Um Sistema de Animação Baseado em Movimento Capturado*

FERNANDO WAGNER DA SILVA[1,2] LUIZ VELHO[2]

[1]LCG - Laboratório de Computação Gráfica, COPPE - Sistemas / UFRJ
21945-970, Rio de Janeiro, RJ, Brasil, Caixa Postal 68511

[2]IMPA-Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110, 22460, Rio de Janeiro, RJ, Brasil
{nando,lvelho}@visgraf.impa.br

Abstract. This work treats the problem of analysis, modification and reutilization of captured motion. A conceptual approach of the Motion Capture technique was made, situating it with respect to Computer Animation. The main contribution of this work is to introduce a "Motion Capture Based Animation System". The prototype of such system was built, implementing several motion processing techniques.

Resumo. Este trabalho aborda o problema de análise, modificação e reutilização de movimentos capturados. Foi feita uma conceituação do processo de captura no âmbito de Animação por Computador. A principal contribuição deste trabalho é introduzir um "Sistema de Animação Baseado em Movimento Capturado". Um protótipo desse sistema foi desenvolvido, implementando várias técnicas para a manipulação de dados capturados.

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, o movimento da câmera virtual.

1.1 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, o que tornava necessário o controle de inúmeros parâmetros e graus de liberdade simultaneamente. Neste trabalho, focalizaremos o estudo da animação de estruturas articuladas que representam seres

*Para maiores informações sobre este trabalho, incluindo versões on-line de artigos e vídeos, visite http://www.visgraf.impa.br/mocap/tese humanos. Uma dificuldade adicional associada a este processo se deve ao fato de seres humanos possuirem uma sensibilidade visual extremamente aguçada para perceber erros relacionados à ações cotidianas como, por exemplo, o movimento de andar.

Diversas técnicas foram desenvolvidas para facilitar a animação de estruturas complexas com diversas articulações e graus de liberdade. 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 e trabalhosa a representação correta de movimentos complexos.

Uma alternativa proposta foi a utilização de simulação dinâmica 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.

1.2 Animação por Movimento Capturado

Ferramenta de animação mais recente, a técnica de Movimento Capturado (MC) 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 e, mais recentemente, nos efeitos especiais dos filmes Titanic (1997), Tropas Estelares (1997) e Parque dos Dinossauros (1995). Essencialmente, a técnica de MC consiste em registrar, através de um processo de amostragem, a posição e orientação das articulações de um ator (ou *performer*) ao longo do tempo. Desta forma, é obtido um conjunto de sinais unidimensionais, que são então mapeados em um ator virtual, no computador. O processo de captura de movimento pode ser feito através da utilização de diferentes tipos de equipamentos, com tecnologias distintas. Uma análise detalhada destes equipamentos pode ser encontrada em [7].

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 à 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.

Nos últimos dois anos começaram a surgir as primeiras ferramentas para a manipulação de movimentos capturados, integradas como plug-ins em sistemas de animação consagrados como o SoftImage. Entretanto, tais ferramentas não abordaram de maneira conceitualmente correta o problema de manipulação de dados capturados, que são essencialmente diferentes dos obtidos através de técnicas tradicionais como *keyframing* e simulação. Por isso, uma abordagem específica deve ser utilizada para manipulá-los.

O objetivo deste trabalho é desenvolver um sistema de animação que tem a técnica de MC 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. Desta forma, o animador tem a possibilidade de criar bibliotecas de movimentos, que poderão ser combinadas e expandidas utilizando-se as ferramentas disponíveis no sistema.

A organização do artigo é descrita a seguir: a seção 2 apresenta o modelo de representação do corpo humano desenvolvido no trabalho. Na seção 3 é feita uma análise dos dados capturados, juntamente com a apresentação do método de extração de ângulos relativos desenvolvido neste trabalho. A seção 4 aborda o problema de processamento de movimentos, com exemplos extraídos do sistema implementado. Na seção 5 é apresentado o sistema de animação desenvolvido neste trabalho. Finalmente, a seção 6 apresenta alguns resultados obtidos através da utilização do sistema, juntamente com as conclusões e considerações finais.

2 Modelos de Representação do Corpo Humano

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 impraticável. Desenvolvemos neste trabalho um modelo simplificado do corpo humano, que consiste em uma estrutura articulada com inúmeros graus de liberdade, capaz de representar adequadamente os movimentos humanos.

O Modelo Utilizado no Sistema

Em nosso sistema de animação, criamos um modelo com 16 segmentos rígidos, conectados por articulações de revolução, com um total de 33 graus de liberdade[1]. A Figura 1 apresenta o modelo de ator virtual, 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 [14].

2.1 Representação Hierárquica

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

[1] O termo Graus de Liberdade, ou DOF, representa o número de posições independentes para rotação e translação que uma estrutura articulada pode assumir no espaço.

