

Um Sistema de Animação Baseado em Movimento Capturado

Fernando Wagner Serpa Vieira da Silva

6 de março de 1998

Sumário

1	Introdução	1
1.1	Um Pouco de História	1
1.2	Animação por Computador	3
1.3	Animação por Movimento Capturado	4
1.4	Organização da Tese	6
2	Modelos de Animação	8
2.1	Representação Geométrica dos Parâmetros da Animação	8
2.1.1	Partículas	9
2.1.2	Objetos Rígidos	9
2.1.3	Objetos Deformáveis	9
2.2	Níveis de Simulação	9
2.3	Exemplo de Simulação	10
2.4	Vínculos	12
3	Ator Virtual	15
3.1	Estruturas Articuladas	15
3.2	Modelos de Representação do Corpo Humano	16
3.2.1	O Modelo Utilizado no Sistema	18
3.3	Definição de Estruturas Articuladas	19
3.3.1	Notação DH	19
3.3.2	Notação APJ	20
3.3.3	Notação APJ Modificada	20
3.4	Representação Hierárquica	20
3.4.1	A Topologia do Ator Virtual do Sistema	22
3.5	Ferramentas Adicionais:	
	Forward e Inverse Kinematics	22
3.5.1	Forward Kinematics (FK)	22
3.5.2	Inverse Kinematics (IK)	23
4	Conceituação de Animação	24
4.1	Visão Geral	24
4.2	Síntese	26
4.3	Análise	26
4.4	Processamento	26
4.5	Reprodução	26

4.6	Captura	26
5	Sistemas de Síntese de Movimento	28
5.1	Keyframing	28
5.2	Animação Procedural	29
5.3	Simulação Dinâmica	30
5.4	Animação Comportamental	30
6	Sistemas de Captura de Movimento	33
6.1	Tecnologia	33
6.1.1	Sistemas Mecânicos	34
6.1.2	Sistemas Magnéticos	35
6.1.3	Sistemas Óticos	36
6.1.4	Sistemas Acústicos	39
6.2	Fonte Emissora vs. Sensores/Marcadores	39
6.2.1	Sistemas Inside-in	40
6.2.2	Sistemas Inside-out	40
6.2.3	Sistemas Outside-In	40
6.3	Aquisição Direta vs. Indireta	40
6.3.1	Sistemas de Aquisição Direta	40
6.3.2	Sistemas de Aquisição Indireta	41
7	Os Dados Capturados	42
7.1	Composição e Representação	42
7.2	Orientação Global vs. Orientação Relativa	43
7.2.1	Extração de Ângulos Relativos	44
7.3	Pós-processamento	48
8	Processamento de Movimentos	49
8.1	Classificação	49
8.2	Operações Básicas	50
8.2.1	Filtragem	50
8.2.2	Concatenação	51
8.2.3	Blending	53
8.3	Outras Operações	55
8.3.1	Warping	55
8.3.2	Filtragem em Multiresolução	56
9	Um Sistema de Animação Baseado em Movimento Capturado	58
9.1	Digressão Histórica: Sistemas de Animação Tradicionais	58
9.2	Um Novo Conceito de Sistema de Animação	59
9.3	Arquitetura do Sistema	60
9.3.1	Módulo de Entrada	60
9.3.2	Módulo de Processamento	61
9.3.3	Módulo de Saída	62
9.3.4	Módulo de Interface	62
9.3.5	Estruturas de Dados	63

9.4	Implementação	63
9.4.1	Operações de Movimento Disponíveis	64
9.4.2	Outras Ferramentas de Animação	64
9.4.3	Detalhes de Implementação	64
10	Interface com o Usuário	65
10.1	O Paradigma de Edição de Vídeo	65
10.2	Representação das Entidades na Interface	65
10.2.1	Ator Virtual	66
10.2.2	Movimentos	67
10.2.3	Operações	69
10.2.4	Outras Ferramentas de Animação	73
10.3	Controle Dinâmico dos Objetos da Interface	74
11	Resultados e Exemplos	76
11.1	Procedimentos Iniciais de Operação do Sistema	76
11.1.1	Definição do Esqueleto do Ator Virtual	76
11.1.2	Escolha dos Movimentos	77
11.1.3	Customização de Parâmetros Auxiliares	78
11.2	Definição dos Exemplos e Soluções Apresentadas	79
11.2.1	Exemplo 1 - Concatenação de Movimentos	79
11.2.2	Exemplo 2 - Alteração da Velocidade (ou Pitch) de um Movimento	80
11.2.3	Exemplo 3 - Warming de Curvas de Movimento	83
12	Conclusão	86
12.1	Conclusões	86
12.2	Trabalhos Futuros	87
	Referências Bibliográficas	89

Lista de Figuras

1.1	Discretização da variação angular do joelho de um performer.	4
2.1	Níveis de simulação de objetos representados no computador.	10
2.2	Configuração inicial do exemplo.	11
2.3	Animação com parâmetros cinemáticos.	11
2.4	Inclusão de parâmetros dinâmicos: gravidade, velocidade e deformação. . .	11
2.5	Situação com parâmetros comportamentais.	12
3.1	Braço robô com três graus de liberdade.	15
3.2	Primeiro estágio do processo de simplificação do corpo humano.	16
3.3	Aglutinação de ossos e representação por meio de um único segmento rígido.	17
3.4	Representação de articulações no computador.	17
3.5	Modelo de ator virtual utilizado no sistema.	18
3.6	Simplificação de articulação por redução de DOF.	19
3.7	Ruptura da estrutura causada por variações na posição de uma articulação.	21
3.8	Representação topológica de uma estrutura articulada hierárquica.	21
3.9	Árvore que representa a estrutura hierárquica do ator virtual no sistema. .	22
3.10	Forward vs. Inverse Kinematics.	23
4.1	O paradigma dos quatro universos e Captura de Movimento.	24
4.2	Etapas principais do processo de animação.	25
4.3	Diagrama completo com sub-etapas (retângulos) do processo de animação.	25
5.1	Interpolação do ângulo (esq.) e interpolação da posição final e inicial (dir.).	29
5.2	Definição de uma série de procedimentos para fazer a roda girar.	29
5.3	A bola caindo: exemplo do emprego de parâmetros dinâmicos na animação.	30
5.4	Papel do módulo comportamental no processo de simulação de peixes. . . .	32
6.1	Utilização de equipamentos mecânicos de captura no filme <i>Starship Troopers</i> .	34
6.2	Conjunto de potenciômetros no sistema de captura mecânico <i>Animatton</i> . .	35
6.3	Sistema magnético de captura <i>Motion Star</i>	36
6.4	Sistema ótico de captura de movimento.	37
7.1	O processo de captura: amostragem uniforme das curvas de movimento. . .	43
7.2	Ângulo relativo entre dois segmentos de uma estrutura articulada hierárquica.	44
7.3	Expansão da composição de transformações em uma estrutura articulada hierárquica.	45
7.4	Utilização do plano formado por três marcadores no cálculo da orientação do quadril.	46

7.5	Cálculo do ângulo relativo entre o quadril e a coxa.	46
7.6	Cálculo do ângulo relativo entre os segmentos <i>upleg</i> e <i>lowleg</i>	47
7.7	Utilização do plano formado por três marcadores no cálculo da orientação do peito.	47
8.1	Eliminação de altas frequências de um sinal utilizando um filtro passa-baixa.	51
8.2	Imagem original e modificada com filtro artístico.	51
8.3	Filtro de suavização aplicado à articulação do cotovelo.	52
8.4	Concatenação de imagens (replicação).	52
8.5	Concatenação: sinal original (em cima) e sinal concatenado (em baixo).	53
8.6	Concatenação com região de transição entre os dois sinais.	53
8.7	“Cross-Dissolve” de dois sinais: (a) sinal <i>f</i> , (b) sinal <i>g</i> e (c) sinal resultante <i>h</i>	54
8.8	Fusão de duas imagens.	54
8.9	Combinação de diferentes curvas de movimento em um mesmo conjunto de articulações.	55
8.10	Combinação de diferentes curvas de movimento em conjuntos diferentes de articulações.	55
8.11	Transformação projetiva de uma imagem: exemplo de warping.	56
8.12	Warping de uma curva de movimento.	56
8.13	Decomposição de uma imagem em bandas de frequência: (a) <i>lowpass pyramid</i> e (b) <i>bandpass pyramid</i>	57
9.1	<i>Screenshot</i> do sistema <i>MC Animator</i>	59
9.2	Fluxograma da arquitetura interna do sistema.	60
9.3	Fluxograma do módulo de entrada.	61
9.4	Fluxograma do módulo de processamento.	62
9.5	Fluxograma do módulo de saída.	63
10.1	<i>Skeleton Graph Window</i> : informações topológicas sobre o esqueleto articulado.	66
10.2	Representação geométrica do ator virtual.	67
10.3	Representação de um movimento na interface gráfica.	67
10.4	Representação de uma curva de movimento no sistema.	68
10.5	A janela <i>Joint Curves</i>	68
10.6	<i>Motion View Window</i> : ligação entre as representações do ator e do movimento.	69
10.7	<i>Motion ScratchPad</i> : conexão entre os módulos de entrada e processamento.	70
10.8	Janela da operação de filtragem.	70
10.9	Concatenação de movimentos (configuração inicial).	71
10.10	Intervalos de transição na concatenação de movimentos (detalhe).	71
10.11	Janela da operação de blending e <i>Time-Markers</i>	72
10.12	Janela da operação de warping.	72
10.13	Manipulação da estrutura articulada hierárquica utilizando Forward Kinematics.	73
10.14	Menu de controle interativo da câmera.	74
10.15	Janela de rendering e painel de controle interativo da animação.	74
10.16	Loop de controle dinâmico dos objetos da interface gráfica.	75

11.1	Tela de entrada do sistema <i>MC Animator</i>	77
11.2	Detalhe do painel principal do sistema.	77
11.3	Ator virtual em posição anatômica (posição inicial).	78
11.4	<i>ScratchPad</i> contendo os movimentos utilizados nos exemplos.	78
11.5	Janelas de customização de parâmetros de visualização e animação do sistema.	79
11.6	Concatenação de movimentos: disposição inicial e região de sobreposição.	80
11.7	Comparação dos sinais: movimentos originais e movimento resultante.	81
11.8	Seqüência com a seleção de alguns quadros da animação resultante (concatenação). A região de sobreposição ocorre entre os quadros 70 e 90.	82
11.9	Detalhe da barra de movimento com marca especial (\leftrightarrow), utilizada na reparametrização uniforme.	82
11.10	Resultado da reparametrização uniforme em uma curva do movimento “caminhar normal”.	82
11.11	Função de reparametrização definida pelo usuário.	83
11.12	Resultado da reparametrização adaptativa em uma curva do movimento “caminhar normal”, utilizando a função definida na figura anterior.	83
11.13	Inserção do movimento “caminhar normal” em um cenário virtual com obstáculo. A colisão da perna esquerda do ator virtual com o obstáculo ocorre no quadro 30.	84
11.14	Etapas do processo de warping de uma curva de movimento.	85
11.15	Seqüência com a seleção de alguns quadros da animação resultante (warping)	85
12.1	O sistema <i>MC Animator</i>	88

Capítulo 1

Introdução

O presente trabalho se enquadra no ramo da Computação Gráfica conhecido como *Animação por Computador*. A definição deste termo pode ser apresentada, em sua forma mais simples, como sendo a geração de uma seqüência de imagens que retrata o movimento relativo dos objetos de uma cena sintética e, possivelmente, do movimento da câmera virtual.

1.1 Um Pouco de História

O conceito de Animação é antigo. Analisando historicamente suas origens, vemos que os primeiros trabalhos envolvendo rudimentos de Animação foram desenvolvidos antes mesmo da invenção do computador. Personalidades importantes da nossa era, como Thomas Edison, George Eastman, Winsor McCay, Irmãos Warner e Walt Disney, contribuíram, de forma decisiva, para o desenvolvimento do que hoje entendemos por Animação. Abaixo, listamos alguns marcos da história da Animação:

1824 Peter Roget apresenta o seu trabalho “The persistence of vision with regard to moving objects” à Sociedade Real Britânica.

1831 O Dr. Joseph Antoine Plateau e o Dr. Simon Ritter constroem uma máquina chamada *phenakistoscope*, que produz a ilusão de movimento permitindo ao observador visualizar um disco giratório contendo pequenas janelas, através das quais é possível observar um outro disco contendo uma seqüência de imagens. Rotacionando-se os discos na velocidade correta, a sincronização das janelas com as imagens cria o efeito animado.

1887 Thomas Edison dá início à sua pesquisa sobre imagens animadas.

1889 Thomas Edison anuncia o desenvolvimento do seu kinetoscópio, que conseguia projetar um filme de 10m de comprimento em aproximadamente 13 segundos.

1889 George Eastman começa a produção de filmes fotográficos utilizando como elemento base a nitro-celulose.

1895 Louis e Auguste Lumière conseguem uma patente sobre o desenvolvimento de um aparelho chamado cinematógrafo, capaz de projetar imagens animadas.

1896 Thomas Armat desenvolve o vitascópio, que foi utilizado para projetar os filmes de Thomas Edison. Esta máquina teve grande influência no desenvolvimento de todos os projetores subsequentes.

1906 J. Stuart Blackton produz o primeiro filme animado chamado “Humorous phases of funny faces”.

1908 Winsor McCay produz uma seqüência animada utilizando o seu personagem cômico de quadrinhos, chamado “Little Nemo”.

1909 Winsor McCay produz o cartoon chamado “Gertie the Trained Dinosaur”, formado por 10.000 películas de desenhos.

1913 Pat Sullivan cria a série animada “Gato Félix”.

1915 Earl Hurd desenvolve o conceito de *Cel Animation*.

1923 Walt Disney utiliza a técnica de combinar atores reais com cartoons no filme “Alice’s Wonderland”.

1926 Lotte Reiniger produz o primeiro filme completamente animado, chamado “Prince Achmed”.

1927 Os Irmãos Warner lançam “The Jazz Singer” que, pela primeira vez, combinava som e imagens.

1928 Walt Disney cria o primeiro cartoon com som sincronizado, chamado “Mickey Mouse”.

1945 Harry Smith produz animação desenhando diretamente na película de filme.

1957 John Whitney cria os primeiros gráficos analógicos gerados por computador.

1961 John Whitney cria a primeira seqüência animada de letreiros para a televisão.

1964 Ken Knowlton, trabalhando nos laboratórios da Bell Company, começa a desenvolver técnicas de computação para produzir filmes animados.

1.2 Animação por Computador

A utilização de computadores como ferramenta de desenho representou um grande avanço nas técnicas de animação. Entretanto, à medida que os recursos computacionais aumentavam, também aumentava a complexidade dos objetos que se pretendia animar.

Objetos complexos como, por exemplo, um braço de um robô, possuem inúmeros graus de liberdade, o que torna a animação do conjunto uma tarefa mais complicada, pois é preciso controlar diversos parâmetros simultaneamente.

Atualmente, a complexidade dos objetos animados é ainda maior. Sistemas de partículas, contendo milhares de objetos, são simulados e controlados por processos baseados na física newtoniana. Seres humanos são representados no computador por segmentos articulados conectados entre si, cada um deles contendo diversos graus de liberdade. O espaço de fase de um esqueleto humano é extremamente complexo, tornando a manipulação do mesmo uma tarefa árdua.

Neste trabalho, focalizaremos o estudo da animação de estruturas articuladas que representam seres humanos. Uma dificuldade adicional no processo de animação de seres humanos se deve ao fato de nós termos uma sensibilidade visual extremamente aguçada para perceber erros relacionados à ações cotidianas como, por exemplo, o movimento de andar. Existem diversos grupos trabalhando com animação de seres humanos. Podemos citar, por exemplo, (Computer Graphics Laboratory, 1997) e (Animation Laboratory, 1997).

Inúmeras técnicas foram desenvolvidas para facilitar a animação de estruturas complexas. A técnica de keyframing, a mais antiga de todas, facilita muito o trabalho do animador, gerando automaticamente as posições intermediárias a partir de quadros-chave (ou keyframes) estabelecidos *a priori*. No entanto, o animador deve posicionar cada articulação do objeto manualmente em cada quadro-chave da animação, o que torna muito difícil a representação correta de movimentos complexos.

Uma alternativa proposta foi a utilização de simulação para calcular as posições das articulações do objeto ao longo do tempo, utilizando, para isso, conceitos de física newtoniana. Apesar dos resultados serem visualmente melhores - e mais corretos - que os obtidos através de keyframing, o uso de simulação requer cálculos extremamente complexos, o que inviabiliza a sua utilização em aplicações onde a interatividade e a possibilidade de controle da animação são fatores fundamentais.

Ferramenta de animação mais recente, a técnica de Captura de Movimento foi desenvolvida, inicialmente, para aplicações em medicina como, por exemplo, o estudo do caminhar de deficientes físicos. No entanto, a alta qualidade do movimento gerado tornou esta técnica também atraente para aplicações na indústria cinematográfica, que a utilizou pela primeira vez na produção dos efeitos especiais do filme “O Exterminador do Futuro 2” (1993), na pele do robô de metal líquido T-1000.

Essencialmente, a técnica de Captura de Movimento consiste em registrar, através de um processo de amostragem, a posição e orientação das articulações de um ator (ou performer) ao longo do tempo, e então mapeá-las em um ator virtual, no computador.

Atualmente existem diversos equipamentos de captura de movimento que conseguem gravar movimentos extremamente complexos e rápidos. Entretanto, uma das dificuldades encontradas por animadores é a falta de ferramentas eficientes para edição de movimentos capturados. Desta forma, para se fazer qualquer modificação no movimento é necessário repetir a sessão de captura, o que ofusca as principais vantagens do processo: velocidade

de produção e interatividade.

1.3 Animação por Movimento Capturado

Como foi citado anteriormente, o processo de Captura de Movimento consiste na amostragem da posição e orientação de marcadores posicionados no corpo do performer. Para cada marcador, o processo gera um conjunto de dados bruto, na forma de um sinal unidimensional que varia ao longo do tempo (isto é, ao longo do tempo de duração da sessão de captura).

Este sinal pode ser representado matematicamente através de uma função $f: U \subset \mathbb{R} \rightarrow \mathbb{R}$, onde U é o seu domínio finito e discreto, e representa a região do espaço onde a função f varia. Na literatura de processamento de sinais (Gomes & Velho, 1997), costuma-se dizer que U é o *domínio do tempo* da função f .

Como exemplo, na Figura 1.1 é apresentado o sinal unidimensional que representa a variação angular do joelho do performer ao longo do número de quadros de uma animação.

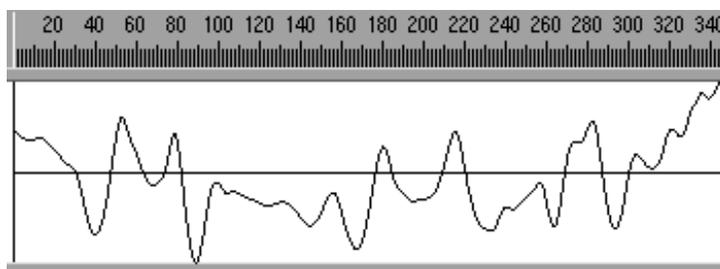


Figura 1.1: Discretização da variação angular do joelho de um performer.

O conjunto de dados amostrados é então mapeado em um ator virtual que, idealmente, deve possuir dimensões proporcionais às do performer. O resultado final é uma animação com um nível de realismo extremamente alto.

Por enquanto, não entraremos em detalhes sobre o processo de captura, que será descrito profundamente nos capítulos seguintes. Nosso objetivo nesta seção é apenas introduzir a técnica e apresentar os problemas existentes na manipulação dos dados capturados, fornecendo assim a motivação do nosso trabalho.

Reutilizar ou Recapturar?

Após a euforia e excitação causadas pelos ótimos resultados iniciais atingidos através da utilização de captura de movimento, começaram a aparecer diversas questões relacionadas ao processo, feitas principalmente por animadores e produtores de filmes.

Tais questões eram relacionadas com o destino do “sub-produto” do processo de captura, ou seja, os dados brutos. O que fazer com eles após utilizá-los?

Além disso, uma outra questão era relacionada com a dificuldade de encontrar ferramentas para a edição ou ajuste dos dados capturados. A existência de tais ferramentas era fundamental, pois caso o movimento capturado não correspondesse exatamente ao esperado, todo o processo de captura teria que ser refeito. Esse tipo de problema é ainda mais grave quando tratamos da questão de produção de efeitos especiais para filmes,

onde o orçamento limitado, o custo extremamente elevado do aluguel do equipamento e a demanda imediata por resultados não permitem erros ou repetições no processo de captura.

A resposta para essas questões ainda é tema de discussão atualmente. Alguns animadores e produtores dizem que é interessante a formação de um banco de dados de movimentos, ou seja, uma biblioteca contendo diversos segmentos de movimentos, que seria expandida ao longo do tempo. Tal biblioteca seria utilizada para a criação de animações de alto-nível, através da combinação de seus elementos. Atualmente, algumas empresas (Datalabs, 1997), (BioVision, 1997) já oferecem pequenas bibliotecas com movimentos capturados à comunidade em geral.

No entanto, apenas nos últimos dois anos começaram a surgir os primeiros trabalhos na área de processamento de movimentos capturados. A abordagem utilizada nos trabalhos seguiu a linha de processamento de sinais, como era de se esperar. Nestes trabalhos foram apresentadas técnicas para a modificação de dados capturados. Podemos citar, por exemplo, filtragem (Bruderlin & Williams, 1995), transição (Rose *et al.*, 1996), blending (Bruderlin & Williams, 1995) e warping (Witkin & Popović, 1995).

A indústria de software também seguiu esta tendência e colocou no mercado algumas ferramentas para a manipulação de movimentos capturados, integradas como plug-ins em sistemas de animação consagrados como 3D Studio, Alias/Wavefront e Softimage. Todavia, talvez devido à pressa de lançar os produtos para atender aos anseios dos animadores, não foi feita uma conceituação correta do problema da manipulação de dados capturados. Como consequência disso, as ferramentas desenvolvidas não conseguiam atingir o seu objetivo de uma forma eficiente, pois tratavam o movimento capturado utilizando os mesmos paradigmas que eram usados nas ferramentas de keyframing e cinemática já disponíveis nos sistemas de animação tradicionais.

Os dados obtidos através de captura de movimento são essencialmente diferentes dos obtidos com as técnicas tradicionais como keyframing e simulação, seja pelo seu processo de obtenção ou pela própria natureza matemática dos dados. Por isso, uma abordagem específica deve ser utilizada para manipulá-los.

Uma interessante analogia sobre esse problema foi feita por Olivier Ozoux, um dos organizadores do Online Panel do SIGGRAPH'97 (SIGGRAPH, 1997) sobre Movimento Capturado, que comparou a indústria de Movimento Capturado à indústria Musical. A técnica de Movimento Capturado necessita de ferramentas de edição semelhantes às encontradas nos estúdios de gravação de música. Seguindo a analogia, atualmente o que temos são apenas os “microfones” (equipamento de captura) e o “gravador”. Ainda precisamos de:

- **Alto-falantes.** Ao gravarmos músicas ou qualquer outro tipo de som, sempre necessitamos de um retorno (feedback) do que estamos gravando, para nos certificarmos que está tudo correto. O mesmo ocorre com Captura de Movimento, onde o feedback visual em tempo-real do personagem virtual é fundamental para que o ator possa ajustar seus movimentos em relação ao traje de captura. Atualmente, apenas alguns tipos de equipamento de captura fornecem um feedback em tempo-real da performance do ator.
- **Mesa de edição.** Um movimento capturado deve ser tratado de forma global, e não local, como acontece com os quadros-chaves de uma animação baseada em

keyframing. Em outras palavras, um movimento capturado é uma animação potencialmente pronta, que deve ser tratada seguindo um paradigma parecido com o utilizado em mesas de edição de áudio.

- **Efeitos.** No campo de processamento de sons e música, existe uma vasta coleção de efeitos (adicionar echo, alterar o *pitch*, eliminar ruídos, etc). Poderíamos ter ferramentas semelhantes para manipular movimentos capturados, pois existe uma grande semelhança entre os dois tipos de dados.

O objetivo final deste trabalho é desenvolver um sistema de animação que tem a técnica de Captura de Movimento como o núcleo de todo o processo de animação, diferentemente dos sistemas atuais, que a tratam apenas como uma ferramenta secundária. O sistema proposto conta com ferramentas para análise, modificação e reutilização de movimentos capturados, permitindo ao animador ajustar, combinar, filtrar e concatenar movimentos previamente capturados, sem a necessidade de repetir o processo de captura. Desta forma, o animador terá a possibilidade de criar bibliotecas de movimentos, que poderão ser combinadas e expandidas utilizando-se as ferramentas disponíveis no sistema.

Como consequência da utilização da técnica de Captura de Movimento no sistema proposto, foi necessário desenvolver uma arquitetura que trate de maneira adequada os principais aspectos deste tipo de tecnologia, levando em conta as potencialidades e deficiências do processo. Além disso, o sistema utiliza um modelo de interface baseado no paradigma de edição de vídeo, que julgamos adequado para tratar o problema em questão.

O sistema também conta com ferramentas tradicionais de animação, como keyframing e cinemática, que têm como objetivo auxiliar o animador na manipulação dos movimentos capturados. Ferramentas adicionais para análise das curvas de movimentos, controle de câmera em tempo-real, e parâmetros da cena também estão presentes no sistema.

1.4 Organização da Tese

A continuação da dissertação seguirá a seguinte organização em capítulos, brevemente descritos a seguir.

- **Capítulo 2 - Modelos de Animação**

Neste capítulo faremos uma classificação dos objetos geométricos que são utilizados em animação por computador, discutindo os tipos de parâmetros a que estão associados. Faremos uma análise dos níveis de manipulação desses objetos geométricos no âmbito de animação. Por último, estudaremos a possibilidade de combinação desses objetos através de vínculos, dando uma maior importância para o caso de estruturas articuladas.

- **Capítulo 3 - Ator Virtual**

No Capítulo 3, apresentaremos o conceito de ator virtual. Discutiremos sua representação no computador através de estruturas articuladas hierárquicas (ou esqueletos), juntamente com uma descrição do processo de mapeamento de dados.

- **Capítulo 4 - Conceituação de Animação**

Neste capítulo, faremos uma conceituação da área de animação por computador. Introduziremos um diagrama que visa posicionar as diversas etapas do processo de animação, de acordo com a sua finalidade, estabelecendo assim relações importantes para o entendimento global do processo. Também situaremos a técnica de Captura de Movimento em relação a esse contexto.

- **Capítulo 5 - Sistemas de Síntese de Movimento**

No Capítulo 5, abordaremos as principais técnicas existentes para a geração (síntese) de movimentos fazendo, sempre que possível, ilustrações comparativas.

- **Capítulo 6 - Sistemas de Captura de Movimento**

Discutiremos aspectos tecnológicos do processo, fazendo uma análise comparativa dos principais equipamentos de captura disponíveis atualmente no mercado.

- **Capítulo 7 - Os Dados Capturados**

Neste capítulo, faremos uma análise dos dados obtidos através de Captura de Movimento. Estudaremos também o problema de conversão dos dados capturados para um modelo hierárquico, que será mapeado então no esqueleto do ator virtual.

- **Capítulo 8 - Processamento de Movimentos**

Faremos uma classificação conceitual das operações existentes para o processamento de movimentos capturados. Analisaremos detalhadamente as principais técnicas disponíveis atualmente.

- **Capítulo 9 - Um Sistema de Animação Baseado em Movimento Capturado**

Nesta seção apresentaremos o sistema desenvolvido, baseado nos conceitos descritos anteriormente. Este sistema tem como objetivo fornecer ferramentas para análise, modificação e reutilização de movimentos capturados, num ambiente conceitualmente correto para tratar as principais entidades do processo de captura. Faremos uma descrição de suas principais características e ferramentas disponíveis, juntamente com a arquitetura e modelo de interface propostos.

- **Capítulo 10 - Interface com o Usuário**

No Capítulo 10, será apresentado o paradigma utilizado para a construção da interface gráfica do sistema, juntamente com a descrição gráfica das principais entidades existentes no sistema.

- **Capítulo 11 - Resultados e Exemplos**

Neste capítulo, apresentaremos alguns resultados e exemplos obtidos através da aplicação das técnicas descritas anteriormente.

- **Capítulo 12 - Conclusão**

Neste capítulo apresentaremos as principais conclusões do trabalho. Faremos uma análise final da técnica de captura de movimento, comparando-a com as principais ferramentas disponíveis atualmente. Faremos também uma análise crítica do sistema, discutindo aspectos funcionais e apontando suas aplicações. Finalmente, apresentaremos propostas para trabalhos futuros.

Capítulo 2

Modelos de Animação

Animar um objeto genérico não consiste simplesmente na definição da sua trajetória ou comportamento ao longo do tempo. Os objetos que serão animados devem ser representados corretamente no computador, de modo a fornecer uma descrição adequada e intuitiva, facilitando assim a sua simulação.

Tal representação depende, basicamente, de dois fatores fundamentais: os parâmetros que pretende-se animar e a representação visual dos objetos.

Os parâmetros de animação relacionados aos objetos estão fortemente ligados ao tipo de simulação que será aplicada a esses objetos. Em outras palavras, se pretendemos simular aspectos dinâmicos de um objeto, o seu conjunto de parâmetros de animação deve conter grandezas físicas como aceleração, massa e torque.

Em relação aos aspectos visuais dos objetos reais, nem sempre é possível representar a sua complexidade total no computador. Devemos lembrar que o computador é uma máquina com precisão e capacidade de representação finitas, enquanto que a maioria dos objetos existentes em nosso mundo são formados por um número grande de detalhes, formas e outros componentes que superam em muito a capacidade de representação dos computadores atuais. Pensando nisso, é necessário estabelecer classes de objetos no computador, que têm como objetivo representar diferentes categorias de objetos reais.

Neste capítulo, analisaremos a relação entre os objetos geométricos, seus parâmetros e os níveis de simulação existentes na área de animação por computador. Veremos também que esses objetos geométricos podem ser combinados por meio de relações - ou vínculos - de modo a criar estruturas mais complexas.

2.1 Representação Geométrica dos Parâmetros da Animação

Dependendo do nível de simulação e do tipo de objeto desejado, diversos parâmetros podem ser associados. Inicialmente consideraremos apenas os parâmetros geométricos (posição e orientação). Depois, podemos também atribuir a esses objetos parâmetros físicos (massa, torque etc.) e até comportamentais (relações, motivações etc.).

Para estabelecer relações entre os objetos no computador e seus parâmetros, podemos classificá-los em categorias, que têm como critério de complexidade o tipo e número de parâmetros que tais objetos podem representar.

Podemos classificar os objetos representados no computador em três categorias:

- **Partículas**
- **Objetos Rígidos**
- **Objetos Deformáveis**

Tais categorias procuram estabelecer níveis de estratificação dos objetos em relação aos parâmetros a que estão associados. Logicamente, dentro de cada categoria, podem existir diferentes níveis de complexidade da geometria desses objetos. Nesta seção, não estaremos interessados nesse tipo de detalhe.

