

MAURICIO FERREIRA

**ANIMAÇÃO DE UM MODELO HUMANÓIDE POR
INTERPOLAÇÃO DE KEYFRAMES**

**PROJETO ORIENTADO
ORIENTADOR: WALDEMAR CELES**

DEPARTAMENTO DE INFORMÁTICA

Rio de Janeiro, 10 de novembro de 2005

Sumário

SUMÁRIO.....	2
ÍNDICE DE FIGURAS	4
1. Introdução	5
2. Animação	7
2.1. Temporização	8
2.2. Suavização do início e do fim	8
2.3. Arcos	8
2.4. Antecipação	9
2.5. Exagero	9
2.6. Esmague e estique	9
2.7. Ação secundária	10
2.8. Ação do seguimento e ultrapassagem	10
2.9. Diretamente x quadro-a-quadro	10
2.10. Enquadramento	11
2.11. Apelo	11
3. Keyframing	12
4. Interpolações	14
4.1. Interpolação linear para translações	14
4.2. Interpolação linear para rotações	16
4.2.1. LERP	16

4.2.2. SLERP	18
4.3. Resultados	20
5. Modelo humanóide	21
6. Conclusão	29
7. Referências Bibliográficas	30

Índice de Figuras

Figura 1 - Gráfico de uma animação em função do tempo	13
Figura 2.a - Interpolação de retas	15
Figura 2.b - Interpolação de curvas	15
Figura 3 - Interpolação linear em uma translação	16
Figura 4 – LERP	17
Figura 5 – SLERP	18
Figura 6 - Etapas de interpolação de translação e rotação de objetos	20
Figura 7 - Modelo de articulações	22
Figura 8 - Modelo de partes rígidas	24
Figura 9 - Hierarquia de movimento	25
Figura 10 - Movimento em função do tempo	26
Figura 11 - Animação de um modelo humanóide	27

1. Introdução

O conceito de Animação é antigo, tendo sido pela primeira vez utilizado pelo cartunista Winsor McKey, em 1911. Winsor conseguiu obter o movimento de um personagem desenhando-o em diversos pedaços de papel, que eram visualizados segundo uma certa taxa de amostragem.

Walt Disney popularizou a arte da animação através de seus filmes animados que até hoje surpreendem pela qualidade técnica. Para produzir o desenho animado "Branca de Neve e os Sete Anões", os estúdios Walt Disney precisaram desenhar cerca 250.000 quadros, o que tornava impraticáveis produções como Winsor fazia.

Com a evolução do computador, foi possível fazer as animações de maneira mais rápida e eficiente. Foi nos estúdios Disney que surgiram técnicas como o *keyframing*, que automatizou a tarefa de gerar quadros através de interpolações. O computador então passou a ser uma nova ferramenta para automatizar tarefas longas e tediosas para a realização de uma animação otimizando além de tempo, custo e confiabilidade.

Além do *keyframing*, os animadores profissionais têm à disposição todo um conjunto de técnicas de animação como simulação e cinemática. No entanto, uma grande dificuldade que os animadores enfrentam até hoje é a reprodução dos movimentos humanos, pois são extremamente complexos para serem representados, devido ao seu alto grau de complexidade como o caminhar humano, devido à enorme quantidade de articulações existentes.

Neste trabalho, daremos ênfase a movimentos humanóides baseados nessa técnica de animação por *keyframes*. Serão apresentados as técnicas de animações estudadas e os seus princípios básicos, álgebra de interpolações e quatérnios, hierarquia de movimento, a técnica de *keyframing* e os resultados obtidos.

2. Animação

Animação é o processo de gerar um filme baseado em quadros que são criados individualmente. Esses quadros podem ser gerados por fotografia de imagens desenhadas ou por computação gráfica. Para gerar o filme final, a seqüência de quadros criados deve ser exibidos numa taxa de 16 ou mais quadros por segundo. Isso gera a ilusão de movimento contínuo por causa da persistência de visão.

Portanto gerar uma animação é um trabalho intensivo e entediante, consumindo bastante tempo para ser produzido. A maioria das animações para a TV e o cinema vem de estúdios de animação profissional. Contudo, o campo de animação independente existe pelo menos desde os anos 50, com a animação sendo produzida algumas vezes por uma única pessoa.

Para criar uma animação existem princípios fundamentais que foram formalizados pela Walt Disney Productions. Esses princípios fazem parte de um conjunto de informações que um animador deve possuir, independente da técnica adotada para a síntese de movimento de personagens.

Estes princípios que determinam as regras da animação estão descritas abaixo:

2.1. Temporização (*Timing*):

A essência de uma animação é a sua relação de tempo. É a temporização quem determina a velocidade a qual algo se movimenta, o que é de extrema importância, pois essa velocidade que determina se um personagem, por exemplo, está cansado ou agitado. Portanto, essas relações de tempo que fornecem a veracidade à animação.

2.2. Suavização do início e do fim (*Slow-in & Slow-out*)

Esse princípio rege o espaçamento dos entremeios (*inbetweens*) entre os quadros-chave que aceleram e desaceleram o movimento.

2.3. Arcos (*Arcs*)

No mundo real quase todas as trajetórias não são retilíneas; os movimentos acontecem em arcos descritos no espaço. Portanto, o percurso realizado por um objeto de um quadro-chave ao outro deve sempre ser uma curva.

