso até que haja colisão com o espaço de enunciação, para depois ser aplicada a cinemática inversa. Cabe acrescentar que a nova pose, derivada da pose de descanso serve apenas para fins de cálculo, não sendo nunca exibida. A figura 5.6 (a) mostra a pose que é sempre exibida quando há a necessidade da pose de descanso. A figura 5.6 (b) mostra como será a configuração inicial (pose de cálculo) das cadeias para o cálculo das novas posições se o espaço de enunciação for 10% maior que o padrão.

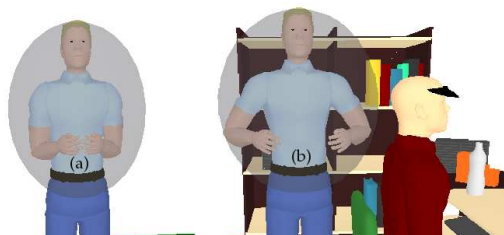


Figura 5.6: Pose de descanso (a) e pose utilizada para o cálculo da cinemática inversa quando o espaço de enunciação é 10% maior que o padrão

5.3 Implementação dos módulos

Os módulos de geração de gestos e de animação foram implementados usando a biblioteca V-ART (*Virtual Articulations for Virtual Reality* (GRUPO DE COMPUTAÇÃO GRÁFICA DA UFRGS, 2007)), desenvolvida com o objetivo de facilitar a criação de programas envolvendo humanóides. Ela se distingue das demais bibliotecas já existentes por permitir a representação de articulações anatomicamente corretas. O V-ART tem suas origens no projeto VPAT (MACIEL; NEDEL; FREITAS, 2002) e era conhecido por esse nome até maio de 2006, quando foi renomeado. Maiores informações sobre a biblioteca podem ser vistas no anexo 7.2. Já o modelador foi desenvolvido com o auxílio do Blender (BLENDER ARTISTS, 2007), software para modelagem 3D, disponível sob licença GPL (*GNU General Public License*).

5.3.1 Modelador

Foi gerado um humanóide com base nos detalhes descritos na seção 2.1. Uma imagem do modelo pode ser vista na figura 2.9.

Foi também desenvolvida uma DTD (ver anexo A.3) a qual define regras para a descrição textual de todos os objetos da cena (câmera, luz, humanóides, etc.). Com base nesta DTD foi desenvolvido um exportador na forma de um *plug-in* para o Blender, que transforma o modelo gerado em um arquivo XML de acordo com os padrões estipulados.

A figura 5.7 mostra um trecho do xml do humanóide. Nela observa-se dois nodos. O primeiro trata da descrição da junta raiz do humanóide. Tem-se como parâmetros sua posição no mundo, seu tipo, limite do DOF (em radianos), além do valor do DOF para a posição de descanso. Além deste nó tem-se o que especifica a malha que representa o membro o qual possui a junta da cintura. O arquivo gerado com toda a descrição do modelo usado pode ser visto no anexo A.3. Classes específicas do V-ART tratam do carregamento do modelo na cena nos módulos seguintes.

```
<?xml version="1.0"?>
<!DOCTYPE scene SYSTEM "vpatScene.dtd">
<scene description="policial">
  <node>
    <joint description="hipjoint" type="uniaxial">
      <dof description="twisthipjoint">
        <position x="0.000000" y="0.000000" z="0.000000"/>
        <axis x="-0.015017" y="-0.024149" z="-0.999596"/>
        <range min="-0.523599" max="0.523599" rest="0.500000"/>
      </dof>
    </joint>
  </node>
  <meshobject filename="man_hands7c.obj" description="hipMesh" type = "obj"/>
</node>
```

Figura 5.7: Trecho do XML que descreve o humanóide

5.3.2 Gerador de ações

Entende-se por gesto todo movimento do corpo para exprimir idéias ou sentimentos (MICHAELIS, 1998). No caso de uma animação, ele indica como as juntas se compor-

tarão com o passar do tempo. Alguns parâmetros estão envolvidos no gesto, tais como as juntas envolvidas no processo, suas posições, sua ciclicidade, duração e velocidade, conforme seção 2.2.

Para que um movimento seja executado na animação, é necessário especificar primeiramente o gesto a ser realizado. Foi desenvolvido então um gerador de gestos chamado *Action Generator*, cuja interface pode ser vista na figura 5.8.

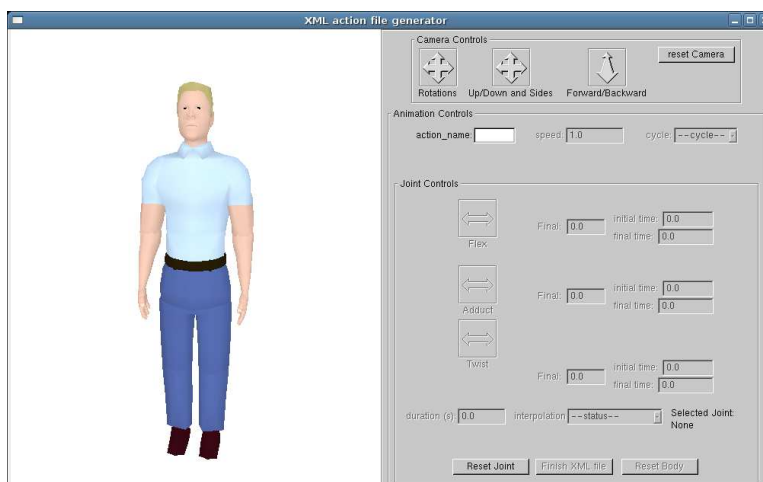


Figura 5.8: Interface do Action Generator

Através do menu existente na interface são determinados o nome do gesto (que será o nome do arquivo XML gerado), a velocidade e a ciclicidade da animação, a duração do gesto (em segundos) e o tipo de interpolação para cada junta escolhida, além das posições finais dos DOFS. A escolha da junta a ser movida é feita através do mouse.

O arquivo XML gerado deve estar de acordo com a DTD (vide anexo A.3) criada, que padroniza o que deve ter em uma ação (no caso em um gesto). A figura 5.9 mostra um trecho do XML que descreve um dos gestos que formam o sinal OBRIGADO. Observa-se que os parâmetros cujos valores são especificados são o nome da ação, a velocidade, a ciclicidade. Estes são seguidos por blocos com o nome da junta que irá se mover, além da duração do movimento e a função de interpolação a ser usada, além dos DOFs que se moverão, representados pela identificação de seu tipo (flexão, adução ou *twist*), tempo inicial e final e posição final (valor entre 0 e 1). Um exemplo completo de XML gerado pode ser visto no apêndice A.3.

```
<?xml version="1.0"?>
<!DOCTYPE action SYSTEM "vartAnimation.dtd">
<action action_name="" speed="1" cycle="false">
  <joint_movement joint_name="R_handjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.779342">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
</action>
```

Figura 5.9: Trecho do XML que descreve um dos gestos que compõem o sinal OBRIGADO

Após criados e armazenados todos os gestos que compõem os sinais estes são utilizados no módulo de animação, bem como o modelo gerado.

5.3.3 Animador

O módulo de animação é responsável por executar os sinais definidos pelo *Action Generator*, levando em conta o tamanho do espaço de enunciação. Este módulo recebe o nome de *Human Libras*.

A figura 5.10 mostra o processo necessário para se animar um sinal. Primeiro o modelo do humanóide, criado segundo visto na subseção 5.3.1, é analisado através de um parser e carregado na cena. Esta é renderizada através da utilização de classes específicas do V-ART. Os arquivos do tipo XMLAction, criados pelo *Action Generator* (visto na subseção 5.3.2), são também submetidos a um parser e carregados na cena.

Escolhida a expressão (ou palavra) a ser animada, os gestos que fazem parte do sinal que a representa sofrem uma adequação em suas amplitudes de acordo com o espaço de enunciação existente e são então animados.

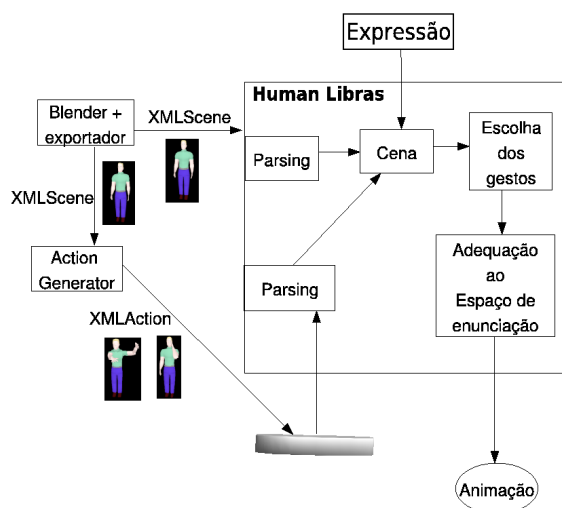


Figura 5.10: Sistema de animação para Libras

A interface do *Human Libras* pode ser vista na figura 5.11. O usuário pode ver a cena de outro ângulo, fazendo com que o sinal seja mais facilmente reconhecido através dos controles de câmera. Ele também pode escolher o sinal a ser animado na lista existente. Ao se escolher o sinal os gestos que compõem o mesmo se adequam ao espaço existente (o que pode ser configurado na aplicação, conforme visto na interface). Cada gesto é executado na ordem correta para compor o sinal e um gesto só começa a ser executado quando o anterior é desativado.

Cabe ressaltar também que após a execução de um gesto, o humanóide retorna a uma pose considerada de descanso (padrão), a qual pode ser vista na figura 5.12.

Com relação à adequação ao espaço de enunciação, no caso da orientação, como o modelo é um corpo articulado hierárquico, o espaço de enunciação pode ser filho da articulação que representa a cintura do humanóide. Assim, se o corpo sofre uma rotação, o espaço de enunciação o acompanhará. Como a parte superior do corpo é filha da cintura, a transformação aplicada à ela consequentemente será aplicada aos seus filhos e assim o gesto vai acompanhar a mudança de orientação.

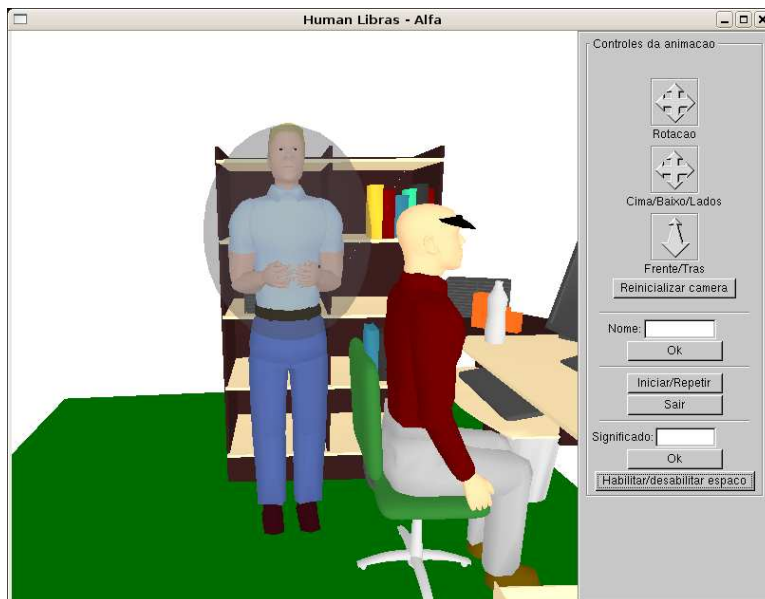


Figura 5.11: Interface do módulo animador (*Human Libras*)



Figura 5.12: Pose de descanso

5.4 Considerações finais

No capítulo 2 foi apresentada uma técnica de animação que leva em consideração a limitação do espaço disponível para a execução do movimento. Esta técnica foi aplicada em LIBRAS e implementada, gerando assim um sistema de animação para LIBRAS.