2.1.1 Partículas

Geometricamente, partículas são objetos sem volume, as quais são associadas posições que variam ao longo do tempo segundo uma determinada regra. Em termos matemáticos, uma partícula pode ser definida como um ponto P no espaço \mathbb{R}^3 , sujeito a transformações de translação. Partículas podem ser utilizadas para representar objetos amorfos e complexos, como nuvens e fogo (Reeves, 1983).

2.1.2 Objetos Rígidos

A grande maioria dos objetos que utilizamos no cotidiano podem ser representados por objetos rígidos no computador. Uma roda, por exemplo, pode ser representada pelo objeto geométrico cilindro.

Matematicamente falando, um objeto rígido pode ser descrito como um conjunto fechado e limitado de pontos $S \subset \mathbb{R}^3$, sendo afetado por transformações de translação e rotação. Assim, este tipo de objeto geométrico não leva em conta fatores de deformação em sua superfície. Em termos geométricos, um objeto rígido possui uma posição e orientação no espaço em que está inserido (Baraff, 1995).

2.1.3 Objetos Deformáveis

Um objeto deformável é um conjunto de pontos suscetível a transformações que alteram a sua geometria.

O processo de deformação de um objeto pode ser visto sob dois aspectos distintos: como um processo de modelagem, onde um modelo geométrico do objeto é criado a cada quadro da animação; ou como um processo de animação, onde os dados geométricos que representam o objeto são alterados ao longo dos quadros da animação (Terzopoulos *et al.*, 1987b).

A utilização de objetos deformáveis no processo de animação introduziu inúmeras possibilidades aos animadores. A principal característica do processo é permitir a aplicação de técnicas tradicionais de deformação de superfícies utilizadas em desenhos animados, como por exemplo o *squash and stretch* (Lasseter, 1987), a objetos representados no computador.

2.2 Níveis de Simulação

Os objetos geométricos descritos na seção anterior podem ser animados em diferentes níveis de simulação. A utilização desses níveis visa estabelecer uma diferenciação entre os tipos de parâmetros que são usados nos objetos geométricos, dependendo do grau de simulação desejado.

Existem basicamente três níveis de simulação do movimento de objetos representados no computador:

- **Cinemático.** No nível cinemático, os parâmetros simulados estão estritamente relacionados com a geometria do objeto. Posição e orientação são parâmetros cinemáticos que podem ser aplicados a um objeto geométrico para simular, por exemplo, sua trajetória no espaço (Magenat-Thalmann & Thalmann, 1990).
- **Dinâmico.** A introdução de parâmetros dinâmicos (ou físicos) como massa e torque, permite a simulação física de fenômenos naturais pelo computador. A utilização de objetos com propriedades dinâmicas pode ser útil para simular, por exemplo, colisões entre veículos ou mesmo sistemas planetários complexos. No caso de objetos deformáveis, por exemplo, poderíamos adicionar fatores físicos como elasticidade, para simular o processo de colisões elásticas (Terzopoulos & Witkin, 1988).
- **Comportamental.** O nível mais alto de simulação - comportamental - lida com parâmetros que relacionam os objetos com o meio em que estão inseridos. Fatores como motivação, desejo e libido podem ser associados a objetos durante uma animação. Em (Terzopoulos, 1994), fatores comportamentais como medo, fome e caça foram aplicados na simulação de um cardume de peixes em um aquário virtual.

A Figura 2.1 mostra um diagrama com a disposição hierárquica dos níveis de simulação dos parâmetros. Embora os parâmetros dos níveis de simulação possam ser aplicados aos objetos de uma forma independente, geralmente eles estão relacionados entre si. Por exemplo, o parâmetro físico aceleração envolve um deslocamento (translação), que por sua vez é um parâmetro cinemático. Em um nível superior, um parâmetro comportamental também pode envolver parâmetros dinâmicos, que por sua vez podem envolver parâmetros cinemáticos.

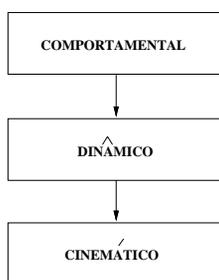


Figura 2.1: Níveis de simulação de objetos representados no computador.

2.3 Exemplo de Simulação

Nas seções anteriores vimos que os objetos geométricos podem ser animados em diferentes níveis de simulação com relação aos seus parâmetros de animação. Nesta seção, apresentaremos um exemplo prático da animação de um objeto geométrico simples - uma bola - nos três níveis de simulação.

Considere o exemplo da Figura 2.2. Neste exemplo, uma bola é colocada, inicialmente, a uma certa altura do solo. Suponha agora que o objetivo da animação seja fazer a bola colidir com o solo segundo uma direção d .

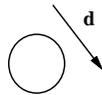


Figura 2.2: Configuração inicial do exemplo.

No nível cinemático, podemos animar a trajetória da bola fazendo variar a sua posição na direção d até o solo, como é mostrado na Figura 2.3.

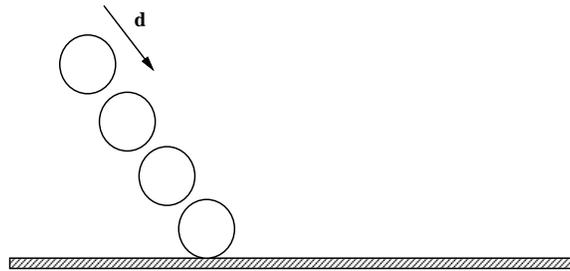


Figura 2.3: Animação com parâmetros cinemáticos.

No nível dinâmico, podemos adicionar fatores como a gravidade e velocidade para simular a trajetória da bola. Se desejarmos incluir características de colisão elástica, precisaremos mudar o tipo de objeto geométrico que representará a bola, utilizando então um objeto deformável (Figura 2.4).

No nível comportamental, podemos considerar que a bola agora seja a representação geométrica de um animal, que avistou comida no solo e pretende alcançá-la se locomovendo na direção d , como mostrado na Figura 2.5.

Como podemos observar, a mudança de representação pode ser feita tanto com relação aos parâmetros de animação (dependendo do nível de simulação desejado), quanto ao tipo de objeto geométrico utilizado (que também pode depender do nível de simulação).

Na Tabela 2.1, apresentamos uma esquematização dos objetos geométricos e seus respectivos parâmetros de animação, nos diferentes níveis de manipulação.

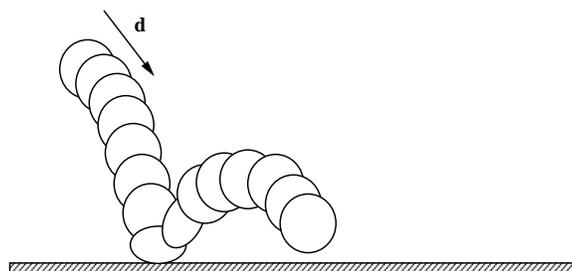


Figura 2.4: Inclusão de parâmetros dinâmicos: gravidade, velocidade e deformação.

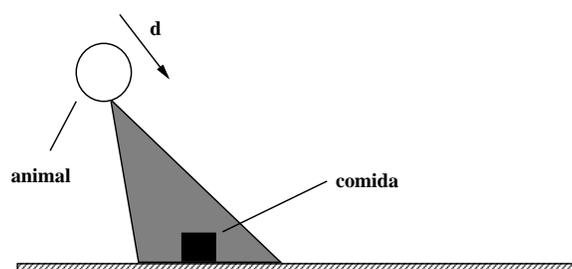


Figura 2.5: Situação com parâmetros comportamentais.

	<i>cinemático</i>	<i>dinâmico</i>	<i>comportamental</i>
<i>partículas</i>	posição	velocidade e massa	relações
<i>objetos rígidos</i>	posição e orientação	velocidade, massa, torque etc.	relações, motivações, desejos etc.
<i>objetos deformáveis</i>	posição, orientação	velocidade, massa, torque, volume etc.	relações, motivações, desejos etc.

Tabela 2.1: Objetos geométricos vs. níveis de manipulação.

2.4 Vínculos

A classificação dos objetos geométricos e a conceituação de níveis de simulação facilita o entendimento do processo de animação, uma vez que estabelece camadas de complexidade entre os objetos geométricos e os parâmetros a que estão associados.

Todavia, à medida que a complexidade dos objetos reais que pretende-se representar no computador aumenta, surge a necessidade de criar estruturas de representação mais complexas e poderosas.

O caminho natural utilizado para solucionar essa questão é a criação de vínculos entre os diferentes tipos de objetos geométricos. No sentido estrito da palavra, vínculo significa relação, interdependência. Pensando nisso, podemos estabelecer relações entre os objetos geométricos de modo a criar estruturas que permitam representar objetos reais com maior

grau de complexidade.

Podemos dividir os tipos de relações (ou vínculos) entre os objetos geométricos em três categorias:

- **Vínculos geométricos (ou estruturais).** Um vínculo estrutural pode ser empregado para criar uma relação geométrica entre dois objetos. Uma articulação, por exemplo, pode ser utilizada como um vínculo geométrico entre dois objetos, criando uma estrutura articulada com maior poder de movimentação no espaço. O braço de um robô é um exemplo prático da utilização de um vínculo geométrico entre dois objetos. Neste trabalho, focalizaremos o estudo de estruturas articuladas que representam seres humanos.
- **Vínculos dinâmicos.** Campos de forças de atração ou repulsão podem ser aplicados a objetos geométricos, estabelecendo um vínculo dinâmico entre eles. Desta forma, a proximidade entre os objetos é definida por uma função que pode variar ao longo do tempo. Um exemplo prático desse tipo de vínculo é o sistema planetário, onde a distância entre os dois corpos é modelada segundo a força gravitacional que existe entre eles. Em um outro exemplo, a simulação de um sistema massa-mola é feita estabelecendo-se uma função que modelará a atração da massa segundo a deformação de uma mola.
- **Vínculos comportamentais.** Este tipo de vínculo é utilizado para estabelecer relações entre parâmetros comportamentais associados a objetos no computador. Podemos ilustrar a utilização desse tipo de vínculo com o exemplo da seção anterior (Figura 2.5). Neste caso, podemos associar a variação do parâmetro comportamental “fome” a uma motivação para a procura de comida.

Existem inúmeras aplicações da utilização de vínculos entre objetos no computador. Listamos abaixo alguns exemplos dessas aplicações.

Exemplo 1 - Simulação de galáxias

Sistemas de partículas podem ser utilizados para simular galáxias. Descartando alguns fatores críticos, como a colisão entre corpos celestes, podemos associar uma partícula a cada planeta ou estrela. Tais partículas “herdariam” algumas características físicas dos corpos celestes como, por exemplo, massa e velocidade. Seriam então estabelecidos vínculos dinâmicos entre as partículas, de modo a simular a atração gravitacional entre elas. Neste caso, a associação entre partículas e objetos rígidos também serve para simplificar o processo de simulação.

Exemplo 2 - Objetos articulados

A composição de diversos objetos geométricos na formação de estruturas articuladas é de fundamental importância para determinados tipos de aplicação, como por exemplo, a animação de animais. Neste caso, vínculos geométricos são empregados para representar as principais articulações do animal, conectando diversos segmentos que representam as partes do corpo.

Exemplo 3 - Modelagem de superfícies

O processo de modelagem de superfícies paramétricas por partes consiste na utilização de pontos de controle, que são empregados na modificação dos coeficientes das funções polinomiais que definem a superfície. É possível associar partículas a conjuntos de pontos de controle. Desta forma, movendo-se a partícula no espaço são produzidas deformações na superfície. Nesse caso, um vínculo geométrico foi estabelecido entre cada partícula e o conjunto de pontos de controle da superfície em questão.

Capítulo 3

Ator Virtual

Como vimos no capítulo anterior, podemos utilizar um vínculo geométrico entre dois ou mais objetos para criar uma estrutura articulada. O interesse na animação de figuras articuladas tem aumentado muito ultimamente devido, principalmente, ao desejo de se criar humanos sintéticos no computador.

3.1 Estruturas Articuladas

Por definição, uma estrutura articulada consiste em um conjunto de objetos rígidos, conectados por articulações (ou *joints*) (Watt & Watt, 1992). Tais articulações servem para formar o vínculo geométrico entre os objetos e, principalmente, para permitir a movimentação relativa entre eles.

Na Figura 3.1 podemos observar um exemplo de estrutura articulada simples. Neste caso, objetos geométricos rígidos (os retângulos) são conectados através de articulações (representadas pelos círculos escuros).

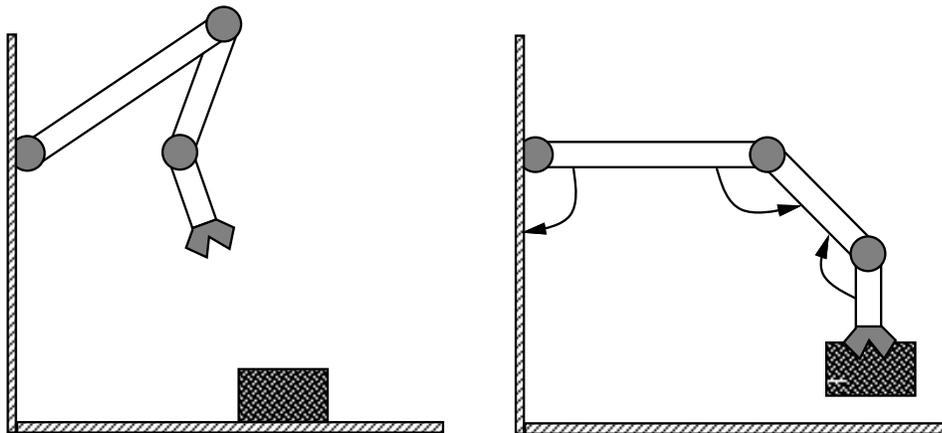


Figura 3.1: Braço robô com três graus de liberdade.

Na estrutura do exemplo acima, cada objeto pode girar em torno da articulação, segundo a direção de rotação permitida pela mesma (representada pelas setas). Em robótica, isso é chamado de grau de liberdade (DOF, *Degrees of Freedom*).

O número de graus de liberdade de uma estrutura articulada está intimamente associado com o poder de movimentação da mesma no espaço. Em outras palavras, quanto maior o grau de liberdade da estrutura, maior o número de configurações possíveis que ela pode formar no espaço. Como veremos adiante, nem sempre é interessante lidar com estruturas com muitos graus de liberdade, pois a complexidade de manipulação computacional da estrutura é diretamente proporcional ao seu número de DOF's.

Um outro termo importante associado à estruturas articuladas (e também extraído da robótica) é o *end-effector*, que representa o último segmento livre de uma estrutura articulada. Geralmente, os *end-effectors* são responsáveis pela interação das estruturas articuladas com outros objetos.

Um caso particular de estrutura articulada é o esqueleto humano. Na próxima seção, veremos como representar de maneira eficiente a estrutura complexa que representa o corpo humano, associando objetos geométricos às principais partes do corpo e conectando-os por meio de articulações.

3.2 Modelos de Representação do Corpo Humano

Na seção anterior, vimos que o esqueleto humano pode ser representado por meio de uma estrutura articulada.

Nosso corpo é uma estrutura extremamente complexa, formada por segmentos rígidos e deformáveis, centenas de músculos, tecidos com diferentes graus de elasticidade, ossos com geometria detalhada, articulações com inúmeros graus de liberdade, líquidos e muitas outras características que têm certa influência no seu movimento. Tentar representar todas essas características no computador seria impossível.

Cientes desta limitação, podemos utilizar modelos simplificados para representar o corpo humano. Para efeitos de simplificação, podemos começar considerando apenas os ossos e as articulações como fatores de influência no movimento do corpo (Figura 3.2). Desta forma, não precisaremos mais nos preocupar, por exemplo, com os efeitos da deformação da pele e fatores biomecânicos dos músculos no processo de animação. No entanto, tais fatores são muito importantes em certas aplicações como a simulação de atividades esportivas (Hodgins *et al.*, 1995) e aplicações biomédicas.

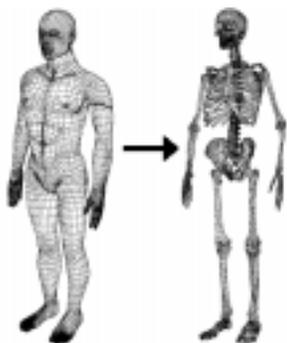


Figura 3.2: Primeiro estágio do processo de simplificação do corpo humano.

O corpo humano possui centenas de ossos. O pé, por exemplo, é formado por cerca de 20 pequenos ossos, que compõem uma estrutura complexa responsável pela nossa sus-

tentação. Além disso, as articulações que conectam tais ossos possuem uma geometria extremamente complicada, com diversos pontos de contato e inúmeros graus de liberdade para rotação e até mesmo translação. Como podemos notar, é preciso simplificar ainda mais o modelo para podermos representá-lo no computador utilizando os objetos e conceitos descritos anteriormente.

Aprofundando um pouco mais o nível de simplificação do esqueleto, podemos “aglutinar” determinados conjuntos de ossos, representando-os no computador por meio de um único segmento rígido. Na Figura 3.3, este processo é exemplificado substituindo-se o conjunto de ossos que formam o pé por um único segmento rígido.

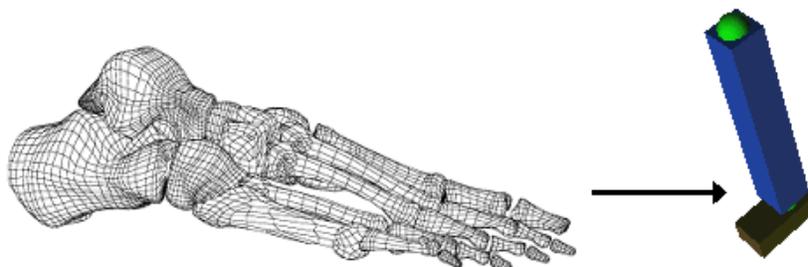


Figura 3.3: Aglutinação de ossos e representação por meio de um único segmento rígido.

Em relação às articulações do esqueleto humano, podemos aproximá-las por articulações de revolução, cada uma podendo ter até quatro graus de liberdade (Figura 3.4).

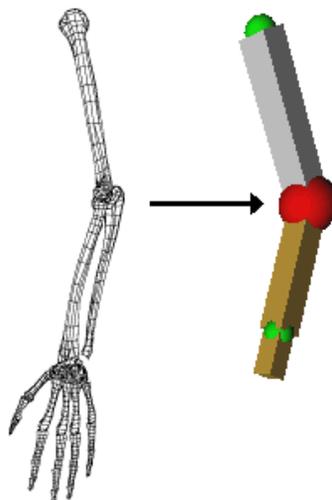


Figura 3.4: Representação de articulações no computador.

Utilizando as simplificações citadas anteriormente, conseguimos criar no computador uma estrutura articulada que consegue simular, com um grau de fidelidade bastante razoável, os movimentos do corpo humano.

A literatura de animação contém diversos exemplos de modelos simplificados do corpo humano. Em todos os casos, o objetivo é o mesmo: obter uma representação simples, capaz de ser manipulada e visualizada interativamente no computador, e que consiga reproduzir, de maneira realista, os movimentos de seres humanos.

Em um dos primeiros trabalhos envolvendo modelos simplificados do corpo humano, Jim Blinn (Blinn, 1987) construiu um modelo contendo 17 segmentos rígidos simples (elipsóides), conectados por meio de articulações de revolução, num total de 54 graus de liberdade (51 deles relativos à rotação das articulações e mais 3 relativos à translação do modelo no espaço).

Em um outro trabalho mais recente, Ken Perlin (Perlin, 1995) utilizou um modelo com 19 elipsóides, também conectados por meio de articulações de revolução (57+3 DOF's). No entanto, Perlin incluiu em seu modelo algumas características naturais inerentes ao corpo humano como, por exemplo, a distensão do músculo adutor da coxa. Para obter esse “efeito”, Perlin desenvolveu um método que produzia variações na posição relativa entre os elipsóides que formavam o segmento que representava a coxa, de forma proporcional ao grau de flexão da perna.

Como vimos anteriormente, o número de articulações (e, conseqüentemente, de graus de liberdade) de uma estrutura articulada determina o seu poder de representação de movimentos no espaço. No caso de estruturas articuladas que representam esqueletos humanos, logicamente existe um limite no que diz respeito ao número mínimo de articulações necessárias para representar movimentos humanos, dependendo da aplicação desejada. Apesar de não existir um número mínimo fixo estabelecido na literatura, acreditamos que este número esteja entre 15 e 20 articulações.

3.2.1 O Modelo Utilizado no Sistema

Em nosso sistema de animação, criamos um modelo com 16 segmentos rígidos, conectados por articulações de revolução. Como veremos adiante, o posicionamento e o número de articulações empregadas no ator virtual do nosso sistema possui uma forte relação com o número de marcadores utilizados no processo de captura de movimento.

A Figura 3.5 apresenta o modelo de ator virtual utilizado no sistema, representado em diferentes estilos de visualização. É possível utilizar objetos mais complexos na representação visual do ator virtual, tornando-o mais realista. Porém, neste trabalho, não nos preocuparemos muito com os aspectos visuais do modelo, que podem ser melhor estudados em (Badler *et al.*, 1993).

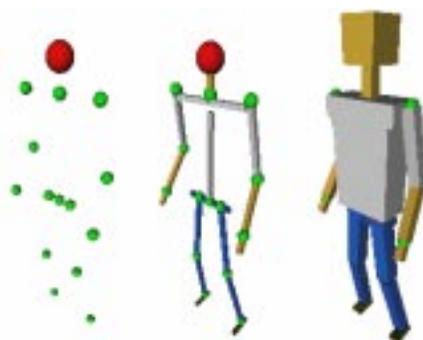


Figura 3.5: Modelo de ator virtual utilizado no sistema.

Para simplificar um pouco mais o modelo, limitamos o número de graus de liberdade de algumas articulações cuja contribuição não era significativa para o processo de animação.

Como exemplo, podemos citar a articulação do cotovelo (Figura 3.6), que permite apenas a rotação em torno do eixo X, perpendicular ao comprimento do segmento superior do braço. Essa simplificação foi feita baseada em observações reais de características do corpo humano. No total, o modelo de ator virtual utilizado no sistema possui 33 DOF's (30+3).

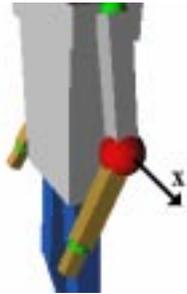


Figura 3.6: Simplificação de articulação por redução de DOF.

3.3 Definição de Estruturas Articuladas

Após a definição do modelo que será utilizado para representar o corpo humano, o passo seguinte é representá-lo sob a forma de estruturas de dados no computador.

Devemos escolher uma forma de representação que seja simples e intuitiva, pois à medida que a complexidade da estrutura aumenta, cresce a dificuldade de representá-la no computador. Nesta seção, descreveremos duas notações utilizadas para representar estruturas articuladas, juntamente com a variante utilizada em nosso sistema de animação.

3.3.1 Notação DH

A notação DH (Denavit & Hartenberg, 1955) foi desenvolvida originalmente para representar manipuladores robóticos, porém também pode ser utilizada para representar determinados tipos de estruturas articuladas.

Esta notação descreve o estado de cada segmento (ou *link*) da estrutura em relação aos segmentos vizinhos, através de parâmetros cinemáticos inseridos em um sistema de coordenadas associado a cada segmento. Matrizes de transformação entre esses sistemas de coordenadas são então criadas e associadas às articulações da estrutura.

Basicamente, as matrizes de transformação possuem quatro parâmetros cinemáticos, relativos ao comprimento, distância relativa, twist e ângulo entre os segmentos da estrutura articulada.

Apesar de ser um esquema de representação econômico, pois especifica cada sistema de coordenadas em relação ao anterior, a notação DH possui algumas desvantagens:

- **Articulações com mais de um DOF.** A notação DH foi desenvolvida originalmente para representar estruturas contendo articulações com apenas um grau de liberdade cada. Articulações com mais graus de liberdade também podem ser representadas, considerando-as como um conjunto de articulações de 1 DOF. No entanto, isso dificulta a manipulação da estrutura e ofusca a característica econômica da notação.

- **Estruturas ramificadas livres.** A representação de estruturas com ramificações não é possível utilizando a notação DH. Isso pode ser um fator crítico quando estamos lidando com estruturas que representam o corpo humano. Além disso, a notação DH supõe que um dos extremos da cadeia articulada esteja fixo, o que também limita o escopo de aplicações possíveis.

3.3.2 Notação APJ

Tendo em mente as limitações da notação DH, Sims e Zeltzer (Zeltzer & Sims, 1988) desenvolveram um esquema de representação menos econômico, porém mais intuitivo e com maior flexibilidade, chamado *Axis-Position Joint*, ou simplesmente APJ.

Desta forma, a notação APJ guarda alguns parâmetros adicionais, que permitem a criação de estruturas articuladas ramificadas e com um maior número de graus de liberdade. Para cada articulação, a notação APJ armazena os seguintes parâmetros:

- Posição da articulação.
- Orientação do sistema de coordenadas da articulação.
- Ponteiros para os segmentos conectados pela articulação.

Com isso, sete parâmetros (três para posição, três para orientação e um para o ângulo da articulação em relação ao seu sistema de coordenadas) são associados a cada articulação da estrutura.

3.3.3 Notação APJ Modificada

Em nosso sistema de animação, utilizaremos uma variante da notação APJ, modificada para receber o tipo de dados produzidos por equipamentos de captura de movimento. Por enquanto, não entraremos em detalhes técnicos sobre os dados capturados, que serão descritos no Capítulo 7.

Além dos parâmetros relacionados com a posição e orientação do sistema de coordenadas da articulação, incluímos mais três parâmetros, que descrevem a sua orientação relativa em relação à articulação anterior. Este tipo de modificação se faz necessária devido ao fato de estarmos utilizando uma representação hierárquica dos segmentos e articulações, tópico que será descrito a seguir.

3.4 Representação Hierárquica

Os esquemas de representação descritos anteriormente consideram o fato de cada articulação ter um conjunto de parâmetros relacionados com sua posição no espaço. No entanto, essa característica pode causar alguns problemas.

A necessidade de especificação da posição de cada articulação da estrutura não garante a integridade da conectividade do conjunto, uma vez que pequenas variações na posição de uma articulação podem “quebrar” as ligações entre os segmentos (Figura 3.7).

Existem métodos baseados em relaxação que podem ser utilizados para evitar tais rupturas, porém esses métodos são custosos e sujeitos a problemas de precisão.

A organização da estrutura segundo um modelo hierárquico resolve o problema da integridade da estrutura. No modelo hierárquico, a posição de cada articulação é definida,

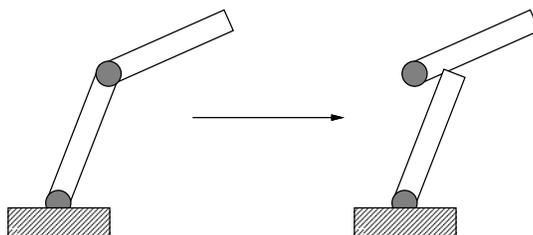


Figura 3.7: Ruptura da estrutura causada por variações na posição de uma articulação.

automaticamente, através da composição em seqüência das matrizes de transformação das articulações anteriores. Desta forma, apenas a primeira articulação da estrutura precisa ser posicionada no espaço, enquanto que o resto da estrutura é comandada apenas pelos ângulos entre as articulações (ângulos relativos).

A construção da estrutura hierárquica é feita posicionando-se as articulações em uma configuração inicial no espaço, utilizando operações de translação e rotação. No entanto, a estrutura deve ser implementada de modo a permitir a acumulação de transformações geométricas entre os sistemas de coordenadas das articulações.

Em termos topológicos, a disposição de uma estrutura articulada hierárquica pode ser comparada a uma árvore, onde os nós internos representam as articulações, as folhas representam os *end-effectors* e as ligações entre esses elementos representam os segmentos rígidos da estrutura, como mostrado na Figura 3.8. Desta forma, ao aplicarmos uma transformação de rotação em uma articulação T , todas as articulações da sub-árvore definida por T serão afetadas por essa transformação, e assim por diante.

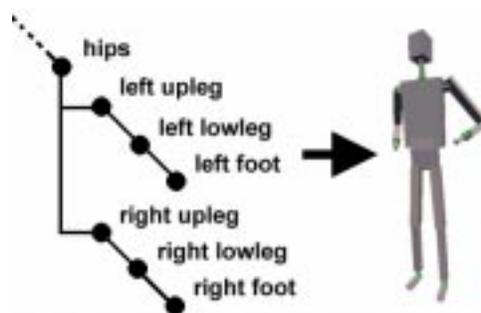


Figura 3.8: Representação topológica de uma estrutura articulada hierárquica.

Sob o ponto de vista da árvore que representa a estrutura articulada, os comandos de manipulação de transformações *Push* e *Pop* funcionam como ferramentas para subir ou descer entre os seus níveis. Além disso, cada comando *Push* empregado é equivalente à criação de uma nova ramificação na estrutura.

Como havíamos dito na seção anterior, utilizamos em nosso sistema de animação uma variante da representação APJ, que inclui parâmetros para a orientação relativa entre as articulações. Esses parâmetros relativos são responsáveis pela configuração do ator virtual no espaço, e serão calculados utilizando-se o método que será descrito no Capítulo 7.

3.4.1 A Topologia do Ator Virtual do Sistema

No ator virtual utilizado em nosso sistema de animação, a árvore que representa a estrutura articulada hierárquica possui uma configuração que tem como objetivo representar as relações estruturais reais existentes no corpo humano.

Como podemos observar na Figura 3.9, a raiz dessa árvore está localizada na região do quadril do ator virtual, e tem como finalidade concentrar e expandir as transformações que são aplicadas ao esqueleto como um todo. Em outras palavras, ao aplicarmos uma transformação geométrica à raiz da árvore, todos os segmentos que fazem parte da estrutura serão afetados por essa transformação. Isso é útil, por exemplo, para movimentar a estrutura inteira pelo espaço.

No segundo nível da árvore, encontramos os nós internos que representam os segmentos do quadril e do peito do ator. Subordinados a esses segmentos, estão as partes mais “externas” do corpo, ou seja, suas pernas e braços.

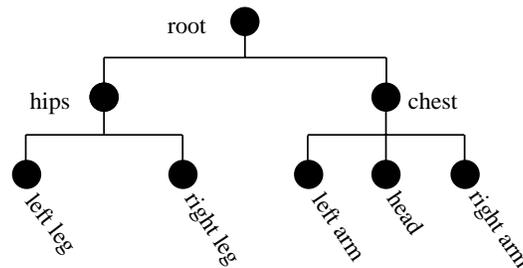


Figura 3.9: Árvore que representa a estrutura hierárquica do ator virtual no sistema.