2.4. Antecipação (*Anticipation*)

Antecipação é a preparação para um movimento, ou seja, um objeto que se pretenda animar deve primeiramente recuar na direção oposta do movimento desejado como uma preparação para o movimento futuro. Sem esta preparação, os movimentos parecem irrealis, duros e mecânicos.

2.5. Exagero (*Exaggeration*)

O exagero é utilizado para acentuar uma ação ou gesto, tornando movimento qualquer mais claro e prontamente identificável. Deve ser usado de maneira cuidadosa e equilibrada para isso.

2.6. Esmague e estique (*Squash & Strech*)

A rigidez e a massa de um objeto pode ser definida pela aplicação de distorções na forma durante uma ação. Um exemplo seria na contração de músculos, ao serem contraídos deverão ser esmagados e quando estendidos esticados.

2.7. Ação secundária (*Secondary Action*)

A ação secundária faz com que a animação fique mais interessante e ao mesmo tempo cria um aspecto mais realista, ou seja, se um personagem caminha numa animação, ele movimenta não só as pernas, mas a cabeça e os braços.

2.8. Ação do seguimento e ultrapassagem (*Follow through & Overlapping Action*)

O *follow through* é o termo empregado para denominar o efeito observado nas extremidades dos objetos flexíveis, que continuam a se movimentar com a frenagem do corpo. O *overlapping action* é o acréscimo de complexidade no *timing* proporcionando naturalidade à ação ao acrescentar variações no timing e na velocidade dos apêndices ou partes flácidas de todo o corpo do personagem.

2.9. Diretamente x quadro-a-quadro (*Straight Ahead Action & Pose-to-Pose Action*)

Esse termo é aplicado a duas aproximações contrastantes na criação do movimento. A cena pode ser construída e editada de maneira fluída, sob a forma de um passeio da câmera, ou em tomadas quadro-a-quadro, quebrando a continuidade de uma seqüência, mas enfatizando algum elemento em especial.

2.10. Enquadramento (*Staging*)

Apresentação de uma idéia de forma clara, facilitando a compreensão do movimento.

2.11. Apelo (*Appeal*)

Apresentação do personagem: capacidade de comunicação de conceitos sobre a individualidade do personagem.

3. Keyframing

Na animação tradicional, o animador desenha cada quadro da animação, baseado numa certa modificação no tempo. Esses quadros são passados para outro desenhista, que então desenhara os quadros intermediários (chamados *inbetweens*) pra cada par de quadros originais de tal forma que seja percebido uma movimentação natural do movimento de objetos, metamorfose, variação de cor e luz, etc.

Entretanto esse processo é extremamente lento e requer muita paciência e habilidade do animador. Para ilustrar uma animação de 5 minutos precisaria de $5(\text{minutos}) \times 60(\text{segundos/minuto}) \times 24(\text{quadros/segundo}) = 7200$ quadros, o que é um número bastante elevado se considerarmos o trabalho que cada quadro proporciona.

O *keyframing* surgiu para automatizar esse *intbetween*, pois ele funciona a partir de interpolação de quadros-chave da animação. O *keyframing* é a transposição da técnica de *intbetween* para os computadores sendo bem menos trabalhosa, permitindo um bom controle da animação.

Resumidamente a técnica de *keyframing* é criar quadros-chaves que consigam representar a animação por completo ao criar os quadros de transição de um *keyframe* a outro através de interpolações.

A Figura 1 abaixo, retirada de [11], exibe uma simples animação do um salto de uma luminária utilizando cinco *keyframes* em relação a sua posição relativa ao tempo da animação.

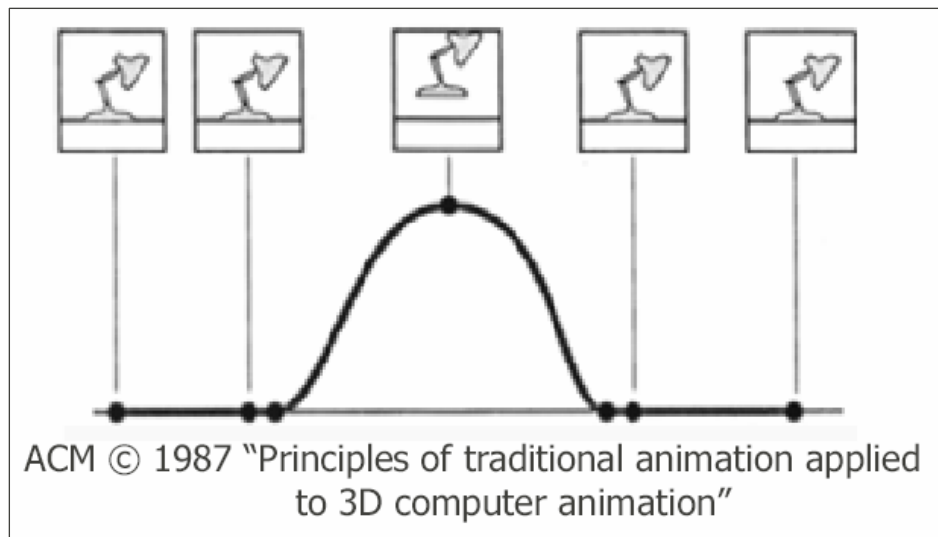


Figura 1

Cada ponto que compõe a curva da animação, na Figura 1, representa um estado do movimento do objeto numa transição de um *keyframe* para outro. Para escrever essa curva é necessário utilizar uma álgebra de interpolações descritas na seção seguinte.