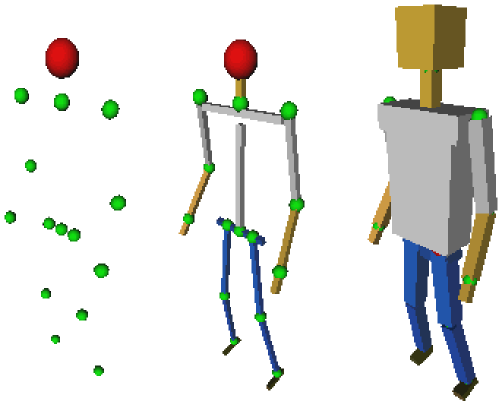


Figura 1: Modelo de ator virtual utilizado no sistema.

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. Embora existam diversos métodos de representação de estruturas articuladas, apenas o modelo hierárquico oferece condições para uma manipulação satisfatória dos parâmetros de animação associados à cada articulação do modelo, preservando ainda a integridade geométrica da estrutura.

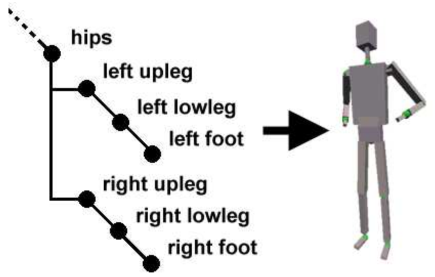
No modelo hierárquico, a posição de cada articulação é definida, automaticamente, através da composição em seqüência das matrizes de transformação das articulações anteriores. Desta forma, apenas a primeira articulação da estrutura precisa ser posicionada no espaço, enquanto que o resto da estrutura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

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 (Figura 2 (a)). Como podemos observar na Figura 2 (b), 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. Na literatura de Robótica [17], os segmentos externos de uma estrutura articulada (no caso do corpo humano, os pés e as mãos) são chamados de *end-effectors*.

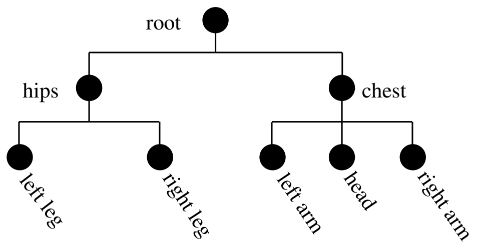
3 Os Dados Capturados

3.1 Composição e Representação

Embora atualmente exista uma boa variedade de tecnologias de captura de movimento [4], o núcleo fundamental do processo é sempre o mesmo: armazenar os movimentos através



(a)



(b)

Figura 2: (a) Representação topológica de uma estrutura articulada hierárquica e (b) Árvore que representa o ator virtual no sistema.

da amostragem uniforme das posições e/ou orientações globais de marcadores posicionados no corpo de um ator. Desta forma é gerado um conjunto de valores seqüenciais relativos à discretização da "curva real" do movimento do marcador em questão. Tais valores são então agrupados para formar os diversos quadros da animação.

3.2 Orientação Global vs. Orientação Relativa

Como vimos anteriormente, os dados capturados são constituídos por valores amostrados das posições e orientações globais de marcadores, ou seja, valores relativos a um sistema de coordenadas fixo (normalmente a origem $(0, 0, 0)$). Este fato torna impraticável qualquer tipo de manipulação dos dados capturados, 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.

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. Tal processamento consiste na conversão da informação de posição dos marcadores para ângulos relativos entre os segmentos da estrutura. No caso do ator virtual utilizado em nosso sistema, 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.

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 3. 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 3, 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*.

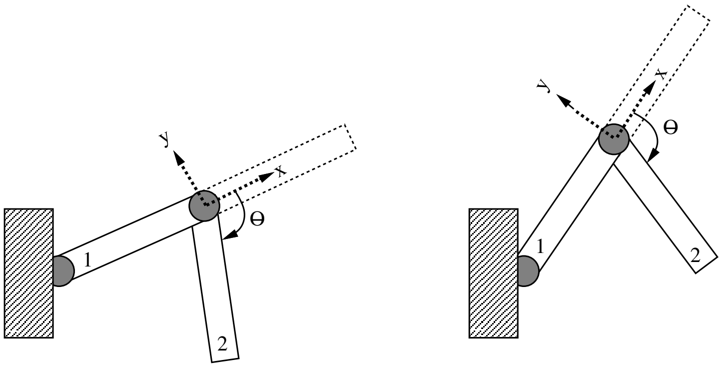


Figura 3: Ângulo relativo entre dois segmentos de uma estrutura articulada hierárquica.

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 4).