3.5 Ferramentas Adicionais: Forward e Inverse Kinematics

Com a utilização de modelos hierárquicos na representação de estruturas articuladas, podemos dispor de ferramentas mais avançadas para a manipulação dessas estruturas.

3.5.1 Forward Kinematics (FK)

Neste tipo de ferramenta, o movimento da estrutura é especificado explicitamente, através da definição dos ângulos relativos de cada uma das articulações. Assim, o movimento do *end-effector* da estrutura é determinado indiretamente pela composição de todas as transformações existentes nas articulações que o precedem na árvore.

Em termos matemáticos, o uso de FK pode ser colocado da seguinte forma: dado um conjunto θ_i de ângulos relativos, a posição X do *end-effector* da estrutura é dada por $X = f(\theta_i)$.

3.5.2 Inverse Kinematics (IK)

Este tipo de ferramenta é geralmente utilizada para aplicações do tipo *goal-directed motion*, onde o objetivo é especificar apenas a posição final de um *end-effector* no espaço, deixando para o computador o cálculo da configuração de ângulos necessária para tal. Em outras palavras, dada a posição X do *end-effector*, a configuração de ângulos θ_i é dada por $\theta_i = f^{-1}(X)$.

Um Exemplo Prático

Considere a estrutura articulada simples da Figura 3.10. A solução por FK, $X = (x, y)$, é dada por:

$$X = (L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2), L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2))$$

Utilizando trigonometria, a solução IK é dada por:

$$\theta_2 = \cos^{-1} \frac{(x^2 + y^2 - L_1^2 - L_2^2)}{2L_1 L_2} \text{ e } \theta_1 = \frac{-(L_2 \sin \theta_2)x + (L_1 + L_2 \cos \theta_2)y}{(L_2 \sin \theta_2)y + (L_1 + L_2 \cos \theta_2)x}$$

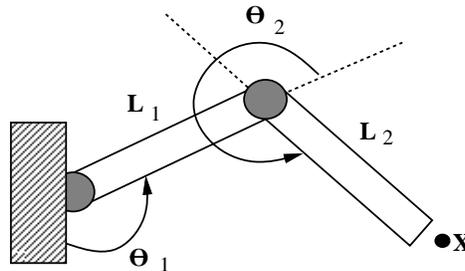


Figura 3.10: Forward vs. Inverse Kinematics.

Embora este seja um exemplo bastante simples, podemos notar que a utilização dessas ferramentas se torna mais difícil à medida que a complexidade (e conseqüentemente o número de DOF's) da estrutura aumenta. No caso de FK, o animador terá que controlar uma quantidade cada vez maior de transformações para atingir o objetivo, enquanto que em IK o problema de achar θ_i pode ficar indeterminado devido à redundância do sistema de equações.

Capítulo 4

Conceituação de Animação

O processo de animação de objetos geométricos envolve uma série de etapas, que vão desde a síntese dos dados de movimento, até a sua análise e reprodução no computador. Tais etapas nem sempre são corretamente descritas na literatura.

Neste capítulo, buscaremos suprir essa deficiência através da conceituação e caracterização das principais etapas do processo de animação, utilizando diagramas ilustrativos. Também localizaremos o processo de Captura de Movimento no âmbito de Animação por Computador.

4.1 Visão Geral

Utilizando o paradigma dos quatro universos (Gomes & Velho, 1994), que procura estabelecer diferentes níveis de abstração de objetos que são utilizados em Computação Gráfica, observamos que as técnicas tradicionais de animação consistem na criação de modelos matemáticos para a síntese no computador de movimentos existentes no mundo real. Sob esse aspecto, a vantagem da técnica de Captura de Movimento se deve pelo fato dela permitir um “caminho direto” do universo físico para o universo de representação, sem a necessidade de processos computacionais custosos ou sujeitos a erros.

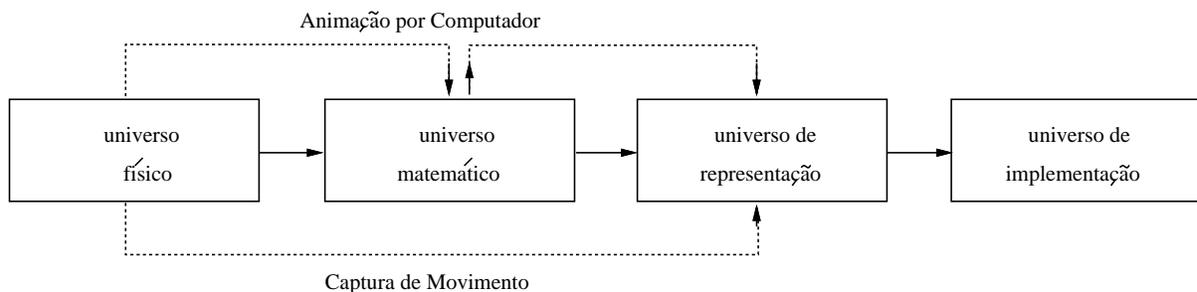


Figura 4.1: O paradigma dos quatro universos e Captura de Movimento.

Como vimos nos capítulos anteriores, o processo de animação consiste na utilização de objetos geométricos no computador, que são comandados por parâmetros cinemáticos, dinâmicos ou comportamentais. O resultado final é uma seqüência de configurações, que

descrevem a variação desses parâmetros ao longo do tempo. Essas etapas podem ser exemplificadas no diagrama apresentado na Figura 4.2.



Figura 4.2: Etapas principais do processo de animação.

Todavia, a passagem de uma etapa para a outra não é direta, e envolve uma série de sub-etapas essenciais para o entendimento do processo de animação. Podemos citar, por exemplo, a etapa de síntese dos parâmetros de animação, que podem ser gerados utilizando-se diversas técnicas existentes atualmente.

Organizando de maneira diferente o diagrama anterior, e inserindo as sub-etapas do processo de animação, chegamos ao diagrama da Figura 4.3. Nas seções seguintes, analisaremos brevemente cada sub-etapa do processo de animação, destacando as suas principais contribuições.

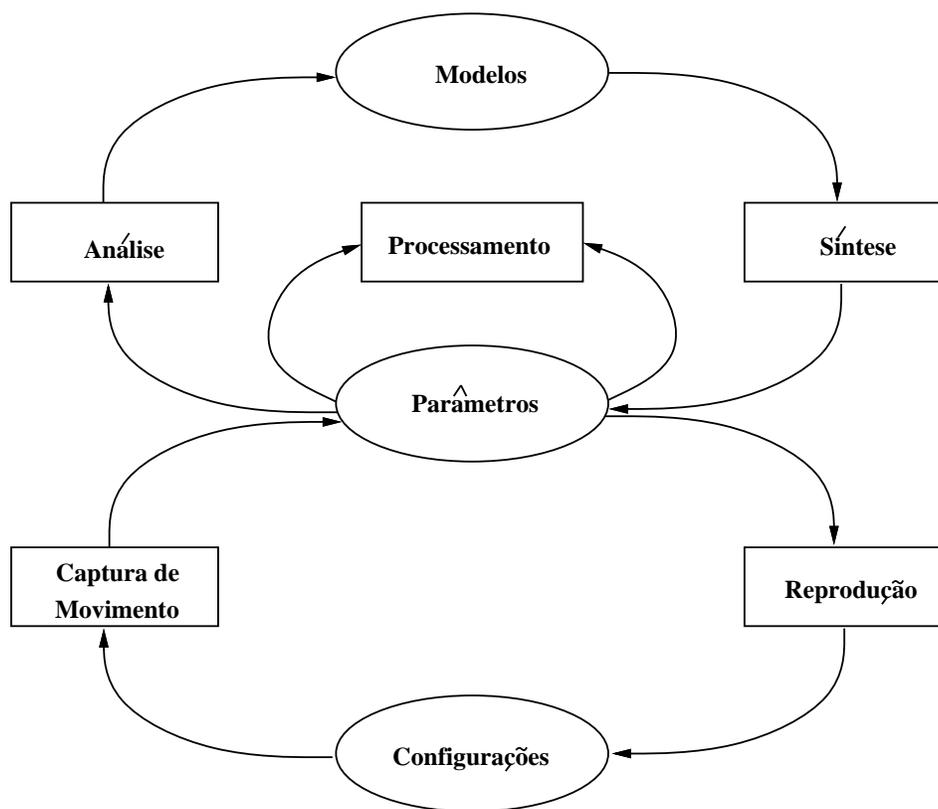


Figura 4.3: Diagrama completo com sub-etapas (retângulos) do processo de animação.

4.2 Síntese

Os parâmetros de animação associados aos objetos geométricos são usualmente gerados (ou sintetizados) através de processos computacionais, onde um modelo matemático é utilizado para representar no computador o movimento (ou mesmo o comportamento) de objetos reais.

A etapa de síntese de parâmetros de animação é a mais estudada atualmente. Ao longo do tempo, diversas técnicas foram desenvolvidas para sintetizar os parâmetros, seja através de métodos cinemáticos (keyframing), dinâmicos (simulação) ou comportamentais (animação comportamental). No Capítulo 5, estudaremos com maior profundidade as principais técnicas desenvolvidas para a síntese de parâmetros de animação.

4.3 Análise

Através da observação de certos aspectos visuais de um objeto, podemos extrair informações importantes sobre o mesmo. Em (Terzopoulos *et al.*, 1987a), foi desenvolvida uma técnica que consegue extrair informações geométricas a partir de uma seqüência de imagens de objetos reais, com o objetivo de inserí-los em um ambiente virtual no computador.

Os conjuntos de técnicas que utilizam seqüências de imagens de objetos reais para extrair informações geométricas e de movimento dos mesmos são conhecidos respectivamente como *Shape from Motion* e *Motion Tracking*, e estão profundamente relacionados com a área de Visão Computacional.

4.4 Processamento

Nem sempre os parâmetros sintetizados são aproveitados de imediato. Ocasionalmente, é necessário fazer alguns ajustes nos dados, processando-os de modo a corrigir ou modificar alguns aspectos, de acordo com o desejo do animador.

Como exemplo prático de processamento de parâmetros de animação, podemos utilizar a técnica de Forward/Inverse Kinematics para modificar a configuração de uma estrutura articulada em um quadro específico de um conjunto de dados gerados por keyframing.

4.5 Reprodução

Na etapa de reprodução, os parâmetros de animação são aplicados aos objetos geométricos no computador. Desta forma, é gerada uma seqüência de configurações dos objetos que estão sendo animados. Posteriormente, essa seqüência de configurações pode ser transformada em imagens, utilizando-se métodos de visualização tradicionais.

4.6 Captura

O processo de Captura de Movimento armazena, ao longo do tempo, o movimento de marcadores especiais posicionados no corpo de um ator. Como veremos no próximo

capítulo, esse processo pode ser feito através da análise de uma seqüência de imagens dos marcadores, feitas por câmeras especiais.

Em uma primeira abordagem, poderíamos pensar em inserir a técnica de Captura de Movimento na etapa de síntese, porém devemos notar que os parâmetros gerados neste processo não são sintetizados, e sim reconstruídos. Conseqüentemente, podemos inserir Captura de Movimento como uma etapa independente do processo de animação.

Capítulo 5

Sistemas de Síntese de Movimento

No Capítulo 4 vimos que os parâmetros de animação podem ser gerados por síntese, através de processos computacionais, ou por reconstrução, através da amostragem do movimento de objetos reais. Neste capítulo, faremos uma breve análise dos principais métodos de síntese de movimento no computador.

5.1 Keyframing

Um dos primeiros métodos de síntese de movimento no computador, a técnica de keyframing, também conhecida como in-betweening, pode ser encarada como a versão computadorizada da técnica desenvolvida por Walt Disney em meados da década de 30.

A técnica de Walt Disney consistia na utilização de desenhistas habilidosos, que criavam os principais quadros de uma animação, assim chamados keyframes. Então, animadores menos habilidosos ficavam responsáveis por criar os desenhos intermediários aos quadros-chave, compondo assim a animação completa.

No computador, o processo não é muito diferente: o animador estabelece os quadros-chave da animação, porém, a geração dos quadros intermediários é feita automaticamente pelo computador através de métodos de interpolação. Devido às limitações do próprio processo de interpolação, esses quadros-chave devem possuir uma certa coerência espacial. Se isso não acontecer, o algoritmo de interpolação pode não ter informação suficiente para completar a seqüência.

A técnica de keyframing foi rapidamente generalizada para permitir a interpolação de qualquer parâmetro cinemático que afete o movimento do objeto, fornecendo assim um maior nível de controle da animação. O processo consiste na interpolação dos parâmetros de animação dos objetos. Através da utilização de funções interpolantes (lineares, quadráticas, splines etc.) ou métodos avançados como quaternions, o parâmetro de interpolação é calculado, gerando as informações que serão utilizadas nos quadros intermediários.

Entretanto, alguns cuidados devem ser tomados durante a escolha do tipo de parâmetro que será interpolado. Um exemplo disto é ilustrado na Figura 5.1. Neste caso, o objetivo da animação consiste na rotação de uma linha pelo fator $\pi/2$. Na imagem da esquerda, o parâmetro de animação que está sendo interpolado é o ângulo de rotação; enquanto que na imagem da direita, o parâmetro interpolado representa a os pontos extremos da linha (este é o processo feito em keyframing tradicional).

Podemos notar neste caso que a interpolação linear entre os pontos extremos da linha

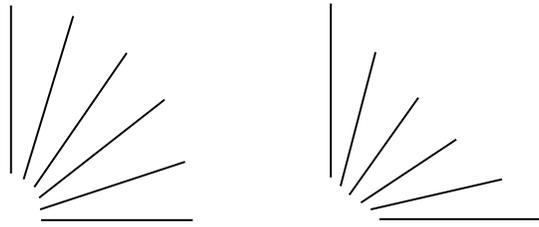


Figura 5.1: Interpolação do ângulo (esq.) e interpolação da posição final e inicial (dir.).

(imagem da direita) gerou um resultado incorreto, sob o ponto de vista visual. Já o resultado da imagem da esquerda, onde foi feita a interpolação do ângulo de rotação, produziu um resultado coerente.

5.2 Animação Procedural

A técnica de animação procedural consiste, por definição, na utilização de funções (ou procedimentos) para controlar algum parâmetro do objeto.

Um exemplo simples de animação procedural é mostrado na Figura 5.2. Nesta animação, o objetivo é fazer a roda girar sobre um plano. Podemos criar uma função cujo objetivo seja aplicar uma composição de transformações para transladar a roda de um fator determinado, fazendo-a girar de forma coerente. Para isso, devemos observar alguns aspectos matemáticos do problema. O comprimento do arco ab definido pelo ângulo \hat{s} é dado por $C_{ab} = \hat{s}r$. Logo, a composição de transformações deve ser a seguinte: para cada passo da animação, girar a roda em torno do eixo que passa pelo seu centro q pelo ângulo \hat{s} , aplicando um fator de translação $\hat{s}r$ na direção paralela ao plano suporte, onde r é o raio da roda.

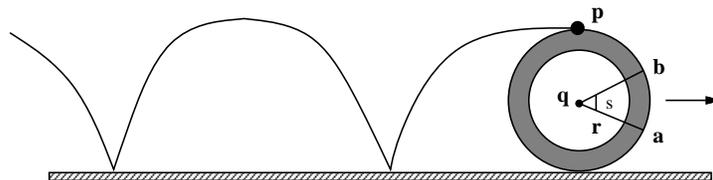


Figura 5.2: Definição de uma série de procedimentos para fazer a roda girar.

Uma aplicação um pouco mais avançada de animação procedural foi desenvolvida em (Perlin, 1995). Neste trabalho, Perlin adaptou técnicas de síntese procedural de texturas, aplicando-as para controlar a animação de atores virtuais. Funções estocásticas de ruído foram utilizadas para “perturbar” a variação angular do movimento das articulações do personagem virtual, conseguindo assim uma aparência mais natural do movimento.

Um exemplo prático de animação procedural foi utilizado no filme *Independence Day* (1996). Neste filme, uma das cenas principais envolvia cerca de 300 aviões e naves extra-terrestres, numa grande batalha aérea sobre o céu de Los Angeles. Para poupar os animadores da criação de cada trajetória manualmente, foram utilizados sistemas de partículas

para animar o movimento dos objetos da cena. Então, funções especiais semelhantes às utilizadas por Perlin foram empregadas para “perturbar” aleatoriamente as trajetórias das partículas, destruindo assim qualquer padrão gerado no movimento original.

5.3 Simulação Dinâmica

A área de animação sempre teve como principal objetivo representar realisticamente o movimento de objetos reais no computador. Com a utilização de keyframing, os animadores têm à sua disposição todo um ferramental para tentar reproduzir tais movimentos. No entanto, nem sempre é possível atingir esse objetivo utilizando apenas parâmetros cinemáticos.

Vejamus por exemplo o caso da bola (Figura 5.3), utilizado no Capítulo 2. Se o objetivo da animação for representar fatores como a gravidade, massa, atrito ou torque, será muito difícil e trabalhoso reproduzir visualmente o resultado real utilizando apenas interpolação de parâmetros cinemáticos.

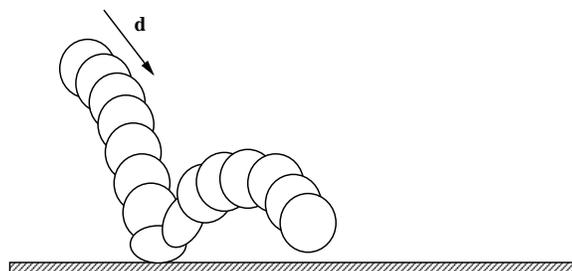


Figura 5.3: A bola caindo: exemplo do emprego de parâmetros dinâmicos na animação.

Neste caso, devemos passar para o nível dinâmico de simulação, incluindo variáveis físicas na representação do movimento. Tais variáveis são descritas por leis físicas, que são encapsuladas através de procedimentos no computador.

De uma forma geral, o processo de simulação dinâmica do movimento de um objeto consiste na especificação de um conjunto de valores (gravidade, coeficiente de atrito, velocidade inicial etc.), que serão utilizados como condições iniciais para as equações dinâmicas. No estágio final do processo, para cada quadro da animação, obtém-se a configuração dos parâmetros regidos pelas equações dinâmicas. Uma desvantagem da técnica de simulação dinâmica é o pouco ou nenhum controle sobre os parâmetros, após o início do processo.

A manipulação dinâmica dos parâmetros de animação possui diversas aplicações, que vão desde a simulação de fenômenos naturais com sistemas de partículas (Reeves, 1983), (Sims, 1990), até a simulação do movimento de humanos sintéticos (Hodgins & Raibert, 1991), (Zeltzer & McKenna, 1990).

5.4 Animação Comportamental

A manipulação de parâmetros comportamentais entre objetos de uma animação consiste, basicamente, na definição de relações entre os objetos e o ambiente onde estão

inseridos. Isto é feito através de procedimentos que têm como objetivo dar ao objeto uma percepção do que está acontecendo à sua volta.

Normalmente, um conjunto de regras é introduzido no comportamento básico do objeto. Tais regras servirão para decidir as reações que serão tomadas pelo objeto com relação aos acontecimentos ao longo da animação. O conjunto de regras não é necessariamente estático, e pode mudar durante a animação, através da utilização de algoritmos evolutivos e conceitos de inteligência artificial.

Um ótimo exemplo de animação comportamental pode ser encontrado em (Terzopoulos, 1994). Neste trabalho, peixes virtuais são inseridos num ambiente aquático tridimensional. Cada peixe possui um sofisticado sistema vital, formado por três módulos principais.

O primeiro módulo - motor - é responsável pela locomoção do peixe no ambiente, e utiliza um sistema massa-mola simplificado para representar a estrutura do peixe. Esse sistema leva em conta características hidrodinâmicas, úteis para simular o movimento do peixe no ambiente aquático.

O módulo seguinte - perceptivo - é responsável pelo recebimento das informações do ambiente ao redor do peixe. Este módulo é formado por sensores especiais, que analisam os objetos e características do ambiente (temperatura, pressão etc.), escolhendo as mais relevantes para posterior processamento no módulo comportamental. A percepção de cada peixe é feita através de um processo de visão artificial, que tem como objetivo simular a visão do peixe no ambiente. A partir de um ponto P (Figura 5.4), localizado na parte frontal do peixe, um ângulo esférico de visão é estabelecido. Dentro desse ângulo, algoritmos especiais avaliam os riscos de colisão do peixe com outros obstáculos ou mesmo a presença de predadores.

No último módulo - comportamental - os dados obtidos pelo módulo perceptivo são processados e analisados. Cada peixe possui um conjunto de parâmetros de comportamento básico, que definem, por exemplo, o grau de fome ou medo que ele possui. No caso do parâmetro “medo”, se o módulo perceptivo avistar um predador no campo visual do peixe, o módulo comportamental avaliará a situação. Caso o predador represente uma ameaça real, o módulo de comportamento enviará ao módulo motor um conjunto de parâmetros dinâmicos, que terão como objetivo desviar o peixe da trajetória do predador. Os procedimentos para a análise dos parâmetros e as tomadas de decisão são compostos por sistemas de equações que levam em conta as variáveis recebidas pelo sistema perceptivo.

As aplicações de animação comportamental são vastas, e vão desde métodos de interação entre humanos virtuais (Perlin & Goldberg, 1996), (Costa & Feijó, 1996), até a simulação de grupos de animais (Reynolds, 1987) e multidões de pessoas (Musse & Thalmann, 1997).

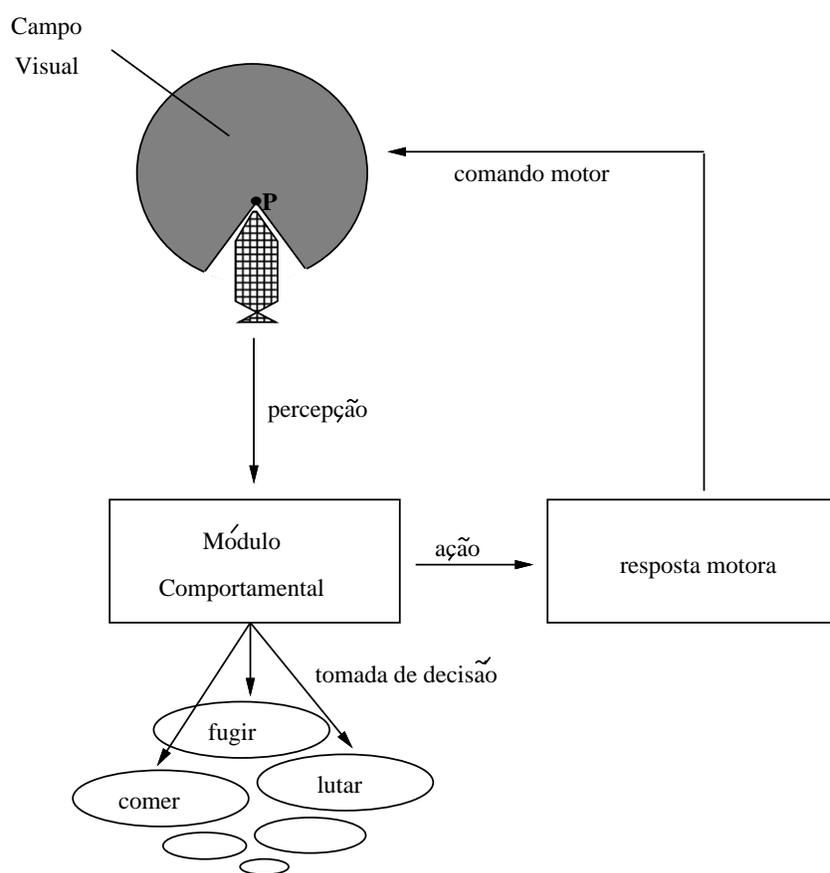


Figura 5.4: Papel do módulo comportamental no processo de simulação de peixes.

Capítulo 6

Sistemas de Captura de Movimento

Neste capítulo, faremos um relato sobre o estágio atual da tecnologia e dos equipamentos utilizados no processo de Captura de Movimento, classificando-os em relação a diversos aspectos tecnológicos.

Sistemas de Captura de Movimento são, por definição, conjuntos de equipamentos que armazenam no computador informações relativas a medidas físicas do movimento de um objeto real (geralmente um ator, ou *performer*). O processo de captura consiste, de uma forma geral, em vestir um ator com uma roupa especial, onde são posicionados refletores (sistemas óticos), transmissores (sistemas magnéticos) ou potenciômetros (sistemas mecânicos). Esses marcadores são geralmente posicionados nas principais articulações do corpo humano, também chamadas de “articulações universais”, que representam as localizações no corpo que fornecem uma precisão razoável para a representação de um movimento humano.

Podemos dividir os sistemas de captura em quatro categorias: mecânicos, óticos, magnéticos e acústicos. Atualmente, os sistemas mais utilizados são os baseados nas tecnologias ótica e magnética, e serão o alvo principal de nosso estudo. Em (da Silva, 1997a) e (Mulder, 1994), é possível encontrar informações mais detalhadas sobre a tecnologia de Captura de Movimento.

Classificação

Podemos utilizar diversas classificações para os sistemas de captura atuais. A seguir, analisaremos três maneiras distintas de classificação: quanto à tecnologia utilizada, quanto à posição da fonte emissora e dos sensores/marcadores, e quanto à forma de obtenção dos dados capturados.

6.1 Tecnologia

Nesta seção, faremos uma análise e classificação dos equipamentos da captura com relação ao método empregado para obter os dados de movimento.

Este tipo de classificação é muito importante pois define explicitamente o processo utilizado para a aquisição dos movimentos. Nesta classificação, estudaremos o estado atual da tecnologia de captura de movimento, juntamente com os prós e contras de cada tipo de sistema.

6.1.1 Sistemas Mecânicos

Sistemas mecânicos são compostos por potenciômetros que, posicionados nas articulações desejadas, fornecem suas posições e orientações em altas taxas de amostragem (tempo real). Uma das vantagens desse tipo de sistema é o fato de possuírem uma interface parecida com as utilizadas em equipamentos de *stop-motion*, amplamente empregados na produção de filmes. Com isso, uma grande quantidade de animadores que utilizavam *stop-motion* puderam migrar para essa nova tecnologia sem problemas, aumentando a popularidade desse tipo de sistema. Como exemplos dessa aplicação, podemos citar os dinossauros do filme *Jurassic Park* (1993) e os insetos invasores do filme *Starship Troopers* (1997) (Figura 6.1), cujos movimentos foram gerados com o auxílio de sistemas mecânicos de captura de movimento.

Os sistemas mecânicos de captura não contam com uma tecnologia muito avançada, porém possuem algumas vantagens que os tornam extremamente atraentes, especialmente para a indústria cinematográfica. Geralmente, esse tipo de sistema é bastante obtrusivo, talvez até mais que os sistemas magnéticos com cabos (analisados a seguir).

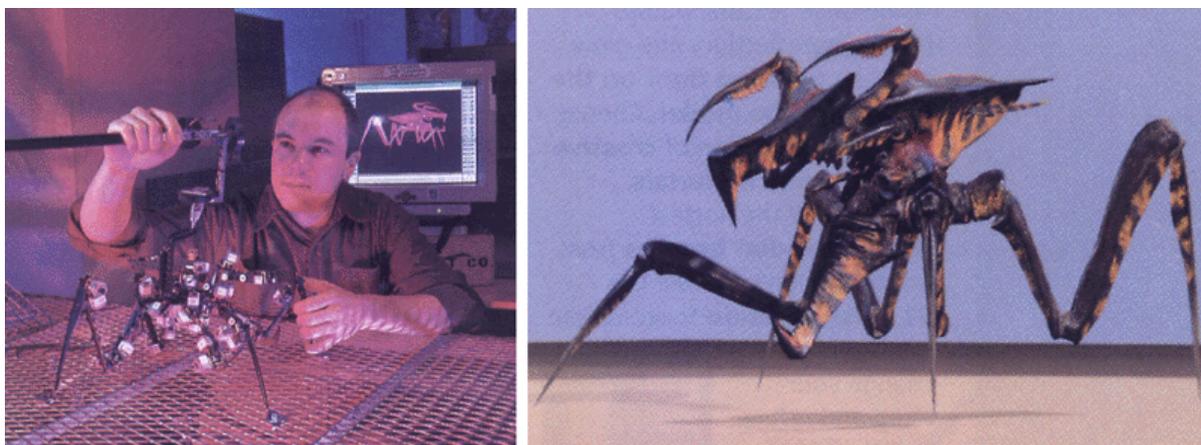


Figura 6.1: Utilização de equipamentos mecânicos de captura no filme *Starship Troopers*.

Sistemas mecânicos são equipamentos de medida absoluta, não sendo afetados por campos magnéticos ou reflexões indesejadas. Conseqüentemente, não necessitam de um processo longo de calibragem, o que torna a sua utilização mais fácil e produtiva.

Como exemplos de sistemas mecânicos de captura, podemos citar o *The Monkey - Digital Poseable Mannequin* e o *Animatton*. O primeiro pode ser considerado como um “*stop-motion* assistido por computador”. Nesse sistema, o “manequim” pode ser qualquer objeto desejado - desde humanóides até dinossauros e insetos. Um esqueleto articulado composto por potenciômetros é então fixado no modelo, fornecendo assim as posições e orientações impostas ao modelo pelo animador. Uma desvantagem deste sistema é o realismo, que depende exclusivamente da habilidade do animador em posicionar cada articulação do modelo corretamente, num processo que requer, além de técnica, um pouco de arte. Esse tipo de sistema mecânico é considerado por muitos como uma interface em hardware para a técnica de keyframing, pois a configuração capturada das articulações do esqueleto mecânico é geralmente utilizada como quadro-chave para posterior interpolação e geração da animação.

Uma nova tecnologia em sistemas mecânicos de captura foi desenvolvida pela Analo-gus, com o sistema *Animatton*. Este sistema, desenvolvido exclusivamente para a captura de movimentos humanos, é composto por articulações e potenciômetros que são fixados diretamente no corpo do ator, formando uma verdadeira “armadura”. A captura é feita em tempo real e o sistema requer pouca ou nenhuma calibragem. Além disso, permite a captura de movimentos de múltiplos atores numa mesma sessão. Apesar de ser extremamente obtrusivo (Figura 6.2), este sistema possui vantagens como o preço (em torno de US\$35.000), que o torna acessível para a maioria das produtoras que se utilizam de Movimento Capturado. Na Tabela 6.1, apresentamos algumas características de sistemas mecânicos de captura.

	<i>sistemas mecânicos</i>
<i>resolução espacial</i>	independente do alcance, $\sim 0.5 \rightarrow 1.0$ graus
<i>acurácia espacial</i>	≤ 5 graus
<i>precisão</i>	média \rightarrow elevada
<i>taxa de amostragem</i>	> 120 FPS <i>Frames per Second</i>
<i>workspace</i>	ilimitado
<i>exemplos de sistemas</i>	<i>Animatton, Digital Monkey, Power Glove</i>
<i>softwares compatíveis</i>	Alias Wavefront, Jack, 3D Studio MAX, Softimage
<i>preço</i>	US\$1,000 a US\$35,000

Tabela 6.1: Informações técnicas de sistemas mecânicos de captura.