4. Interpolações

A técnica de interpolação pode ser dividida em dois padrões de curvas. O primeiro padrão, o de interpolação linear (Figura 2.a), corresponde a uma velocidade constante entre quadros chave. O segundo padrão, o de interpolação de curvas (Figura 2.b), proporciona diversos parâmetros que podem alterar a velocidade de interpolação de quadros entre quadros-chave.

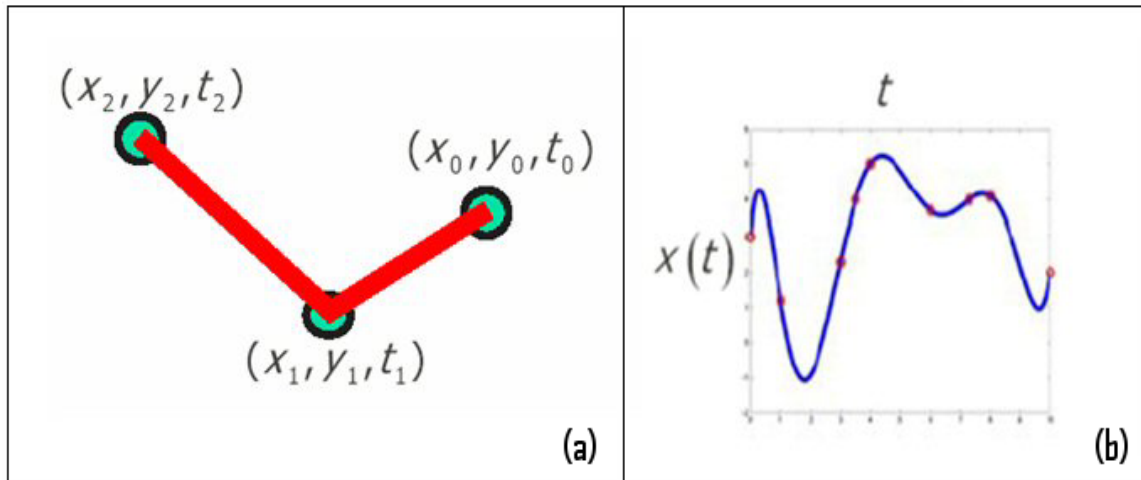


Figura 2

4.1. Interpolação linear para translações

A interpolação linear é a maneira mais simples de calcular o *inbetweening*, pois dado dois *keyframes* conhecidos e supondo que a taxa de variação de um frame para outro é constante, a interpolação linear cria novos valores de distâncias iguais em uma reta entre esses frames. Esses intervalos são determinados pelo usuário, onde é ele quem decide o tempo que dois keyframes estão separados. Se, por exemplo, os frames

estiverem separados por um segundo e deseja se obter uma taxa de exibição da animação de 10 quadros por segundo, então, o artista deve representar 10 quadros para cada intervalo de *keyframes*.

Nesse trabalho foi utilizada a interpolação linear para a translação do movimento vertical do modelo humano andando no plano, ou seja, a sua altura relativa ao chão varia dependendo se a sua perna está totalmente esticada ou não. A figura 3 ilustra uma interpolação linear de translação do cubo.

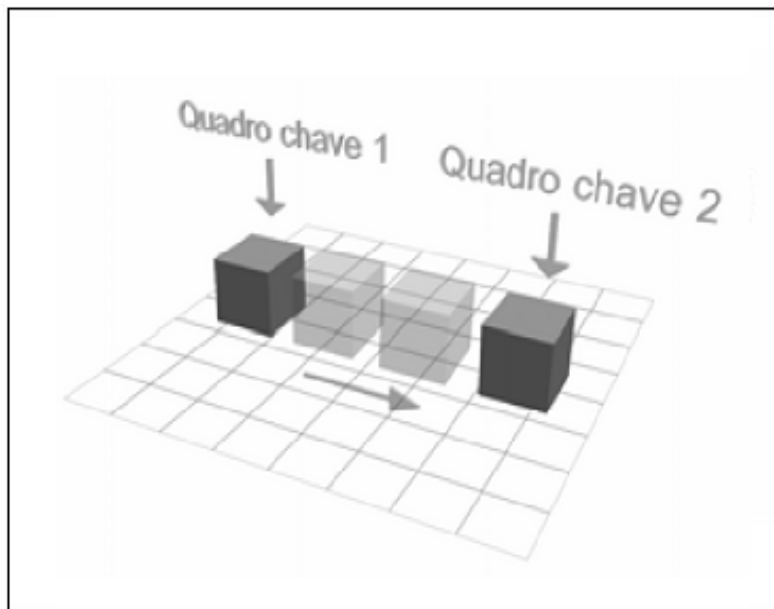


Figura 3

Um pseudo código para a interpolação linear na translação seria:

```
static int subFrameActual = 0; //Numero do frame interpolado  
Vector interpolTranslation ( Vector firstPos, Vector secondPos )  
{  
    float distanceFrame1, distanceFrame2;  
    Vector resultPos;
```

```

int frameDivision = 10; //Numero de subframes entre 2 frames

distanceFrame2 = (( 100.0f * subFrameActual ) / frameDivision) /100.0f;
distanceFrame1 = fabs(distanceFrame2 - 1.0f);

resultPos.x = (firstPos.x * distanceFrame1) + (secondPos.x * distanceFrame2);
resultPos.y = (firstPos.y * distanceFrame1) + (secondPos.y * distanceFrame2);
resultPos.z = (firstPos.z * distanceFrame1) + (secondPos.z * distanceFrame2);
resultPos.w = 1;

return resultPos;
}

```

4.2. Interpolação linear para rotações

Para interpolar as rotações de objetos foram utilizadas interpolações de quatérnios. Resumidamente, quatérnio é uma representação algébrica simples e elegante para representar as rotações no espaço, criado por Hamilton (1843), através da extensão dos números complexos. Depois, Shoemake (1985) introduziu os quatérnios para a Computação Gráfica.