Este sistema criado é composto por 3 módulos: o modelador, que tem como saída um arquivo XML com a descrição do humanóide a descrição do humanóide (XMLScene). Este modelo é passado ao Gerador de gestos, que é responsável pela geração dos gestos que compõem os sinais. Estes gestos são descritos através de um arquivo XML (XMLAction) e assim como o modelo do humanóide são passados ao animador que é o módulo responsável por animar os sinais, adequando-os ao espaço de enunciação existente.

6 VALIDAÇÃO

Para que a técnica seja válida é necessário confrontar seus resultados com a realidade. Por causa disto foram realizados testes com pessoas observando os gestos e informando se a animação é passível de entendimento e reflete o que foi determinado (animação do sinal deve fazer o usuário entender qual palavra está sendo gesticulada).

A implementação descrita no capítulo 5 foi utilizada nos testes. Como o interesse era na identificação dos gestos, a interface do módulo animador (*Human LIBRAS*) foi modificada. A nova interface pode ser vista na figura 6.1.

Nesta nova interface tanto os sinais a serem exibidos quanto a ordem de exibição dos mesmos e o tamanho do espaço de enunciação são determinados no programa e não podem ser modificados pelo usuário. Assim não há mais uma lista de sinais na qual o usuário pode escolher qual será exibido, nem a opção de ser modificar o espaço de enunciação.

O usuário deve observar a execução do sinal (podendo repetir a animação sempre que necessário) e dar o significado do mesmo (qual a tradução do sinal em português). Para que outro sinal seja animado é necessário utilizar o botão da interface chamado “próximo sinal”. Os sinais vão sendo exibidos até que todos os sinais pré-estabelecidos tenham sido animados, o que faz com que o programa termine sua execução.

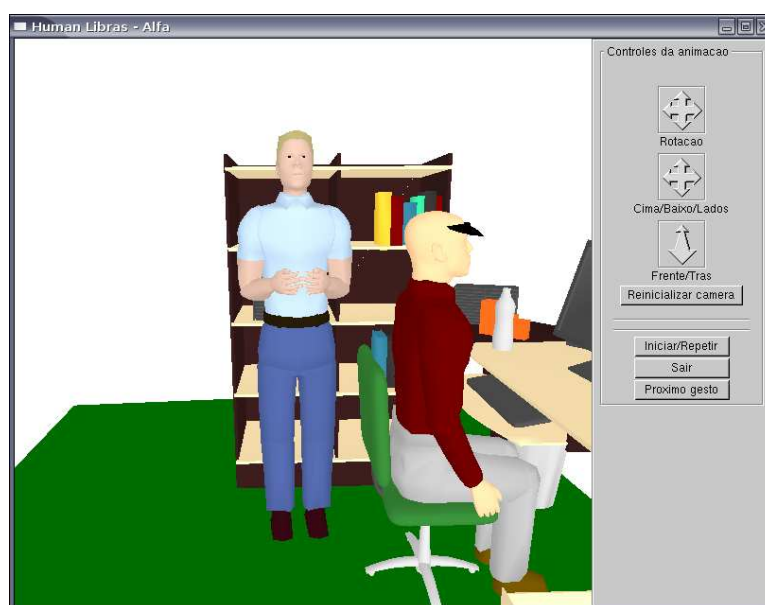


Figura 6.1: Interface modificada do Animador de LIBRAS para os testes

Foram apresentados quatro sinais: QUERER (composto por dois gestos), AMAR (composto por um gesto), ACHAR (composto por dois gestos) e OBRIGADO (composto por dois gestos), além de uma frase, ou expressão, composta de dois sinais, cada um com dois gestos (BOM DIA). Todos estes gestos foram mostrados com três espaços de enunciação diferentes (padrão, 10% maior e 40% menor). Para o espaço de enunciação padrão o sinal modelado foi animado. Para os demais houve a adequação automática do sinal usando cinemática inversa. Os vídeos com a animação dos sinais podem ser vistos em (SCHNEIDER, 2008).

Na figura 6.2 pode-se observar a posição final dos dois gestos que compõem o sinal ACHAR nos três espaços especificados (padrão, menor e maior, respectivamente). Percebe-se a diferença entre as posições das juntas do braço e pulso, sem entretanto mudar drasticamente a configuração da mão ou o ponto de articulação.

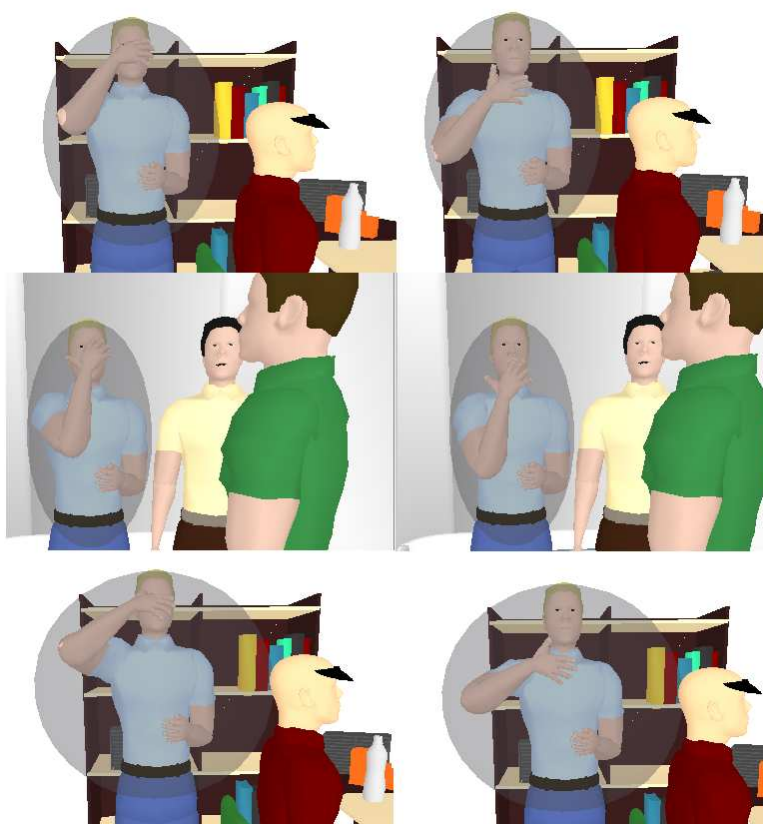


Figura 6.2: Gestos ACHAR em três configurações

A quantidade de iterações para que o resultado desejado ocorresse foi fixada em 300, mas no máximo foram realizadas 75 iterações para que o *end effector* convergisse, o que garantiu o tempo real na animação. A tabela 6.1 mostra quantas iterações foram necessárias para cada gesto dos sinais, para os espaços de enunciação maior e menor que o padrão.

Os testes realizados deveriam verificar se os gestos animados pelo humanóide eram passíveis de reconhecimento, mesmo com a mudança do espaço de enunciação. Cabe ressaltar que a ordem em que os gestos foram apresentados foi aleatória.

Para que fosse verificado o conhecimento do usuário quanto ao reconhecimento dos gestos, foi apresentado após o teste da ferramenta, um vídeo contendo os sinais apre-

Tabela 6.1: Quantidade de iterações até a convergência do *end effector*

Sinais	Gesto	Espaço de enunciação maior	Espaço de enunciação menor
AMAR	Gesto único	75	72
QUERER	Gesto 1	62	10
	Gesto 2	67	9
ACHAR	Gesto 1	62	67
	Gesto 2	10	9
OBRIGADO	Gesto 1	48	12
	Gesto 2	46	15
BOM	Gesto 1	28	1
	Gesto 2	30	1
DIA	Gesto 1	72	4
	Gesto 2	77	8

sentados (vide (SCHNEIDER, 2008)), para que o mesmo pudesse dar seus significados. Além disso foi aplicado um questionário (vide F.2) com perguntas sobre conhecimento em LIBRAS, além dos significados dos sinais exibidos tanto na animação quanto nos vídeos.

Foram voluntárias quatro pessoas para avaliar a técnica. O baixo número de usuários deu-se por causa do fato da exigência de que os mesmos tinham que ter conhecimento de LIBRAS. Os usuários se dividiram entre surdos e ouvintes (uma surda e três ouvintes) com conhecimento avançado em LIBRAS, sendo metade do sexo feminino e metade masculino. A idade variou de 25 a 37 anos. Os mesmos declararam conhecer todos os sinais a serem avaliados.

Dos sinais apresentados foram reconhecidos em todos os testes os sinais para AMAR (ou AMOR), ACHAR e OBRIGADO. O sinal para QUERER não foi reconhecido em um teste (foi confundido com o sinal CANSADO) e BOM DIA não foi reconhecido em dois testes (em um foi reconhecido como BOA MANHÃ e em outro como BOA (OU FALAR) LIBERDADE).

Foi exposta a dificuldade em se avaliar os sinais por falta de expressões faciais, mas observa-se que o reconhecimento dos mesmos não foi prejudicado por isso.

Outra coisa a se destacar é que os sinais que não foram bem reconhecidos através da animação não o foram também através dos vídeos, o que faz crer que o problema não é da técnica em si, e sim da modelagem dos gestos.

7 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma proposta de animação computacional de sinais em LIBRAS, o mais próximo possível da realidade e podendo ser entendidos independente do espaço disponível para sua execução. A seguir serão apresentadas algumas conclusões importantes, além de trabalhos futuros.

7.1 Conclusões

A principal contribuição deste trabalho é na avaliação de critérios para animação de gestos em LIBRAS, levando-se em conta um espaço de enunciação dinâmico. Foi necessário para isso analisar todos os parâmetros que compõem o sinal padrão, buscando sempre uma maior proximidade do sinal virtual (humanóide) para o real (humano), o que é uma tarefa complicada, haja visto que muitos parâmetros devem ser combinados.

Como o espaço de enunciação deve ser dinâmico, ele pode sofrer transformações em sua direção ou em seu tamanho e os gestos devem se adequar a estas mudanças. Se houver uma mudança de direção do corpo do humanóide, automaticamente o espaço também sofre essa transformação, já que o espaço faz parte do corpo articulado. Com relação ao tamanho, o rearranjo do gesto se dá através da modificação, no gesto padrão, das posições e orientações de determinadas juntas, posições estas calculadas através de cinemática inversa.

A tabela 7.1 compara a técnica proposta com as técnicas vistas no capítulo 4. Nela percebe-se que a técnica apresentada é a única que explora o espaço de enunciação e é voltada para LIBRAS, além de tratar muitos dos parâmetros das demais.