Figura 6.2: Conjunto de potenciômetros no sistema de captura mecânico *Animatton*.

6.1.2 Sistemas Magnéticos

Os sistemas magnéticos de captura se caracterizam pela velocidade em tempo real de processamento dos dados capturados. Neste tipo de sistema, emprega-se um conjunto de receptores que são posicionados nas articulações do ator. Tais receptores medem a posição e orientação das articulações no espaço, em relação a uma antena transmissora que emite um sinal de pulso. Cada receptor necessita de um cabo para se conectar à antena.

Os sistemas magnéticos são mais acessíveis em termos de preço. Sua tecnologia também é menos sofisticada que a utilizada nos sistemas óticos. Existem sistemas magnéticos simples e sofisticados, que vão desde US\$ 5,000 (*Polhemus InsideTrak*) a US\$ 33,000 (*Ascension Flock of Birds*). O preço desse tipo de sistema depende basicamente do número de receptores que conseguem ser processados simultaneamente. Pode-se, por exemplo, adquirir um *Flock of Birds* com poucos receptores por até US\$ 3,000.

Os sistemas magnéticos mais conhecidos são o *Ultratrak Pro*, da Polhemus e o *Flock of Birds*, da Ascension. Neste último, por exemplo, um conjunto de até 30 receptores é conectado ao ator, fornecendo a orientação e posição de suas principais articulações.

Algumas vantagens dos sistemas magnéticos são o baixo custo computacional para o processamento dos dados, maior precisão (não existem problemas de oclusão de receptores) e o baixo custo do equipamento. Com uma taxa de amostragem que gira em torno de 100 quadros por segundo, os sistemas magnéticos são ideais para a captura de movimentos mais simples.

A maior desvantagem deste tipo de equipamento são os diversos cabos que conectam os receptores à antena. Tais cabos restringem o movimento do ator, não permitindo que movimentos complexos e rápidos possam ser representados com naturalidade. Felizmente, algumas empresas já estão desenvolvendo sistemas magnéticos que não possuem cabos.



Figura 6.3: Sistema magnético de captura *Motion Star*.

Uma outra desvantagem do processo magnético é a interferência causada por objetos de metal próximos ao local de captura. Campos magnéticos são extremamente sensíveis a objetos metálicos. Até mesmo a estrutura do prédio pode causar alguma interferência. Na Tabela 6.2, apresentamos algumas características desse tipo de sistema.

6.1.3 Sistemas Óticos

Neste tipo de sistema, o ator veste uma roupa especial coberta com refletores (em geral, esferas revestidas com material super-refletivo), posicionados nas suas principais articulações (Figura 6.4). Câmeras especiais são então posicionadas estrategicamente para fazer o tracking desses refletores durante o movimento do ator. Cada câmera gera as coordenadas bidimensionais para cada refletor (obtidas via processo de segmentação). O conjunto dos dados 2D capturados pelas câmeras independentes é então analisado

	<i>sistemas magnéticos</i>
<i>resolução espacial</i>	~ 0.005 a 8 mm; ~ 0.025 a 0.1 graus
<i>acurácia espacial</i>	~ 0.8 a 5 mm; ~ 0.1 a 3 graus
<i>precisão</i>	elevada
<i>taxa de amostragem</i>	~ 100 FPS
<i>workspace</i>	raio de 1 a 2 m
<i>exemplos de sistemas</i>	<i>Flock of Birds, ULTRATRAK</i>
<i>softwares compatíveis</i>	Alias Wavefront, Jack, 3D Studio MAX, Softimage
<i>preço</i>	US\$5,000 a US\$70,000

Tabela 6.2: Informações técnicas de sistemas magnéticos de captura.

por um software que, através de um processo de triangulação, fornecerá as coordenadas tridimensionais dos refletores.



Figura 6.4: Sistema ótico de captura de movimento.

Existem algumas variações entre diferentes sistemas óticos de captura. No sistema *MultiTrax*, da Adaptive Optics Associates, os marcadores posicionados nas articulações são refletores que, iluminados por flashes sincronizados de luz infra-vermelho, são capturados por uma ou mais câmeras. Neste tipo de equipamento ótico também podem ser utilizados marcadores de outros tipos (focos de laser e retro-prismas).

O sistema *Optotrak*, da Northern Digital, utiliza marcadores LED sincronizados, que são rastreados por três câmeras sensíveis ao espectro infra-vermelho. O software presente no sistema utiliza os dados fornecidos pelas câmeras para gerar as coordenadas tridimensionais, orientações e momentos dos marcadores. Um problema gerado pelos marcadores LED é o aquecimento, o que dificulta a sua utilização por longos períodos de tempo.

O sistema *ELITE*, da Bioengineering Tech., utiliza marcadores refletivos passivos (hemisférios de plástico cobertos com material refletivo) e câmeras eletrônicas CCD com LED's infra-vermelhos em volta das lentes. O software fornecido no sistema utiliza algoritmos de reconhecimento de formas (do tipo *shape from shading*) para auxiliar o processo

de tracking dos marcadores (Azarbayerjani & Wren, 1996). Existem ainda muitos outros sistemas óticos no mercado. Entre eles, podemos citar o *HiRES ExpertVision 3D* (Motion Analysis Corp.) e o *MacReflex* (Optikon Corp.).

Os equipamentos óticos de captura são os mais caros existentes atualmente, podendo chegar a até US\$250,000. Seu preço elevado é resultado da alta tecnologia utilizada no processo, que conta com câmeras de alta resolução e software de última geração.

Uma das vantagens da utilização de sistemas óticos é a alta taxa de amostragem, que permite a captura de movimentos rápidos como os utilizados em artes marciais e esportes olímpicos. A taxa de amostragem depende basicamente da capacidade de definição das câmeras utilizadas no processo. Em outras palavras, quanto maior resolução tiverem as câmeras, maior será a taxa de amostragem. Taxas de amostragem de até 200 quadros por segundo são conseguidas neste tipo processo quando se utilizam câmeras especiais de alta resolução e velocidade. Na Tabela 6.3, apresentamos algumas informações técnicas complementares sobre os sistemas óticos.

	<i>sistemas óticos</i>
<i>resolução espacial</i>	~ 0.0015 a 0.2% do FOV
<i>acurácia espacial</i>	~ 0.004 a 0.5% do FOV
<i>precisão</i>	~ 0.0055 a 0.02% do FOV
<i>taxa de amostragem</i>	> 200 FPS
<i>workspace</i>	raio de 1 a 4 m
<i>exemplos de sistemas</i>	<i>Multitrax, OPTOTRAK</i>
<i>softwares compatíveis</i>	ZoeTrax, Softimage, PowerAnimator V7
<i>preço</i>	US\$20,000 a US\$150,000

Tabela 6.3: Informações técnicas de sistemas óticos de captura.

Outra vantagem dos sistemas óticos é a liberdade que oferecem ao ator durante a execução do movimento. Ao contrário dos sistemas magnéticos, onde o ator é coberto de transmissores e fios, os refletores não oferecem resistência aos movimentos do ator. Além disso, não existe limite para o número de refletores posicionados no ator, o que permite alcançar um maior nível de detalhe na representação dos movimentos. O espaço de trabalho (*workspace*) do processo de captura para sistemas óticos é muito superior ao dos sistemas magnéticos e depende, basicamente, do campo de visão (FOV) das câmeras utilizadas no processo. Um cuidado, porém, deve ser tomado: a iluminação global do ambiente deve ser calibrada de modo a não causar interferências no processo de tracking dos refletores.

A grande desvantagem dos sistemas óticos é a oclusão de um ou mais refletores durante o processo de captura. Este tipo de problema é mais freqüente durante a captura de movimentos de objetos pequenos (dedos das mãos) ou de vários atores interagindo muito próximos uns dos outros. Neste caso, a recuperação da posição tridimensional do refletor pode ser impossível, o que compromete todo o processo.

O problema de oclusão pode ser minimizado com a utilização de um número maior de câmeras e de refletores. Porém existe um balanço entre esses fatores: um maior número de câmeras acarreta um maior tempo de processamento para a CPU, durante o tracking dos refletores. Ao se aumentar o número de refletores, surge o problema de “*tracking*”

confusion”, ou seja, a dificuldade de identificar os refletores que estão muito próximos. Este problema é diretamente influenciado pela resolução das câmeras - câmeras de maior resolução conseguem identificar melhor refletores próximos.

Existe ainda a necessidade de um processamento via software dos dados obtidos pelas câmeras para a obtenção das informações finais, o que não permite a interatividade, importante para a velocidade de produção e redução de custos operacionais. A utilização de software para o processamento dos dados obtidos pelas câmeras introduz erros de precisão (*offset*) no processo de obtenção dos dados. Em alguns casos, é necessário um processo de filtragem dos dados finais para a eliminação dos erros de precisão e dos ruídos introduzidos pelo tracking das câmeras.

6.1.4 Sistemas Acústicos

Neste tipo de sistema, um conjunto de emissores sonoros é colocado nas principais articulações do ator, enquanto que três receptores sensíveis são posicionados no local de captura. Os transmissores são então seqüencialmente acionados para produzir um ruído característico, que será captado pelos transmissores, calculando assim as suas posições no espaço.

O cálculo da posição de cada transmissor é feito da seguinte forma: utilizando como dados iniciais o tempo decorrido entre a emissão do ruído pelo transmissor e o seu recebimento pelo receptor, e a velocidade do som no ambiente, consegue-se calcular a distância percorrida pelo som, do transmissor até cada um dos receptores. Para calcular a posição tridimensional de cada transmissor, é feita uma triangulação das distâncias deles em relação aos três receptores.

Um dos problemas deste método é a dificuldade de obter uma descrição correta dos dados num instante desejado, devido ao caráter seqüencial do disparo dos transmissores no corpo do ator. Além disso, os sistemas acústicos sofrem do mesmo tipo de problema existente em sistemas magnéticos: os incômodos cabos, que prejudicam a movimentação do ator, reduzindo assim o escopo de movimentos que podem ser executados. O número de transmissores que podem ser utilizados simultaneamente também é limitado, o que pode não fornecer uma descrição suficientemente correta do movimento capturado.

Sistemas acústicos estão sujeitos a problemas causados pela própria tecnologia: reflexões do som emitido pelos transmissores ou ruídos externos podem afetar o processo de captura e destruir os dados obtidos. Em compensação, este tipo de sistema não possui problemas de oclusão, típico de sistemas óticos; e interferência por objetos metálicos, percebido em sistemas magnéticos.

Somando vantagens e desvantagens, os sistemas acústicos devem ser considerados como a “última alternativa” na hora da aquisição de um equipamento de captura de movimento.

6.2 Fonte Emissora vs. Sensores/Marcadores

Uma classificação interessante, proposta em (Mulder, 1994), divide os sistemas de captura em três categorias: *inside-in*, *inside-out* e *outside-in*. A semântica desta classificação é simples: a primeira palavra refere-se à localização do sensor ou marcador no sistema (*inside*, se o sensor está no corpo do ator; e *outside*, caso esteja fora). Já a segunda

palavra indica a posição da fonte emissora (por exemplo, a câmera dos sistemas óticos), em relação ao corpo do ator.

6.2.1 Sistemas Inside-in

Neste tipo de sistema, os sensores e a fonte transmissora estão localizados no próprio corpo do performer. Um exemplo de equipamento deste tipo é a *Power Glove* (Nintendo-Mattel), que utiliza sensores flexíveis para captar orientações. Os sensores deste tipo de sistema geralmente possuem um baixo “fator de forma”¹ e, conseqüentemente, são os mais indicados para a captura de movimentos de objetos pequenos (dedos das mãos, olhos etc.). Seu campo de trabalho é ilimitado devido à sua restrição espacial ao corpo do ator, porém são considerados extremamente obtrusivos. As coordenadas obtidas por sistemas deste tipo são, geralmente, desprovidas de informação posicional, fornecendo somente orientações. Sistemas mecânicos de captura, como o *Digital Monkey*, estão inseridos nesta categoria.

6.2.2 Sistemas Inside-out

São sistemas onde os sensores estão conectados ao corpo do ator, respondendo a sinais emitidos por uma fonte emissora externa. Sistemas magnéticos e acústicos estão nesta categoria. O espaço de trabalho e a precisão de sistemas *inside-out* são limitados devido ao uso de fontes externas, mas os dados obtidos fornecem uma boa descrição espacial dos sensores/marcadores. Além disso, o seu “fator de forma” mais alto restringe sua utilização à captura de partes maiores do corpo.

6.2.3 Sistemas Outside-In

Em sistemas deste tipo, a fonte emissora está localizada no corpo do ator, enquanto que os sensores externos capturam seus sinais. Em sistemas óticos, a fonte emissora são os marcadores cobertos com material refletivo, que refletem a luz emitida por flashes. Tais reflexões são captadas pelos sensores, que são as câmeras. Sistemas deste tipo sofrem do problema de oclusão e possuem um espaço de trabalho limitado, porém possuem o menor grau de obtrusividade. Outro exemplo de sistemas *outside-in* são os utilizados para o tracking da pupila e da córnea.

6.3 Aquisição Direta vs. Indireta

Essa é uma classificação importante, pois fornece uma indicação de como são obtidos os dados capturados: direta ou indiretamente.

6.3.1 Sistemas de Aquisição Direta

São aqueles cujos dados obtidos não necessitam de nenhum processamento posterior à captura. Em geral, sistemas de aquisição direta são os mais obtrusivos e possuem

¹Medida quantitativa de tamanho.

uma baixa taxa de amostragem. Estão incluídos nessa categoria os sistemas acústicos, mecânicos e magnéticos.

6.3.2 Sistemas de Aquisição Indireta

Os sistemas inseridos nessa classificação permitem uma maior liberdade ao performer e possuem uma alta taxa de amostragem. Porém, devido a sua alta tecnologia, são extremamente caros. Nesse tipo de sistema, os dados obtidos durante o processo são agrupados e analisados, geralmente através de um software poderoso, de modo a gerar o produto final. Este fato inviabiliza a utilização de sistemas deste tipo em aplicações que necessitem do fator interatividade, porém os tornam mais precisos e mais indicados para a captura de movimentos complexos e rápidos. Sistemas óticos se encaixam nesta classificação.

Capítulo 7

Os Dados Capturados

Nos capítulos anteriores, vimos que os equipamentos de captura armazenam os dados de movimento através de um processo de amostragem. Neste capítulo, estudaremos a composição interna e a estrutura desses dados, dando especial importância à técnica desenvolvida para mapeá-los em uma estrutura articulada hierárquica, componente fundamental do ator virtual utilizado em nosso sistema.

7.1 Composição e Representação

Embora atualmente exista uma boa variedade de tecnologias de captura de movimento, o núcleo fundamental do processo é sempre o mesmo: armazenar os movimentos através da amostragem uniforme (Figura 7.1¹) da posição ou orientação de marcadores posicionados no corpo de um ator. Este processo é semelhante ao utilizado na amostragem de funções matemáticas contínuas. Porém no caso dos movimentos o equipamento registra, em intervalos de tempo uniforme, a configuração instantânea de cada marcador, gerando assim um conjunto de valores seqüenciais relativos à discretização da “curva real” do movimento do marcador em questão.

No computador, esse conjunto de valores seqüenciais pode ser reconstruído de modo a gerar uma representação contínua das curvas de movimento dos marcadores. Tal reconstrução pode ser feita utilizando-se funções de interpolação (linear, quadrática, cúbica etc.). Em nosso sistema de animação, utilizaremos o processo de interpolação linear para obter a aproximação das curvas de movimento dos marcadores.

Os dados registrados por equipamentos de captura de movimento são, geralmente, armazenados no computador sob a forma de arquivos-texto comuns. Cada arquivo contém, para cada marcador utilizado no processo, a seqüência de valores relativos à amostragem de suas curvas de movimento.

De acordo com o tipo de tecnologia utilizada, os dados obtidos podem conter apenas informação posicional dos marcadores (sistemas óticos), ou conter informação posicional e angular dos marcadores (sistemas magnéticos). Esses dados são estruturados no arquivo de modo a fornecer, para cada marcador, o conjunto de valores amostrados em cada passo do processo de captura. Cada conjunto de valores é geralmente associado a um quadro de animação durante a visualização das curvas de movimento no computador.

¹A imagem desta figura foi gentilmente cedida pelo autor de (de Figueiredo, 1996).

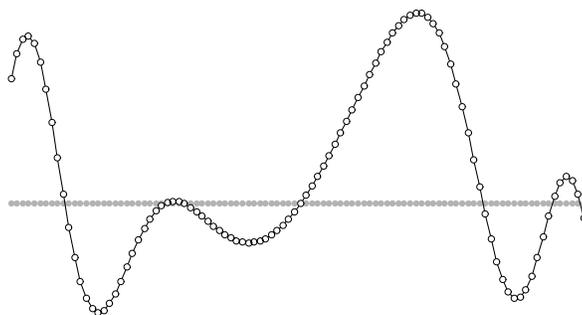


Figura 7.1: O processo de captura: amostragem uniforme das curvas de movimento.

Posteriormente, esse conjunto de valores será traduzido para a forma de estruturas de dados mais adequadas para o processo de animação.

O fragmento de arquivo apresentado a seguir contém informações sobre o movimento do marcador posicionado na articulação do ombro de um ator, obtida por um equipamento magnético de captura.

```
Segment:    LeftUpArm
Frames:     3
Frame Time: 0.033333
XTRAN      YTRAN      ZTRAN      XROT      YROT      ZROT
INCHES     INCHES     INCHES     DEGREES   DEGREES   DEGREES
3.33       50.98      13.71      6.44      -51.28    51.02
1.72       50.93      20.70      14.81     -55.30    59.06
0.27       51.08      27.83      22.49     -58.50    67.15
```

Embora cada tipo de equipamento de captura armazene os dados nos arquivos de uma forma diferente, a estrutura básica de tais arquivos permanece a mesma. No exemplo anterior, a identificação do marcador, o número de amostras (*frames*) e o tempo decorrido em segundos entre cada amostra (*frame time*) são fornecidos no cabeçalho que antecede cada bloco de dados relativos aos marcadores. Os dados de movimento são divididos verticalmente em canais (no caso do exemplo, três canais para translação e três para orientação global do marcador), onde cada linha do bloco de dados representa a configuração do marcador em um instante do processo de captura.

7.2 Orientação Global vs. Orientação Relativa

Dependendo do tipo de equipamento utilizado, os dados de movimento de cada marcador podem ser capturados em relação a um sistema global fixo de coordenadas (sistemas óticos, magnéticos e acústicos), ou em relação a um sistema de coordenadas local relativo aos marcadores vizinhos (sistemas mecânicos).

Em ambos os casos, a visualização do movimento é facilmente obtida posicionando-se objetos geométricos utilizando os valores existentes no arquivo de dados capturados. No entanto, para o caso da captura em relação a um sistema de coordenadas global (o que acontece na grande maioria dos equipamentos existentes atualmente), qualquer tipo de manipulação para modificar os dados capturados se torna impraticável. Isto ocorre pois

não é possível manter o vínculo entre os valores dos marcadores utilizados para orientar os segmentos que representam o ator virtual. Com isso, qualquer mudança na posição ou orientação de um segmento causará uma ruptura na estrutura articulada.

Para contornar esse problema, a melhor solução é mapear os dados em uma estrutura articulada hierárquica. Desta forma, a existência de vínculos topológicos e geométricos garante a integridade da estrutura e permite a utilização de diversas técnicas tradicionais para a manipulação de cadeias articuladas.

Todavia, os dados originais de movimento não podem ser diretamente aplicados na estrutura hierárquica. Isto acontece pois esse tipo de estrutura necessita de dados que possuam uma certa relação entre si, da mesma forma que os segmentos são relacionados hierarquicamente na topologia do esqueleto. Os dados fornecidos nos arquivos de movimento não possuem qualquer tipo de relação implícita, portanto é necessário algum tipo de processamento para criar tais relações, baseadas na topologia da estrutura hierárquica.

7.2.1 Extração de Ângulos Relativos

O processamento necessário para aplicação dos dados na estrutura hierárquica consiste na conversão da informação de posição dos marcadores para ângulos relativos entre os segmentos da estrutura.

Em uma estrutura articulada hierárquica, um ângulo relativo entre dois segmentos pode ser descrito matematicamente como a configuração angular do segundo segmento em relação ao sistema de coordenadas do primeiro segmento. Este processo pode ser visualizado na Figura 7.2. Neste exemplo, o ângulo relativo θ representa a rotação do segmento 2 em relação ao sistema de coordenadas definido pelo segmento 1. Ao girarmos o segmento 1 (Figura 7.2, direita), seu sistema de coordenadas também será rotacionado. Desta forma, a orientação do segmento 2, que é relativa ao sistema de coordenadas do primeiro segmento, também será afetada pelo fator de rotação aplicado em 1.

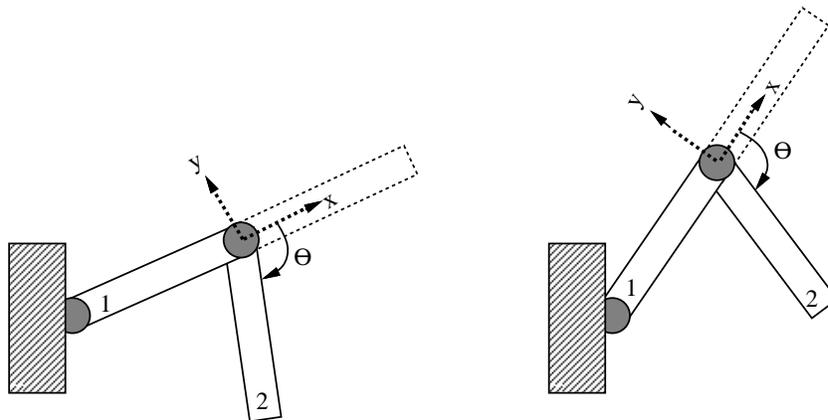


Figura 7.2: Ângulo relativo entre dois segmentos de uma estrutura articulada hierárquica.

Este processo pode ser estendido para estruturas hierárquicas com diversos segmentos e inúmeros graus de liberdade. Assim, uma matriz de rotação M_{rot} aplicada a um nó interno A da árvore que representa a estrutura hierárquica, será também aplicada em toda a sub-árvore de A , e assim por diante (Figura 7.3).

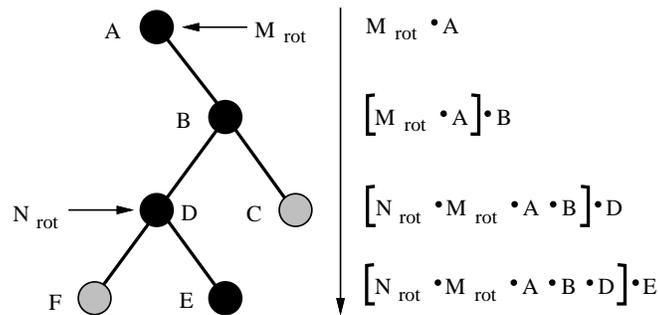


Figura 7.3: Expansão da composição de transformações em uma estrutura articulada hierárquica.

No caso do ator virtual utilizado em nosso sistema de animação, cada segmento do corpo possui um sistema de coordenadas tridimensional associado à sua articulação, contendo uma matriz que representa o conjunto de transformações geométricas aplicadas ao segmento.

O método de cálculo de ângulos relativos para o esqueleto do ator virtual, desenvolvido neste trabalho, utiliza conceitos de geometria computacional (O'Rourke, 1994). Podemos dividir o método em duas etapas, descritas a seguir.

• ETAPA 1: Orientação do quadril e complexo inferior do corpo

A articulação do quadril é responsável pelo controle de toda a parte inferior do corpo do ator virtual, ou seja, as pernas. Além disso, os dados relativos ao quadril podem ser utilizados para controlar a posição e orientação do esqueleto inteiro no espaço.

O cálculo da orientação do quadril utilizando apenas a informação posicional relativa a essa articulação é impossível, uma vez que não faz sentido falar em orientação para um ponto no espaço. Para contornar esse problema, podemos utilizar a própria estrutura do corpo humano como ferramenta para calcular a orientação do quadril.

Como podemos observar na Figura 7.4, a orientação do quadril pode ser definida pelo plano formado por três marcadores: os dois marcadores posicionados na parte superior das pernas do ator, mais o marcador posicionado no próprio quadril. Calculando a orientação deste plano em relação a um sistema de coordenadas fixo na origem, obtemos a orientação do quadril no espaço.

A perna do ator virtual é composta por três segmentos: *upleg*, *lowleg* e *foot*, como mostrado na Figura 7.4. Para as pernas, o cálculo dos ângulos relativos é feito utilizando-se processos geométricos. Para exemplificar o processo, consideraremos apenas a perna direita do ator. O cálculo para a perna esquerda é exatamente igual.

O ângulo relativo entre os segmentos *hips* e *upleg* é calculado através da projeção do vetor unitário \vec{ab} , definido pelas articulações dos segmentos *upleg* e *lowleg* (Figura 7.5), no sistema de coordenadas definido pela orientação do quadril, calculada anteriormente. Projetando-se \vec{ab} nos planos coordenados, obtemos os ângulos de Euler relativos entre os dois segmentos em questão. No exemplo da Figura 7.5, o ângulo relativo θ_1 foi obtido projetando-se \vec{ab} no plano XY.

O cálculo do ângulo relativo entre os segmentos *upleg* e *lowleg* é facilitado devido às simplificações impostas ao modelo que representa o corpo humano, descritas no Capítulo 3.

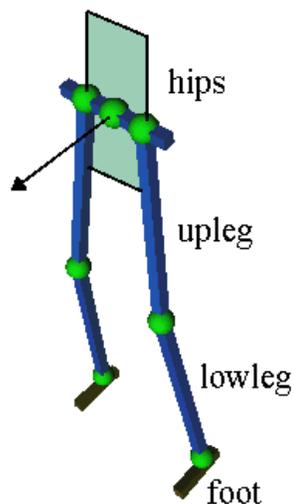


Figura 7.4: Utilização do plano formado por três marcadores no cálculo da orientação do quadril.

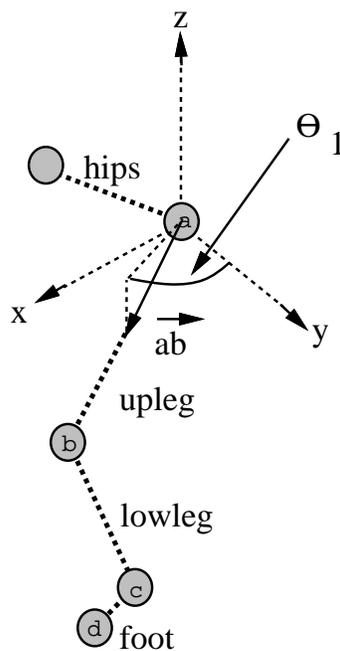


Figura 7.5: Cálculo do ângulo relativo entre o quadril e a coxa.

Nesse caso, a articulação que liga esses dois segmentos possui apenas 1 DOF, relativo à rotação em torno do eixo X que a atravessa perpendicularmente em relação ao seu comprimento (Figura 7.6). Desta forma, o ângulo relativo θ_2 é obtido através do produto escalar entre os vetores \vec{ba} e \vec{bc} , ou seja, $\theta_2 = \cos^{-1}(\|\vec{ba}\| \cdot \|\vec{bc}\|)$.

Finalmente, o ângulo relativo θ_3 entre os segmentos *lowleg* e *foot* é calculado através de um processo semelhante ao utilizado no cálculo de θ_1 . No entanto, alguns tipos de

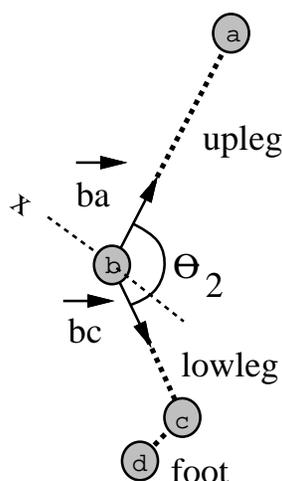


Figura 7.6: Cálculo do ângulo relativo entre os segmentos *upleg* e *lowleg*.

arquivo de movimento não incluem a informação do último marcador do corpo do ator (na Figura 7.6, a articulação *d*). Desta forma, a orientação do pé fica indeterminada. Para contornar esse problema de forma razoável, e assim obter uma orientação “falsa” (porém aceitável) para o pé, podemos fixar o seu ângulo relativo em 90 graus, fazendo com que ele fique sempre perpendicular ao segmento *lowleg*.

- **ETAPA 2: Orientação do peito e complexo superior do corpo**

A articulação do peito é responsável pelo controle de todo o complexo superior do corpo do ator virtual, ou seja, braços, pescoço e cabeça.

O processo utilizado para obter a orientação do peito é semelhante ao aplicado na extração do ângulo do quadril. Porém desta vez o plano que define este ângulo é formado pelos marcadores posicionados na parte superior dos braços e do marcador do próprio peito, como mostra a Figura 7.7.

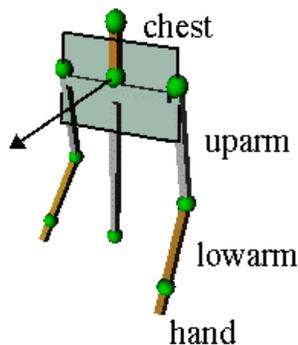


Figura 7.7: Utilização do plano formado por três marcadores no cálculo da orientação do peito.

Para os segmentos *uparm*, *lowarm* e *hand* o processo é o mesmo que foi utilizado no cálculo dos ângulos relativos das pernas, na etapa 1.

Mapeamento no Ator Virtual

O processo de extração dos ângulos relativos descrito anteriormente é feito para cada quadro de animação definido no arquivo de movimento. Embora para um único quadro o custo computacional não seja grande, no total o processo possui um certo custo computacional, pois um arquivo comum de movimento contém geralmente cerca de 600 quadros. Por isso, em nosso sistema de animação, a etapa de extração de ângulos relativos é feita logo após o carregamento do arquivo de movimento pelo usuário. Após isso, os ângulos são mapeados nas devidas articulações da estrutura articulada hierárquica.

7.3 Pós-processamento

Após a obtenção dos dados de movimento, geralmente é necessário fazer uma verificação da integridade dos valores obtidos no processo. Isto ocorre pois a maioria dos equipamentos de captura introduz algum tipo de ruído no conjunto de dados obtidos, quase sempre causado por interferência externa. Para minimizar os efeitos dos ruídos são utilizados filtros de passa-baixa, especialmente construídos pelos fabricantes dos equipamentos. Tais filtros têm como objetivo “suavizar” o conjunto de dados através da eliminação de altas frequências específicas, causadas pelo ruído.