Duas possíveis interpolações de quatérnios são: a interpolação linear (LERP) e a interpolação linear esférica (SLERP).

4.2.1. LERP

O LERP (do inglês, *Linear IntERPolation*) é um método de interpolação através de uma linha reta (secante) através de duas posições (quatérnios). A Figura 4.a abaixo

representa dois *keyframes* representados pelos quatérnios q_0 e q_1 , a Figura 4.b representa o resultado do LERP pela linha vermelha.

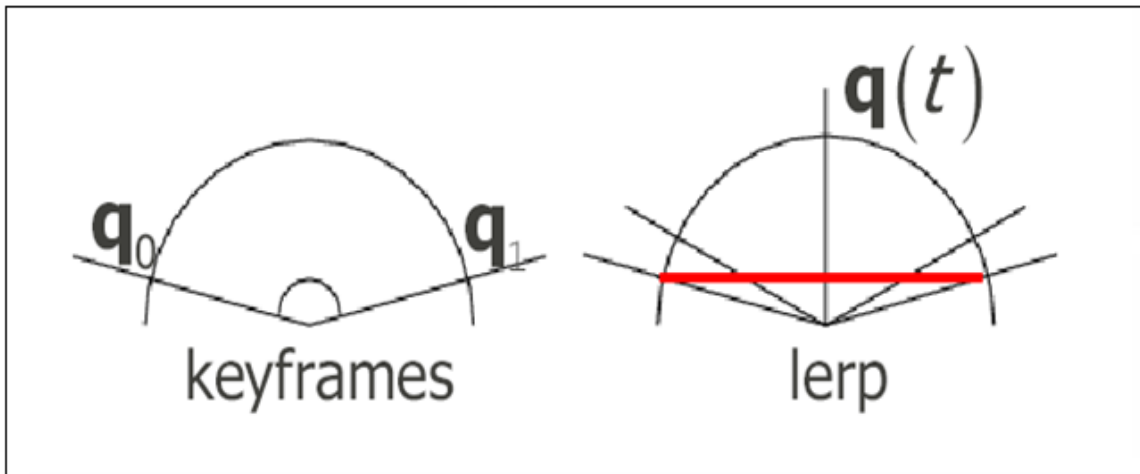


Figura 4

A representação matemática do LERP é:

$$\text{lerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}(t) = \mathbf{q}_0 (1 - t) + \mathbf{q}_1 t$$

Onde q_0 é o quatérnio inicial, q_1 é o quatérnio final, t é a “distância” em tempo do q_0 para o q_1 .

O resultado de uma animação através da interpolação LERP, é algo não muito agradável pois em algumas seções tem variações de velocidade. Portanto para o caso da animação do boneco esse tipo de interpolação não pode ser usado. Para resolver o problema do LERP, foi utilizado nesse trabalho o SLERP (do inglês, *Spherical Linear IntERPolation*).

4.2.2. SLERP

O SLERP ao invés de fazer interpolação linear simples, a curva segue ao longo de um grande arco na esfera unitária do quatérnio. A Figura abaixo mostra o resultado do SLERP entre dois *keyframes*:

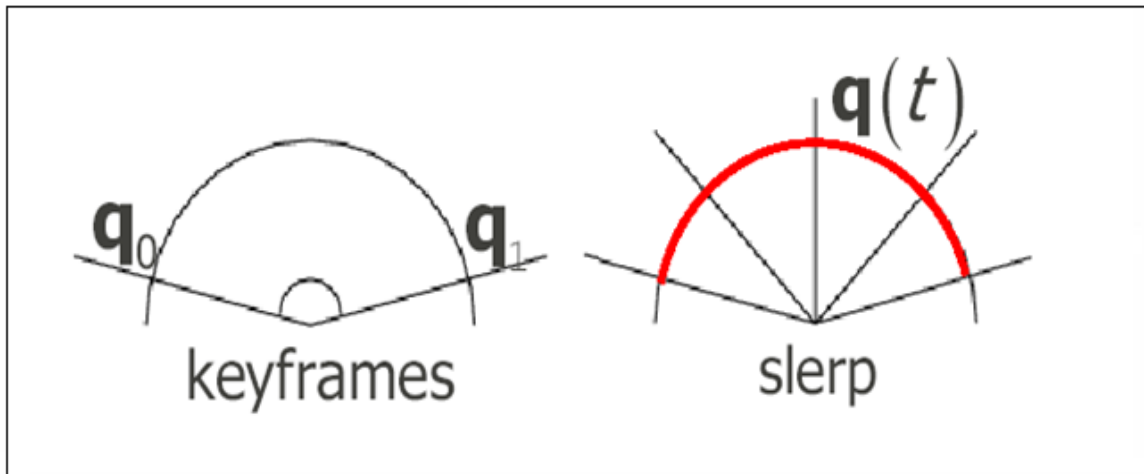


Figura 5

A representação matemática do SLERP é:

$$\text{slerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}(t) = \frac{\mathbf{q}_0 \sin((1-t)\omega) + \mathbf{q}_1 \sin(t\omega)}{\sin(\omega)}$$