Tabela 7.1: Comparação entre as técnicas, inclusive a proposta

Parâmetro	HST	VSigs	Visicast	Técnica proposta
Uso de sistema simbólico de representação	não	sim	não	não
Exploração do espaço de enunciação	não	não	não	sim
Realismo na animação	não	não	sim	sim
Uso de LIBRAS	não	não	não	sim

Para que se tivesse a animação computacional alguns passos foram seguidos:

1. Determinação do modelo do humanóide
2. Geração dos gestos
3. Animação

Primeiro foi desenvolvido no Blender um modelo cujas juntas possuem limites definidos através de pesquisa na área da biomecânica. Este modelo foi então exportado para um arquivo XML (XMLScene), cuja descrição está de acordo com os parâmetros definidos no V-ART. Ele é então utilizado para se gerar os gestos através de uma ferramenta desenvolvida para isso (*Action Generator*). O resultado desta aplicação é um arquivo XML (XMLAction), com todos os parâmetros para a geração da animação (tempo de duração, velocidade, posição das juntas, etc.) definidos. Os gestos são formados por poses, todas essas definidas através do gerador de ações e arquivadas em uma base de dados. O humanóide gerado é ainda utilizado na renderização da cena do módulo de animação (*Human LIBRAS*). Informada uma expressão (ou palavra) para a animação, o tratamento das poses é feito de acordo com o espaço de enunciação dado e as mesmas são animadas na ordem correta.

Alguns atributos, tais como a velocidade do movimento e sua duração podem ser modificados via módulo da animação através de funções já existentes na biblioteca V-ART. Assim, se é desejável que o humanóide faça uma determinada ação mais rápido (indicando raiva, pressa, etc.) não há necessidade de se gerar novamente a ação e sim apenas modificar esses atributos na aplicação.

Com relação à validação da técnica proposta, apesar do baixo número de usuários nos testes, já foi possível notar a validação da mesma através do reconhecimento dos gestos apresentados. Apesar da falta de movimentos faciais, o que resultou em dificuldades no reconhecimento de alguns sinais para os usuários, observa-se pelos resultados que foi possível determinar o significado dos sinais.

7.2 Trabalhos futuros

Conforme dito no capítulo 2 a animação facial é de extrema importância, auxiliando na semântica das frases. Porém ela não foi analisada neste trabalho, por ser tão complexa a ponto de merecer um estudo à parte. Assim, para que os gestos se tornem mais realistas ainda há a necessidade de se definir uma técnica de animação facial, fixando quais pontos da malha da face devem se mover, quais são os limites dos movimentos, como será a deformação da malha para que se observe com clareza as modificações das poses, etc.

É interessante também que a malha do modelo possa sofrer deformação para que se diminuam as irregularidades no momento em que uma junta se move, fazendo com que possam aparecer buracos no encaixe das partes corpóreas.

Na parte dos gestos, como os mesmos são formados por configuração de mãos, ponto de locação, etc. conforme visto no capítulo 3, todos estes aspectos já tendo sido determinados através da gramática da língua, é interessante então que se tenha uma forma de se criar os gestos mais automaticamente, de maneira que o usuário informasse todos os parâmetros textualmente e o gesto fosse gerado, baseado na imagem gráfica gerada e armazenada anteriormente. Acredita-se que desta forma será mais fácil criar os gestos.

Espera-se que este trabalho seja o início da criação de um tradutor virtual para LIBRAS, dando assim mais acessibilidade aos surdos. Para tal é necessário definir uma forma de transformar uma frase em Português para LIBRAS, o que parece ser uma tarefa complexa, haja visto que o mapeamento de frases de uma língua para a outra não é direto.

REFERÊNCIAS

- ACCORSI, F. Animação bidimensional para World Web baseada em autômatos finitos. In: WORKSHOP DE MÉTODOS FORMAIS, WMF, 3., 2000. **Anais...** João Pessoa: CEFET-PB, 2000.
- BANGHAM, J. A. et al. Virtual signing: capture, animation, storage and transmission - an overview of the ViSiCAST Project. In: IEE SEMINAR ON SPEECH AND LANGUAGE PROCESSING FOR DISABLED AND ELDERLY PEOPLE, 2000. **Proceedings...** [S.l.]: IEE, 2000.
- BANGHAM, J.A. et al. Signing for the deaf using virtual humans. In: IEE COLLOQUIUM ON SPEECH AND LANGUAGE PROCESSING FOR DISABLED AND ELDERLY, 2000. **Proceedings...** [S.l.]: IEE, 2000.
- BATTISON, R. Phonological deletion in american sign language. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 20., 2000, Curitiba. **Anais...** Curitiba: PUCPR, 2000.
- BIDEAU, B. et al. Virtual reality applied to sports: do handball goalkeepers react realistically to simulated synthetic opponents? In: ACM SIGGRAPH INTERNATIONAL CONFERENCE ON VIRTUAL REALITY CONTINUUM AND ITS APPLICATIONS IN INDUSTRY, VRCAI, 2004. **Proceedings...** New York: ACM Press, 2004. p.210–216.
- BLENDER ARTISTS. **Broken Shadows**: Portrait, Grunge theme. Disponível em: <<http://blenderartists.org/forum/showthread.php?t=103053>>. Acesso em: 21 nov. 2007.
- BORBA CAMPOS, M. de; GIRAFFA, L. M. M.; SANTAROSA, L. M. C. SIGNSIM: uma ferramenta para auxílio à aprendizagem da língua brasileira de sinais. In: CONGRESSO IBERO-AMERICANO DE INFORMÁTICA NA EDUCAÇÃO - RIBIE, 5., 2000, Vinã del Mar. **Anais...** [S.l.: s.n.], 2000.
- BRITO, L. F. Uma abordagem fonológica dos sinais da LSCB. **Espaço: Informativo técnico científico do INES**, Rio de Janeiro, v.1, n.1, p.20–43, 1974.
- DENARDI, R. M.; MENEZES, P. B.; COSTA, A. C. R. AGA-Sign: Animador de Gestos aplicado à Língua de Sinais. **INFOCOMP, Journal of Computer Science**, Lavras, v.4, n.1, p.47–53, 2005.
- EMMELKAMP, P. et al. Virtual Reality Treatment versus exposure in vivo: A Comparative Evaluation in Acrophobia. **Behavior Research & Therapy**, [S.l.], v.40, n.5, p.509–516, 2002.

FENEIS. **Dicionário de Libras**. Disponível em: <<http://www.dicionariolibras.com.br>>. Acesso em: 1 ago. 2007.

GRIEVE-SMITH, A. B. **SignSynth**: A Sign Language Synthesis Application Using Web3D and Perl. Disponível em: <<http://www.unm.edu/grvsmth/portfolio/ss010710.pdf>>. Acesso em: 01 nov. 2005.

GRUPO DE COMPUTAÇÃO GRÁFICA DA UFRGS. **V-ART**: Virtual Articulations for Virtual Reality. Disponível em: <<http://www.codeplex.com/vart>>. Acesso em: 21 nov. 2007.

HOLDEN, E. J.; ROY, G. G. The graphical translation of English text into signed English in the hand sign translator system. **Computer Graphics Forum**, Amsterdam, v.11, n.3, p.357–366, 1992.

HOPPENFELD, S. **Propedêutica ortopédica- coluna e extremidades**. [S.l.]: Atheneu, 2004.

HULST, H. Atoms of segmental structure: components, gestures and dependency. **Phonology**, Cambridge, v.5, p.1–19, 1989.

ISO/IEC. **ISO/IEC 14496-2**: Committee Draft ISO/IEC JTC1/SC29/WG11 N1902, Information Technology - Coding of Audio-Visual Objects: Visual. Fribourg, Switserland, 1997.

KAMATA, K. et al. An approach to Japanese-sign language translation system. In: SYSTEMS, MAN AND CYBERNETICS, 1989. **Proceedings...** [S.l.: s.n.], 1989. v.3, p.1089–1090.

KARNOPP, L. B. **Aquisição da Linguagem por crianças surdas**. Disponível em: <http://www.ines.org.br/ines_livros/36/36_PRINCIPAL.HTM>. Acesso em: 21 jun. 2006.

KENDALL, F. P.; MCCREARY, E. K.; PROVANCE, P. G. **Músculos provas e funções**. 4.ed. [S.l.]: Manole, 1995.

KENNAWAY, R. Synthetic Animation of Deaf Signing Gestures. In: INTERNATIONAL GESTURE WORKSHOP, GW, 2001, London. **Gesture and Sign Language in Human-computer Interaction**: revised papers. Berlin: Springer, 2001. p.146-157. (Lecture Notes in Computer Science, v.2298).

KLIMA, E.; BELLUGI, U. Wit and poetry in american sign language. **Sign Language Studies**, [S.l.], p.203–224, 1975.

LINCOLN, M.; COX, S. J.; NAKISA, M. The development and Evaluation of a speech to sign translation system to assist transactions. **Journal of Human-computer Studies**, Tunísia, 1998.

LIU, X.; ZHUANG, Y.; PAN, Y. Video based human animation technique. In: ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA (PART 1), MULTIMEDIA, 7., 1999. **Proceedings...** New York: ACM, 1999. p.353–362.

- MACIEL, A.; NEDEL, L. P.; FREITAS, C. M. D. S. Anatomy Based Joint Models for Virtual Humans Skeletons. In: INTERNATIONAL CONFERENCE ON COMPUTER ANIMATION, 2002, Geneva. **Proceedings...** Los Alamitos: IEEE Computer Society, 2002. p.220–224.
- MARCATO, S. A.; ROCHA, H. V. da; LIMA, M. C. M. P. Um Ambiente para a Aprendizagem da Língua de Sinais. **Sign Language Studies**, [S.l.], n.32, 1990.
- MARSHAL, I. et al. On The Real Time Elision of Text. In: INTERNATIONAL WORKSHOP ON EXTRACTION, FILTERING AND AUTOMATIC SUMMARIZATION, 1998. **Proceedings...** [S.l.]: CNRS, 1998.
- MARSHALL, I.; SAFAR, E. A prototype text to British Sign Language (BSL) translation system. In: MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, ACL, 41., 2003. **Proceedings...** [S.l.]: Association for Computational Linguistics, 2003. p.113–116.
- MAUREL, W.; THALMANN, D. Human Sholder Modeling Including Scapulo-Thoracic Constraint and Joint Sinus Cones. **Computer and Graphics**, [S.l.], v.24, n.2, p.203–218, 2000.
- MICHAELIS. **Moderno dicionário da Língua Portuguesa**. São Paulo, Brasil: Melhoramentos, 1998.
- PAPADOGIORGAKI, M. et al. Synthesis of Virtual Reality Animations from SWML using MPEG-4 Body Animation Parameters. In: WORKSHOP ON THE REPRESENTATION AND PROCESSING OF SIGN LANGUAGES; INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION, LREC, 4., 2004, Lisboa, Portugal. **Proceedings...** [S.l.: s.n.], 2004. p.43–50.
- PONTES, A. M.; ORTH, A. I. **Uma Proposta de Interface de Software Orientada à Linguagen de Sinais**. Disponível em: <<http://www.unicamp.br/~ihc99/Ihc99/AtasIHC99/art31.pdf>>. Acesso em: 21 nov. 2006.
- QUADROS, R. de; KARNOPP, L. **Língua de Sinais Brasileira: Estudos Linguísticos**. [S.l.]: Artmed, 2004.
- ROCHA COSTA, A. C. da; DIMURO, G. P. **The (current version of) SWML DTD**. Disponível em: <<http://swml.ucpel.tche.br>>. Acesso em: 21 nov. 2006.
- SCHNEIDER, A. R. A. **Vídeos usados na validação do Human Libras**. Disponível <<http://www.inf.ufrgs.br/araschneider/videos>>. Acesso em: 21 jan. 2008.
- SLEATOR, D.; TEMPERLEY, D. **Parsing english with a Link Grammar**. Pittsburgh, PA: School Computer Science, Carnegie Mellon University, 1991.
- SUTTON, V. **SignWriter Manual**. La Jolla, CA: Deaf Action Committee for Singwriting, 1995.
- UGHINI, C.; NEDEL, L. P. Adaptable Deterministic Roadmaps for Motion Planning of Bodies with many DOFs. In: INTERNATIONAL CONFERENCE ON COMPUTER ANIMATION AND SOCIAL AGENTS, 20., 2007, hassel, Belgium. **Proceedings...** Diepenbeek, Belgium: Hasselt University, 2007. v.1, p.19-26.