Um outro tipo de processamento também integrado a sistemas de captura é a “eliminação de buracos” nos dados, que consiste em detectar falhas no processo de amostragem e corrigi-las através de algoritmos de interpolação e filtros de predição. Tais filtros podem levar em conta até mesmo características biomecânicas e anatômicas do ator e do conjunto de marcadores que estão sendo capturados para auxiliar no processo de preenchimento. Esse tipo de falha pode ocorrer, por exemplo, quando as câmeras de um sistema ótico não conseguem fazer o tracking de um marcador, devido à sua oclusão por um outro objeto da cena ou mesmo por uma parte do corpo do ator.

Capítulo 8

Processamento de Movimentos

Uma dificuldade existente na técnica de Captura de Movimento é a grande complexidade dos dados capturados, o que torna difícil a sua manipulação no computador. Utilizando ferramentas adequadas para processamento, podemos criar novos movimentos sem a necessidade de uma recaptura dos dados.

Através da utilização de estruturas articuladas hierárquicas, comandadas por ângulos relativos, é possível desenvolver uma série de ferramentas (ou operações) para o processamento de dados de movimento. Neste capítulo, analisaremos algumas dessas ferramentas. Para um estudo mais aprofundado sobre o assunto, sugerimos a leitura de (Bruderlin & Williams, 1995), (Rose *et al.*, 1996), (Witkin & Popović, 1995) e (Unuma *et al.*, 1996).

8.1 Classificação

O processamento de dados capturados consiste na utilização de primitivas básicas - os movimentos - que são combinadas ou modificadas por algum tipo de operação. Este mecanismo é semelhante ao utilizado em sistemas de modelagem por CSG, onde primitivas são combinadas através de operações booleanas.

As operações de processamento de movimentos podem ser classificadas com relação ao número de operandos a que estão relacionadas:

- **Operações Unárias.** Este tipo de operação tem um único movimento como operando, e é útil para modificar aspectos específicos de um movimento. Como exemplo, podemos citar a filtragem das curvas de articulações e o warping de parâmetros das curvas de movimentos.
- **Operações Binárias.** Neste tipo de operação, o operando é formado por dois movimentos. Seu principal objetivo é unir ou agrupar movimentos diferentes, criando um novo movimento mais longo e mais complexo. Exemplos de operações binárias são concatenação, ciclificação e transição. Note que, embora a concatenação possa ser aplicada a diversos movimentos em seqüência, ela pode ser encarada localmente como uma operação entre dois movimentos. As aplicações de operações binárias são vastas, e vão desde jogos de lutas virtuais (Page, 1995), até ambientes virtuais multi-usuários (Virtual, 1997).

- **Operações N-árias.** Operações n-árias trabalham com dois ou mais movimentos como operandos. Tais movimentos podem ser parcialmente ou totalmente combinados, de modo a misturar suas características básicas. Da mesma forma que em processamento de imagens, em geral é possível reduzir uma operação n-ária a um conjunto de operações binárias. A operação de *blending* é um exemplo de operação inserida nesta categoria.

8.2 Operações Básicas

Como vimos no capítulo anterior, os dados capturados dos marcadores são formados por sinais unidimensionais (1D) que representam curvas de movimento. No entanto, devido ao fato das articulações do ator estarem intrinsecamente relacionadas através de vínculos geométricos, ao pensarmos no movimento como um todo devemos encará-lo como sendo um “sinal” n -dimensional (nD), onde n representa o número de DOF’s das articulações utilizadas no processo. Tendo isso em mente, é natural pensar na utilização de técnicas tradicionais de processamento de sinais para manipular esses dados.

De fato, a grande maioria dessas técnicas podem ser aplicadas a dados de movimento. No entanto, nem todas produzem como resultado um movimento coerente e válido. Nas próximas seções, apresentaremos algumas técnicas de processamento de dados capturados existentes na literatura, estabelecendo sempre que possível um paralelo com aplicações de processamento de imagens e sinais. Para isso, assumiremos que o leitor possui alguma familiaridade com os conceitos básicos da área.

8.2.1 Filtragem

No campo de processamento de sinais, as aplicações de filtros são vastas. Filtros podem ser utilizados para eliminar ou atenuar frequências específicas de um sinal. Como exemplo, podemos citar os filtros do tipo passa-altas e passa-baixas (Figura 8.1), que têm como objetivo eliminar, respectivamente, baixas e altas frequências de um sinal.

No caso de imagens (sinais bidimensionais, 2D), filtros podem ser utilizados em processos de suavização, detecção de bordas ou mesmo para a geração de efeitos artísticos (Figura 8.2).

A utilização de filtros em movimentos nem sempre produz resultados satisfatórios, sob o ponto de vista visual. Isto ocorre pois a complexidade do movimento que está sendo representado é muito grande, e envolve inúmeros aspectos visuais, geométricos e temporais, que são profundamente afetados por pequenas modificações impostas ao conjunto de dados de uma articulação do ator virtual.

Podemos aplicar um filtro de suavização em uma curva (ou em um conjunto de curvas) de movimento, eliminando assim as altas frequências e reduzindo a variação dos dados do sinal. O resultado da aplicação de um filtro como esse no movimento de articulações é uma redução no número de oscilações de seus parâmetros angulares ao longo do tempo. Na Figura 8.3, foi aplicado um filtro de suavização no movimento da articulação do cotovelo do ator virtual. As imagens representam a configuração angular do movimento da articulação em um mesmo quadro da animação, no movimento original e no movimento filtrado.

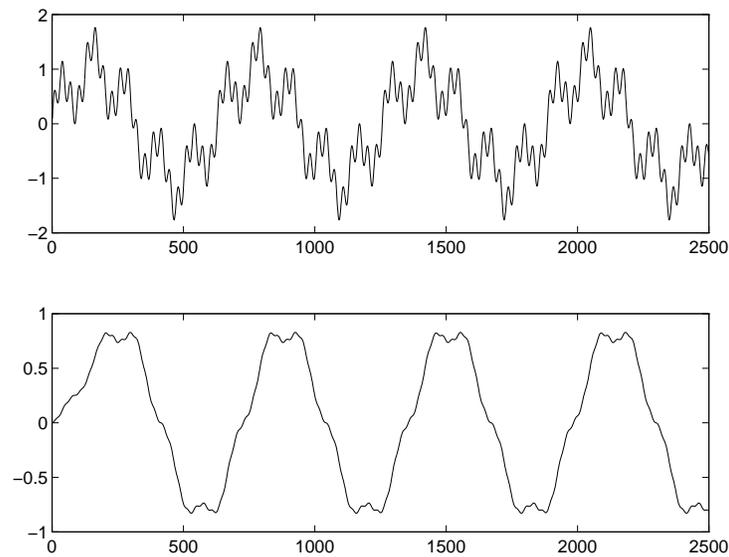
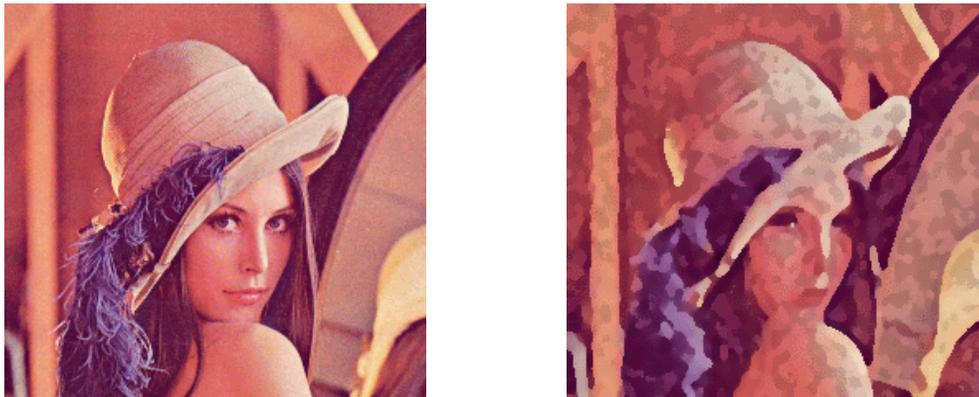


Figura 8.1: Eliminação de altas frequências de um sinal utilizando um filtro passa-baixa.



(a)

(b)

Figura 8.2: Filtragem: (a) imagem original e (b) modificada com filtro artístico.

8.2.2 Concatenação

No sentido estrito da palavra, concatenação significa união, junção. Podemos concatenar dois sinais quaisquer simplesmente posicionando o início do segundo sinal logo após o final do primeiro. Uma aplicação prática de concatenação de sinais 1D é a junção seqüencial de diversas palavras para formar uma frase em sistemas automáticos de consulta por telefone.

Em imagens, o processo de concatenação de sinais é utilizado na operação de *tiling*, onde uma imagem é replicada de modo a ocupar uma determinada região no espaço (Figura 8.4).

Podemos concatenar vários movimentos em seqüência, através da junção dos sinais que representam o movimento de cada articulação. Esse processo é utilizado em jogos de lutas

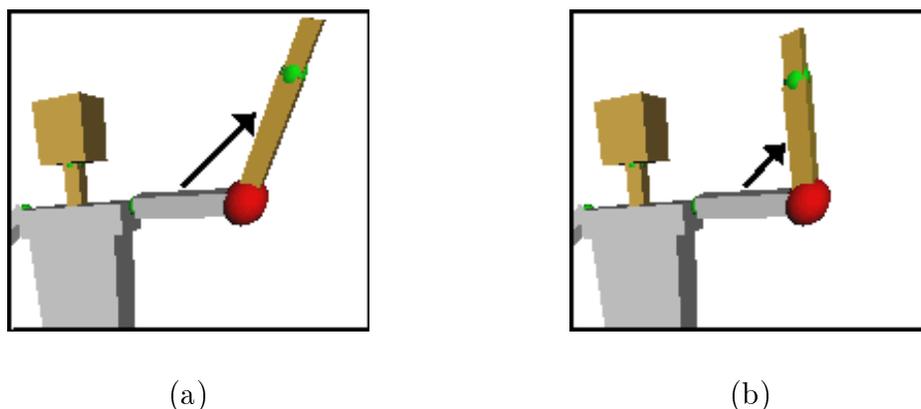


Figura 8.3: Articulação do cotovelo: (a) movimento original e (b) movimento filtrado.



Figura 8.4: Replicação de imagens: (a) imagem original e (b) imagem replicada.

virtuais, como o *Virtua Fighter* (SEGA, 1995). Neste tipo de jogo, os movimentos dos lutadores virtuais são gerados através da concatenação de diversos movimentos básicos de luta armazenados na memória, de acordo com a ação estabelecida pelo usuário.

Apesar de sua utilidade e facilidade de implementação para sinais de qualquer tipo, o processo de concatenação sofre do problema de descontinuidade na região da “fronteira” dos sinais combinados. Em sinais 2D, isto pode ser observado facilmente na Figura 8.4, onde existe uma descontinuidade visível na transição entre as imagens.

Em movimentos, a descontinuidade é melhor observada no decorrer da animação, e acontece sob a forma de um “salto” abrupto entre dois quadros. Em outras palavras, no momento da descontinuidade existe uma variação brusca na configuração angular das articulações do ator virtual, causada pela junção de seus sinais.

A Figura 8.5 mostra a concatenação de dois sinais idênticos de movimento¹. A junção dos dois sinais causa uma região de grande variação do parâmetro angular (apontada pela

¹Este processo também é conhecido como ciclificação (Rose *et al.*, 1996).

seta). A régua de medição existente sobre cada sinal marca o número de quadros da animação.

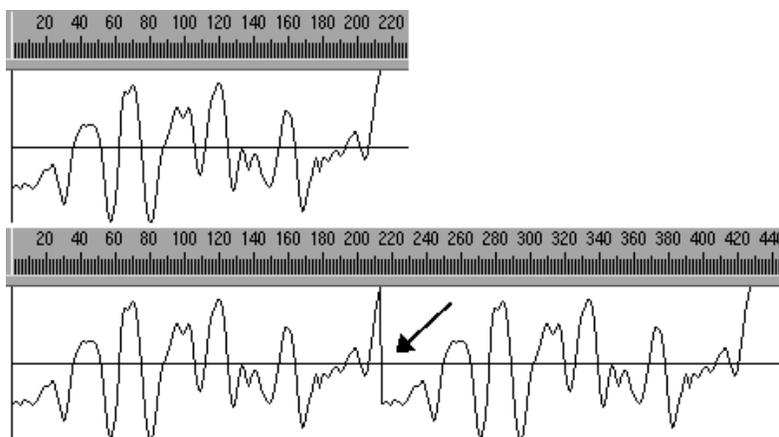


Figura 8.5: Concatenação: sinal original (em cima) e sinal concatenado (em baixo).

Para reduzir o efeito de descontinuidade na concatenação, podemos estabelecer uma pequena região de sobreposição entre os sinais, onde será feita uma transição suave entre os seus valores. Tal transição pode ser feita, por exemplo, através de interpolação linear ou média aritmética dos valores dos sinais.

Na Figura 8.6, foi feita a concatenação de dois sinais idênticos, estabelecendo-se uma região de transição entre os sinais (indicada na figura). Nesta região, os valores são formados pela média dos valores dos dois sinais, processo que reduz a descontinuidade do sinal resultante.

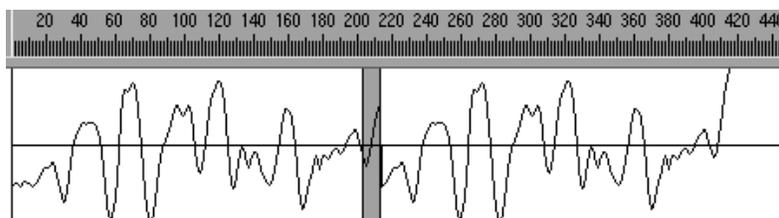


Figura 8.6: Concatenação com região de transição entre os dois sinais.

8.2.3 Blending

Este tipo de operação é útil para combinar (ou misturar) características de dois sinais distintos, gerando assim um terceiro sinal diferente. Isto é feito utilizando-se algum tipo de parametrização entre os valores dos sinais originais, de modo a criar os valores do novo sinal.

Um exemplo prático de blending entre dois sinais é utilizado durante a mixagem de dois sons. Este processo é também conhecido como *cross-dissolve*, e consiste na transformação

dos atributos de dois sinais através de algum tipo de parametrização. Na Figura 8.7², é apresentado o *cross-dissolve* entre dois sinais f e g , gerando um terceiro sinal h . A parametrização utilizada foi linear ($h = (1 - \lambda)f + (\lambda)g$), com $\lambda = 0.5$.

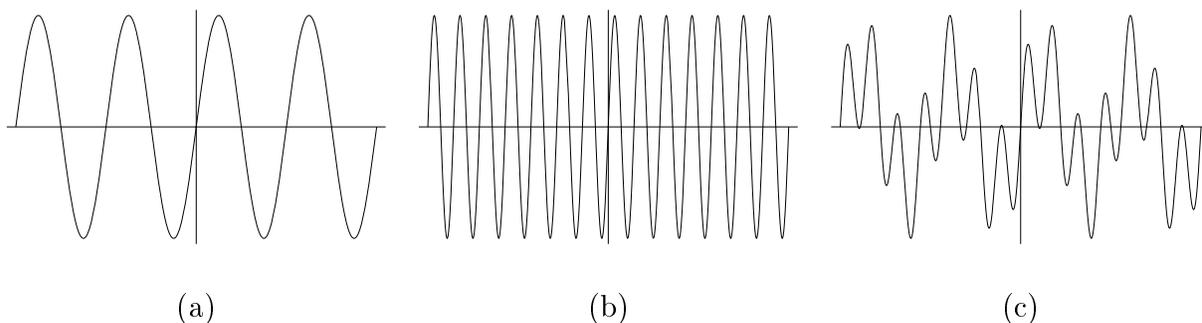


Figura 8.7: “Cross-Dissolve” de dois sinais: (a) sinal f , (b) sinal g e (c) sinal resultante h .

Em imagens, o processo de blending dos parâmetros da imagem é semelhante ao utilizado para sinais 1D, e é popularmente conhecido como fusão. A Figura 8.8 exemplifica o processo de fusão de duas imagens.

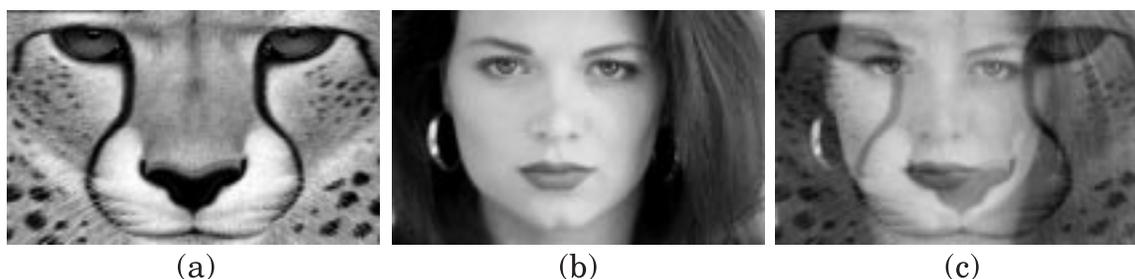


Figura 8.8: Fusão de duas imagens.

Existe ainda um outro tipo de fusão entre imagens, conhecido como *morphing*. Neste caso, é feita uma fusão adaptativa baseada em características geométricas das duas imagens, de modo a relacionar certos aspectos visuais. Este processo pode ser melhor estudado em (Gomes & Velho, 1997).

O processo de blending de movimentos segue a mesma linha de raciocínio utilizada em sinais 1D e 2D. As curvas de movimento das articulações envolvidas no processo são combinadas através de algum tipo de parametrização. Entretanto, o processo pode ser feito de duas formas distintas: a primeira consiste em combinar as curvas de articulações coincidentes nos dois movimentos, gerando um novo movimento, intermediário em relação aos originais (Figura 8.9). A outra forma consiste em combinar as curvas de articulações diferentes nos dois movimentos. Assim, o primeiro conjunto de articulações do ator virtual executará um determinado tipo de movimento, enquanto que o outro conjunto de articulações executará outro movimento (Figura 8.10).

²As imagens das Figuras 8.1, 8.7, 8.8 e 8.11 foram gentilmente cedidas pelo autor de (Goldenstein, 1997)

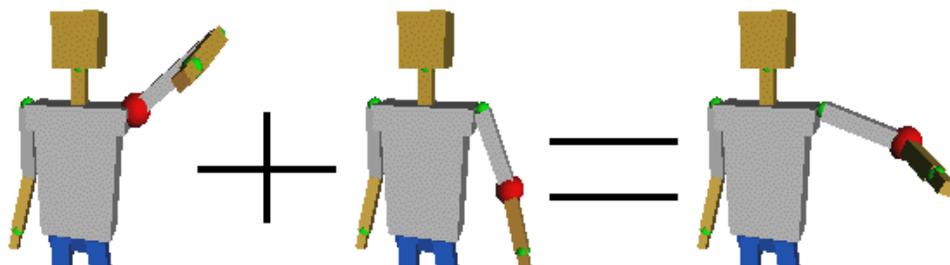


Figura 8.9: Combinação de diferentes curvas de movimento em um mesmo conjunto de articulações.

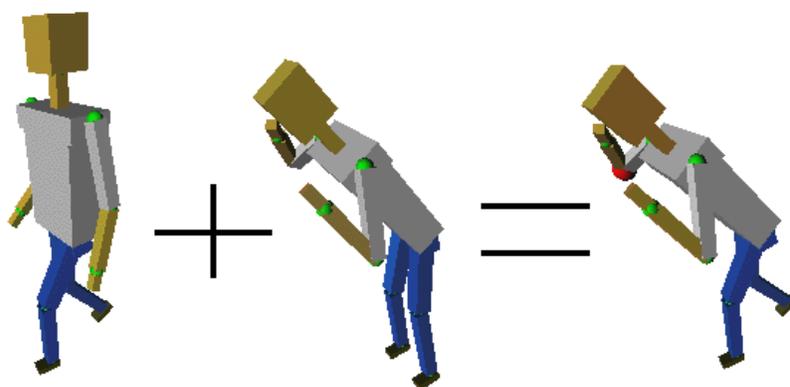


Figura 8.10: Combinação de diferentes curvas de movimento em conjuntos diferentes de articulações: conjunto 1 - articulações da parte superior do corpo; conjunto 2 - articulações da parte inferior do corpo.

8.3 Outras Operações

8.3.1 Warping

O processo de Warping consiste na geração de uma seqüência contínua de transformações de um objeto gráfico qualquer. Esse objeto gráfico pode ser um som (1D), imagem (2D), sólido geométrico (3D) ou mesmo um movimento (nD).

Em sinais 1D, o processo de warping pode ser empregado para expandir ou comprimir temporalmente um som, de modo a aumentar ou reduzir a sua duração.

No caso de imagens, uma transformação projetiva (Figura 8.11) é um exemplo de warping, pois consiste na transformação do suporte geométrico da imagem original.

Em movimentos, warping pode ser utilizado para “deformar” as curvas dos parâmetros de articulações do ator virtual. Tal deformação causa uma mudança na orientação dos segmentos afetados pela transformação. O processo consiste na alteração do valor da curva de movimento em um quadro específico da animação (*warp frame*). Então, é estabelecido algum tipo de deformação para transformar a curva original, incluindo o novo quadro modificado (Figura 8.12).

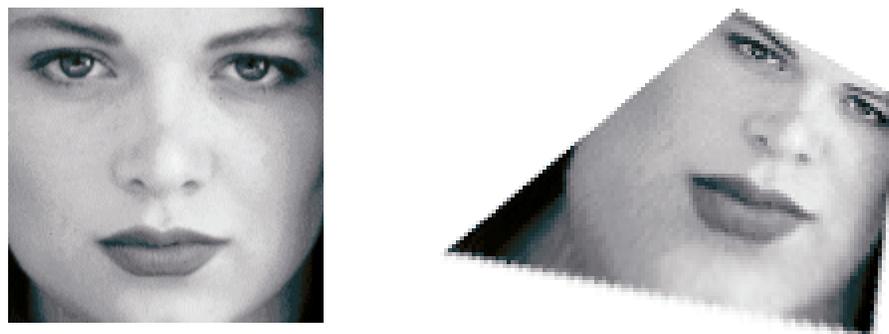


Figura 8.11: Transformação projetiva de uma imagem: exemplo de warping.

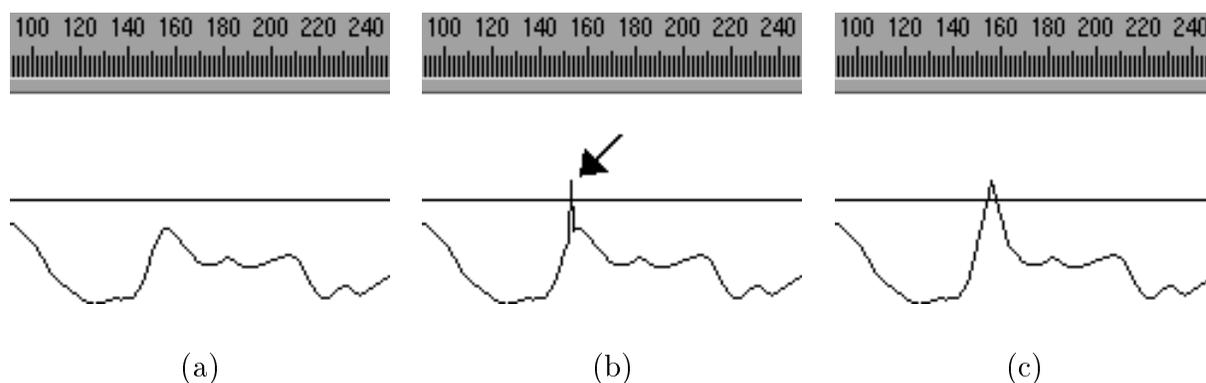


Figura 8.12: Warping de uma curva de movimento: (a) curva original, (b) pico causado pela modificação do valor do parâmetro em um quadro específico (apontado pela seta) e (c) warping da curva englobando o novo valor.

8.3.2 Filtragem em Multiresolução

As aplicações do princípio de multiresolução são vastas, e incluem a simplificação de objetos geométricos, visualização em diversos níveis de detalhe e compressão de imagens.

Em imagens, a filtragem em multiresolução é realizada através da divisão recursiva da imagem original em duas pirâmides, *bandpass pyramid* e *lowpass pyramid*, contendo respectivamente a decomposição da imagem original em bandas de frequência, cada uma delas representando um conjunto diferente de frequências espaciais (Figura 8.13). Uma aplicação prática desse método é a combinação de mosaicos de imagens, apresentada em (Burt & Adelson, 1983).

Os princípios de filtragem em multiresolução de imagens também podem ser aplicados a movimentos. Em (Bruderlin & Williams, 1995), foi estabelecida uma relação entre as altas e baixas frequências das curvas de articulações com características qualitativas do movimento. Através da observação e análise dos dados capturados, as altas frequências foram relacionadas com os detalhes e sutilezas existentes em um movimento, enquanto que as baixas frequências foram ligadas ao padrão básico contido no movimento.

Calculando as bandas de frequência de cada sinal, e separando-as em diferentes níveis na pirâmide, é possível modificar características específicas de um movimento (por exem-

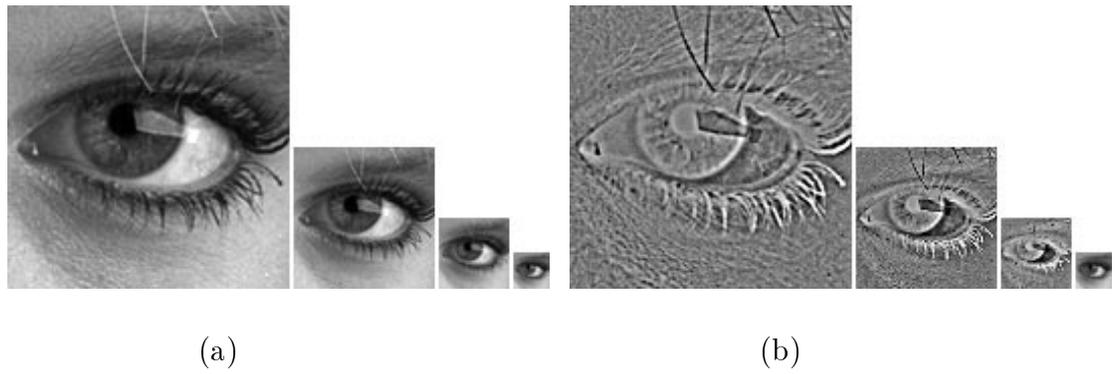


Figura 8.13: Decomposição de uma imagem em bandas de frequência: (a) *lowpass pyramid* e (b) *bandpass pyramid*.

plo, exagerar um movimento de andar) simplesmente alterando as curvas em cada banda separadamente.

Capítulo 9

Um Sistema de Animação Baseado em Movimento Capturado

Neste capítulo, apresentaremos o protótipo do sistema de animação desenvolvido no presente trabalho. A principal característica desse sistema é ter a técnica de Captura de Movimento como o núcleo de todo o processo de animação, diferentemente dos sistemas atuais, analisados no Capítulo 5. Desta forma, foi necessário desenvolver um novo conceito de sistema de animação, com estruturas de dados, arquitetura interna e interface gráfica adequadas para tratar a técnica de Captura de Movimento.

9.1 Digressão Histórica: Sistemas de Animação Tradicionais

Os sistemas de animação possuem uma história relativamente recente. O primeiro sistema de animação por computador, chamado EXPLOR, foi desenvolvido no laboratório de pesquisa da *Bell Telephone*, por Ken Knowlton em 1970. Esse sistema foi utilizado para auxiliar artistas na criação de simples imagens, formadas por padrões que eram repetidos com diferentes níveis de randomização.

O passo seguinte foi dado através da utilização de linguagens simples, onde o animador tinha à sua disposição uma série de comandos (ROTATE, TRANSLATE, DRAW etc.), que eram empregados e combinados para animar objetos geométricos no computador. Esse tipo de sistema também era conhecido como *scripting system*, e seu conceito básico ainda é utilizado em muitos sistemas de animação disponíveis atualmente.

Desde então, os sistemas de animação foram ficando cada vez mais sofisticados, acompanhando o desenvolvimento das técnicas de animação e dos equipamentos de Computação Gráfica. Hoje em dia, sistemas de animação como o Alias/Wavefront, SoftImage, 3D Studio MAX e LightWave possuem recursos de última geração para animação e rendering, incluindo módulos para animação procedural, simulação dinâmica e movimento capturado (Capítulos 5 e 6).

9.2 Um Novo Conceito de Sistema de Animação

Como vimos nos capítulos anteriores, a técnica de Captura de Movimento possui algumas características peculiares, fato que a torna essencialmente diferente das técnicas de animação existentes atualmente. Tal diferenciação existe tanto no que diz respeito à composição dos dados, quanto à forma de manipulá-los e visualizá-los.

A maioria dos sistemas profissionais de animação possui ferramentas para manipular dados capturados. No entanto, tais sistemas tratam a técnica de Captura de Movimento como uma ferramenta secundária, um *plug-in*. Conseqüentemente, os dados capturados são manipulados utilizando-se os mesmos paradigmas de interface e arquitetura interna empregados nas técnicas de animação já existentes, o que torna o processo pouco intuitivo e produtivo.

Tomando esses fatos como motivação, desenvolvemos o protótipo de um sistema de animação baseado em Movimento Capturado, chamado *MC Animator* (Figura 9.1). Este sistema foi implementado de modo a tratar de maneira adequada os principais aspectos deste tipo de tecnologia de animação.

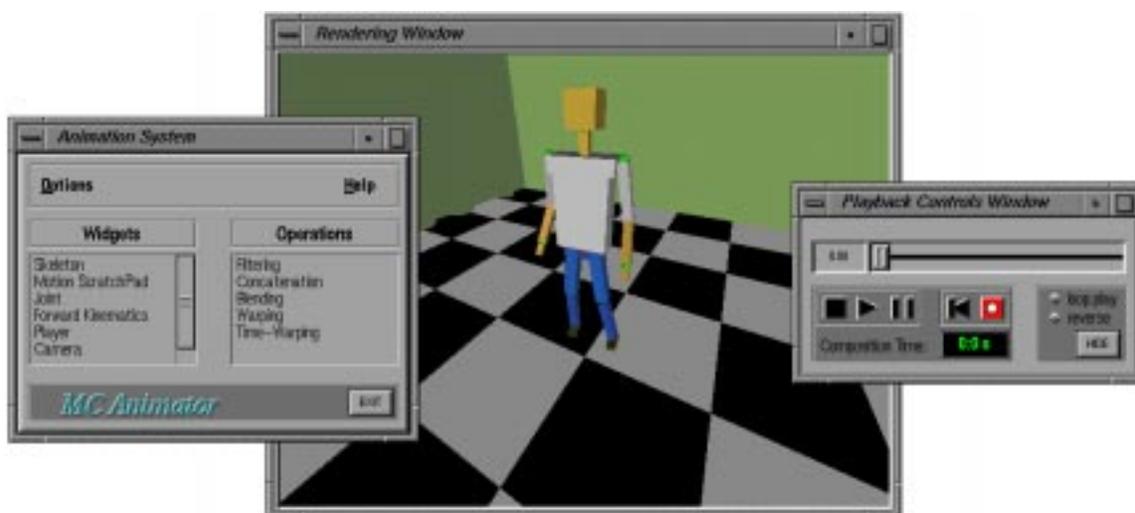


Figura 9.1: *Screenshot* do sistema *MC Animator*.