Onde q0 é o quatérnio inicial, q1 é o quatérnio final, t é a “distância” em tempo do q0 para o q1 e ω é dado por:

$$\omega = \cos^{-1}(\mathbf{q}_0 \cdot \mathbf{q}_1)$$

Um pseudo código para o caso da interpolação linear na rotação seria:

```
static int subFrameActual = 0; //Numero do frame interpolado

Quat interpolRotation ( Quat q1, Quat q2)
{
    float distanceFrame1, resultAngle;
    Quat resultQuaternion;
    Vector resultAxis;

    int frameDivision = 10; //Numero de subframes entre 2 frames

    //Pegando a distancia do Frame1
    distanceFrame1 = (( 100.0f * subFrameActual ) / frameDivision) /100.0f;

    //Fazendo interpolação dos Quatérnios
    resultQuaternion = algQuatSLERP(q1, q2, distanceFrame1);

    resultAngle = (float)algQuatRotAngle(resultQuaternion);
    resultAxis = algQuatRotAxis(resultQuaternion);

    return resultQuaternion;
}
```

4.3. Resultados

Nesse trabalho, foram implementados testes básicos das interpolações. No exemplo abaixo foram criados dois objetos simples com dois *keyframes*, um inicial e outro final. A esfera sofreu somente translação da posição inicial à final, portanto foi utilizada a interpolação linear de translação. E o paralelepípedo está apenas rotacionando em torno de um eixo, portanto foi utilizado o SLERP.

Na Figura 6 abaixo ilustra um exemplo desses testes básicos. A Figura 6.a é o *keyframe* inicial, a figura 6.d é o *keyframe* final. As Figuras 6.b e 6.c são resultados intermediários entre o *keyframe* inicial e final.

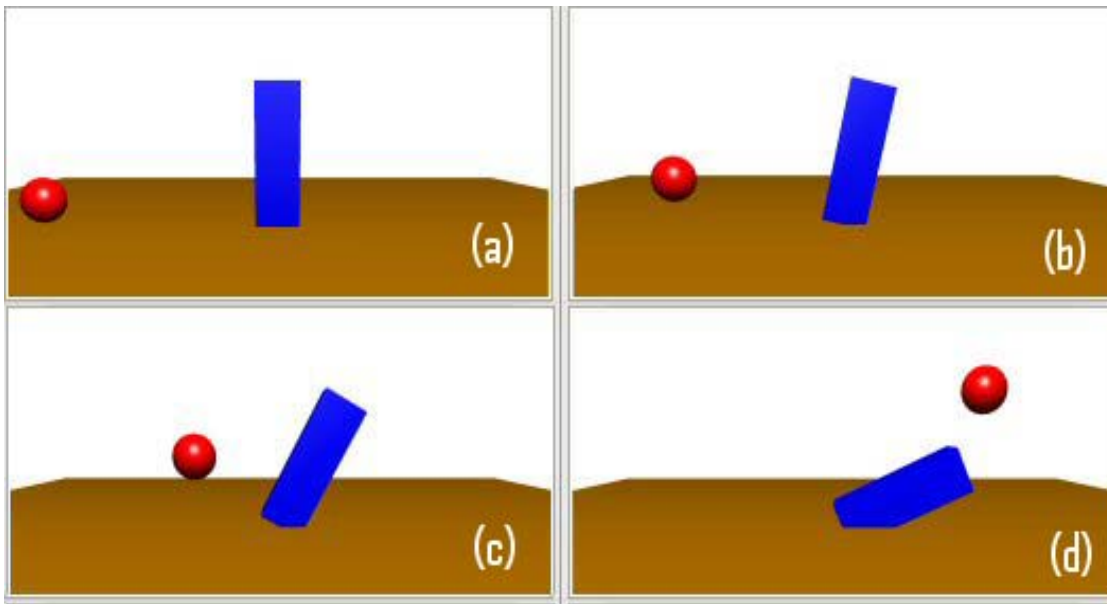


Figura 6

5. Modelo humanóide

O modelo proposto para o boneco humanóide é um modelo articulado que foi discretizado e classificado em juntas e partes rígidas. As juntas são os pontos de articulação do modelo. São esses pontos que compõem os graus de liberdade de movimento do boneco. As partes rígidas são os “ossos” do modelo, são elas que representam o boneco em si.

As juntas definidas foram:

- Joelho Esquerdo (*Left Knee*)
- Joelho Direito (*Right Knee*)
- Ombro Esquerdo (*Left Shoulder*)
- Ombro Direito (*Right Shoulder*)
- Tornozelo Esquerdo (*Left Ankle*)
- Tornozelo Direito (*Right Ankle*)
- Quadril, Lado Esquerdo (*Left Hip*)
- Quadril, Lado Direito (*Right Hip*)
- Scapula (Scapula)
- Cotovelo Esquerdo (Left Elbow)
- Cotovelo Direito (Right Elbow)
- Pelvis (Pelvis)

A figura ao lado exhibe as 12 articulações do modelo proposto.