O método de cálculo de ângulos relativos para o esqueleto do ator virtual, desenvolvido neste trabalho, utiliza conceitos de geometria computacional [16]. 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

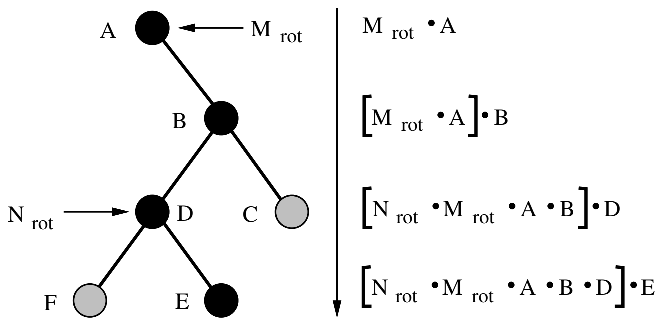


Figura 4: Expansão da composição de transformações em uma estrutura articulada hierárquica.

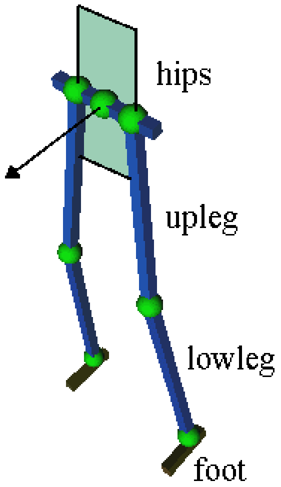
problema, podemos utilizar a própria estrutura do corpo humano como ferramenta para calcular a orientação do quadril.

Como podemos observar na Figura 5 (a), 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 5 (a). 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 feito de forma equivalente.

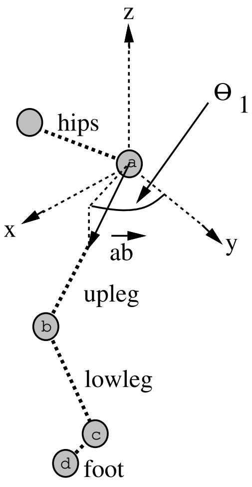
O ângulo relativo entre os segmentos *hips* e *upleg* é calculado através da projeção do vetor unitário $\overrightarrow{ab}$, definido pelas articulações dos segmentos *upleg* e *lowleg* (Figura 5 (b)), no sistema de coordenadas definido pela orientação do quadril, calculada anteriormente. Projetando-se $\overrightarrow{ab}$ nos planos coordenados, obtemos os ângulos de Euler relativos entre os dois segmentos em questão. No exemplo da Figura 5 (b), o ângulo relativo θ_1 foi obtido projetando-se $\overrightarrow{ab}$ 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. Nesse caso, a articulação que liga esses dois segmentos possui apenas 1 grau de liberdade, relativo à rotação em torno do eixo X que a atravessa perpendicularmente em relação ao seu comprimento (Figura 6). Desta forma, o ângulo relativo θ_2 é obtido através do produto escalar entre os vetores $\overrightarrow{ba}$ e $\overrightarrow{bc}$, ou seja, $\theta_2 = cos^{-1}(||\overrightarrow{ba}||.||\overrightarrow{bc}||)$.

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 arquivo de movimento não incluem a informação do último marcador do corpo do ator (na Figura 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*.



(a)



(b)

Figura 5: (a) Utilização do plano formado por três marcadores no cálculo da orientação do quadril (b) Cálculo do ângulo relativo entre o quadril e a coxa.

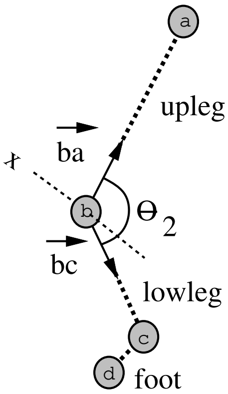


Figura 6: Cálculo do ângulo relativo entre os segmentos *upleg* e *lowleg*.

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. 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.

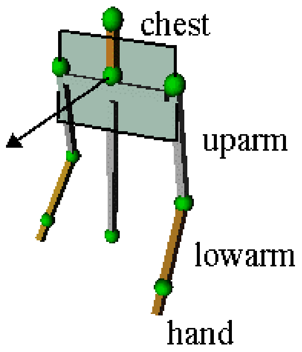


Figura 7: Utilização do plano formado por três marcadores no cálculo da orientação do peito.

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 pode se tornar bastante complexo, 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 em memória do arquivo de movimento pelo usuário. Após isso, os ângulos são mapeados nas devidas articulações da estrutura articulada hierárquica.

4 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. Nesta seção, analisaremos brevemente algumas dessas ferramentas. Para um estudo mais aprofundado sobre o assunto, sugerimos a leitura de [1].