Como veremos nas próximas seções, as principais entidades básicas, estruturas de dados, arquitetura interna e interface com o usuário do sistema desenvolvidas de modo a oferecer uma forma conceitualmente correta de lidar com os dados capturados.

As aplicações da utilização de um sistema com características deste tipo são importantes. Além de oferecer ferramentas avançadas para a manipulação e análise de movimentos capturados, tal sistema pode ser utilizado junto a um equipamento de captura, formando um ambiente poderoso onde os dados são analisados e, se necessário, modificados logo após a sua aquisição, contribuindo assim para a redução do tempo e, conseqüentemente, dos custos de produção.

9.3 Arquitetura do Sistema

O sistema *MC Animator* possui uma arquitetura interna (da Silva *et al.*, 1997b) desenvolvida tendo em vista os seguintes objetivos principais:

- Oferecer um conjunto básico de ferramentas para a manipulação, análise e reutilização de movimentos capturados.
- Permitir a criação de animações complexas utilizando bibliotecas de movimentos.
- Prover alternativas para algumas limitações técnicas existentes no processo.
- Expansibilidade: permitir a inclusão de novas técnicas para a manipulação de movimentos.

Pensando nisso, dividimos a arquitetura em três módulos básicos: entrada, processamento e saída (Figura 9.2). Tais módulos são amparados pelo módulo especial de interface gráfica, que contém a representação visual das principais entidades e operações existentes no sistema. Além disso, todos os módulos compartilham de uma estrutura de dados que encapsula, entre outras, as entidades Ator Virtual e Movimento, descritas nos Capítulos 3 e 7, respectivamente.

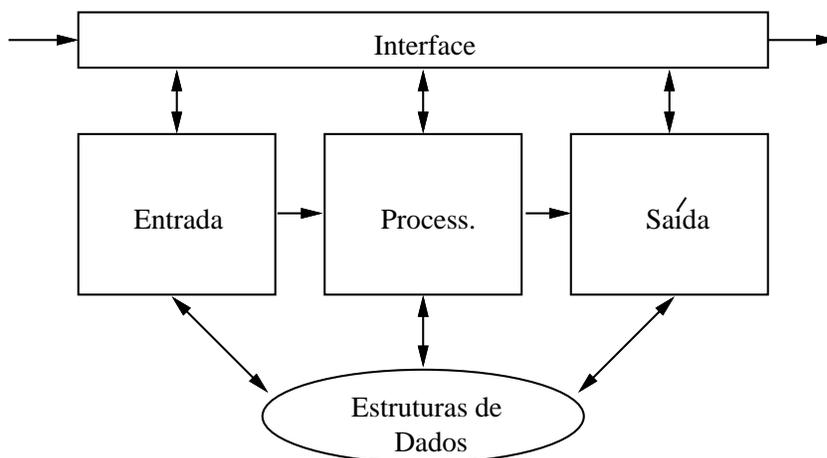


Figura 9.2: Fluxograma da arquitetura interna do sistema.

9.3.1 Módulo de Entrada

Este módulo é responsável pela interpretação e pré-processamento dos dados de movimento. Seu fluxograma interno é apresentado na Figura 9.3. A etapa de interpretação consiste na escolha pelo usuário do esqueleto que vai ser utilizado como alvo no mapeamento dos movimentos. Na versão atual do sistema, apenas um tipo de esqueleto (descrito no Capítulo 3) é oferecido ao usuário. No futuro, pretendemos adicionar outros tipos de esqueletos, o que aumentará o número de formatos de dados capturados

que poderão ser utilizados no sistema, oferecendo uma maior flexibilidade na criação de animações de atores sintéticos.

Após a definição do esqueleto do ator virtual, a etapa seguinte consiste na escolha do(s) movimento(s) que serão analisados ou processados. Após a escolha, os movimentos serão então processados internamente pelo sistema, que fará automaticamente a conversão dos dados capturados e da definição do esqueleto para as estruturas de dados do sistema, adequando-os para o processamento. Neste processo está incluído o método de extração de ângulos relativos, descrito no Capítulo 7.

Após essas etapas, os dados são finalmente mapeados no ator virtual e, estando prontos para o processamento, são então armazenados no objeto gráfico de interface chamado *Motion ScratchPad*. Este objeto gráfico funciona como uma espécie de container onde os movimentos podem ser visualizados e escolhidos pelo usuário para posterior processamento. O *Motion ScratchPad* será analisado em mais detalhes no próximo capítulo.

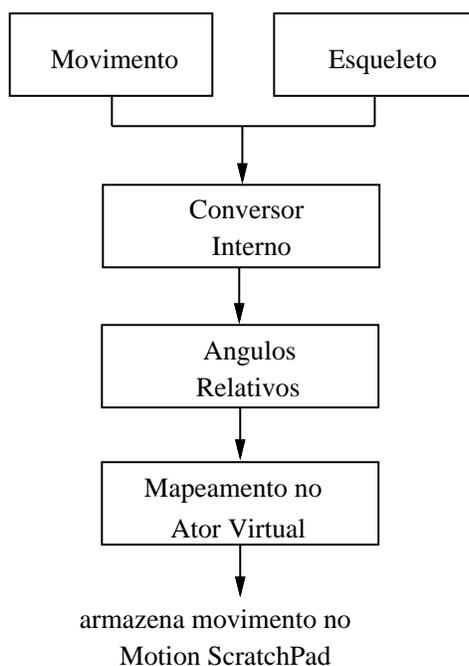


Figura 9.3: Fluxograma do módulo de entrada.

9.3.2 Módulo de Processamento

Neste módulo estão presentes as ferramentas para análise, processamento e reutilização de movimentos. O sistema foi desenvolvido de modo a permitir a inclusão de novas técnicas para o processamento de movimentos, sob a forma de *plug-ins*, que podem ser criados utilizando o ferramental de interface e estruturas de dados presentes no sistema. Desta forma, é possível utilizar o sistema como uma plataforma para o desenvolvimento de novas técnicas e algoritmos para o processamento de dados capturados.

O fluxograma da Figura 9.3 exemplifica o funcionamento deste módulo. Após passarem pelo módulo de entrada, os movimentos armazenados no *ScratchPad* são selecionados pelo usuário, que direciona-os para as operações disponíveis no sistema. Os movimentos

processados podem ainda voltar ao *ScratchPad*, para serem novamente processados no futuro, ou então podem ser armazenados na biblioteca de movimentos do sistema, utilizando os recursos do módulo de saída.

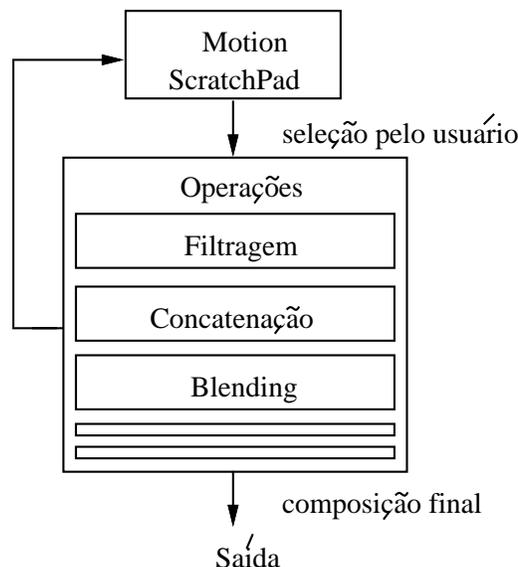


Figura 9.4: Fluxograma do módulo de processamento.

9.3.3 Módulo de Saída

Após o processamento dos movimentos, é fundamental oferecer ao usuário uma forma de registrar a composição final, permitindo assim uma posterior visualização ou reprocessamento. No módulo de saída, são oferecidas ferramentas para o armazenamento dos dados de movimento processados pelo sistema. Tais ferramentas possibilitam ao usuário escolher entre armazenar os dados sob a forma de curvas de movimento, podendo até mesmo salvar a configuração de ângulos relativos das articulações; ou então gerar uma seqüência de imagens renderizadas, que poderão ser utilizadas para posterior visualização em equipamentos profissionais de vídeo.

Em versões posteriores do sistema, será possível ler arquivos de movimentos provenientes de uma variedade maior de equipamentos de captura, processá-los e então fazer a conversão para outros tipos de arquivos de movimento ou mesmo direcioná-los para sistemas profissionais de rendering, como RenderMan (RenderMan, 1997) e PovRay (of Vision Ray-Tracer, 1997). Será possível ainda processar os dados capturados de modo a convertê-los para um esquema de representação compatível com keyframing. Assim, o sistema de animação também servirá como um poderoso conversor entre arquivos de movimento.

9.3.4 Módulo de Interface

Uma grande desvantagem dos sistemas de animação tradicionais é a falta de uma descrição adequada, no nível de interface com o usuário, das principais abstrações da

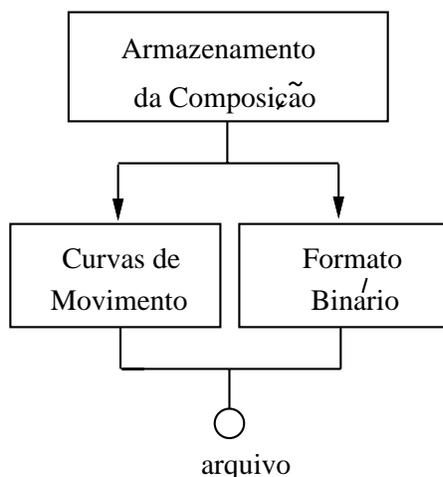


Figura 9.5: Fluxograma do módulo de saída.

técnica de Captura de Movimento. Isto contribui para tornar o processo de manipulação dos dados mais difícil.

Em nosso sistema, decidimos adotar um novo paradigma de interface gráfica para tratar movimentos capturados, fornecendo uma representação visual de entidades como o Ator Virtual, os Movimentos e as Operações de Movimentos. A descrição da interface gráfica do sistema será abordada separadamente no Capítulo 10.

9.3.5 Estruturas de Dados

As estruturas de dados utilizadas neste trabalho têm como objetivo fornecer uma representação no computador das principais entidades manipuladas pelo sistema, organizando-as logicamente e encapsulando-as através de procedimentos algorítmicos. Além disso, tais estruturas fornecem um controle para determinados aspectos do sistema como, por exemplo, o controle dinâmico dos objetos da interface gráfica e o gerenciamento das funções de rendering.

Como vimos no Capítulo 3, o sistema *MC Animator* utiliza uma variante da notação APJ, adaptada para o uso em estruturas articuladas hierárquicas comandadas por ângulos relativos. Também estudamos, no Capítulo 7, a composição dos dados capturados.

As estruturas de dados do sistema *MC Animator* organizam os dados capturados de acordo com as articulações do ator virtual e com os quadros de animação presentes no arquivo de movimento, facilitando assim o processo de mapeamento e visualização.

9.4 Implementação

Nesta seção, descreveremos alguns aspectos relacionados com a implementação do sistema desenvolvido neste trabalho.

9.4.1 Operações de Movimento Disponíveis

O protótipo do sistema de animação conta com quatro operações para o processamento de movimentos: filtragem, concatenação, blending e warping. Os fundamentos dessas operações foram apresentados no Capítulo 8.

No Capítulo 11 apresentaremos exemplos de utilização do sistema, juntamente com os resultados obtidos.

9.4.2 Outras Ferramentas de Animação

As técnicas de animação tradicionais têm papel importante no sistema de animação desenvolvido neste trabalho. Ferramentas como forward kinematics e keyframing são amplamente utilizadas em conjunto com as técnicas de processamento de movimentos, auxiliando o processo de manipulação das estruturas articuladas. Essa integração poderá ser melhor observada no Capítulo 11.

9.4.3 Detalhes de Implementação

O sistema protótipo, desenvolvido neste trabalho, foi implementado utilizando a linguagem C, tendo como plataforma base uma estação de trabalho Silicon Graphics Indigo 2 rodando um sistema operacional baseado em UNIX.

Foi empregado OpenGL (Neider *et al.*, 1993) como biblioteca básica de rendering. Fizemos essa escolha pois o OpenGL vem se tornando um padrão em termos de biblioteca gráfica, sendo suportado pelos principais sistemas operacionais e placas gráficas existentes no mercado. Além disso, o OpenGL oferece poderosos recursos de rendering, com ferramentas avançadas e otimizadas para a visualização de objetos geométricos no computador, sendo, portanto, particularmente interessante para aplicações que necessitam de interatividade, situação que se aplica ao nosso caso.

Para a geração dos objetos básicos de interface gráfica adotamos a biblioteca XForms (Zhao & Overmars, 1996), muito utilizada no meio acadêmico por fornecer rotinas de alto-nível para o controle dos eventos e funções de interface do X. No entanto, para atender às necessidades de projeto do sistema protótipo, foi necessário desenvolver objetos gráficos de interface avançados, que logicamente não eram fornecidos pela biblioteca original. Tais objetos, descritos no capítulo a seguir, foram modelados e implementados separadamente, e então integrados à biblioteca do XForms.

Capítulo 10

Interface com o Usuário

A comunicação entre o usuário e o software é uma etapa fundamental em qualquer projeto de sistema de animação. Nos sistemas interativos, essa comunicação se faz por meio de uma interface gráfica, que tem como objetivo fornecer uma representação visual das principais entidades e funções presentes no sistema.

Os sistemas de animação tradicionais foram desenvolvidos tendo como base as entidades e conceitos existentes em técnicas conhecidas como, por exemplo, keyframing. Por isso, seus modelos de interface gráfica foram desenvolvidos de modo a tratar adequadamente o tipo de dados existente nessas técnicas. No entanto tal modelo de interface não é eficiente para lidar com os dados capturados, seja devido à sua complexidade ou por características específicas desse tipo de dado.

Em nosso sistema, decidimos adotar um modelo de interface que trate de forma coerente e eficiente as principais abstrações da técnica de Captura de Movimento. Assim, escolhemos como ponto de partida no desenvolvimento da interface o modelo empregado em alguns sistemas digitais de edição de vídeo, como por exemplo o Video Cube (IMIX, 1996).

10.1 O Paradigma de Edição de Vídeo

Atualmente, sistemas digitais de edição de vídeo são utilizados pelas principais produtoras comerciais pois oferecem uma maneira rápida e eficiente de manipular segmentos de vídeo.

Muitos desses sistemas empregam paradigmas semelhantes em termos de interface gráfica com o usuário. Nestes modelos, os segmentos de vídeo são tratados como barras horizontais, alinhadas em um objeto contendo diversos “trilhos” que permitem uma movimentação relativa entre elas. Desta forma, é possível combinar facilmente os segmentos de vídeo para formar a composição final simplesmente movimentando as barras que os representam, alinhando-as temporalmente. Além disso, este tipo de representação permite o acesso individual a cada quadro de um segmento de vídeo qualquer, possibilitando a sua modificação.

10.2 Representação das Entidades na Interface

O sistema *MC Animator* conta com uma interface gráfica baseada no modelo existente em algumas ilhas digitais de edição de vídeo. Essa escolha foi feita baseada na observação de que um movimento capturado é uma animação potencialmente pronta e, portanto, deve ser tratada de forma macroscópica, diferentemente da maneira que são tratados os dados das técnicas tradicionais de animação.

O objetivo principal da interface utilizada neste trabalho é fornecer uma forma coerente de descrever graficamente as principais entidades e conceitos existentes na técnica de Movimento Capturado. Além disso, tal interface deve ser extensível, permitindo a inclusão de novos objetos e técnicas ao longo do tempo. Nas próximas seções, apresentaremos as principais entidades existentes na interface gráfica do sistema.

10.2.1 Ator Virtual

No paradigma de interface utilizado no sistema, o ator virtual é representado através de seus principais componentes: a topologia e geometria do esqueleto articulado.

A topologia do esqueleto é visualizada sob a forma de um grafo, em uma janela com informações sobre cada uma de suas articulações (Figura 10.1). A disposição das informações na janela sugere as relações hierárquicas entre os segmentos e articulações do ator virtual. Além disso, neste objeto gráfico são oferecidos recursos para a seleção ou agrupamento de articulações, ferramenta que será útil em determinadas situações.

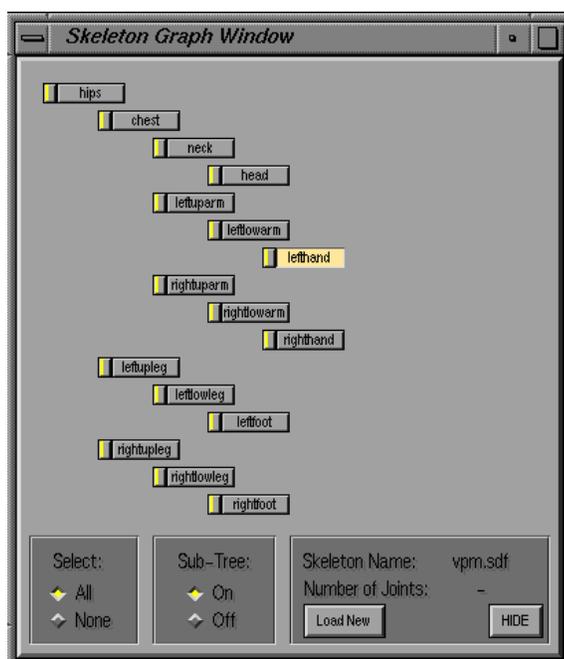


Figura 10.1: *Skeleton Graph Window*: informações topológicas sobre o esqueleto articulado.

A representação geométrica do ator virtual é visualizada na janela de rendering, como mostrado na Figura 10.2. Conceitualmente, existe uma importante relação entre o componente topológico e geométrico no ator virtual. Tal relação também é implementada no

sistema: ao selecionar uma articulação na janela *Skeleton Graph*, a mesma aparece em destaque com cor e tamanho diferentes na janela de rendering.

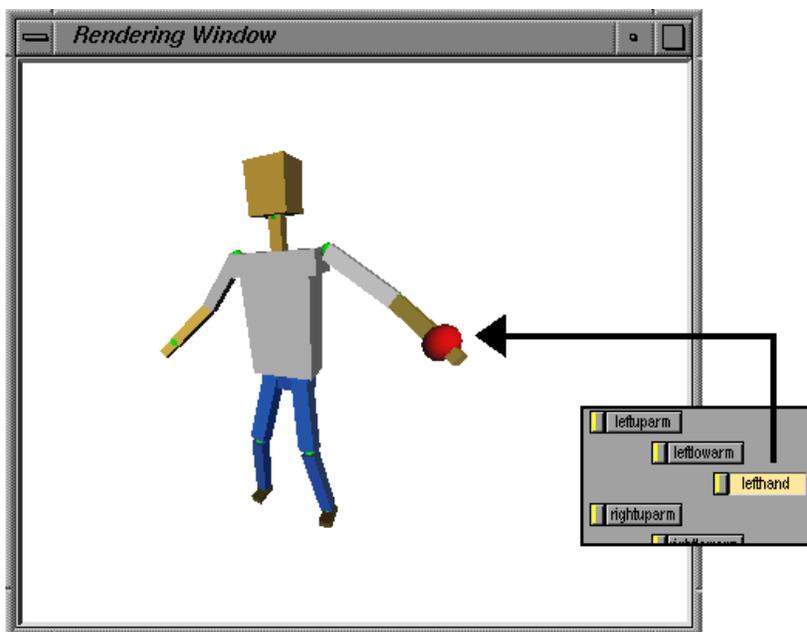


Figura 10.2: Representação geométrica do ator virtual. Seleção de articulação (detalhe da janela *Skeleton Graph*) e destaque na janela de rendering.

10.2.2 Movimentos

O modelo de interface utilizado no sistema trata cada movimento como uma barra horizontal (Figura 10.3). O comprimento da barra é determinado pela duração (em quadros de animação) dos dados que representam o movimento em questão. O design da barra foi desenvolvido de modo a oferecer informações adicionais sobre o movimento, como por exemplo o nome. Além disso, suas extremidades possuem marcas especiais que servem para posicionar e reparametrizar temporalmente o movimento, aspectos que serão analisados posteriormente.



Figura 10.3: Representação de um movimento na interface gráfica.

Usualmente, utilizaremos uma régua de medição de quadros de animação (*frame ruler*) associada a cada movimento. Esta associação tem como objetivo fornecer uma melhor percepção temporal e espacial dos movimentos.

Curvas de Movimento

As curvas das articulações são as componentes básicas de um movimento. Em nosso sistema, tais curvas são representadas através de um objeto gráfico que descreve o seu

comportamento ao longo dos quadros da animação (Figura 10.4). Este objeto gráfico também fornece informações numéricas sobre os canais de dados em cada quadro de animação (canais X, Y e Z de posição, orientação global e relativa).



Figura 10.4: Representação de uma curva de movimento no sistema.

Utilizando esse tipo de representação podemos pensar na implementação de diversas técnicas para a edição de curvas existentes na literatura, fornecendo um controle preciso e interativo da forma da curva.

Como vimos nos capítulos anteriores, cada articulação do ator virtual possui um conjunto de curvas (uma para cada DOF). No sistema *MC Animator* tais curvas são agrupadas em um objeto especial de interface chamado *Joint Curves Window*, que oferece uma visão global das curvas e possui relações diretas com outras janela do sistema. Quando uma articulação é selecionada na janela *Skeleton Graph* as suas curvas são visualizadas na janela *Joint Curves*, como mostrado na Figura 10.5.

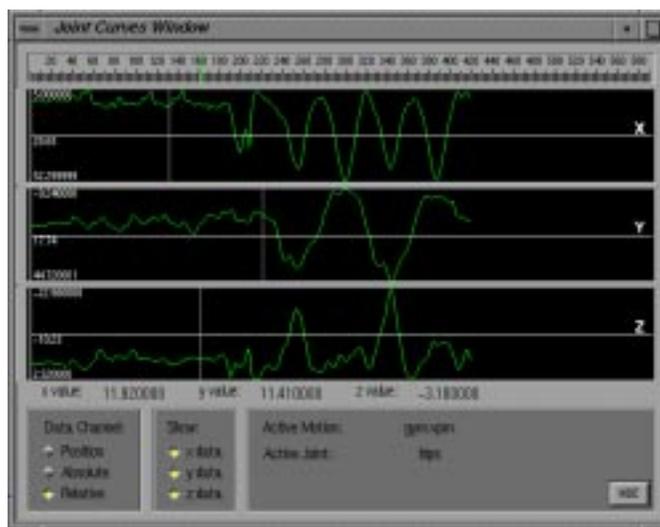


Figura 10.5: A janela *Joint Curves*.

A Ligação Ator - Movimento

Para estabelecer uma ligação entre as descrições do ator virtual e dos movimentos, decidimos criar um objeto de interface chamado *Motion View Window*, acessível através de um clique duplo em qualquer barra de movimento. Neste objeto, pequenos botões horizontais representam as componentes do movimento em relação às articulações do ator

virtual, e podem ser selecionados ou agrupados. Visualmente, o objeto *Motion View* parece como um zoom em uma barra de movimento. Esta semelhança foi proposital, pois os movimentos são compostos por curvas associadas às articulações de um ator.

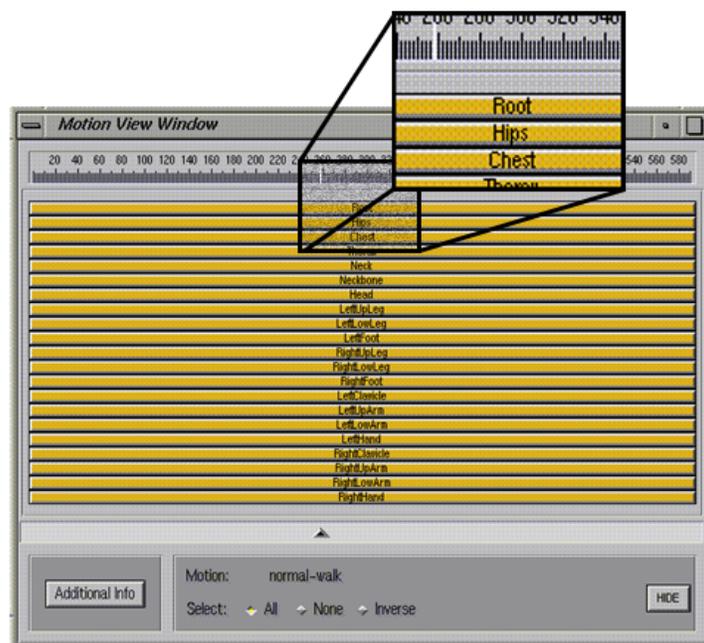


Figura 10.6: *Motion View Window*: ligação entre as representações do ator e do movimento.

O objeto gráfico *Motion View* possui importantes aplicações no sistema. Uma delas é permitir a aplicação de uma operação em um conjunto específico de curvas de movimento, como foi mostrado na Figura 8.10 do Capítulo 8.

Motion ScratchPad

Em nosso sistema, foi criado um objeto de interface que tem o objetivo de organizar os movimentos selecionados pelo usuário, fornecendo uma percepção global de todos os movimentos nele armazenados. No *Motion ScratchPad* (Figura 10.7), os movimentos selecionados pelo usuário são posicionados seqüencialmente em um canvas especial, contendo diversas outras informações úteis como, por exemplo, a duração em segundos de cada movimento.

O *Motion ScratchPad* permite a visualização individual de cada movimento. Além disso, é nesse objeto gráfico que o usuário direciona os movimentos para as janelas contendo as operações de processamento, através de um menu localizado no canto inferior direito da janela. Como podemos observar, o *Motion ScratchPad* estabelece uma conexão de alto-nível entre os módulos de entrada e processamento.

10.2.3 Operações

As operações de movimento são peças fundamentais em nosso sistema. Conseqüentemente, devem ser representadas graficamente de modo a tornar intuitivo o processamento

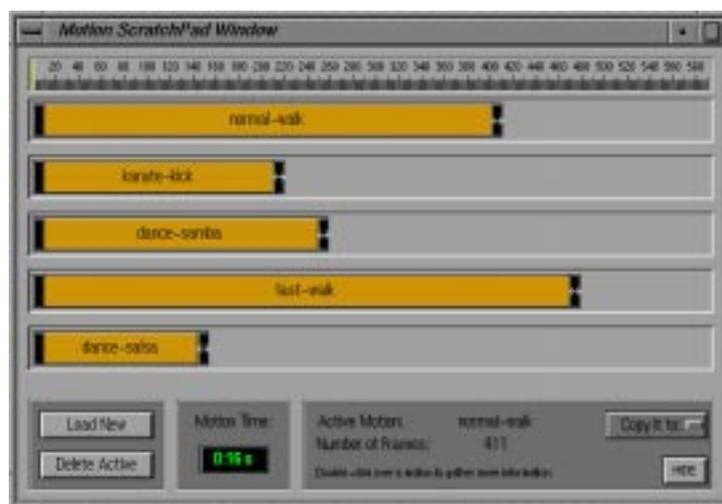


Figura 10.7: *Motion ScratchPad*: conexão entre os módulos de entrada e processamento.

dos dados capturados.

No sistema de animação desenvolvido neste trabalho, cada operação de movimento possui a sua própria janela. Quando necessário, objetos gráficos auxiliares podem ser utilizados para auxiliar o processo e controlar alguns parâmetros da operação. Para cada janela de operação, é fornecido um conjunto básico de objetos auxiliares - um visualizador interativo e uma régua de medição (*frame ruler*).

Filtragem

A operação de filtragem é representada em uma janela com ferramentas para permitir a seleção de uma região específica do movimento a ser filtrada, além da lista com os filtros disponíveis.



Figura 10.8: Janela da operação de filtragem.

Concatenação

O objeto de interface que representa a operação de concatenação foi desenvolvido de modo a oferecer uma boa percepção temporal da composição final.

Os movimentos selecionados pelo usuário são, inicialmente, posicionados na janela desta operação de modo a executar uma concatenação direta, isto é, sem intervalos de transição. Esta configuração é apresentada na Figura 10.9.

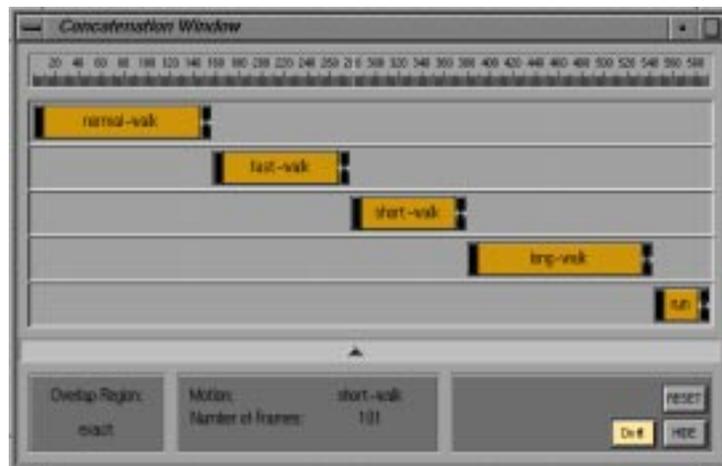


Figura 10.9: Concatenação de movimentos (configuração inicial).

Utilizando as ferramentas para o posicionamento relativo entre os movimentos é possível estabelecer regiões de sobreposição, que executarão uma transição suave entre os movimentos concatenados (na Figura 10.10, representadas pelas regiões escuras entre os movimentos). Os parâmetros de transição (curvas de *easy-in* / *easy-out*) entre os movimentos podem ser acessados / alterados através de um duplo-clique no intervalo de transição desejado.

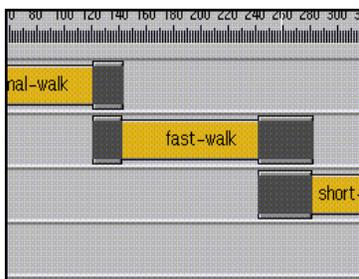


Figura 10.10: Intervalos de transição na concatenação de movimentos (detalhe).

Blending

Na janela de blending, assim como na operação de concatenação, os movimentos selecionados pelo usuário são agrupados em um canvas especial que oferece uma visão global da composição. Para a operação de blending, entretanto, novos objetos gráficos foram introduzidos para auxiliar o processo.

Tais objetos são os *Time-Markers*, cujo propósito é sincronizar acontecimentos importantes nos movimentos a serem combinados. A importância desse sincronismo é melhor

compreendida com o seguinte exemplo: ao combinarmos dois tipos diferentes de movimento de andar, é fundamental que os pés do ator toquem o solo no mesmo instante nos dois movimentos. Se isto não acontecer, não poderemos nem mesmo garantir que os pés tocarão o solo no movimento resultante.