MODELO DE ARTICULAÇÕES

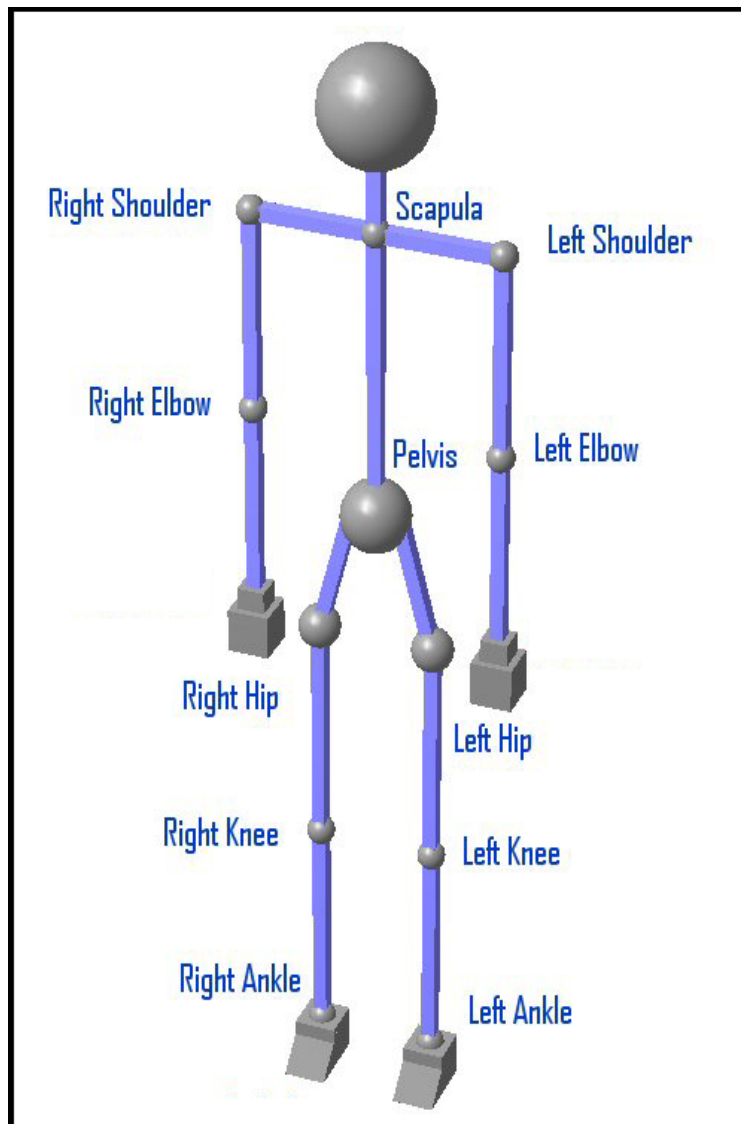


Figura 7

As partes rígidas definidos foram:

- Cabeça (*Head*)
- Coluna Servical (*Servical Collumn*)
- Humerus Esquerdo (*Left Humerus*)
- Humerus Direito (*Right Humerus*)
- Radio Esquerdo (*Left Radio*)
- Radio Direito (*Right Radio*)
- Mão Esquerda (*Left Hand*)
- Mão Direita (*Right Hand*)
- Fêmur Esquerdo (*Left Femur*)
- Fêmur Direito (*Right Femur*)
- Fíbula Esquerda (*Left Fibula*)
- Fíbula Direita (*Right Fibula*)
- Pé Esquerdo (*Left Feet*)
- Pé Direito (*Right Feet*)

A figura ao lado exibe as 14 partes rígidas do modelo proposto.