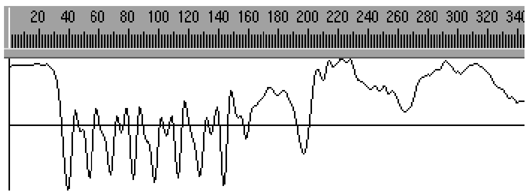
4.1 Exemplos de Operações

Como vimos anteriormente, 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 graus de liberdade 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 [1].

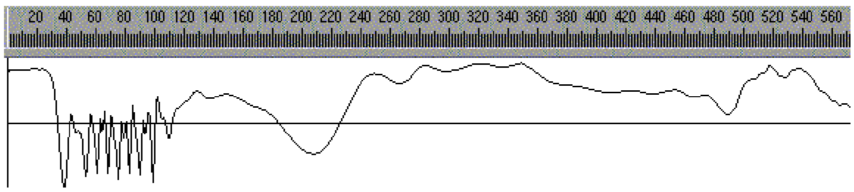
De fato, a grande maioria dessas técnicas pode ser aplicada a dados de movimento. No entanto, nem todas produzem como resultado um movimento coerente e válido. A seguir, apresentaremos alguns exemplos de técnicas de processamento de dados capturados existentes na literatura, e que foram implementadas em nosso sistema.

Filtragem

No campo de processamento de sinais, as aplicações de filtros são vastas. Filtros podem ser utilizados para eliminar ou atenuar freqüências específicas de um sinal. Podemos aplicar um filtro de suavização em uma curva (ou em um conjunto de curvas) de movimento, eliminando assim as altas freqüências e reduzindo a variação dos dados que formam o 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. Filtros também podem ser utilizados na ampliação de características ou duração de um movimento, através da reparametrização temporal do sinal de cada articulação [15]. Na Figura 8 é apresentado um exemplo de reparametrização temporal de uma curva de movimento. Note a expansão/compressão de regiões específicas do sinal original, causando efeitos de *slow-motion* e *accelerated-time*, respectivamente.



(a)



(b)

Figura 8: Reparametrização temporal: (a) curva de movimento original; (b) expansão/compressão de partes da curva através de uma função de reparametrização temporal.

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.

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 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. Na Figura 9 é apresentada uma animação gerada pelo sistema desenvolvido neste trabalho, resultante da concatenação de dois movimentos distintos: “caminhar normal” e “caminhar mancando”.

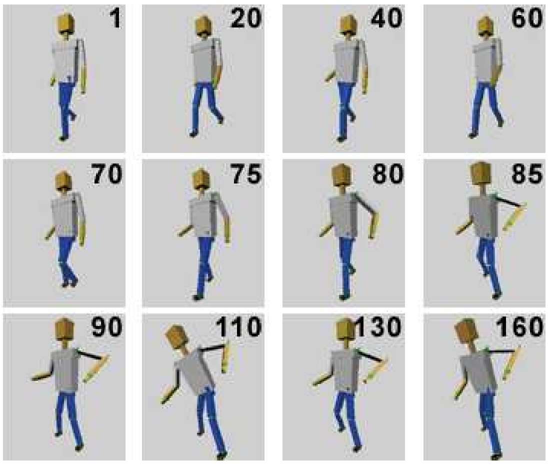


Figura 9: Concatenação: transição do movimento “caminhar normal” para o movimento “caminhar mancando”. A região de interpolação dos movimentos ocorre entre os quadros 70 e 90.

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.

No caso de movimentos, a operação de *blending* pode ser empregada para combinar características distintas de diferentes movimentos em um único movimento, como mostrado na Figura 10.

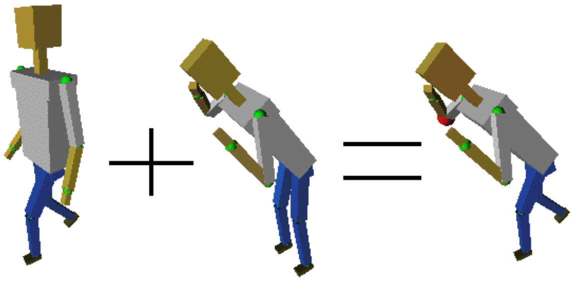


Figura 10: Combinação de diferentes curvas de movimento em conjuntos diferentes de articulações: conjunto 1 - articulações da parte superior do corpo; conjunto 2 - articulações da parte inferior do corpo.

Warping

Movimentos também podem ser "deformados" através do *warping* das curvas que contêm os dados capturados. 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 estabelecida uma deformação para transformar a curva original de modo a incluir o novo quadro modificado. No final deste artigo, apresentaremos um exemplo complexo da utilização desta técnica de processamento de movimentos.

5 Um Sistema de Animação Baseado em Movimento Capturado

Nesta seção, 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, que a tratam apenas como uma ferramenta secundária. 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.