W3C. **Extensible Markup Language (XML)**. [S.l.], 2000.

WANG, L. C. T.; CHEN, C. C. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. **IEEE Transactions on Robotics and Automation**, [S.l.], v.7, n.4, p.489–499, 1991.

WELMAN, C. **Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation**. Masters Thesis, Simon Fraser University, Ottawa.

APÊNDICE A V-ART

V-ART (*Virtual Articulations for Virtual Reality*) é uma biblioteca desenvolvida para manipulação e animação de objetos articulados 3D (preferencialmente humanos virtuais). Ela foi desenvolvida em C++ e a vantagem é que a mesma possui um sistema de suporte a animações e permite a representação de articulações biologicamente corretas. Suas origens estão no VPAT (MACIEL; NEDEL; FREITAS, 2002), que propôs um modelo baseado em anatomia para representar o esqueleto humano.

A seguir, encontram-se descritas as principais características do V-ART, do ponto de vista de implementação.

A.1 A estrutura do V-ART

Como a biblioteca é desenvolvida de acordo com o paradigma de programação orientado a objetos, os tipos de dados existentes na mesma estão dispostos em classes, as quais são organizadas de acordo com uma hierarquia. Dentre estas classes pode-se citar:

- objetos gráficos (cilindros, esferas, cubos, malhas de polígonos, etc.)
- transformações geométricas
- luzes
- câmeras
- cenas
- janelas
- animações

Qualquer aplicação gráfica 3D deve ter um cena para ser visualizada. Esta por sua vez deve possuir luzes, câmeras e objetos a serem visualizados. Pode-se representar o exposto acima através de uma estrutura hierárquica, conforme figura A.1.

Pode-se dizer então que uma cena é composta por nós, e dentre estes nós tem-se objetos gráficos, luzes, câmeras, janelas, etc. Como todas estas representações já estão definidas na biblioteca, o programador foca mais na funcionalidade de sua aplicação.

O desenvolvimento do V-ART é um trabalho colaborativo, em constante desenvolvimento. Algumas informações adicionais podem ser encontradas em (GRUPO DE COMPUTAÇÃO GRÁFICA DA UFRGS, 2007).

O maior objetivo desta biblioteca é trabalhar com humanos virtuais. Assim classes específicas que tratam da manipulação de humanóides devem existir. Pensando no esqueleto humano de uma maneira simplificada, o mesmo é composto por ossos e estes são

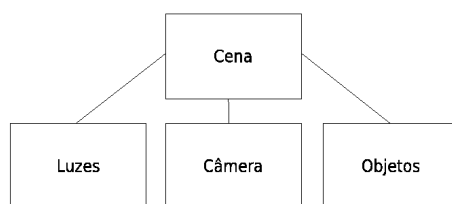


Figura A.1: Cena

ligados uns nos outros através de juntas (ou articulações), as quais podem ser móveis ou não.

A.2 Articulações

A maioria das articulações, sejam mecânicas (de robôs) e biológicas (humanas) são tradicionalmente representadas por transformações geométricas simples (rotações e translações). Algumas juntas mais complexas, como a do joelho, não são facilmente representadas por transformações simples. Além disso elas possuem limites funcionais e em alguns casos a mudança no comportamento de uma articulação gera alterações de limites em outras.

O V-ART é desenvolvido para tratar desses detalhes, fazendo com que o programador consiga tratar as articulações em alto nível e obtendo um alto grau de realismo. A definição das juntas está baseada no trabalho de mestrado de Anderson Maciel (MACIEL; NEDEL; FREITAS, 2002). Nele as articulações são representadas por um conjunto hierárquico de DOFs (*Degrees Of Freedom*). Cada DOF é definido através de uma transformação controlada em função de um valor escalar entre 0 e 1 (limites mínimo e máximo do mesmo), o qual representa a configuração do DOF. Alterando-se este valor, altera-se o estado do DOF. A figura A.2 mostra um uma junta definida por um DOF, além do espaço que esta pode percorrer entre o intervalo [0..1]. No V-ART as juntas podem ser classificadas em três grandes grupos: uniaxiais (um DOF), biaxiais (2 DOFs) e poliaxiais (3 DOFs).

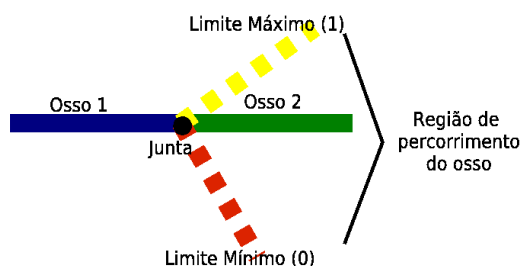


Figura A.2: Configuração possível de um DOF

Como dito anteriormente um esqueleto é representado por uma estrutura hierárquica de juntas e ossos (objetos gráficos). Por sua vez uma junta é definida por um conjunto de DOFs. Como qualquer estrutura hierárquica, os nós filhos acabam sofrendo modificações de acordo com a nova configuração do nó pai. Os DOFs, também representados hierarquicamente, sofrem modificações dependendo das mudanças nos DOFs anteriores.

A.3 Animações

As animações representadas no V-ART são feitas através de modificações nas juntas em um intervalo de tempo. A classe `Action` trata dessa parte. Ela é definida como um conjunto de funções que alteram as articulações, controladas por um objeto da classe `Joint Mover`. Cada uma dessas alterações da articulação é um conjunto das modificações de seus DOFS em tempos pré-definidos. Essas modificações são tratadas por objetos da classe `DOF Mover`. De acordo com a figura A.3 percebe-se que uma ação é estruturada da seguinte forma: toda ação possui um conjunto de juntas que se moverão com o tempo. Cada uma delas irá mover um conjunto de DOFs até uma certa configuração, desde um tempo inicial até um final (valores entre 0 e 1, por indicarem a porcentagem do tempo- "quando tiver passado x por cento do tempo o dof em questão irá ser modificado até y por cento do tempo ter passado").

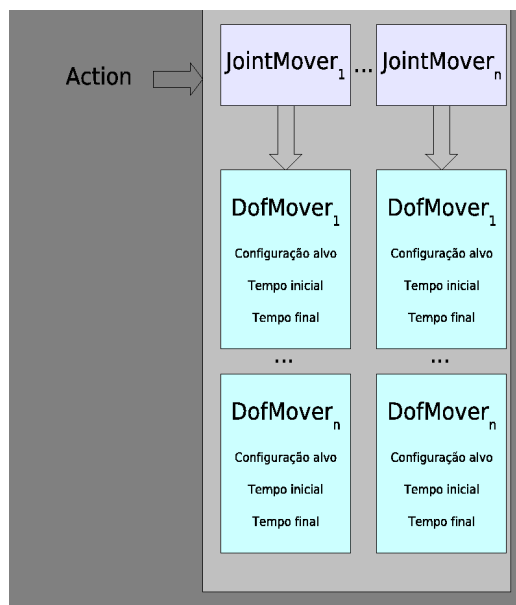


Figura A.3: Estrutura de uma ação

É possível se fazer também animação baseada em uma posição alvo através de cinemática inversa (CCD). São dados uma posição e uma orientação alvo, além de um *end effector* e de um erro posicional e de orientação, além de uma cadeia cinemática. Os DOFs dessa cadeia são movidos um por vez, através de uma função objetiva e ao se chegar no alvo (ou próximo a ele, de acordo com o erro definido) para-se o movimento. Maiores detalhes sobre isso estão na subseção 2.3.

Quando uma ação é definida, define-se também um tempo total de duração, chamado de tempo neutro. Esse valor é usado para calcular o tempo normalizado, de acordo com a equação

$$n = v.r/t \quad (\text{A.1})$$

onde n é o tempo normalizado, t é o tempo neutro, r é o tempo decorrido a partir do início da ação (tempo real) e v é a velocidade da ação.

As ações também têm prioridades que é verificada e armazenada na atualização da configuração de cada DOF, permitindo assim que ações possam ser executadas simultaneamente e que a de maior prioridade se sobressaia às demais. Por exemplo, se o

humanóide estiver correndo e quer pegar algo sem parar pode-se mesclar as duas ações através da determinação de prioridade.

A execução das ações é baseada em posições-chave para os DOFs do modelo a ser animado. A reprodução da animação é feita pela interpolação do estado de um grau de liberdade em função do tempo decorrido entre duas atualizações da imagem da cena. A interpolação é feita por objetos da classe `Interpolator` que podem ser criados sob medida para cada aplicação, desde que respeitem uma interface pré-definida.