MODELO DE PARTES RÍGIDAS

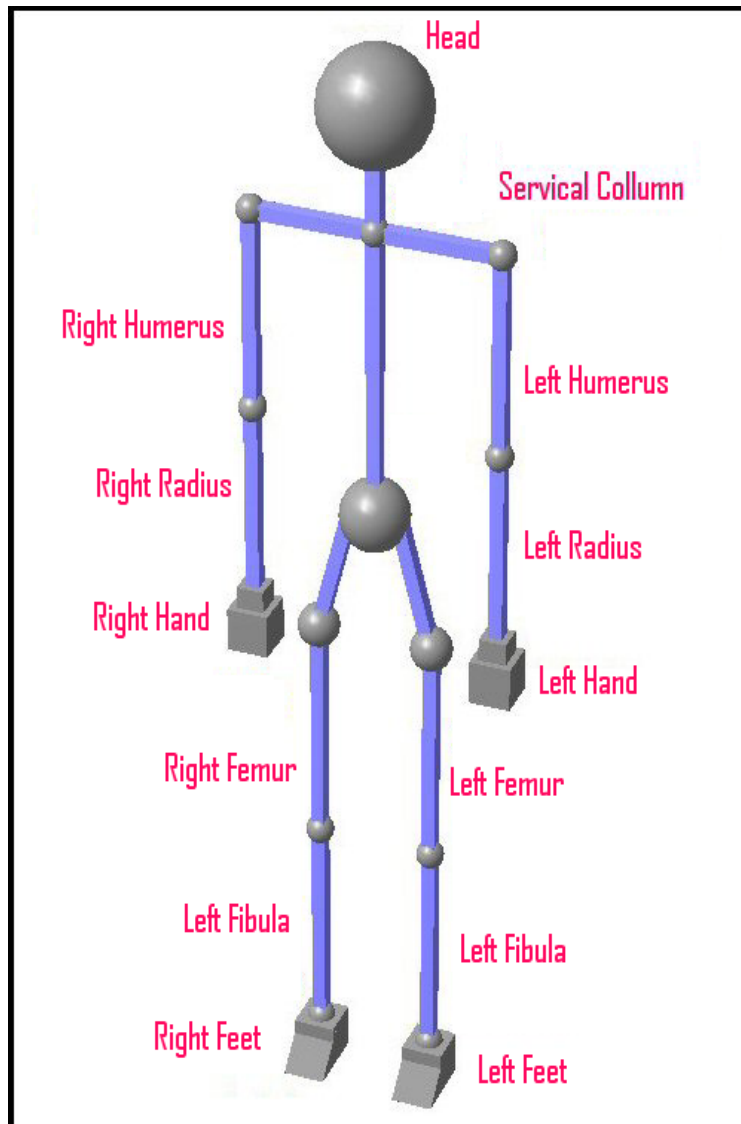


Figura 8

O boneco foi modelado no OpenGL usando apenas chamadas da glut sem qualquer ajuda de malhas prontas. Ele é composto, obviamente, de uma hierarquia dessas articulações e sólidos onde, por exemplo, ao levantar um braço, a mão acompanhará essa transformação de posição. Essa hierarquia segue na árvore representada na Figura 9 abaixo:

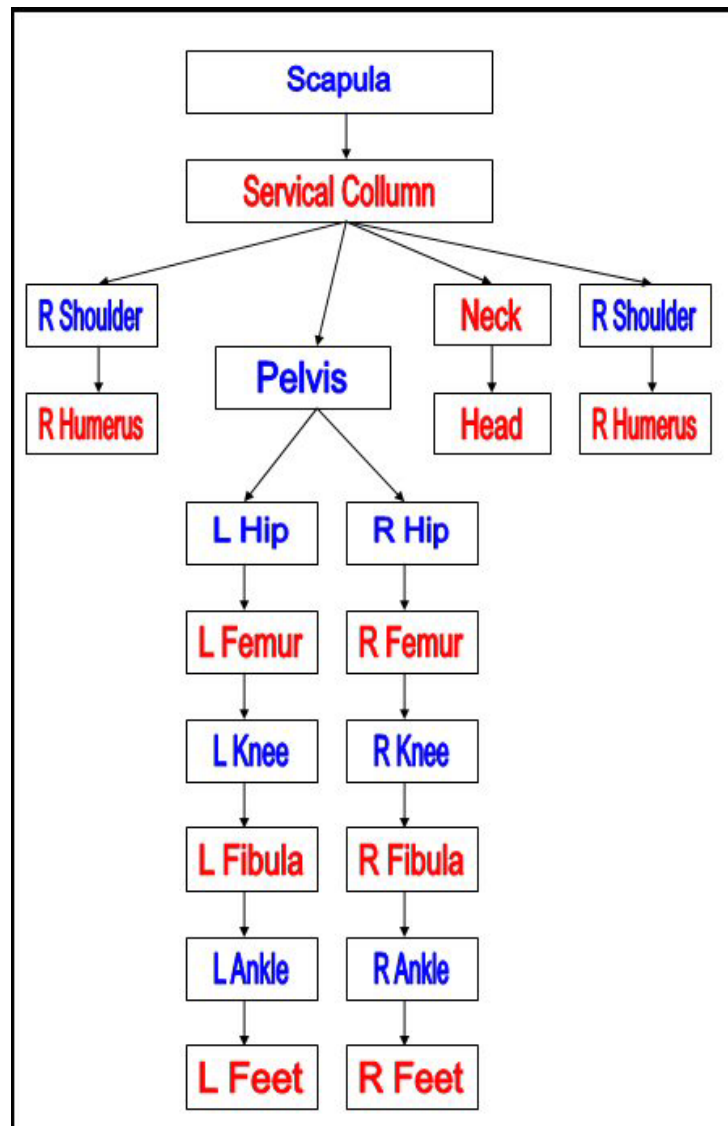


Figura 9

Um modelo, como esse, bastante discretizado, dá ao animador bastante liberdade de representar qualquer movimento para o boneco.

Um movimento é criado, a grosso modo, especificando ângulos para juntas do modelo como função do tempo (figura 10).