5.1 Arquitetura

O sistema *MC Animator* possui uma arquitetura interna [3] desenvolvida tendo como objetivo principal oferecer um conjunto básico de ferramentas para o processamento de movimentos, permitindo a criação de animações complexas a partir de bibliotecas de movimentos básicos. Além disso, tal arquitetura fornece alternativas para algumas limitações tecnológicas do processo de captura. Pensando nisso, dividimos a arquitetura em três módulos básicos: entrada, processamento e saída (Figura 11). 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 básicas do sistema: o Ator Virtual e os Movimentos.

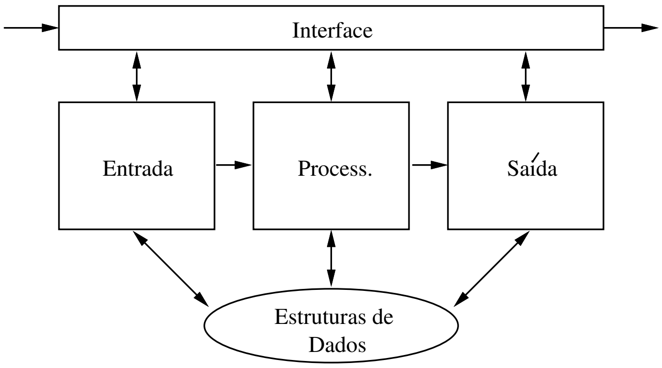


Figura 11: Fluxograma da arquitetura interna do sistema.

Cada módulo é responsável por um conjunto de tarefas específicas. No módulo de *Entrada* é feita a interpretação e pré-processamento dos dados de movimento, que inclui a conversão para ângulos relativos. No de *Processamento*, estão disponíveis operações de manipulação de dados capturados (filtragem, concatenação, blending, warping, reparametrização e outras). O módulo de *Saída* é responsável pelo armazenamento da composição feita pelo usuário, em formaro binário ou sob a forma de curvas de movimento.

5.2 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 software.

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 [2], diferentemente dos sistemas de animação existentes no mercado. Assim, escolhemos como ponto de partida no desenvolvimento da interface o modelo empregado em alguns sistemas digitais de edição de vídeo [12]. Esta 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. 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.

Representação das Entidades na Interface

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

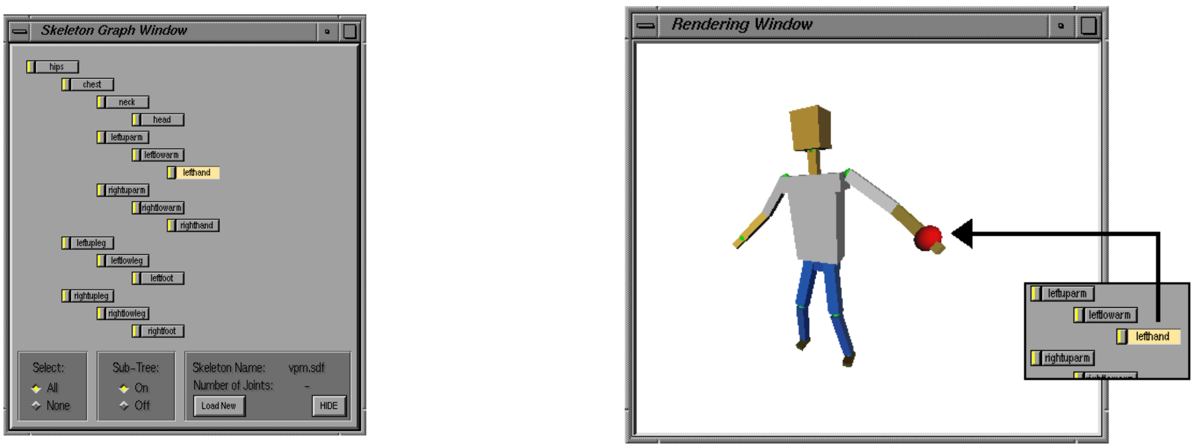
Capturado. Além disso, tal interface deve ser extensível, permitindo a inclusão de novos objetos e técnicas ao longo do tempo. Nas próximas seções, apresentaremos as principais entidades existentes na interface gráfica do sistema.

Ator Virtual

No paradigma de interface utilizado no sistema, o ator vitual é representado através de seus principais componentes: 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 12 (a)). 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 util em determinadas situações.

A representação geométrica do ator virtual é visualizada na janela de rendering, como mostrado na Figura 12 (b). 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.



(a) (b)

Figura 12: (a) *Skeleton Graph Window*: informações topológicas sobre o esqueleto articulado; (b) Representação geométrica do ator virtual. Seleção de articulação (detalhe da janela *Skeleton Graph*) e destaque na janela de rendering.