APÊNDICE B XML DO HUMANÓIDE

```

<?xml version="1.0"?>
<!DOCTYPE scene SYSTEM "vpatScene.dtd">
<scene description="policial">
  <node>
    <joint description="hipJoint" type="uniaxial">
      <dof description="twisthipJoint">
        <position x="0.000000" y="0.000000" z="0.000000"/>
        <axis x="-0.015017" y="-0.024149" z="-0.999596"/>
        <range min="-0.523599" max="0.523599" rest="0.500000"/>
      </dof>
    </joint>
    <node>
      <meshobject filename="man_hands7c.obj" description="hipMesh" type = "obj"/>
    </node>
  <node>
    <joint description="bellyJoint" type="polyaxial">
      <dof description="flexbellyJoint">
        <position x="0.000239" y="0.174288" z="-0.054122"/>
        <axis x="1.000000" y="0.000351" z="0.000323"/>
        <range min="-0.349066" max="1.483530" rest="0.190476"/>
      </dof>
      <dof description="adductbellyJoint">
        <position x="0.000239" y="0.174288" z="-0.054122"/>
        <axis x="0.000323" y="-0.996548" z="0.083016"/>
        <range min="-0.959931" max="0.959931" rest="0.500000"/>
      </dof>
      <dof description="twistbellyJoint">
        <position x="0.000239" y="0.174288" z="-0.054122"/>
        <axis x="-0.000351" y="0.083016" z="0.996548"/>
        <range min="-1.308997" max="1.308997" rest="0.500000"/>
      </dof>
    </joint>
    <node>
      <meshobject filename="man_hands7c.obj" description="bellyMesh" type = "obj"/>
    </node>
  <node>
    <joint description="chestJoint" type="polyaxial">
      <dof description="flexchestJoint">
        <position x="-0.000401" y="0.325582" z="1.762064"/>
        <axis x="1.000000" y="0.000298" z="0.000320"/>
        <range min="-0.174533" max="0.174533" rest="0.500000"/>
      </dof>
      <dof description="adductchestJoint">
        <position x="-0.000401" y="0.325582" z="1.762064"/>
        <axis x="0.000321" y="-0.997440" z="-0.071508"/>
        <range min="-0.698132" max="0.698132" rest="0.500000"/>
      </dof>
      <dof description="twistchestJoint">
        <position x="-0.000401" y="0.325582" z="1.762064"/>
        <axis x="-0.000298" y="-0.071509" z="0.997440"/>
        <range min="-0.523599" max="0.523599" rest="0.500000"/>
      </dof>
    </joint>
    <node>
      <meshobject filename="man_hands7c.obj" description="chestMesh" type = "obj"/>
    </node>
  <node>
    <joint description="R_armJoint" type="polyaxial">

```

```

<dof description="flexR_armJoint">
  <position x="-1.463741" y="0.100822" z="4.023921"/>
  <axis x="0.973471" y="0.047945" z="-0.223729"/>
  <range min="-3.141593" max="0.785398" rest="0.800000"/>
</dof>
<dof description="adductR_armJoint">
  <position x="-1.463741" y="0.100822" z="4.023921"/>
  <axis x="-0.061993" y="0.996494" z="-0.056189"/>
  <range min="-0.698132" max="1.745329" rest="0.285714"/>
</dof>
<dof description="twistR_armJoint">
  <position x="-1.463741" y="0.100822" z="4.023921"/>
  <axis x="-0.220251" y="-0.068568" z="-0.973030"/>
  <range min="-0.959931" max="0.785398" rest="0.550000"/>
</dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_armMesh" type = "obj"/>
</node>
<node>
  <joint description="R_forearmJoint" type="biaxial">
    <dof description="flexR_forearmJoint">
      <position x="-2.093631" y="-0.095273" z="1.241181"/>
      <axis x="0.991499" y="0.076837" z="-0.105000"/>
      <range min="-2.356194" max="0.000000" rest="1.000000"/>
    </dof>
    <dof description="twistR_forearmJoint">
      <position x="-2.093631" y="-0.095273" z="1.241181"/>
      <axis x="-0.076841" y="-0.305432" z="-0.949108"/>
      <range min="-1.570796" max="1.570796" rest="0.500000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_forearmMesh" type = "obj"/>
  </node>
<node>
  <joint description="R_handJoint" type="biaxial">
    <dof description="adductR_handJoint">
      <position x="-2.286975" y="-0.863793" z="-1.146941"/>
      <axis x="0.106261" y="0.965602" z="-0.237322"/>
      <range min="-1.396263" max="1.221730" rest="0.533333"/>
    </dof>
    <dof description="flexR_handJoint">
      <position x="-2.286975" y="-0.863793" z="-1.146941"/>
      <axis x="0.990874" y="-0.082925" z="0.106261"/>
      <range min="-0.349066" max="0.523599" rest="0.400000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_handMesh" type = "obj"/>
  </node>
<node>
  <joint description="R_indexf_baseJoint" type="biaxial">
    <dof description="flexR_indexf_baseJoint">
      <position x="-2.272439" y="-1.244422" z="-1.835075"/>
      <axis x="0.185913" y="0.918014" z="-0.350265"/>
      <range min="-1.570796" max="0.523599" rest="0.750000"/>
    </dof>
  </joint>
</node>

```

```

<dof description="adductR_indexf_baseJoint">
  <position x="-2.272439" y="-1.244422" z="-1.835075"/>
  <axis x="-0.958213" y="0.090520" z="-0.271354"/>
  <range min="-0.523599" max="0.174533" rest="0.750000"/>
</dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_indexf_baseMesh" type = "obj"/>
</node>
<node>
<joint description="R_indexf_midJoint" type="uniaxial">
  <dof description="flexR_indexf_midJoint">
    <position x="-2.213129" y="-1.349748" z="-2.079645"/>
    <axis x="0.247729" y="0.935896" z="-0.250458"/>
    <range min="-1.483530" max="0.261799" rest="0.850000"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_indexf_midMesh" type = "obj"/>
</node>
<node>
<joint description="R_indexf_tipJoint" type="uniaxial">
  <dof description="flexR_indexf_tipJoint">
    <position x="-2.120706" y="-1.418202" z="-2.244024"/>
    <axis x="0.259721" y="0.931165" z="-0.255883"/>
    <range min="-1.745329" max="0.261799" rest="0.869565"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_indexf_tipMesh" type = "obj"/>
</node>
</node>
</node>
</node>
<node>
<joint description="R_ringf_baseJoint" type="biaxial">
  <dof description="flexR_ringf_baseJoint">
    <position x="-2.252250" y="-0.871156" z="-1.839935"/>
    <axis x="0.232803" y="0.967088" z="0.102685"/>
    <range min="-1.570796" max="0.523599" rest="0.750000"/>
  </dof>
  <dof description="adductR_ringf_baseJoint">
    <position x="-2.252250" y="-0.871156" z="-1.839935"/>
    <axis x="-0.966886" y="0.241513" z="-0.082484"/>
    <range min="-0.261799" max="0.436332" rest="0.375000"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_ringf_baseMesh" type = "obj"/>
</node>
<node>
<joint description="R_ringf_midJoint" type="uniaxial">
  <dof description="flexR_ringf_midJoint">
    <position x="-2.220718" y="-0.847009" z="-2.138847"/>
    <axis x="0.183721" y="0.974110" z="0.131743"/>
    <range min="-1.483530" max="0.261799" rest="0.850000"/>
  </dof>
</joint>

```

```

<node>
  <meshobject filename="man_hands7c.obj" description="R_ringf_midMesh" type = "obj"/>
</node>
<node>
  <joint description="R_ringf_tipjoint" type="uniaxial">
    <dof description="flexR_ringf_tipjoint">
      <position x="-2.133136" y="-0.835251" z="-2.347921"/>
      <axis x="0.224699" y="0.961504" z="0.158175"/>
      <range min="-1.745329" max="0.261799" rest="0.869565"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_ringf_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
<node>
  <joint description="R_thumb_basejoint" type="polyaxial">
    <dof description="adductR_thumb_basejoint">
      <position x="-2.287868" y="-0.942916" z="-1.179148"/>
      <axis x="0.980898" y="0.191221" z="0.035682"/>
      <range min="-0.349066" max="1.221730" rest="0.222222"/>
    </dof>
    <dof description="flexR_thumb_basejoint">
      <position x="-2.287868" y="-0.942916" z="-1.179148"/>
      <axis x="0.068483" y="-0.511164" z="0.856751"/>
      <range min="-0.174533" max="0.698132" rest="0.200000"/>
    </dof>
    <dof description="twistR_thumb_basejoint">
      <position x="-2.287868" y="-0.942916" z="-1.179148"/>
      <axis x="0.182068" y="-0.837942" z="-0.514495"/>
      <range min="-1.570796" max="1.570796" rest="0.500000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_thumb_baseMesh" type = "obj"/>
  </node>
</node>
<node>
  <joint description="R_thumb_midjoint" type="uniaxial">
    <dof description="flexR_thumb_midjoint">
      <position x="-2.202965" y="-1.333668" z="-1.419070"/>
      <axis x="0.922083" y="-0.100481" z="0.373720"/>
      <range min="-0.174533" max="0.872665" rest="0.166667"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_thumb_midMesh" type = "obj"/>
  </node>
</node>
<node>
  <joint description="R_thumb_tipjoint" type="uniaxial">
    <dof description="flexR_thumb_tipjoint">
      <position x="-2.134612" y="-1.517349" z="-1.637105"/>
      <axis x="0.944999" y="0.088538" z="0.314861"/>
      <range min="-0.349066" max="1.570796" rest="0.181818"/>
    </dof>
  </joint>
</node>

```

```

    <meshobject filename="man_hands7c.obj" description="R_thumb_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
<node>
  <joint description="R_littlef_baseJoint" type="biaxial">
    <dof description="flexR_littlef_baseJoint">
      <position x="-2.253239" y="-0.714117" z="-1.749564"/>
      <axis x="0.167601" y="0.948972" z="0.267135"/>
      <range min="-1.570796" max="0.523599" rest="0.750000"/>
    </dof>
    <dof description="adductR_littlef_baseJoint">
      <position x="-2.253239" y="-0.714117" z="-1.749564"/>
      <axis x="-0.961671" y="0.217022" z="-0.167599"/>
      <range min="-0.174533" max="0.523599" rest="0.250000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="R_littlef_baseMesh" type = "obj"/>
</node>
<node>
  <joint description="R_littlef_midJoint" type="uniaxial">
    <dof description="flexR_littlef_midJoint">
      <position x="-2.202066" y="-0.660166" z="-1.973328"/>
      <axis x="0.012870" y="0.978036" z="0.208040"/>
      <range min="-1.483530" max="0.261799" rest="0.850000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="R_littlef_midMesh" type = "obj"/>
</node>
<node>
  <joint description="R_littlef_tipJoint" type="uniaxial">
    <dof description="flexR_littlef_tipJoint">
      <position x="-2.127818" y="-0.624687" z="-2.144716"/>
      <axis x="0.063981" y="0.945029" z="0.320667"/>
      <range min="-1.745329" max="0.261799" rest="0.869565"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="R_littlef_tipMesh" type = "obj"/>
</node>
</node>
</node>
</node>
<node>
  <joint description="R_middlef_baseJoint" type="biaxial">
    <dof description="flexR_middlef_baseJoint">
      <position x="-2.269276" y="-1.052933" z="-1.888912"/>
      <axis x="0.225511" y="0.962420" z="-0.151300"/>
      <range min="-1.570796" max="0.523599" rest="0.750000"/>
    </dof>
    <dof description="adductR_middlef_baseJoint">
      <position x="-2.269276" y="-1.052933" z="-1.888912"/>
      <axis x="-0.960937" y="0.194156" z="-0.197238"/>
      <range min="-0.436332" max="0.261799" rest="0.625000"/>
    </dof>
  </joint>