A Figura 10.11 mostra um exemplo de combinação de três movimentos distintos. Os *Time-Markers* (detalhe da Figura 10.11) foram utilizados para estabelecer a correspondência entre os momentos-chave nos movimentos combinados.

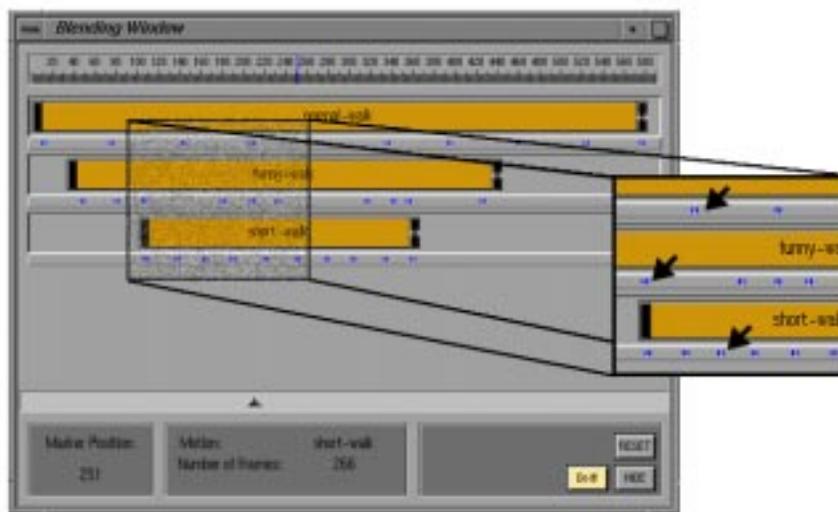


Figura 10.11: Janela da operação de blending e *Time-Markers*.

O processo de sincronismo consiste na reparametrização dos movimentos de acordo com a posição dos *Time-Markers*, relacionando os eventos no tempo.

Warping

Como vimos no Capítulo 8, a operação de warping consiste na deformação da curva de movimento de uma articulação, através da alteração do valor em um determinado quadro da animação e posterior modificação da curva original, ajustando-a para incluir o valor alterado.

A janela de warping oferece recursos para a seleção do quadro que terá o valor alterado, bem como para a definição da região de warp, onde será feito o ajuste da curva original (Figura 10.12).

10.2.4 Outras Ferramentas de Animação

Forward Kinematics

A técnica de Forward Kinematics é muito útil no contexto de estruturas articuladas hierárquicas, pois permite a manipulação dos ângulos relativos de suas articulações.

Em nosso sistema de animação a técnica de Forward Kinematics possui uma janela própria, com potenciômetros que permitem a manipulação interativa dos ângulos relativos de articulações escolhidas pelo usuário, como mostra a Figura 10.13.

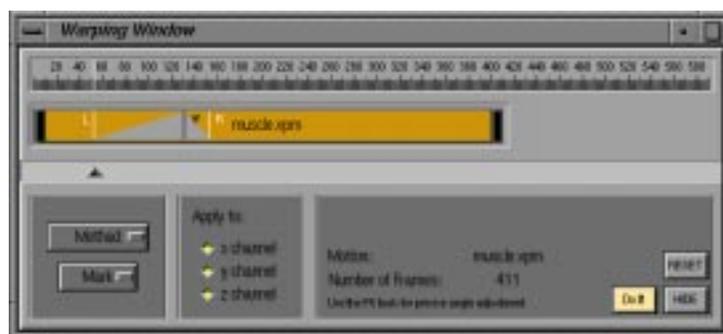


Figura 10.12: Janela da operação de warping.

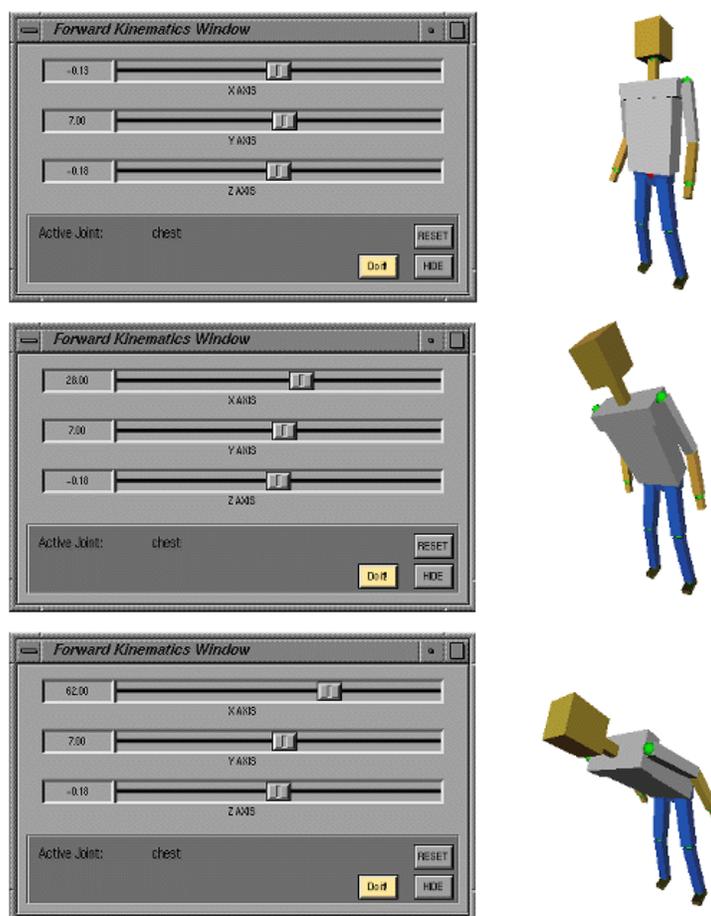


Figura 10.13: Manipulação da estrutura articulada hierárquica utilizando Forward Kinematics.

Câmera

A interface desenvolvida no sistema oferece recursos para o controle interativo de câmera durante o playback de animações. Tais controles são adaptados para o uso com dados capturados. As opções disponíveis permitem direcionar o alvo da câmera para articulações específicas do ator virtual, além de oferecer recursos de acompanhamento

automático do deslocamento do ator e opções de zoom.



Figura 10.14: Menu de controle interativo da câmera. Da esquerda para a direita: zoom in, zoom out, follow mode, lock joint mode, circle camera.

Controles de Playback

Seguindo o paradigma de interface utilizado nos sistemas de edição de vídeo, desenvolvemos um painel de controle para o playback de movimentos e composições criadas pelo usuário. Esse painel de controle contém um potenciômetro que permite um posicionamento interativo preciso da animação em um quadro especificado pelo usuário. Além disso, são oferecidos controles de avanço, pausa, retorno e gravação (em formato raster) de animações, semelhantes aos encontrados em vídeo-cassetes.

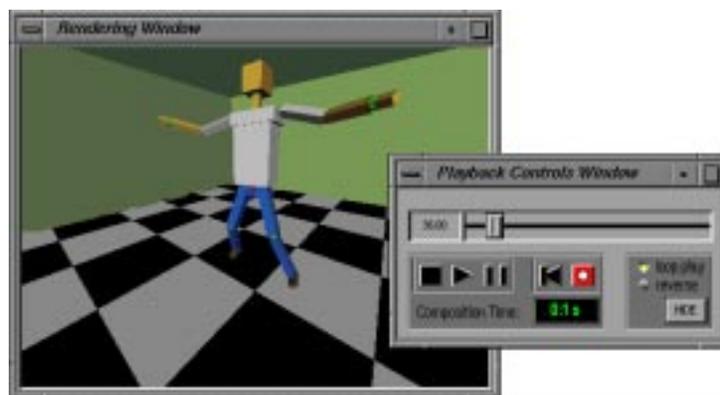


Figura 10.15: Janela de rendering e painel de controle interativo da animação.

10.3 Controle Dinâmico dos Objetos da Interface

Internamente, a interface gráfica é controlada através de um loop contínuo (Figura 10.16) que verifica o estado de cada objeto gráfico, reportando qualquer modificação feita pelo usuário para uma função especial que executa e gerencia as mudanças que precisam ser feitas nos objetos de interface. Isso se faz necessário pois todos os objetos de interface estão intimamente relacionados, compartilhando diversas informações ao mesmo tempo.

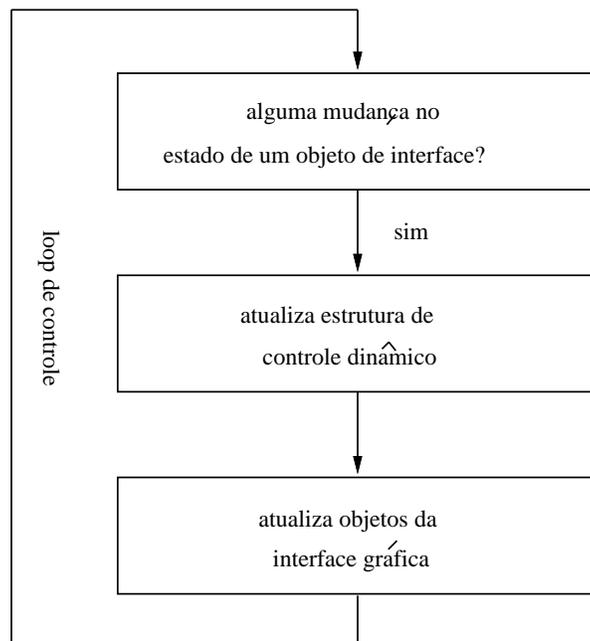


Figura 10.16: Loop de controle dinâmico dos objetos da interface gráfica.

Capítulo 11

Resultados e Exemplos

Neste capítulo, apresentaremos alguns exemplos de utilização das ferramentas de processamento de dados capturados existentes no sistema. Tais ferramentas são úteis para a criação de novos movimentos a partir de movimentos previamente capturados.

Em cada exemplo, será definida uma situação onde o processamento dos dados capturados se faz necessário para atingir o objetivo final. Utilizando os recursos existentes no sistema, apresentaremos as soluções obtidas em cada caso.

11.1 Procedimentos Iniciais de Operação do Sistema

Antes de começar a utilizar as ferramentas para o processamento de dados capturados, é necessário executar um conjunto de procedimentos iniciais, comuns a todas as situações e, particularmente, às apresentadas a seguir. Nesta seção, procuraremos descrever tais procedimentos de forma clara e objetiva.

A inicialização do sistema *MC Animator* apresenta três objetos de interface (Figura 11.1), que representam, da esquerda para a direita, o painel principal, a janela de rendering e o painel de controle interativo da animação. O funcionamento e operação da maioria dos objetos apresentados nesta seção foram descritos no Capítulo 10.

O painel principal do sistema, apresentado em detalhe na Figura 11.2, concentra as referências para as entidades e operações existentes no sistema, divididas em duas colunas. A existência desse painel facilita o acesso aos objetos do sistema e reduz a poluição e confusão visual causadas por inúmeras janelas abertas ao mesmo tempo na tela do computador.

11.1.1 Definição do Esqueleto do Ator Virtual

O primeiro passo na operação do sistema é definir o esqueleto que será utilizado pelo ator virtual. Isto é feito através do menu do painel principal. Selecionando-se a opção *Skeleton*, é oferecido um menu com os tipos de esqueleto suportados pelo sistema. Em sua atual versão, o sistema suporta apenas um tipo de esqueleto, compatível com os movimentos existentes na biblioteca que o acompanha. No futuro, pretende-se incluir novos tipos de esqueletos, o que permitirá a aplicação de movimentos capturados em estruturas articuladas com diferentes topologias.



Figura 11.1: Tela de entrada do sistema *MC Animator*.

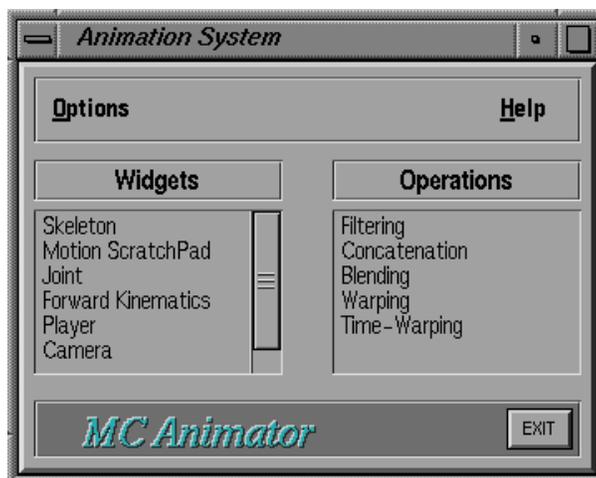


Figura 11.2: Detalhe do painel principal do sistema.

Após a escolha do esqueleto o ator virtual é automaticamente visualizado, em posição anatômica (Figura 11.3), na janela de rendering.

11.1.2 Escolha dos Movimentos

Definido o esqueleto a ser utilizado pelo ator virtual, o usuário deve então selecionar os movimentos que pretende manipular. Como vimos nos capítulos anteriores, as componentes básicas desses movimentos (sinais unidimensionais) são mapeadas nas articulações do ator virtual, gerando assim a representação do movimento capturado.

Através do *Motion ScratchPad*, descrito no Capítulo 10, o usuário pode selecionar e visualizar os movimentos disponíveis na biblioteca que acompanha o sistema. Como vimos anteriormente, o *ScratchPad* funciona como uma ligação entre os módulos de entrada e processamento.

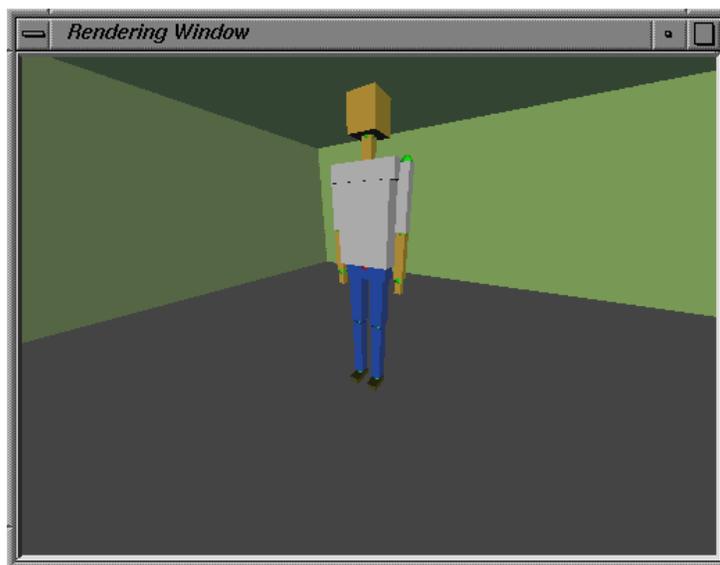


Figura 11.3: Ator virtual em posição anatômica (posição inicial).

Na Figura 11.4 é apresentada uma visão do *ScratchPad*, contendo os movimentos que serão utilizados nos exemplos da próxima seção. O usuário pode selecionar cada movimento individualmente, visualizando-o na janela de rendering e, se desejar, pode também direcioná-lo para uma das operações de processamento.

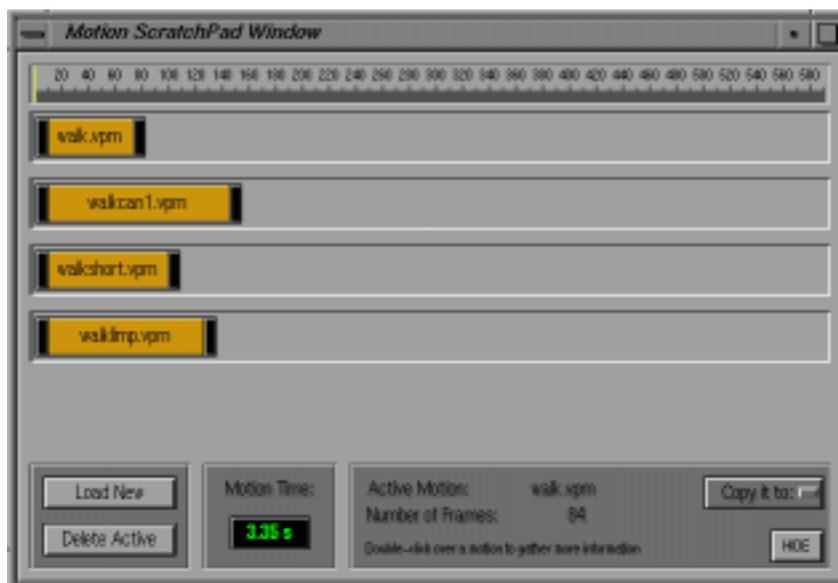


Figura 11.4: *ScratchPad* contendo os movimentos utilizados nos exemplos.

11.1.3 Customização de Parâmetros Auxiliares

Através de um menu localizado no painel principal do sistema, é possível definir em tempo real diversos parâmetros associados à visualização da cena (complexidade, texturas

etc.) e à geometria do ator virtual, bem como parâmetros gerais de animação.

Embora a alteração dos valores iniciais desses parâmetros não seja fundamental para o entendimento e operação do sistema, em certos casos ela pode se mostrar útil para uma melhor compreensão de aspectos específicos de um movimento.

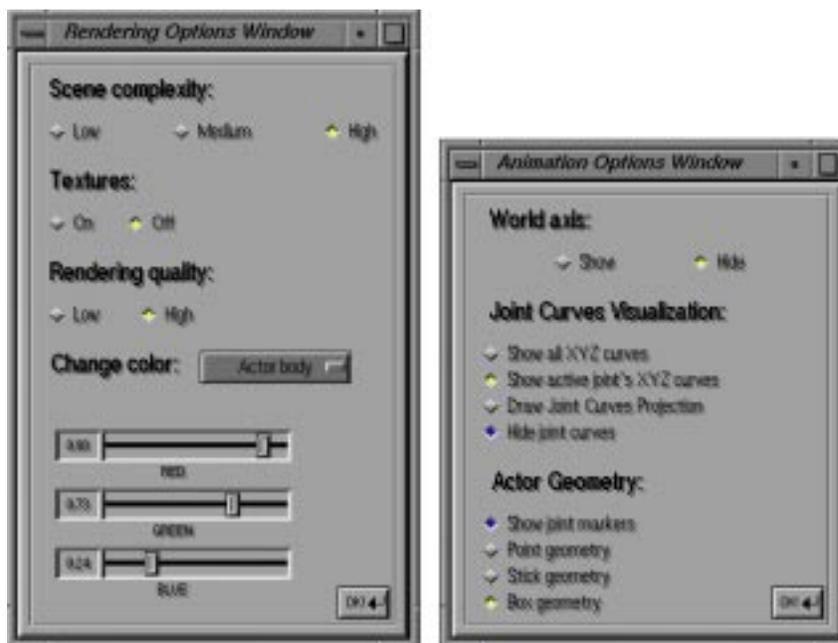


Figura 11.5: Janelas de customização de parâmetros de visualização e animação do sistema.

11.2 Definição dos Exemplos e Soluções Apresentadas

Nesta seção serão apresentados alguns exemplos de utilização das ferramentas de processamento de dados capturados. Em cada exemplo, será definida uma situação com um objetivo a ser alcançado, juntamente com a descrição da solução obtida através da utilização do sistema.

Assumiremos que o leitor esteja familiarizado com os objetos de interface utilizados nesta seção. Para um melhor entendimento do funcionamento de tais objetos, recomendamos a leitura do Capítulo 10.

11.2.1 Exemplo 1 - Concatenação de Movimentos

Neste exemplo utilizaremos dois movimentos distintos: “caminhar normal” e “caminhar mancando”. O objetivo neste caso é criar um novo movimento onde o ator virtual inicia um “caminhar normal” e então executa uma transição para um “caminhar mancando”.

Para atingir este objetivo, os sinais que representam os dados das articulações dos movimentos originais devem ser combinados de modo a garantir uma transição suave entre eles.

Utilizaremos a operação de concatenação para atingir o objetivo desejado. Após o direcionamento dos movimentos para a janela de concatenação, os mesmos são posicionados automaticamente pelo sistema de modo a executar uma concatenação direta (Figura 11.6, esquerda). Embora este tipo de concatenação seja útil em um grande número de casos, neste exemplo necessitaremos de uma concatenação com transição suave entre os dois movimentos, de modo a obter um resultado razoável.

A janela de concatenação permite ao usuário estabelecer, interativamente, uma região de sobreposição entre os movimentos que serão combinados (Figura 11.6, direita). A transição suave é feita então interpolando-se os valores das curvas de movimento na região de sobreposição definida pelo usuário. Esse processo é equivalente a um morphing entre os dois movimentos na região em questão.

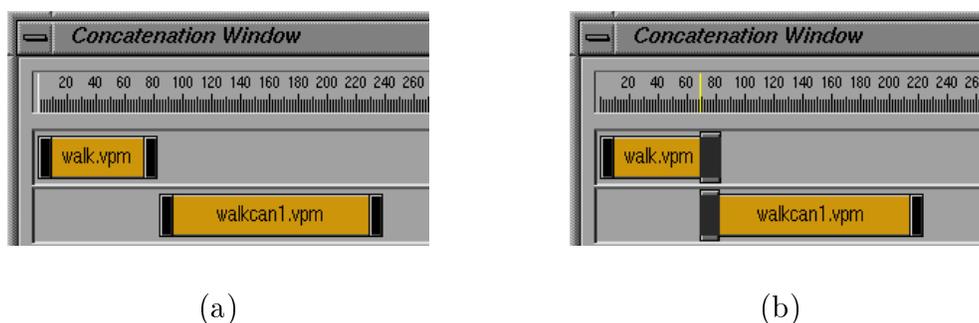


Figura 11.6: Concatenação de movimentos: (a) disposição inicial e (b) região de sobreposição.

Na Figura 11.7 podemos observar as curvas de uma determinada articulação, nos movimentos originais e no movimento resultante da operação de concatenação. Finalmente, na Figura 11.8, apresentamos alguns quadros da seqüência de animação gerada pelo sistema.

11.2.2 Exemplo 2 - Alteração da Velocidade (ou Pitch) de um Movimento

Neste exemplo, o objetivo é alterar a velocidade de um movimento (ou de partes de um movimento), tornando-as mais lentas ou mais rápidas. Uma aplicação prática deste tipo de operação é o sincronismo entre um movimento capturado e um som: podemos “esticar” ou “comprimir” temporalmente partes de um movimento de modo a sincronizá-lo com um som previamente gravado.

Conceitualmente, podemos tornar um sinal mais longo ou mais curto através da reamostragem da curva contínua que o representa. Todavia, no nosso caso teremos que trabalhar no domínio discreto, uma vez que os arquivos de dados capturados não fornecem descrições contínuas das curvas de movimento. Logicamente, seria possível obter uma representação contínua dos dados capturados através de um processo de interpolação.

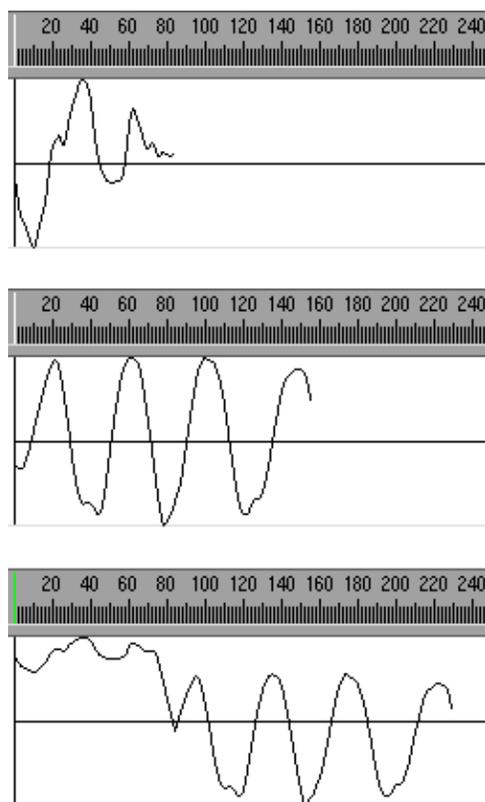


Figura 11.7: Comparação das curvas dos movimentos (de cima para baixo): “caminhar normal”, “caminhar mancando” e “novo movimento” (concatenação com sobreposição).

A mudança de velocidade (ou reparametrização) de um movimento pode ser feita de forma uniforme ou adaptativa. No primeiro caso, as curvas são reamostradas uniformemente ao longo do movimento. Já no caso adaptativo, uma função de reparametrização é utilizada para guiar o processo de reamostragem ao longo do movimento.

No sistema *MC Animator*, a reparametrização uniforme de um movimento é feita aumentando-se ou reduzindo-se interativamente o tamanho do objeto gráfico que o representa, utilizando para isso uma marca especial localizada no lado direito do objeto (Figura 11.9).

A alteração do tamanho do objeto gráfico acarreta uma alteração no número de frames (e, conseqüentemente, uma mudança na duração) do novo movimento. Internamente, o sistema utiliza essa informação para calcular e distribuir novas amostras que serão inseridas no movimento original. Na Figura 11.10, apresentamos o resultado de uma reparametrização uniforme do movimento “caminhar normal” (foi aplicado um fator de expansão de aproximadamente 1.9 no movimento original).

Para executar uma reparametrização adaptativa, o usuário terá que definir uma função de reparametrização. Tal função é definida em uma janela especial, que pode ser acessada através de um clique com o botão direito do mouse sobre o objeto que representa o movimento. Na Figura 11.11, é apresentada a janela de reparametrização adaptativa. Note a existência de duas escalas. A escala horizontal representa a duração do movimento original, em sua velocidade normal. A escala vertical é relacionada com a velocidade

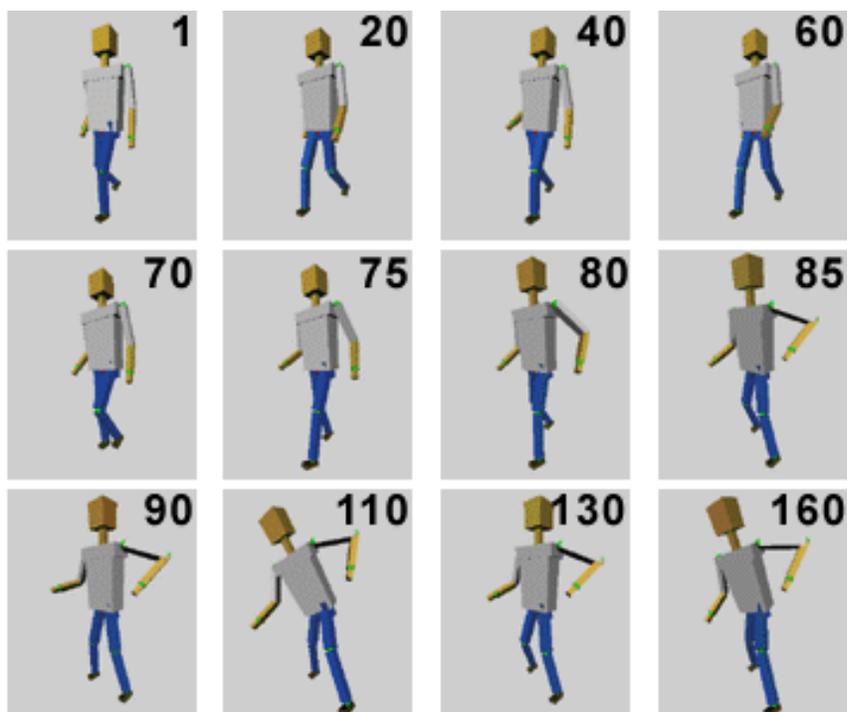


Figura 11.8: Sequência com a seleção de alguns quadros da animação resultante (conca-tenação). A região de sobreposição ocorre entre os quadros 70 e 90.



Figura 11.9: Detalhe da barra de movimento com marca especial (\leftrightarrow), utilizada na reparametrização uniforme.

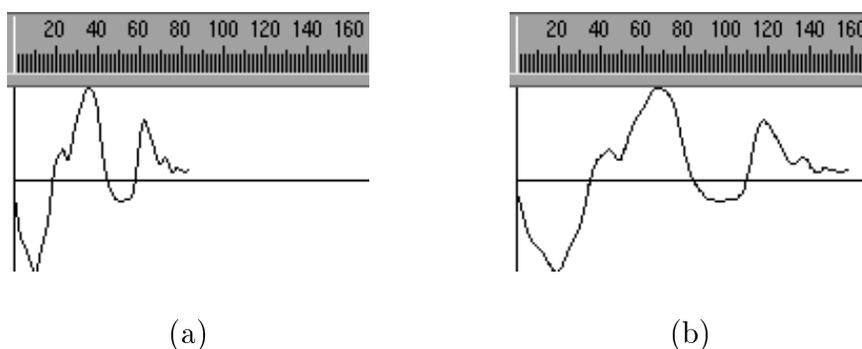


Figura 11.10: Resultado da reparametrização uniforme em uma curva do movimento “caminhar normal”: (a) curva original e (b) curva reparametrizada uniformemente (fator de expansão $\cong 1.9$).

(quadros por segundo) que pode ser associada a cada instante do movimento.

No exemplo da Figura 11.11, foi utilizada como função de reparametrização uma

poligonal, que passa por pontos de controle definidos pelo usuário. Essa função associa a cada instante de tempo no movimento original uma velocidade, que será utilizada no cálculo e na distribuição temporal dos quadros do novo movimento. Para obter resultados mais suaves, é possível utilizar outros tipos de funções de interpolação como, por exemplo, splines.

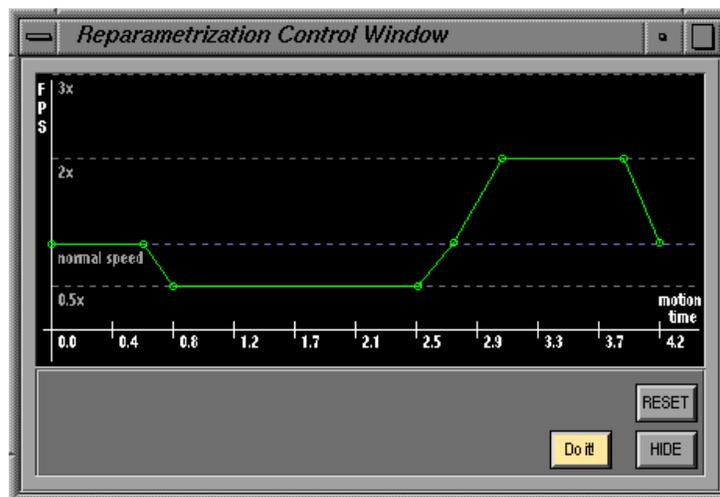


Figura 11.11: Função de reparametrização definida pelo usuário.