Movimentos

O modelo de interface utilizado no sistema trata cada movimento como uma barra horizontal (Figura 13). 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.

Figura 13: 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 14). 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).

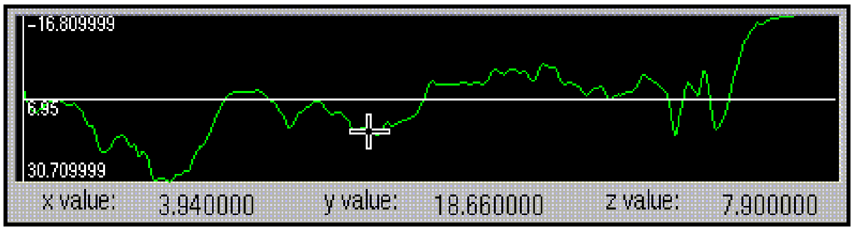
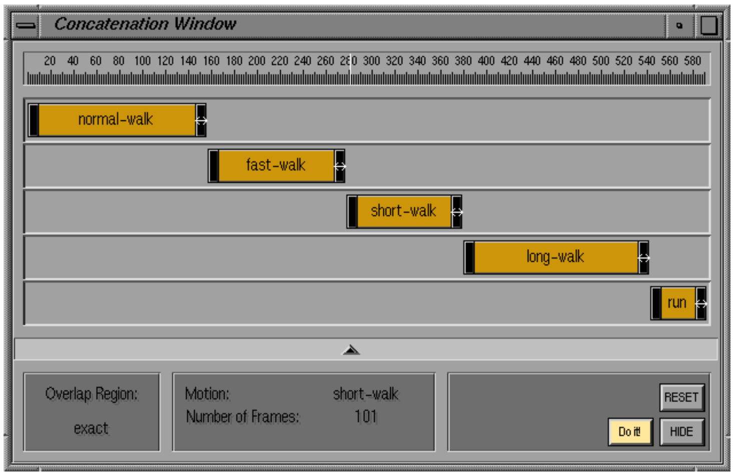


Figura 14: Representação de uma curva de movimento no sistema.

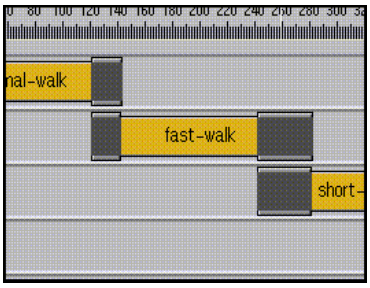
Utilizando esse tipo de representação podemos pensar na implementação de diversas técnicas para a edição de curvas existentes na literatura, fornecendo um controle preciso e interativo da forma da curva.

Processamento de Movimentos

O sistema de animação implementado neste trabalho conta com duversas janelas específicas para o processamento de movimentos. Em todos os casos, o paradigma de edição digital de vídeo foi utilizado como base no *design* de tais objetos de interface. Com isso, conseguimos alcançar o objetivo de tornar o processo de processamento de movimentos mais intuitivo e agradável para o usuário. Na Figura 15 é apresentada a janela onde é feita a concatenação de movimentos selecionados pelo usuário. Note o posicionamento relativo dos movimentos nos diversos “trilhos” (a), oferecendo a possibilidade de movimentação relativa entre eles de modo a definir as regiões de interpolação (b).



(a)



(b)

Figura 15: (a) - Concatenação de movimentos (configuração inicial); (b) - Intervalos de transição na concatenação de movimentos (detalhe).

Outros Elementos da Interface

O sistema proposto conta ainda com diversos outros elementos de interface, incluindo controles de playback de movimentos capturados e janelas específicas para o processamento de movimentos utilizando as ferramentas disponíveis no sistema. Para uma análise mais detalhada de tais elementos, sugerimos a leitura de [2]. Na Figura 16 é apresentada uma visão geral do sistema.