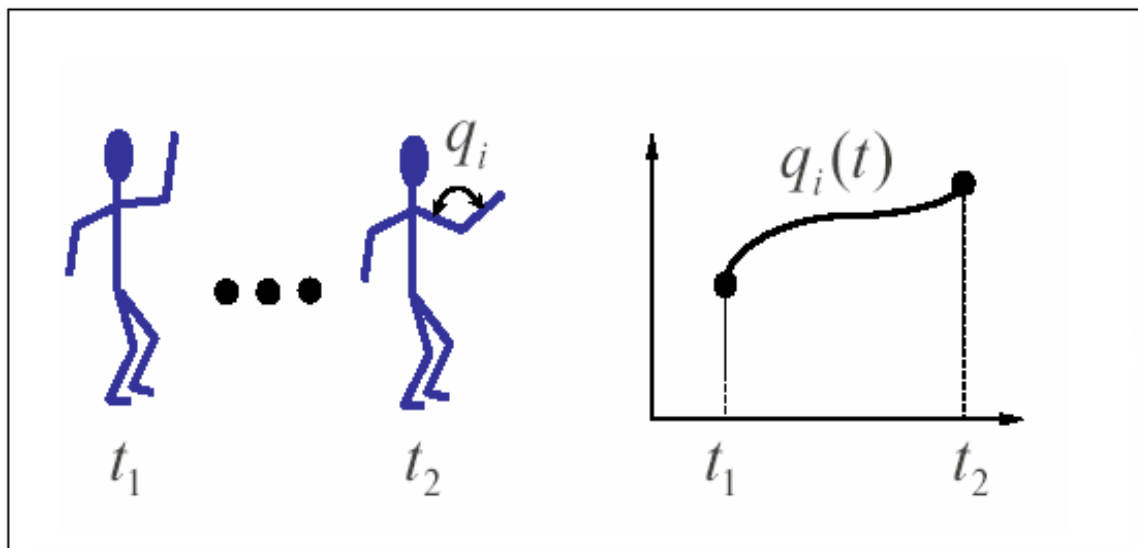


Figura 10

Nesse trabalho, o único movimento modelado foi o caminhar. Essa tarefa de modelagem de movimento foi bastante trabalhosa e que requereu uma certa habilidade artística, o que confirma tudo o que foi dito anteriormente sobre a dificuldade que se tem se gerar os keyframes.

Foi desenvolvido um editor para fazer a modelagem do movimento. Esse editor funciona de uma maneira bem simples: nele é possível rotacionar e transladar qualquer “pedaço” do boneco, e para cada frame desejado era possível salvar os ângulos e posições de cada parte do modelo. Com essa técnica foi possível modelar o movimento até que ele o parece “suave”.

No total foram gerados 24 frames que compõem a animação completa. Entre esses 24 frames não existe nenhuma interpolação, pois através deles é possível representar a animação completa com efeito “suave” da movimentação do boneco.

A próxima etapa foi escolher alguns quadros desses 24 e utilizá-los como *keyframes*. Uma solução encontrada que deu bons resultados foi utilizar 8 dos 24 quadros. Com esses 8 keyframes foram criados 10 frames para cara par de keyframes, gerando um total de 80 frames utilizando as técnicas de interpolação. O resultado foi bastante interessante, pois a animação ficou bem mais realista do que o resultado de 24 quadros.

A figura abaixo ilustra os 8 keyframes da animação:

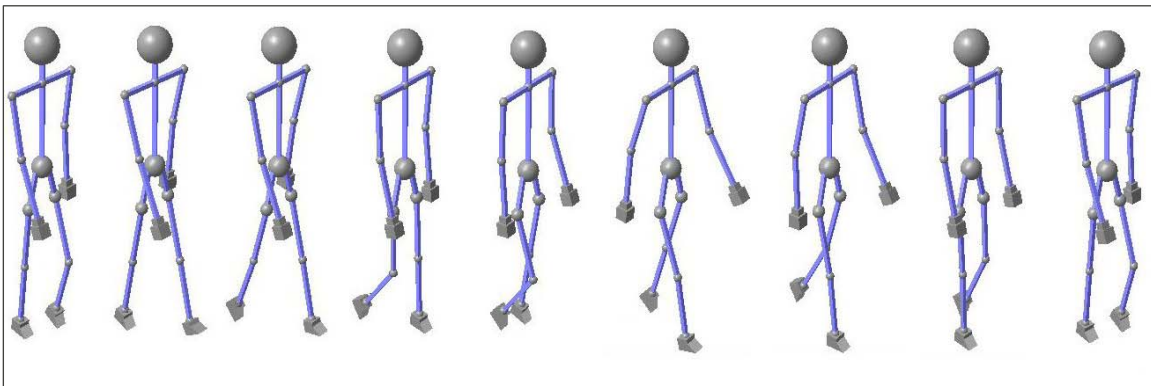


Figura 11

Uma característica interessante do modelo é que, por ele estar na forma de esqueleto, é possível associar aos vértices das juntas e das partes rígidas do modelo a uma malha de triângulos para compor uma “pele” de qualquer tipo de modelo humanóide.

Além disso, criando outras seqüências de movimentos, é possível realizar praticamente todos os movimentos básicos de um modelo humanóide como: saltar, agachar, deitar, etc. Com isso seria possível controlar o boneco semelhante como os jogos fazem, dando total controle ao usuário com a animação.

6. Conclusão

A maior dificuldade de gerar animações como essa ainda é gerar os modelos e gerar os keyframes, pois requer habilidades artísticas sendo um processo ainda um pouco lento.

O modelo proposto é bastante adaptativo podendo ser adaptado para diversos movimentos e facilmente expansível, já que cada articulação e parte rígida são independentes entre si.

O resultado da animação gerada por keyframes de um movimento humanóide nesse trabalho foi satisfatório, pois se mostrou bastante robusta e com a mudança de quadros bastante suave.

7. Referências Bibliográficas

- [1] • N. M. Thalmann. “Computer animation: Theory and Practice”. Springer, 1990
- [2] • M. Giambruno. “3D Graphics & Animation”. New Riders, 2002
- [3] • Watt, M. Watt. “Advanced Animation and Rendering Techniques”. Addison-Wesley
- [4] • GreenHouse - "Using Linear Interpolation"
- [5] • W. Dutra da Silveira. “Técnicas de animação em ambientes tridimensionais”.
- [6] • F.Faure, G.Debunne. “Dynamic Analysis of Human Walking”
- [7] • Leonard McMillan and Jovan Popovic. “Animation”
- [8] • M. Gattass. “Transformações Geométricas”
- [9] • P. Zhao. “Human Walking Control”
- [10] • J. D. Foley, A. van Dam, Steven K. Feiner, John F. Hughes, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Company, 1990
- [11] • J. Lasseter, “Principles of Traditional Animation Applied to 3D Computer Animation”, *Computer Graphics*, Vol. 21, no. 4, July 1987, pp. 35-44.