```

```

</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="R_middlef_baseMesh" type = "obj"/>
</node>
<node>
  <joint description="R_middlef_midJoint" type="uniaxial">
    <dof description="flexR_middlef_midJoint">
      <position x="-2.218334" y="-1.113216" z="-2.196445"/>
      <axis x="0.211615" y="0.974679" z="-0.072249"/>
      <range min="-1.483530" max="0.261799" rest="0.850000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_middlef_midMesh" type = "obj"/>
  </node>
<node>
  <joint description="R_middlef_tipJoint" type="uniaxial">
    <dof description="flexR_middlef_tipJoint">
      <position x="-2.134881" y="-1.146823" z="-2.405392"/>
      <axis x="0.204120" y="0.974447" z="-0.093748"/>
      <range min="-1.745329" max="0.261799" rest="0.869565"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="R_middlef_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
</node>
</node>
</node>
<node>
  <joint description="L_armJoint" type="polyaxial">
    <dof description="flexL_armJoint">
      <position x="1.463742" y="0.100839" z="4.023920"/>
      <axis x="0.973471" y="-0.047957" z="0.223728"/>
      <range min="-3.141593" max="0.785398" rest="0.800000"/>
    </dof>
    <dof description="adductL_armJoint">
      <position x="1.463742" y="0.100839" z="4.023920"/>
      <axis x="-0.062003" y="-0.996493" z="0.056184"/>
      <range min="-0.698132" max="1.745329" rest="0.285714"/>
    </dof>
    <dof description="twistL_armJoint">
      <position x="1.463742" y="0.100839" z="4.023920"/>
      <axis x="-0.220249" y="0.068566" z="0.973031"/>
      <range min="-0.959931" max="0.785398" rest="0.550000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_armMesh" type = "obj"/>
  </node>
<node>
  <joint description="L_forearmJoint" type="biaxial">
    <dof description="flexL_forearmJoint">
      <position x="2.093629" y="-0.095250" z="1.241167"/>
      <axis x="0.991498" y="-0.076846" z="0.105005"/>
    </dof>
  </joint>
</node>

```

```

    <range min="-2.356194" max="0.000000" rest="1.000000"/>
  </dof>
  <dof description="twistL_forearmJoint">
    <position x="2.093629" y="-0.095250" z="1.241167"/>
    <axis x="-0.076843" y="0.305442" z="0.949105"/>
    <range min="-1.570796" max="1.570796" rest="0.500000"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_forearmMesh" type = "obj"/>
</node>
<node>
  <joint description="L_handjoint" type="biaxial">
    <dof description="adductL_handjoint">
      <position x="2.286978" y="-0.863792" z="-1.146941"/>
      <axis x="0.106207" y="-0.965655" z="0.237131"/>
      <range min="-1.396263" max="1.221730" rest="0.533333"/>
    </dof>
    <dof description="flexL_handjoint">
      <position x="2.286978" y="-0.863792" z="-1.146941"/>
      <axis x="0.990882" y="0.082901" z="-0.106207"/>
      <range min="-0.349066" max="0.523599" rest="0.400000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="L_handMesh" type = "obj"/>
</node>
<node>
  <joint description="L_littlef_baseJoint" type="biaxial">
    <dof description="flexL_littlef_baseJoint">
      <position x="2.253232" y="-0.714111" z="-1.749566"/>
      <axis x="0.167599" y="-0.948973" z="-0.267135"/>
      <range min="-1.570796" max="0.523599" rest="0.750000"/>
    </dof>
    <dof description="adductL_littlef_baseJoint">
      <position x="2.253232" y="-0.714111" z="-1.749566"/>
      <axis x="-0.961672" y="-0.217021" z="0.167600"/>
      <range min="-0.174533" max="0.523599" rest="0.250000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="L_littlef_baseMesh" type = "obj"/>
</node>
<node>
  <joint description="L_littlef_midJoint" type="uniaxial">
    <dof description="flexL_littlef_midjoint">
      <position x="2.202060" y="-0.660160" z="-1.973330"/>
      <axis x="0.012846" y="-0.978024" z="-0.208097"/>
      <range min="-1.483530" max="0.261799" rest="0.850000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="L_littlef_midMesh" type = "obj"/>
</node>
<node>
  <joint description="L_littlef_tipjoint" type="uniaxial">
    <dof description="flexL_littlef_tipjoint">
      <position x="2.127810" y="-0.624668" z="-2.144721"/>

```



```

    <axis x="0.063967" y="-0.945014" z="-0.320714"/>
    <range min="-1.745329" max="0.261799" rest="0.869565"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_littlef_tipMesh" type = "obj"/>
</node>
</node>
</node>
<node>
  <joint description="L_indexf_baseJoint" type="biaxial">
  <dof description="flexL_indexf_basejoint">
    <position x="2.272436" y="-1.244391" z="-1.835099"/>
    <axis x="0.185964" y="-0.917971" z="0.350353"/>
    <range min="-1.570796" max="0.523599" rest="0.750000"/>
  </dof>
  <dof description="adductL_indexf_basejoint">
    <position x="2.272436" y="-1.244391" z="-1.835099"/>
    <axis x="-0.958201" y="-0.090534" z="0.271392"/>
    <range min="-0.523599" max="0.174533" rest="0.750000"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_indexf_baseMesh" type = "obj"/>
</node>
<node>
  <joint description="L_indexf_midJoint" type="uniaxial">
  <dof description="flexL_indexf_midjoint">
    <position x="2.213127" y="-1.349740" z="-2.079647"/>
    <axis x="0.247737" y="-0.935892" z="0.250466"/>
    <range min="-1.483530" max="0.261799" rest="0.850000"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_indexf_midMesh" type = "obj"/>
</node>
<node>
  <joint description="L_indexf_tipJoint" type="uniaxial">
  <dof description="flexL_indexf_tipjoint">
    <position x="2.120704" y="-1.418196" z="-2.244025"/>
    <axis x="0.259721" y="-0.931165" z="0.255884"/>
    <range min="-1.745329" max="0.261799" rest="0.869565"/>
  </dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_indexf_tipMesh" type = "obj"/>
</node>
</node>
</node>
</node>
<node>
  <joint description="L_ringf_baseJoint" type="biaxial">
  <dof description="flexL_ringf_basejoint">
    <position x="2.252244" y="-0.871155" z="-1.839934"/>
    <axis x="0.232799" y="-0.967084" z="-0.102732"/>
    <range min="-1.570796" max="0.523599" rest="0.750000"/>
  </dof>

```

```

<dof description="adductL_ringf_baseJoint">
  <position x="2.252244" y="-0.871155" z="-1.839934"/>
  <axis x="-0.966887" y="-0.241513" z="0.082473"/>
  <range min="-0.261799" max="0.436332" rest="0.375000"/>
</dof>
</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_ringf_baseMesh" type = "obj"/>
</node>
<node>
  <joint description="L_ringf_midJoint" type="uniaxial">
    <dof description="flexL_ringf_midJoint">
      <position x="2.220711" y="-0.846992" z="-2.138856"/>
      <axis x="0.183741" y="-0.974109" z="-0.131721"/>
      <range min="-1.483530" max="0.261799" rest="0.850000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_ringf_midMesh" type = "obj"/>
  </node>
</node>
<node>
  <joint description="L_ringf_tipJoint" type="uniaxial">
    <dof description="flexL_ringf_tipJoint">
      <position x="2.133129" y="-0.835241" z="-2.347924"/>
      <axis x="0.224698" y="-0.961504" z="-0.158179"/>
      <range min="-1.745329" max="0.261799" rest="0.869565"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_ringf_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
<node>
  <joint description="L_middlef_baseJoint" type="biaxial">
    <dof description="flexL_middlef_baseJoint">
      <position x="2.269272" y="-1.052901" z="-1.888938"/>
      <axis x="0.225511" y="-0.962417" z="0.151321"/>
      <range min="-1.570796" max="0.523599" rest="0.750000"/>
    </dof>
    <dof description="adductL_middlef_baseJoint">
      <position x="2.269272" y="-1.052901" z="-1.888938"/>
      <axis x="-0.960938" y="-0.194152" z="0.197238"/>
      <range min="-0.436332" max="0.261799" rest="0.625000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_middlef_baseMesh" type = "obj"/>
  </node>
</node>
<node>
  <joint description="L_middlef_midJoint" type="uniaxial">
    <dof description="flexL_middlef_midJoint">
      <position x="2.218330" y="1.113192" z="-2.196475"/>
      <axis x="0.211693" y="0.974658" z="0.072302"/>
      <range min="-1.483530" max="0.261799" rest="0.850000"/>
    </dof>
  </joint>

```

```

</node>
  <meshobject filename="man_hands7c.obj" description="L_middlef_midMesh" type = "obj"/>
</node>
<node>
  <joint description="L_middlef_tipJoint" type="uniaxial">
    <dof description="flexL_middlef_tipJoint">
      <position x="2.134877" y="-1.146816" z="-2.405395"/>
      <axis x="0.204121" y="-0.974447" z="0.093748"/>
      <range min="-1.745329" max="0.261799" rest="0.869565"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_middlef_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
<node>
  <joint description="L_thumb_baseJoint" type="polyaxial">
    <dof description="adductL_thumb_baseJoint">
      <position x="2.287859" y="-0.942883" z="-1.179171"/>
      <axis x="0.980896" y="-0.191238" z="-0.035649"/>
      <range min="-0.349066" max="1.221730" rest="0.222222"/>
    </dof>
    <dof description="flexL_thumb_baseJoint">
      <position x="2.287859" y="-0.942883" z="-1.179171"/>
      <axis x="0.068519" y="0.511154" z="-0.856754"/>
      <range min="-0.174533" max="0.698132" rest="0.200000"/>
    </dof>
    <dof description="twistL_thumb_baseJoint">
      <position x="2.287859" y="-0.942883" z="-1.179171"/>
      <axis x="0.182066" y="0.837944" z="0.514492"/>
      <range min="-1.570796" max="1.570796" rest="0.500000"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_thumb_baseMesh" type = "obj"/>
  </node>
<node>
  <joint description="L_thumb_midJoint" type="uniaxial">
    <dof description="flexL_thumb_midJoint">
      <position x="2.202957" y="-1.333639" z="-1.419093"/>
      <axis x="0.922068" y="0.100495" z="-0.373752"/>
      <range min="-0.174533" max="0.872665" rest="0.166667"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_thumb_midMesh" type = "obj"/>
  </node>
<node>
  <joint description="L_thumb_tipJoint" type="uniaxial">
    <dof description="flexL_thumb_tipJoint">
      <position x="2.134597" y="-1.517324" z="-1.637131"/>
      <axis x="0.945002" y="-0.088540" z="-0.314852"/>
      <range min="-0.349066" max="1.570796" rest="0.181818"/>
    </dof>
  </joint>
</node>

```

```

    <meshobject filename="man_hands7c.obj" description="L_thumb_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
</node>
</node>
</node>
<node>
  <joint description="headJoint" type="polyaxial">
    <dof description="adductheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="0.000112" y="-0.998992" z="-0.044886"/>
      <range min="-0.785398" max="0.785398" rest="0.500000"/>
    </dof>
    <dof description="flexheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="1.000000" y="0.000107" z="0.000100"/>
      <range min="-1.396263" max="1.134464" rest="0.551724"/>
    </dof>
    <dof description="twistheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="-0.000095" y="-0.044886" z="0.998992"/>
      <range min="-1.570796" max="1.570796" rest="0.500000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="headMesh" type = "obj"/>
</node>
</node>
</node>
</node>
<node>
  <joint description="R_thighJoint" type="polyaxial">
    <dof description="flexR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="0.999999" y="0.000895" z="-0.000894"/>
      <range min="-2.181662" max="0.523599" rest="0.806452"/>
    </dof>
    <dof description="adductR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="-0.000895" y="1.000000" z="-0.000135"/>
      <range min="-0.436332" max="0.785398" rest="0.357143"/>
    </dof>
    <dof description="twistR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="-0.000894" y="-0.000135" z="-1.000000"/>
      <range min="-0.610865" max="0.610865" rest="0.500000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="R_thighMesh" type = "obj"/>
</node>
</node>
<node>
  <joint description="R_legJoint" type="uniaxial">
    <dof description="flexR_legJoint">
      <position x="-0.835772" y="0.154783" z="-5.285148"/>
      <axis x="0.999068" y="-0.039214" z="0.018057"/>
    </dof>
  </joint>
</node>