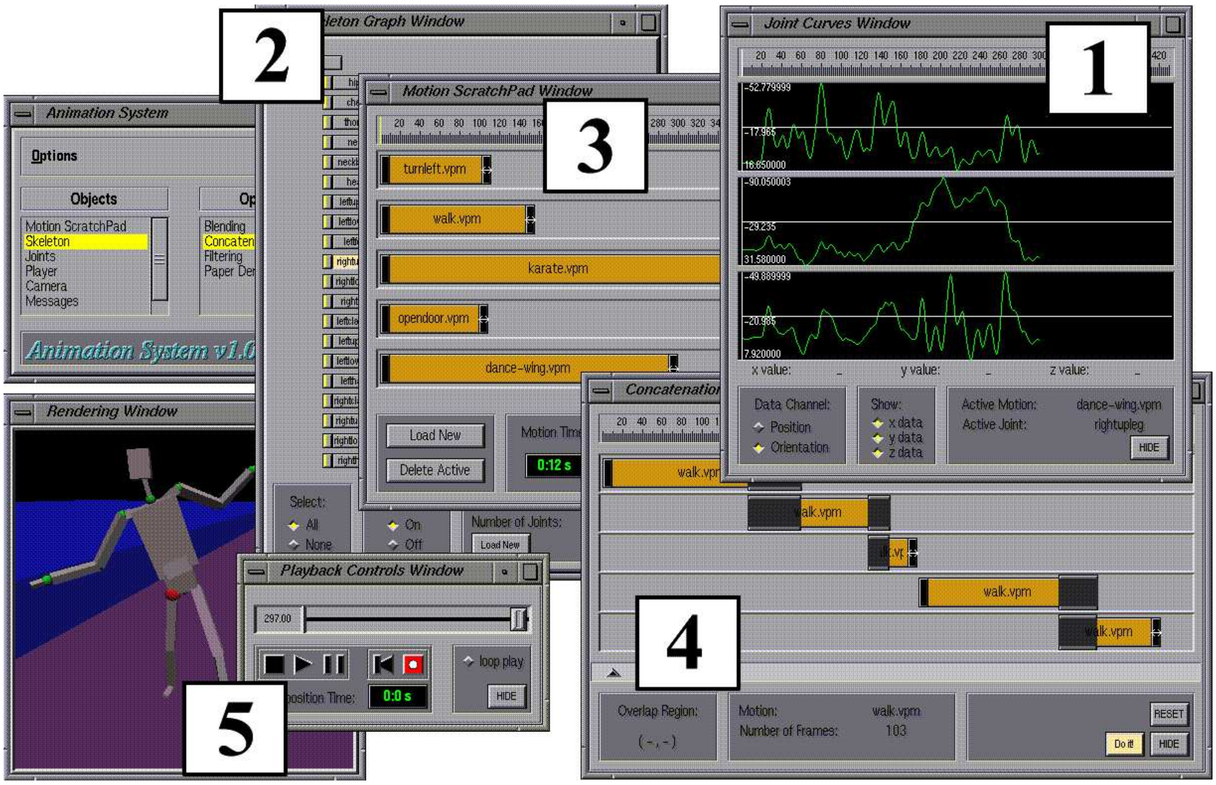


Figura 16: Alguns objetos de interface do sistema: *Joint Curves Window*, $\boxed{1}$; *Skeleton Graph Window*, $\boxed{2}$; *Motion ScratchPad Window*, $\boxed{3}$; *Concatenation Window*, $\boxed{4}$ e *Playback and Rendering Windows*, $\boxed{5}$.

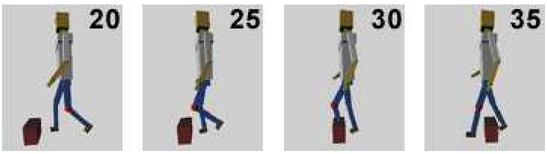
5.3 Ferramentas Adicionais e Detalhes de Implementação

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. 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 como biblioteca básica de rendering e o XForms como biblioteca básica de interface. Os objetos avançados de interface foram modelados e implementados separadamente, e então integrados à biblioteca do XForms.

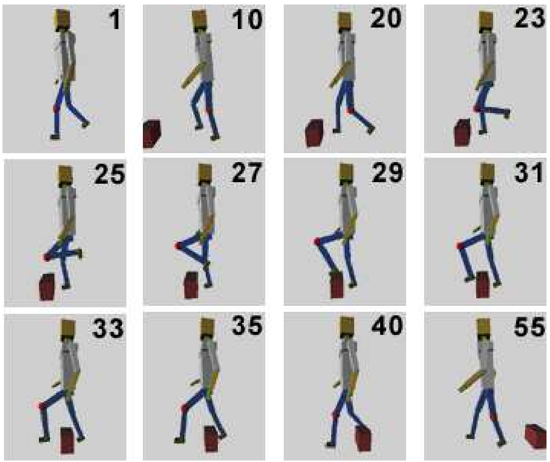
6 Resultados e Conclusões

A seguir apresentamos um resultado da utilização do sistema em uma situação real. Um grande problema do processo de captura acontece quando o movimento capturado não corresponde às expectativas iniciais, o que pode ser causado tanto devido a falhas do ator durante a execução do movimento, quanto a problemas técnicos ocorridos durante o processo de captura. Neste caso é importante ressaltar que o custo de produção de uma cena envolvendo movimentos capturados é bastante elevado. Assim, para reduzir os custos de produção é fundamental o sucesso do processo de captura logo nas primeiras tentativas.