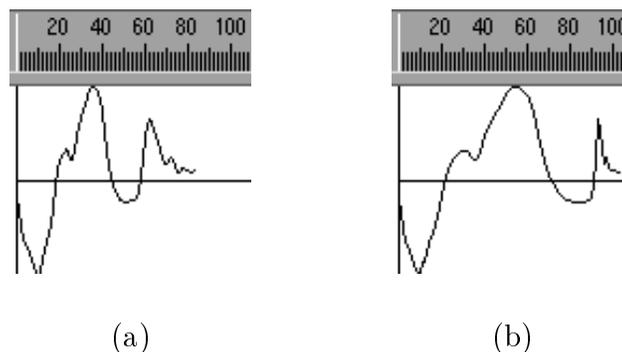


Figura 11.12: Resultado da reparametrização adaptativa em uma curva do movimento “caminhar normal”, utilizando a função definida na figura anterior: (a) curva original e (b) curva reparametrizada adaptativamente. Note a expansão e compressão de determinadas regiões do novo sinal em relação ao original.

11.2.3 Exemplo 3 - Warping de Curvas de Movimento

Um grande problema do processo de captura acontece quando o movimento capturado não corresponde às expectativas iniciais. Isto pode ser causado tanto devido a falhas do ator durante a execução do movimento, quanto a problemas técnicos ocorridos durante o processo de captura. Com isso, todo o processo tem que ser repetido, o que consome mais tempo e aumenta os custos de produção.

Entretanto, na maioria dos casos é possível “modificar” as curvas de movimento de articulações específicas de modo a alterar o seu comportamento no espaço e, com isso, corrigir o movimento capturado. Tal alteração do comportamento das curvas de movimento pode ser conseguido utilizando-se diferentes técnicas de processamento de sinais. Neste exemplo, utilizaremos a técnica de warping, descrita no Capítulo 8.

Considere a situação onde é preciso mapear o movimento “caminhar normal” em um ator virtual inserido em um cenário (também virtual) que possua um obstáculo fixo. Originalmente, o movimento “caminhar normal” foi capturado em um cenário real plano, com o ator caminhando em linha reta e sem qualquer tipo de obstáculo para transpor. Conseqüentemente, ao inserir o movimento no cenário virtual obteremos um efeito visual incorreto, com o ator virtual simplesmente ignorando o obstáculo, atravessando-o como mostra a Figura 11.13.

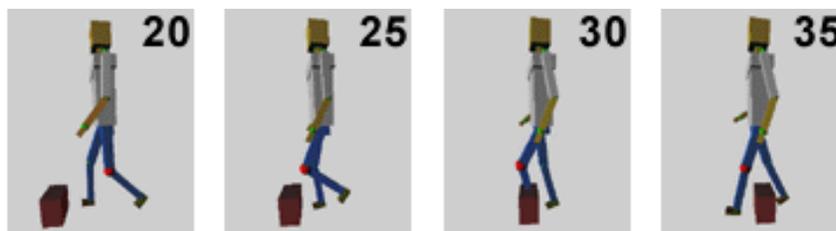


Figura 11.13: Inserção do movimento “caminhar normal” em um cenário virtual com obstáculo. A colisão da perna esquerda do ator virtual com o obstáculo ocorre no quadro 30.

Utilizando as ferramentas de warping existentes no sistema, é possível alterar o comportamento das curvas de movimento de uma articulação. No caso do exemplo, foi necessário alterar as curvas de movimento das articulações da perna esquerda do ator virtual, desviando-a do obstáculo.

O processo consiste em escolher um *warp frame* da curva de movimento desejada (Figura 11.14, centro), cujo valor angular será alterado pelo usuário através dos recursos de forward kinematics oferecidos pelo sistema. Após isso, o usuário deve definir a região de warping, onde será feita uma distorção da curva de movimento original de modo a incluir o quadro alterado (*warp frame*). Então, o sistema utilizará as informações definidas pelo usuário para incorporar o novo quadro à curva da articulação (Figura 11.14, direita).

Na Figura 11.15, apresentamos alguns quadros da seqüência final gerada pelo sistema. Note a mudança no comportamento da perna esquerda do ator virtual na região de warping (entre os quadros 20 e 40), em comparação ao movimento original (Figura 11.13).

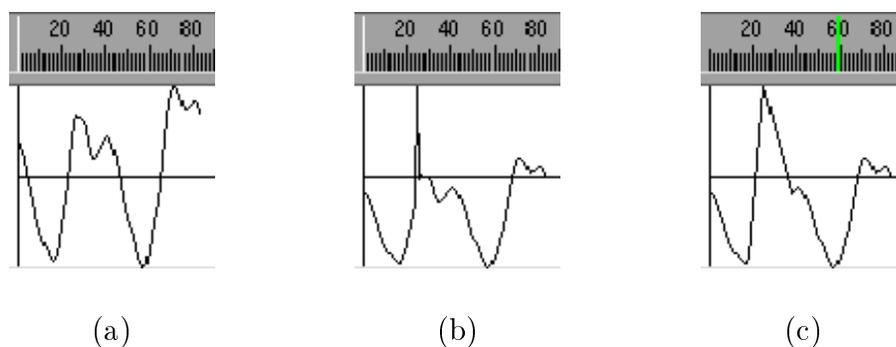


Figura 11.14: Etapas do processo de warping de uma curva de movimento: (a) curva original, (b) *warp frame* introduzido pelo usuário e (c) warping da curva original, englobando o *warp frame*.

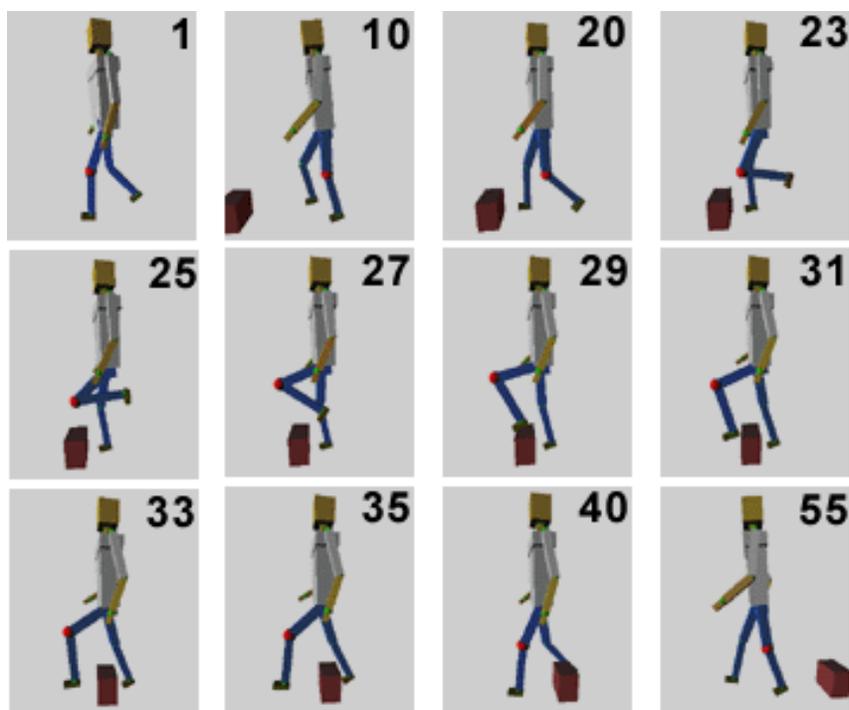


Figura 11.15: Seqüência com a seleção de alguns quadros da animação resultante (warping). As curvas de movimento das articulações da perna esquerda do ator virtual foram distorcidas, desviando-a do obstáculo. A região de warping ocorre entre os quadros 20 e 40.

Capítulo 12

Conclusão

12.1 Conclusões

Neste trabalho abordamos o problema de análise, modificação e reutilização de movimentos capturados, desenvolvendo o protótipo de um sistema de animação que tem Captura de Movimento como principal técnica de animação.

No Capítulo 2 estabelecemos uma conceituação para os diversos modelos de animação utilizados atualmente, definindo níveis de estratificação em relação aos objetos geométricos e aos parâmetros a eles associados.

No Capítulo 3 fizemos uma análise detalhada do Ator Virtual utilizado no sistema desenvolvido neste trabalho, abordando o processo de simplificação do corpo humano de modo a adequá-la ao uso em estruturas articuladas. Estudamos também diversos tipos de notação para estruturas articuladas existentes na literatura propondo, inclusive, uma variante adequada para a utilização com dados capturados. Abordamos ainda a representação de estruturas articuladas sob a forma hierárquica.

O Capítulo 4 teve como objetivo estabelecer uma conceituação geral do processo de Animação, dividindo-a em etapas que foram brevemente analisadas. No âmbito desta conceituação, inserimos o processo de Captura de Movimento como uma etapa independente.

O Capítulo 5 foi dedicado a uma breve revisão das principais técnicas de síntese de movimentos disponíveis atualmente.

Nos Capítulos 6 e 7 fizemos uma análise dos principais equipamentos de captura disponíveis comercialmente, classificando-os de acordo com diversos critérios. Estudamos também os dados obtidos por esses equipamentos, observando a necessidade de um processo de conversão desses dados para um tipo especial, mais adequado para a utilização em estruturas articuladas hierárquicas. Neste ponto, introduzimos um novo método para a extração de ângulos relativos a partir da posição dos marcadores do processo de captura.

No Capítulo 8 fizemos um estudo das principais técnicas de processamento de movimentos existentes atualmente propondo, inclusive, uma classificação conceitual.

Os Capítulos 9 e 10 foram dedicados à apresentação do sistema *MC Animator*, que utiliza os fundamentos descritos nos capítulos anteriores. O sistema *MC Animator* introduz um novo conceito de sistemas de animação: sistemas baseados em movimento capturado. Para isso, foi necessário desenvolver uma arquitetura interna e interface gráfica adequadas para tratar as principais características e entidades relacionadas à técnica de Captura de Movimento.

Finalmente, no Capítulos 11, foram apresentados alguns exemplos de utilização do sistema, juntamente com os resultados obtidos.

Resumidamente, poderíamos dizer que as principais contribuições deste trabalho foram:

- **Conceituação do processo de Animação.** Estabelecemos uma análise do processo de Animação, ressaltando a diferença entre os objetos geométricos utilizados em Animação, seus parâmetros e os níveis de simulação desses parâmetros. Criamos também um diagrama ilustrativo do processo de Animação, inserindo a técnica de Captura de Movimento como uma etapa independente deste processo.
- **Análise da técnica de Captura de Movimento.** Foi feita uma análise detalhada do processo de captura, ressaltando suas diferenças em relação às principais técnicas disponíveis atualmente e discutindo aspectos tecnológicos relevantes. Estudamos métodos de representação de estruturas articuladas mais adequados à manipulação de dados capturados (representação hierárquica) e propusemos um método para a conversão de dados capturados para esse tipo de representação.
- **Um novo conceito de sistema de animação.** Desenvolvemos um sistema de animação voltado para a manipulação de dados capturados. Esse tipo de sistema é essencialmente diferente dos existentes atualmente, pois trata a técnica de Captura de Movimento utilizando paradigmas de arquitetura interna e interface inéditos. Esse tipo de sistema possui aplicações importantes, tanto na indústria de efeitos especiais quanto na área de pesquisa.

12.2 Trabalhos Futuros

O campo de análise e processamento de movimentos é bem recente e, por isso, oferece inúmeras possibilidades para o desenvolvimento de pesquisa. A seguir, apresentamos algumas sugestões para trabalhos futuros, seguindo a linha de Captura de Movimento:

- **Novas técnicas de processamento de movimentos.** A grande maioria das técnicas de processamento de sinais pode ser aplicada a dados capturados. Uma investigação mais detalhada dessas técnicas pode resultar em novas ferramentas para a manipulação de movimentos capturados.
- **Combinação de Movimento Capturado com outras técnicas de animação.** É interessante estudar a possibilidade de utilização de movimentos capturados em conjunto com outras técnicas de animação como, por exemplo, Animação Procedural e Simulação Dinâmica. Movimentos capturados poderiam ser utilizados, por exemplo, para auxiliar processos dinâmicos ou para “guiar” atores virtuais comandados por funções procedurais.
- **Mapeamento de dados capturados em esqueletos com dimensão variável.** Um problema interessante, não abordado neste trabalho, é o desenvolvimento de técnicas de adaptação de um movimento capturado para esqueletos com diferentes tamanhos. Além disso, seria igualmente interessante desenvolver métodos para o mapeamento de dados capturados de seres humanos em objetos manufaturados como, por exemplo, uma cadeira.

Especificamente em relação ao sistema desenvolvido neste trabalho, podemos citar as seguintes sugestões, planejadas para futuras implementações:

- **Incorporação de outras técnicas de processamento de movimentos.** A implementação de novas técnicas de processamento de movimentos existentes na literatura aumentaria a flexibilidade do sistema.
- **Expansão da biblioteca de movimentos.** A implementação de rotinas para a leitura e o mapeamento de novos formatos de dados capturados aumentaria a variedade de movimentos disponíveis para o usuário.
- **Implementação de métodos mais eficientes para a manipulação de ângulos.** O uso de quaternions (Watt & Watt, 1992) poderia facilitar o processo de manipulação interna dos ângulos relativos, pois fornece um método mais robusto e eficiente de parametrizar a orientação dos segmentos da estrutura articulada.

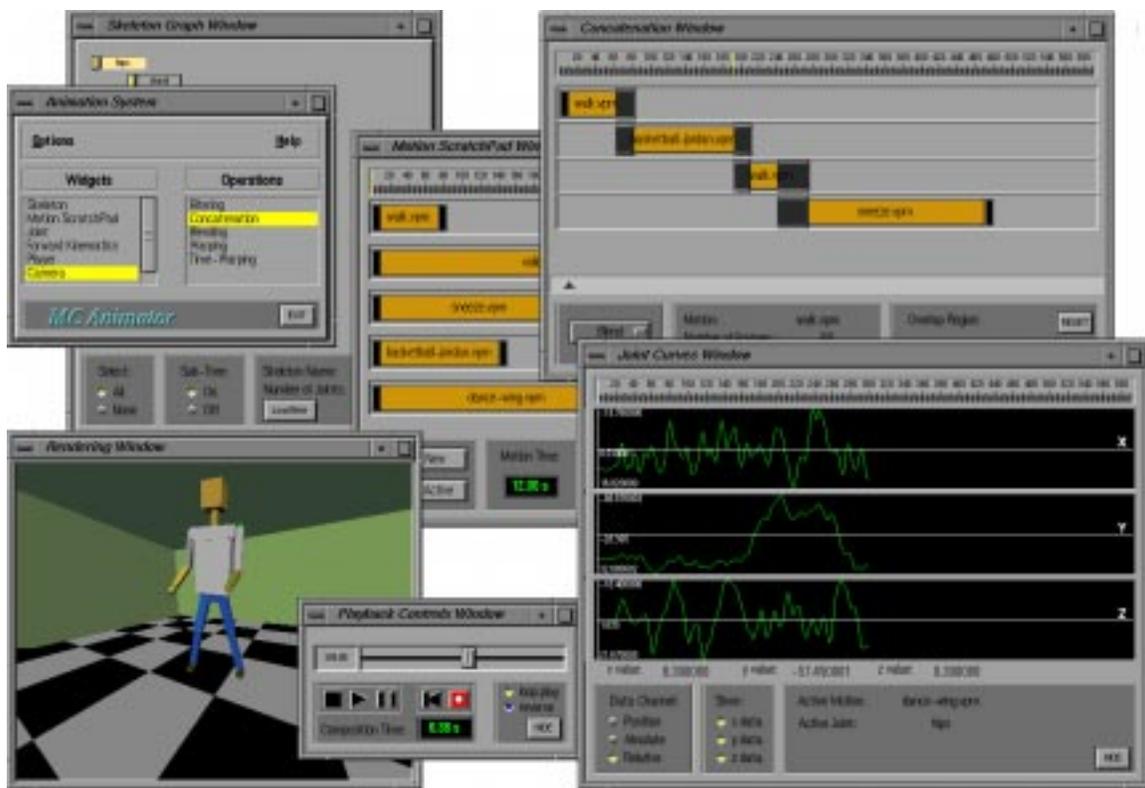


Figura 12.1: O sistema *MC Animator*.

()

Referências Bibliográficas

- Amaya, Kenji; Bruderlin, Armin & Calvert, Tom. 1996. Emotion from Motion. *In: Proceedings of Computer Animation (CA '96)*.
- Animation Laboratory, GVU. 1997. <http://www.cc.gatech.edu/gvu/animation/>.
- Azarbayerjani, A. & Wren, C. 1996 (May). Real-Time 3D Tracking of the Human Body. *In: Proceedings of IMAGE'COM 96*.
- Badler, N. I.; Barsky, B. & Zeltzer, D. 1991. *Making Them Move*. Morgan Kaufmann Publishers Inc.
- Badler, N. I.; Phillips, Cary B. & Webber, Bonnie L. 1993. *Simulating Humans*. Oxford University Press.
- Baraff, David. 1995. Interactive Simulation of Solid Rigid Bodies. *IEEE Computer Graphics and Applications*, **15**(3), 63–75.
- BioVision. 1997. <http://www.biovision.com>.
- Blinn, Jim. 1987. Nested Transformations and Blobby Man. *IEEE Computer Graphics and Applications*, October, 59–65.
- Bruderlin, Armin & Williams, Lance. 1995 (August). Motion Signal Processing. *Pages 97–104 of: Proceedings of Computer Graphics (SIGGRAPH'95)*.
- Burt, P. & Adelson, E. 1983. A Multiresolution Spline with Application to Image Merging. *ACM Transactions on Graphics*, **2**(4), 217–236.
- Carlson, Deborah & Hodgins, Jessica. 1997. Simulation Levels of Detail for Real-time Animation. *In: Proceedings of Graphics Interface'97*.
- Computer Graphics Laboratory, LIG. 1997. <http://ligwww.epfl.ch/>.
- Costa, Mônica & Feijó, Bruno. 1996 (October). An Architecture for Concurrent Reactive Agents in Real-Time Animation. *Pages 281–288 of: Proceedings of IX SIBGRAPI*.
- Costa, Mônica; Feijó, Bruno & Schwabe, Daniel. 1995 (October). Reactive Agents in Behavioral Animation. *Pages 159–166 of: Proceedings of VIII SIBGRAPI*.
- da Silva, Fernando Wagner. 1996 (December). *Apostila de Referências sobre Tecnologias de Motion Capture*. Report. VISGRAF Laboratory - IMPA.

- da Silva, Fernando Wagner. 1997a (April). *Motion Capture - Introdução à Tecnologia*. Report. VISGRAF- Laboratory - IMPA.
- da Silva, Fernando Wagner. 1997b (April). *Um Sistema de Animação Baseado em Movimento Capturado - Descrição Técnica*. Report. VISGRAF Laboratory - IMPA.
- da Silva, Fernando Wagner & Cavalcanti, P. Roma. 1996 (October). Animações Complexas em Tempo Real Usando Movimento Capturado. *Pages 333–334 of: Velho, Luiz; de Albuquerque, Alnaldo & Lotufo, Roberto A. (eds), Anais do IX SIBGRAPI*. Sociedade Brasileira de Computação, Caxambú, Brazil.
- da Silva, Fernando Wagner & Oliveira, Luís Fernando. 1996 (March). *Visão Computacional Aplicada à Animação por Computador*. Report ES-375/96. LCG - COPPE/UFRJ. (In Portuguese).
- da Silva, Fernando Wagner; Velho, Luiz; Cavancanti, Paulo Roma & Gomes, Jonas. 1997a (Outubro). An Architecture for Motion Capture Based Animation. *In: de Figueiredo, Luiz Henrique (ed), Anais do X SIBGRAPI*. Sociedade Brasileira de Computação, Campos do Jordão, Brazil.
- da Silva, Fernando Wagner; Velho, Luiz; Cavancanti, Paulo Roma & Gomes, Jonas. 1997b. A New Interface Paradigm for Motion Capture Based Animation Systems. *Pages 19–36 of: Thallmann, Daniel & van de Panne, Michiel (eds), Proceedings of the 8th EUROGRAPHICS Workshop on Animation and Simulation (CAS'97)*. Budapest, Hungary: Springer-Verlag Wien.
- Datalabs, Viewpoint. 1997. <http://www.viewpoint.com>.
- de Figueiredo, Luiz Henrique. 1996 (February). *Métodos Eficientes para Aproximação de Curvas e Superfícies*. Tech. rept. IMPA. Notas de Aula do Curso de Verão.
- Denavit, J. & Hartenberg, R. S. 1955. A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *Applied Mechanics*, june, 215–21.
- Doyle, Audrey. 1997. The Great Animation Debate. *Digital Magic Magazine*, **1**(1), 3–10.
- Dyer, S.; Martin, J. & Zulauf, J. 1995 (December). *Motion Capture White Paper*. Report. Silicon Graphics.
- Esposito, Chris & Paley, W. Bradford. 1996. Of Mice and Monkeys: A Specialized Input Device for Virtual Body Animation. *In: Proceedings of the 1995 Symposium of Interactive 3D Graphics*.
- Foley, J. & Dam, A. Van. 1993. *Computer Graphics - Principles and Practice*. New York, USA: Addison-Wesley.
- Fournier, Alan. 1986. A Simple Model of Ocean Waves. *Pages 75–84 of: Proceedings of Computer Graphics (SIGGRAPH'86)*.
- Ginsberg, Carol Marsha. 1983 (May). *Human Body Motion as Input to an Animated Graphical Display*. M.Phil. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Massachusetts, USA.

- Glaeser, Georg. 1994. *Fast Algorithms for 3D-Graphics*. first edn. Berlin, Germany: Springer-Verlag.
- Gleicher, Michael. 1997. Motion Editing with Spacetime Constraints. *In: Proceedings of the 1997 Symposium of Interactive 3D Graphics*.
- Gleicher, Michael & Litwinowicz, Peter. 1996 (June). *Constraint-Based Motion Adaptation*. Report 96-153. Apple Computer.
- Goldenstein, Siome Klein. 1997. *Metamorfose de Sons e Aplicações*. M.Phil. thesis, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, PUC-RIO.
- Gomes, Jonas & Velho, Luiz. 1994. *Computação Gráfica: Imagem*. first edn. Rio de Janeiro, Brazil: IMPA - Instituto de Matemática Pura e Aplicada.
- Gomes, Jonas & Velho, Luiz. 1997. *Image Processing for Computer Graphics*. first edn. Rio de Janeiro, Brazil: Springer-Verlag.
- Graphics, Silicon. 1995 (December). *Virtual Sets*. Web document. Silicon Graphics.
- Hodgins, J. K. & Raibert, M. H. 1991 (July). Animation of Dynamic Legged Locomotion. *Pages 349–358 of: Proceedings of Computer Graphics (SIGGRAPH'91)*.
- Hodgins, Jessica K.; Wooten, W. L.; Brogan, D. C. & O'Brien, J. F. 1995. Animating Human Athletics. *Pages 71–78 of: Proceedings of Computer Graphics (SIGGRAPH'95)*.
- IMIX. 1996. *Turbo Cube / Video Cube - User's Guide*. IMIX Company.
- Lasseter, John. 1987 (July). Principles of Traditional Animation Applied to 3D Computer Animation. *Pages 35–44 of: Stone, Maureen C. (ed), Computer Graphics (SIGGRAPH '87 Proceedings)*, vol. 21.
- Laszlo, Joseph; de Panne, Michiel Van & Fiume, Eugene. 1996 (August). Limit Cycle Control and its Application to the Animation of Balancing and Walking. *Pages 155–162 of: Proceedings of Computer Graphics (SIGGRAPH'96)*.
- Magnenat-Thalmann, Nadia & Thalmann, Daniel. 1990. *Computer Animation: Theory and Practice*. Second edn. Tokio, Japan: Springer-Verlag.
- Manocha, Dincsh & Canny, John F. 1995. Efficient Inverse Kinematics for General 6R Manipulators. *IEEE Transactions on Robotics and Automation*, **1**(1).
- Manocha, Dincsh & Zhu, Yunshan. 1994. A Fast Algorithm and System for the Inverse Kinematics of General Serial Manipulators. *In: Proceedings of IEEE Conference on Robotics and Automation'94*. IEEE.
- Maxwell, Delle Rae. 1983 (June). *Graphical Marionette: A Modern-Day Pinocchio*. M.Phil. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Massachusetts, USA.

- Molet, Tom; Huang, Zhiyong; Boulic, Ronan & Thalmann, Daniel. 1997. An Animation Interface Designed for Motion Capture. *Pages 77–85 of: Proceedings of Computer Animation'97*. IEEE CS Press.
- Mulder, S. 1994 (July). *Human Movement Tracking Technology*. Report 94-1. Hand Centered Studies of Human Movement Project, Simon Fraser University.
- Musse, Soraia Raupp & Thalmann, Daniel. 1997. A Model of Human Crowd Behavior. *In: Thalmann, Daniel & van de Panne, Michiel (eds), Proceedings of the 8th EURO-GRAPHICS Workshop on Animation and Simulation (CAS'97)*. Budapest, Hungary: Springer-Verlag Wien.
- Neider, Jackie; Davis, Tom & Woo, Mason. 1993. *OpenGL Programming Guide - The Official Guide to Learning OpenGL*. First edn. New York, USA: Addison-Wesley.
- of Vision Ray-Tracer, Persistence. 1997. <http://www.povray.org>.
- O'Rourke, Joseph. 1994. *Computational Geometry in C*. First edn. Cambridge, USA: Cambridge University Press.
- Ousterhout, John K. 1994. *Tcl and the Tk Toolkit*. first edn. New York, USA: Addison-Wesley.
- Page, FX Fighter. 1995. <http://www.im.gte.com/FXF/xfwhat2.html>.
- Perlin, Ken. 1995. Real Time Responsive Animation with Personality. *IEEE Transactions on Visualization and Computer Graphics*, **1**(1), 5–15.
- Perlin, Ken & Goldberg, Athomas. 1996 (August). Improv: A System for Scripting Interactive Actors in Virtual Worlds. *In: Proceedings of Computer Graphics (SIGGRAPH'96)*.
- Reeves, W. T. 1983. Particle Systems – a Technique for Modeling a Class of Fuzzy Objects. *ACM Trans. Graphics*, **2**(apr), 91–108.
- RenderMan, Pixar's. 1997. <http://www.pixar.com/renderman>.
- Reynolds, C. W. 1987 (July). Flocks, Herds and Schools: a Distributed Behavioural Model. *Pages 25–34 of: Proceedings of Computer Graphics (SIGGRAPH'87)*.
- Rogers, David F. 1991. *Procedural Elements for Computer Graphics*. New York, USA: McGraw-Hill.
- Rose, Charles; Guenter, Brian; Bodenheimer, Bobby & Cohen, Michael F. 1996 (August). Efficient Generation of Motion Transitions using Spacetime Constraints. *Pages 147–154 of: Proceedings of Computer Graphics (SIGGRAPH'96)*.
- Sederberg, T. W. 1986. Free-Form Deformation of Solid Geometric Models. *Pages 151–160 of: Proceedings of Computer Graphics (SIGGRAPH'86)*.
- SIGGRAPH. 1994a (August). Character Motion Systems. *In: Proceedings of Computer Graphics (SIGGRAPH'94), Course Notes no. 9*.

- SIGGRAPH. 1994b (August). An Introduction to Physically Based Modeling. *In: Proceedings of Computer Graphics (SIGGRAPH'94), Course Notes.*
- SIGGRAPH. 1997 (July). *Online Panel on Motion Capture and CG Character Animation.* <http://www.siggraph.org/cgi-bin/s97/WebX.cgi>.
- Sims, Karl. 1990 (aug). Particle Animation and Rendering Using Data Parallel Computation. *Pages 405–413 of: Baskett, Forest (ed), Computer Graphics (SIGGRAPH '90 Proceedings)*, vol. 24.
- Terzopoulos, Demetri. 1994. Artificial Fishes with Autonomous Locomotion, Perception, Behavior and Learning, in a Physical World. *In: Proceedings of the Artificial Life IV Workshop.* MIT Press.
- Terzopoulos, Demetri & Witkin, Andrew. 1988. Physically Based Models with Rigid and Deformable Components. *IEEE Computer Graphics and Applications*, **8**(6), 41–51.
- Terzopoulos, Demetri; Witkin, A.; Kass, M. & Fleisher, K. 1987a. *Cooking with Kurt.* Computer Animation in ACM SIGGRAPH Video Review Issue 36.
- Terzopoulos, Demetri; Platt, John; Barr, Alan & Fleischer, Kurt. 1987b (July). Elastically Deformable Models. *Pages 205–214 of: Stone, Maureen C. (ed), Computer Graphics (SIGGRAPH '87 Proceedings)*, vol. 21.
- Terzopoulos, Demetri; Witkin, A. & Kass, M. 1988. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence*, **36**(1), 91–123.
- Unuma, Munetoshi; Anjyo, Ken & Takaeuchi, Ryozi. 1996. Fourier Principles for Emotion-based Human Figure Animation. *Pages 75–84 of: Proceedings of Computer Graphics (SIGGRAPH'96).*
- van de Panne, Michiel. 1996. Parametrized Gait Synthesis. *IEEE Computer Graphics and Applications*, **1**(1), 40–49.
- Velho, Luiz. 1986 (June). *SCRIPTS: On the Description of Computer Animated Images.* M.Phil. thesis, MIT Media Lab., Michigan, USA.
- Vince, John. 1992. *3-D Computer Animation.* first edn. Addison-Wesley Publishing Company.
- Virtual, OZ. 1997. <http://www.oz.com/ov>.
- Watt, Allan & Watt, Mark. 1992. *Advanced Animation and Rendering Techniques - Theory and Practice.* First edn. New York, USA: Addison-Wesley.
- Wiley, Douglas J. & Hahn, James K. 1997. Interpolation Synthesis for Articulated Figure Motion. *Pages 156–160 of: Proceedings of the 1997 IEEE Virtual Reality Annual International Symposium - VRAIS'97.* IEEE.
- Witkin, Andrew & Kass, Michael. 1988 (August). Spacetime Constraints. *Pages 159–168 of: Dill, John (ed), Proceedings of Computer Graphics (SIGGRAPH'88)*, vol. 22.

- Witkin, Andrew & Popović, Zoran. 1995 (August). Motion Warping. *Pages 105–108 of: Proceedings of Computer Graphics (SIGGRAPH'95)*.
- World, Computer Graphics. 1997. Real-Time Motion Capture. *Page 14 of: Computer Graphics World*, vol. 20. Computer Graphics World.
- Zeltzer, David & McKenna, M. 1990 (August). Dynamic Simulation of Autonomous Legged Locomotion. *Pages 29–38 of: Proceedings of Computer Graphics (SIGGRAPH'90)*.
- Zeltzer, David & Sims, F. A. 1988 (August). A Figure Editor and Gait Controller for Task Level Animation. *Pages 164–181 of: Proceedings of Computer Graphics (SIGGRAPH'88), Course Notes no. 4*.
- Zhao, T. C. & Overmars, Mark. 1996 (July). *Forms Library - A Graphical User Interface Toolkit for X*. Report. University of Wisconsin-Milwaukee. <http://bragg.phys.uwm.edu/xform>.