```

```

    <meshobject filename="man_hands7c.obj" description="L_thumb_tipMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
</node>
</node>
</node>
<node>
  <joint description="headJoint" type="polyaxial">
    <dof description="adductheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="0.000112" y="-0.998992" z="-0.044886"/>
      <range min="-0.785398" max="0.785398" rest="0.500000"/>
    </dof>
    <dof description="flexheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="1.000000" y="0.000107" z="0.000100"/>
      <range min="-1.396263" max="1.134464" rest="0.551724"/>
    </dof>
    <dof description="twistheadJoint">
      <position x="-0.001331" y="0.102096" z="4.879372"/>
      <axis x="-0.000095" y="-0.044886" z="0.998992"/>
      <range min="-1.570796" max="1.570796" rest="0.500000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="headMesh" type = "obj"/>
</node>
</node>
</node>
</node>
<node>
  <joint description="R_thighJoint" type="polyaxial">
    <dof description="flexR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="0.999999" y="0.000895" z="-0.000894"/>
      <range min="-2.181662" max="0.523599" rest="0.806452"/>
    </dof>
    <dof description="adductR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="-0.000895" y="1.000000" z="-0.000135"/>
      <range min="-0.436332" max="0.785398" rest="0.357143"/>
    </dof>
    <dof description="twistR_thighJoint">
      <position x="-0.831825" y="0.155381" z="-0.870014"/>
      <axis x="-0.000894" y="-0.000135" z="-1.000000"/>
      <range min="-0.610865" max="0.610865" rest="0.500000"/>
    </dof>
  </joint>
</node>
  <meshobject filename="man_hands7c.obj" description="R_thighMesh" type = "obj"/>
</node>
</node>
<node>
  <joint description="R_legJoint" type="uniaxial">
    <dof description="flexR_legJoint">
      <position x="-0.835772" y="0.154783" z="-5.285148"/>
      <axis x="0.999068" y="-0.039214" z="0.018057"/>
    </dof>
  </joint>
</node>

```

```

</joint>
<node>
  <meshobject filename="man_hands7c.obj" description="L_legMesh" type = "obj"/>
</node>
<node>
  <joint description="L_footJoint" type="polyaxial">
    <dof description="flexL_footJoint">
      <position x="0.752662" y="0.437652" z="-9.270053"/>
      <axis x="0.999998" y="-0.000081" z="0.001954"/>
      <range min="-0.872665" max="0.349066" rest="0.714286"/>
    </dof>
    <dof description="twistL_footJoint">
      <position x="0.752662" y="0.437652" z="-9.270053"/>
      <axis x="-0.000081" y="0.996530" z="0.083230"/>
      <range min="-0.349066" max="0.349066" rest="0.500000"/>
    </dof>
    <dof description="adductL_footJoint">
      <position x="0.752662" y="0.437652" z="-9.270053"/>
      <axis x="-0.001954" y="-0.083230" z="0.996528"/>
      <range min="-0.349066" max="0.174533" rest="0.666667"/>
    </dof>
  </joint>
  <node>
    <meshobject filename="man_hands7c.obj" description="L_footMesh" type = "obj"/>
  </node>
</node>
</node>
</node>
</node>
</scene>

```


APÊNDICE C XML DO GESTO OBRIGADO

```

<?xml version="1.0"?>
<!DOCTYPE action SYSTEM "vartAnimation.dtd">
<action action_name="obrigado_ex_1" speed="1" cycle="false">
  <joint_movement joint_name="hipjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.5">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="bellyjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.190476">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.5">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.5">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="chestjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.75">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="R_forearmjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.160782">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.757796">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="R_armjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.488116">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.501694">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.444022">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="R_handjoint" duration="0.5">
    <interpolation type="ease-in_ease-out"/>
    <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.597463">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
    <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.467994">
      <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
    </dof_movement>
  </joint_movement>
  <joint_movement joint_name="headjoint" duration="0.5">

```



```

<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.573999">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.627723">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_littlef_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.75">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.805997">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_littlef_midJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.714002">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_littlef_tipJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.667568">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_ringf_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.75">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.696998">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_ringf_midJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.634003">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_ringf_tipJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.659568">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="R_middlef_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.75">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.585001">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>

```

```

</joint_movement>
<joint_movement joint_name="R_middlef_tipJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.645568">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_middlef_midJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.606003">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_indexf_baseJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.75">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.290301">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_indexf_midJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.628003">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_indexf_tipJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.637568">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_thumb_baseJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.292222">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.2">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.5">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_thumb_tipJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.425819">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_thumb_midJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.270667">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>

```

```

<joint_movement joint_name="R_thighJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.806452">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.357143">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.5">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_thighJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.806452">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.357143">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.5">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_legJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_legJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="R_footJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.714286">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.5">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.666667">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_footJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.714286">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.5">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.666667">
    <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>

```

```

</dof_movement>
</joint_movement>
<joint_movement joint_name="L_armJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.831994"/>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.361697"/>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.250063"/>
</joint_movement>
<joint_movement joint_name="L_forearmJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.310074"/>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.5"/>
</joint_movement>
<joint_movement joint_name="L_handJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.533333"/>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.4"/>
</joint_movement>
<joint_movement joint_name="L_thumb_baseJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.292222">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.2">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
  <dof_movement dofID="TWIST" initialTime="0" finalTime="1" finalPosition="0.5">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_thumb_tipJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.425819">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_thumb_midJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.270667">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_littlef_midJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.714002">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_littlef_tipJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.667568">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>
</joint_movement>
<joint_movement joint_name="L_littlef_baseJoint" duration="0.5">
  <interpolation type="ease-in_ease-out"/>
  <dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.648001">
  <error overshoot="0.0" offset="0.0" peak_time="1.0"/>
  </dof_movement>

```

```

</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.41">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_ringf_midJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.634003">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_ringf_tipJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.659568">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_ringf_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.564002">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.447">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_middlef_midJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.606003">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_middlef_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.506003">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.603">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_middlef_tipJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.645568">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_indexf_baseJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.526003">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
<dof_movement dofID="ADDUCTION" initialTime="0" finalTime="1" finalPosition="0.586002">
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>
</dof_movement>
</joint_movement>
<joint_movement joint_name="L_indexf_midJoint" duration="0.5">
<interpolation type="ease-in_ease-out"/>

```

```
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.628003">  
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>  
</dof_movement>  
</joint_movement>  
<joint_movement joint_name="L_indexf_tipjoint" duration="0.5">  
<interpolation type="ease-in_ease-out"/>  
<dof_movement dofID="FLEXION" initialTime="0" finalTime="1" finalPosition="0.637568">  
<error overshoot="0.0" offset="0.0" peak_time="1.0"/>  
</dof_movement>  
</joint_movement>  
</action>
```


APÊNDICE D DTD DA CENA

A seguir tem-se a DTD criada para descrever cenas no V-ART.

```

<!-- DTD for a VART scene -->
<!-- Author: Andreia Schneider, Feb 2006 -->

<!ELEMENT scene (camera*, node+)>
<!ELEMENT camera (position,target,up_vector)>
<!ELEMENT position EMPTY>
<!ELEMENT target EMPTY>
<!ELEMENT up_vector EMPTY>
<!ELEMENT node ((bezier|sphere|cylinder|meshobject|directionallight|spotlight|pointlight|transform|joint),
node*)>
<!ELEMENT bezier (ponto1, ponto2, ponto3, ponto4)>
<!ELEMENT ponto1 EMPTY>
<!ELEMENT ponto2 EMPTY>
<!ELEMENT ponto3 EMPTY>
<!ELEMENT ponto4 EMPTY>
<!ELEMENT sphere (radius, material)>
<!ELEMENT cylinder (radius, height, material)>
<!ELEMENT height EMPTY>
<!ELEMENT radius EMPTY>
<!ELEMENT material EMPTY>
<!ELEMENT meshobject (material?)>
<!ELEMENT directionallight (intensity, ambientIntensity, color, enabled, position)>
<!ELEMENT pointlight (intensity, ambientIntensity, color, enabled, position, attenuation)>
<!ELEMENT spotlight (intensity, ambientIntensity, color, enabled, position, attenuation, beam_Width,
cut_Off_Angle)>
<!ELEMENT intensity EMPTY>
<!ELEMENT ambientIntensity EMPTY>
<!ELEMENT color EMPTY>
<!ELEMENT enabled EMPTY>
<!ELEMENT attenuation EMPTY>
<!ELEMENT beam_Width EMPTY>
<!ELEMENT cut_Off_Angle EMPTY>

```



```

<!ELEMENT transform (translation|scale|rotation|matrix)>
<!ELEMENT translation EMPTY>
<!ELEMENT scale EMPTY>
<!ELEMENT rotation EMPTY>
<!ELEMENT matrix EMPTY>
<!ELEMENT joint (dof+)>
<!ELEMENT dof ( position, axis, range )>
<!ELEMENT axis EMPTY>
<!ELEMENT range EMPTY>
<!ATTLIST scene description CDATA #REQUIRED>
<!ATTLIST camera description CDATA #REQUIRED
  type CDATA #REQUIRED>
<!ATTLIST joint description CDATA #REQUIRED
  type (biaxial|polyaxial|uniaxial) #REQUIRED >
<!ATTLIST dof description CDATA #REQUIRED >
<!ATTLIST position x CDATA #REQUIRED
  y CDATA #REQUIRED
  z CDATA #REQUIRED >
<!ATTLIST target x CDATA #REQUIRED
  y CDATA #REQUIRED
  z CDATA #REQUIRED >
<!ATTLIST up_vector x CDATA #REQUIRED
  y CDATA #REQUIRED
  z CDATA #REQUIRED >
<!ATTLIST translation x CDATA #REQUIRED
  y CDATA #REQUIRED
  z CDATA #REQUIRED >
<!ATTLIST scale x CDATA #REQUIRED
  y CDATA #REQUIRED
  z CDATA #REQUIRED >
<!ATTLIST rotation axis CDATA #REQUIRED
  radians CDATA #REQUIRED>
<!ATTLIST matrix m00 CDATA #REQUIRED
  m01 CDATA #REQUIRED
  m02 CDATA #REQUIRED
  m03 CDATA #REQUIRED
  m10 CDATA #REQUIRED
  m11 CDATA #REQUIRED