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 17, (a). Utilizando as ferramentas de *warping* e reparametrização temporal 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 (Figura 17, (b)).



(a)



(b)

Figura 17: (a) 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. (b) Seqüência com a seleção de alguns quadros da animação resultante (warping + reparametrização temporal). As curvas de movimento das articulações da perna esquerda do ator virtual foram deformadas, desviando-a do obstáculo. A região de warping ocorre entre os quadros 20 e 40.

De maneira resumida, podemos concluir que a utilização em larga escala de sistemas especializados em movimento capturado oferece inúmeros atrativos. Na área de produção cinematográfica e de televisão, tais sistemas poderiam ser empregados para reduzir drasticamente os custos de produção dos efeitos especiais que utilizam movimentos capturados. Na área de pesquisa em computação gráfica, tais sistemas podem ser utilizados como plataforma para experiências no desenvolvimento de novas tecnologias de processamento de dados capturados.

De fato, várias pesquisas têm sido publicadas tendo como base o sistema desenvolvido neste trabalho ([2], [3], [4], [15] e [5]). Além disso, os conceitos aqui apresentados também têm sido empregados em projetos de universidades e grupos de pesquisa no exterior [18].

7 Referências

[1] WILLIAMS, L. AND BRUDELIN, A., Motion Signal Processing. In *Computer Graphics (SIGGRAPH'95 Proceedings)*(August 1995), pp. 97-104.

[2] SILVA, F., VELHO, L., GOMES, J. AND CAVALCANTI, P., A New Interface Paradigm for Motion Capture Based Animation Systems. In *Proceedings of the 8th EUROGRAPHICS Workshop on Computer Animation and Simulation - CAS'97*.

[3] SILVA, F., VELHO, L., GOMES, J. AND CAVALCANTI, P., An Architecture for Motion Capture Based Animation. In *Proceedings SIBGRAPI'97, X Brazilian Symposium of Computer Graphics and Image Processing*, pp. 49-56, october 1997.

[4] SILVA, F., Um Sistema de Animação Baseado em Movimento Capturado. Tese de M.Sc., COPPE/Sistemas, UFRJ, 1998. http://www.lcg.ufrj.br/∼nando/publ.html#thesis

[5] SILVA, F., VELHO, L. AND GOMES, J., Motion Cyclification by Time × Frequency Warping. Submitted to SIBGRAPI'99, may 1999.

[6] SILVA, F., CAVALCANTI, P., Animações em Tempo Real Utilizando Movimentos Capturados. In *Proceedings of SIBGRAPI'96, IX Brazilian Symposium of Computer Graphics and Image Processing*, pp. 333–334, 1996.

[7] SILVA, F., Movimento Capturado - Introdução à Tecnologia. Relatório Interno - Laboratório VISGRAF, IMPA, April 1997.
Available on–line at *http://www.visgraf.impa.br/Projects/mcapture/publ/mc-tech.*

[8] WITKIN, A. AND POPOVIC, Z., Motion Warping. In *Computer Graphics (SIGGRAPH'95 Proceedings)*(August 1995), pp. 105-108.

[9] UNUMA, M., ANJYO, K., TAKAEUCHI, R., Fourier Principles for Emotion-based Human Figure Animation. In *Computer Graphics (SIGGRAPH'96 Proceedings)* (August 1996), pp. 75–84.

[10] GLEICHER, M., Motion Editing with Space-Time Constraints. In *Proceedings of 1997 Symposium on Interactive 3D Graphics* (April 1997), pp. 139–148.

[11] COHEN, M., ROSE, C., GUENTER, B., BODENHEIMER, B., Efficient Generation of Motion Transitions Using Spacetime Constraints. In *Computer Graphics (SIGGRAPH'96 Proceedings)* (August 1996), pp. 147–154.

[12] IMIX, Turbo Cube / Video Cube - User's Guide, IMIX Company, 1996.

[13] Character Motion Systems. In *Computer Graphics (SIGGRAPH'94)*, Course no. 9, 1994.

[14] WATT, A. AND WATT, M., Advanced Animation and Rendering Techniques - Theory and Practice. Addison-Wesley, 1992.

[15] SILVA, F., VELHO, L. AND GOMES, J., Motion Reparametrization. In *EUROGRAPHICS'98 (short-papers proc.)*. September 1998.

[16] O'ROURKE, J., Computational Geometry in C. Cambridge University Press. 1994.

[17] CRAIG, J. J., *Introduction to Robotics - Mechanics and Control*. Addison-Wesley publishing, 1985.

[18] FASE PROJECT, THE NETHERLANDS, http://www.cwi.nl/FASE/, 1998.

[19] Biovision Inc., *http://www.biovision.com*

[20] Viewpoint Datalabs, *http://www.viewpoint.com*