```

```

m12 CDATA #REQUIRED
m13 CDATA #REQUIRED
m20 CDATA #REQUIRED
m21 CDATA #REQUIRED
m22 CDATA #REQUIRED
m23 CDATA #REQUIRED
m30 CDATA #REQUIRED
m31 CDATA #REQUIRED
m32 CDATA #REQUIRED
m33 CDATA #REQUIRED>

<!ATTLIST bezier description CDATA #REQUIRED >

<!ATTLIST ponto1 x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED >

<!ATTLIST ponto2 x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED >

<!ATTLIST ponto3 x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED >

<!ATTLIST ponto4 x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED >

<!ATTLIST sphere description CDATA #REQUIRED >

<!ATTLIST cylinder description CDATA #REQUIRED >

<!ATTLIST radius value CDATA #REQUIRED >

<!ATTLIST height value CDATA #REQUIRED >

<!ATTLIST material r CDATA #REQUIRED
g CDATA #REQUIRED
b CDATA #REQUIRED >

<!ATTLIST meshobject filename CDATA #REQUIRED
type CDATA #REQUIRED
description CDATA #REQUIRED>

<!ATTLIST transform description CDATA #REQUIRED>

<!ATTLIST axis x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED >

<!ATTLIST range min CDATA #REQUIRED
max CDATA #REQUIRED
rest CDATA #REQUIRED>

<!ATTLIST directionallight description CDATA #REQUIRED>

<!ATTLIST spotlight description CDATA #REQUIRED>

```

```
<!ATTLIST pointlight description CDATA #REQUIRED>
<!ATTLIST intensity value CDATA #REQUIRED>
<!ATTLIST ambientIntensity value CDATA #REQUIRED>
<!ATTLIST color red CDATA #REQUIRED
green CDATA #REQUIRED
blue CDATA #REQUIRED
alpha CDATA #REQUIRED>
<!ATTLIST enabled value CDATA #REQUIRED>
<!ATTLIST attenuation x CDATA #REQUIRED
y CDATA #REQUIRED
z CDATA #REQUIRED>
<!ATTLIST beam_Width value CDATA #REQUIRED>
<!ATTLIST cut_Off_Angle value CDATA #REQUIRED>
```


APÊNDICE E DTD DA ANIMAÇÃO

A seguir tem-se a DTD criada para descrever ações no V-ART.

```

<!-- DTD for a VART animation -->
<!-- Author: Andreia Schneider, Feb 2006 -->

<!ELEMENT action ( joint_movement+ ) >

<!ATTLIST action action_name CDATA #REQUIRED
                speed CDATA #REQUIRED
                cycle CDATA #REQUIRED>

<!ELEMENT joint_movement (interpolation, dof_movement+)>
<!ATTLIST joint_movement joint_name CDATA #REQUIRED
                duration CDATA #REQUIRED>

<!ELEMENT interpolation (noise?, persistency?, overshoot?, offset?)>
<!ATTLIST interpolation type CDATA #REQUIRED>

<!ELEMENT noise EMPTY>
<!ATTLIST noise value CDATA #REQUIRED>

<!ELEMENT persistency EMPTY>
<!ATTLIST persistency value CDATA #REQUIRED>

<!ELEMENT overshoot EMPTY>
<!ATTLIST overshoot value CDATA #REQUIRED>

<!ELEMENT offset EMPTY>
<!ATTLIST offset value CDATA #REQUIRED>

<!ELEMENT dof_movement EMPTY> <!-- several dof_movements for the same dof may exist -->
<!ATTLIST dof_movement dofID CDATA #REQUIRED
                initialTime CDATA #REQUIRED
                finalTime CDATA #REQUIRED
                finalPosition CDATA #REQUIRED>

```


APÊNDICE F O MÉTODO CCD

F.1 Definição

CCD (*Cyclic coordinate descent*) é um método iterativo de cinemática inversa no qual se move um DOF da cadeia cinemática por vez procurando minimizar a função objetiva desenvolvida em F.2.

F.2 Determinação da função objetiva

Seja a posição corrente do *end-effector* e a orientação (cujas componentes são vetores ortonormais definidores dos eixos de rotação do *end-effector*) respectivamente definidos pelas funções F.1 e F.2 abaixo.

$$P_c = (x_c; y_c; z_c) \quad (\text{F.1})$$

$$O_c = \begin{bmatrix} u_{1c} \\ u_{2c} \\ u_{3c} \end{bmatrix} \quad (\text{F.2})$$

Este *end-effector* deve ser levado para uma posição e orientação mais próxima possível do alvo. Deve existir então um vetor q que minimize o erro da equação F.3.

$$E(q) = E_p(q) + E_o(q) \quad (\text{F.3})$$

A equação F.3 nada mais é que a soma do erro de posição visto na equação F.4 e orientação (equação F.5) definidas abaixo.

$$E_p(q) = \|P_d - P_c\|^2 \quad (\text{F.4})$$

$$E_o(q) = \sum_{j=1}^3 ((u_{jd} * u_{jc}) - 1)^2 \quad (\text{F.5})$$

A figura F.1 ilustra a situação da junta rotacional i em uma das iterações do processo.

Dados o vetor P_{ic} (vetor entre a posição da junta i e a posição corrente do *end-effector*) e o vetor P_{id} (entre a posição da junta i e a posição do alvo), pode-se rotacionar P_{ic} em torno de um eixo, com um ângulo θ . Este vetor de rotação é determinado por

$$P'_{ic}(\theta) = Rot_{eixo_i}(\theta)P_{ic} \quad (\text{F.6})$$

Enquanto θ varia, P'_{ic} descreve um semi-círculo centrado em J_i . O ponto neste círculo mais próximo do alvo é o ponto de intersecção entre o círculo e o vetor P_{id} . A idéia então

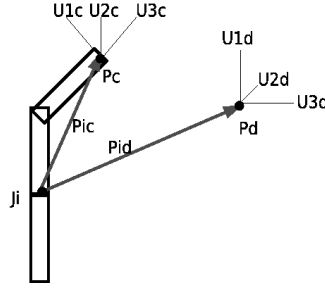


Figura F.1: ccd

é que $P'_{ic}(\theta)$ seja alinhado com P_{id} . Isto implica que devemos então maximizar a função

$$g_1(\theta) = P_{id} \cdot P'_{ic}(\theta) \quad (F.7)$$

Analogamente a função que deve ser maximizada para que seja encontrada a orientação do *end-effector* é

$$g_2(\theta) = \sum_{j=1}^3 u_{jd} \cdot u'_{jc}(\theta) \quad (F.8)$$

Unindo as duas equações, tem-se a função objetiva a ser maximizada dada por

$$g(\theta) = w_1 \cdot g_1(\theta) + w_2 \cdot g_2(\theta) \quad (F.9)$$

w_1 e w_2 são pesos dados às funções e indicam o quanto cada uma delas influencia na equação final.

De acordo com (WELMAN, 1993), esse valores são $w_1 = 1$ e $w_2 = \alpha(1 + \rho)$, onde α é um fator de escala que é inversamente proporcional à escala do mundo (W) ($\alpha = k/W$). Isso faz com que mesmo que o modelo do humano sofra transformação de escala, o algoritmo continue a ser válido. ρ por sua vez tem seu valor sugerido por (WANG; CHEN, 1991) como

$$\rho = \frac{\min(\|p_{id}\|, \|P_{ic}\|)}{\max(\|p_{id}\|, \|P_{ic}\|)} \quad (F.10)$$

Tem-se que uma rotação pode ser definida de acordo com a fórmula de rotação de Rodrigues, sendo u o eixo no qual o vetor v irá rotacionar.

$$v_{rot} = v \cdot \cos(\theta) + u \times v \cdot \sin(\theta) + \langle u, v \rangle \cdot u \cdot (1 - \cos(\theta)) \quad (F.11)$$

Substituindo os valores de P'_{ic} da equação F.7 de acordo com a equação F.11, tem-se que

$$p'_{ic} = p_{ic} \cdot \cos(\theta) + p_{ic} \times axis_i \cdot \sin(\theta) + (p_{ic} \cdot axis_i) \cdot axis_i \cdot (1 - \cos(\theta)) \quad (F.12)$$

Logo a equação F.7 será

$$g_1(\theta) = (w_p \cdot P_{id} \cdot P_{ic} \cdot \cos(\theta)) + (axis_i \cdot [w_p (p_{id} \times p_{ic})] \cdot \sin(\theta)) + ((p_{ic} \cdot axis_i) \cdot (p_{id} \cdot axis_i) \cdot (1 - \cos(\theta))) \quad (F.13)$$

g_2 é calculado da mesma forma e seu valor fica

$$g_2(\theta) = \left(w_o \cdot \sum_{j=1}^3 (u_{jd} \cdot axis_i)(u_{jc} \cdot axis_i)(1 - \cos(\theta)) + (\cos(\theta)) \cdot (w_o \cdot \sum_{j=1}^3 (u_{jd} \cdot u_{jc})) + \left(\sin(\theta) \cdot axis_i \cdot \left[w_o \cdot \sum_{j=1}^3 (u_{jd} X u_{jc}) \right] \right) \right) \quad (F.14)$$

A função objetiva pode ser representada pela junção e simplificação das equações F.13 e F.14

$$g(\theta) = k_1(1 - \cos(\theta)) + k_2 \cdot \cos(\theta) + k_3 \cdot \sin(\theta) \quad (F.15)$$

Onde

$$k_1 = w_p(p_{id} \cdot axis_i)(p_{ic} \cdot axis_i) + \sum_{j=1}^3 (u_{jd} \cdot axis_i)(u'_{jc} \cdot axis_i) \quad (F.16)$$

$$k_2 = w_p(p_{id} \cdot p_{ic}) + \sum_{j=1}^3 (u_{jd} \cdot u_{jc}) \quad (F.17)$$

$$k_3 = axis_i \cdot \left[w_p(p_{id} X p_{ic}) + \sum_{j=1}^3 (u_{jd} \cdot u_{jc}) \right] \quad (F.18)$$

Como queremos maximizar esta função, para o intervalo $-\phi \leq \theta \leq \phi$, é necessário que sua primeira derivada seja zero e a segunda derivada seja negativa. Assim, sua primeira derivada é

$$g'(\theta) = (k_1 - k_2) \sin(\theta) + k_3 \cos(\theta) \quad (F.19)$$

Como seu valor deve ser zero, tem-se que

$$\frac{\sin(\theta)}{\cos(\theta)} = \frac{k_3}{k_2 - k_1} \quad (F.20)$$

A equação F.20 também pode ser vista como

$$\tan(\theta) = \frac{k_3}{k_2 - k_1} \quad (F.21)$$

θ então será, no intervalo $-\phi/2 \leq \theta \leq \phi/2$

$$\theta = \tan^{-1} \left(\frac{k_3}{k_2 - k_1} \right) \quad (F.22)$$

Como a função tangente é periódica em π , existem dois candidatos à θ , além do anterior: $(\theta + \pi)$ e $(\theta - \pi)$. Todos os três candidatos devem então serem submetidos à segunda derivada da função objetiva. O que der resultado negativo é o valor procurado. Temos então que θ é o valor a ser adicionado na junta i , para sua movimentação, logo:

$$q_i = q_i + w_i \cdot \theta \quad (F.23)$$

Onde w_i é um coeficiente de rigidez da junta, o qual determina se ela terá mais o menos facilidade de convergir para o valor calculado.

APÊNDICE G QUESTIONÁRIO APLICADO NOS TESTES

Questionário

1. Nome:
2. Sexo:
3. Idade:
4. Conhecimento de LIBRAS: () Básico () Intermediário () Avançado
5. Das seguintes palavras conhece a representação em LIBRAS de:
 - () Amar
 - () Querer
 - () Obrigado
 - () Bom dia

Escreva a representação dos sinais que verá em sequência:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.

Escreva a representação dos sinais que verá em sequência:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.