

ANIMADO: Um Protótipo de um Sistema de Animação Modelada por Dinâmica

M. ANDRÉIA F. RODRIGUES¹
LÉO P. MAGALHÃES²

Grupo de Computação de Imagens
Departamento de Engenharia de Computação e Automação Industrial
Faculdade de Engenharia Elétrica
C.P. 6101, Unicamp
13081 - Campinas - SP

¹e-mail : andreia@dca.fee.unicamp.br

²e-mail : leopini@dca.fee.unicamp.br

Abstract. The aim of this work is to present the ANIMADO, a first example of using Dynamic Simulation in the Computer Animation field, a research branch area of the group ProSim, in the DCA at UNICAMP. Generally speaking, the ANIMADO is divided into five totally independent, though strongly inter-related modules: the EGESP, a module that solves Systems of Linear Equations by standard Gauss-Jordan elimination with maximal pivoting, the FRED, a module that contains a 4th variant Runge-Kutta Numerical Integration Method, the COLISÃO, a module to Detect and Response collisions, the CONTROLE, a module to control the animation, and the INTEREXPSERV, an interpreter of expressions based in the client-server model using sockets.

1 Introdução

Nos últimos anos, a animação modelada por computador vem se tornando um campo de pesquisa fascinante. "Aqui, o computador é mais do que um suporte, atuando na criação de um mundo 3D" [TT85]. Uma abordagem típica e simples para o processo de animação por computador consiste na especificação da posição e orientação de uma coordenada geométrica 3D do objeto, como uma função do tempo, sem levar em consideração as **forças** ou **torques** que causaram o seu movimento. Ou seja, os objetos participantes de uma cena são considerados entidades geométricas desprovidas de propriedades físicas. Isto faz com que o animador tenha que usar sua intuição sobre o mundo físico existente, objetivando um maior realismo no movimento dos objetos.

Infelizmente, esta técnica é pouco satisfatória, visto que, movimentos realistas geralmente são complexos de serem representados. Este modelo de animação é conhecido no ambiente de Computação de Imagens como Animação Modelada por Cinemática [HOU92] e [ROD92].

Tentando-se amenizar o problema de "intuir" o comportamento físico de um corpo, o animador necessita considerar as entidades participantes da cena de animação como objetos do mundo real, tendo **massa**, **momento de inércia**, **elasticidade**, **coeficiente**

de fricção, etc, características estas, advindas da **Dinâmica**.

Alguns sistemas de simulação [HAH88] são capazes de modelar, realisticamente, a Dinâmica de uma classe geral de **corpos rígidos**. O estado dinâmico de um corpo rígido é completamente descrito por sua massa, inércia rotacional, posição, orientação, momento linear e momento angular.

Neste contexto, foi criado o ANIMADO (ANIMAção por Dinâmica de Objetos) [ROD93] que introduzirá o uso da Simulação Dinâmica no âmbito da Animação de Objetos sintetizados por computador no projeto ProSim¹, como ilustrado na figura 1, cujo propósito é o de tornar-se um sistema de Computação de Imagens de domínio público.

O ANIMADO contém uma biblioteca de ferramentas independentes, suficientes para a composição de Animações Modeladas por Dinâmica. O animador produzirá os movimentos desejados selecionando **forças** e **torques** de controle que estarão diretamente relacionados a leis de Mecânica de Movimento. Facilmente extensível e modular, foi idealizado com a intenção de servir como apoio para outras abordagens, tais como a Dinâmica Inversa e a Cinemática Inversa, visto que possui algumas ferramentas para Cálculo

¹Prototipação e Síntese de IMagens foto-realistas e animação

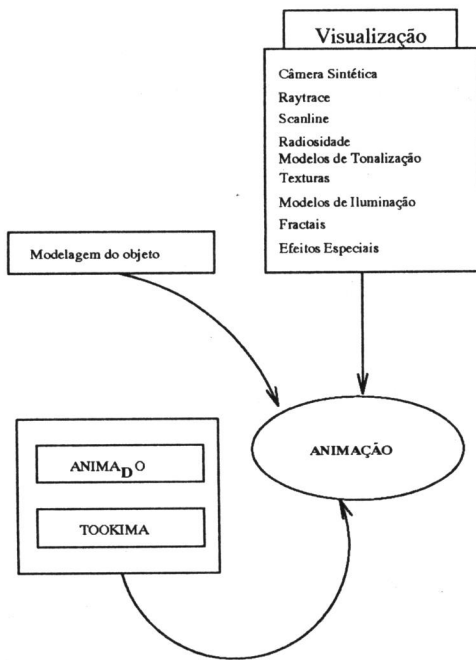


Figura 1: O Projeto ProSim.

Numérico bastante interessantes.

Espera-se com o uso do ANIMA_DO diminuir as limitações encontradas na especificação cinemática, em especial, a nível de realismo do movimento.

2 A formulação utilizada no ANIMA_DO: o Método de Euler

Em geral, o movimento de um corpo rígido pode ser decomposto em um movimento linear de um ponto localizado no centro de massa de um corpo (devido à uma força externa) mais um movimento rotacional sobre o centro de massa do corpo (devido a um torque externo). Assim, o Método de Euler é composto por seis equações: três equações para a translação que relacionam as **acelerações lineares** e **massas** às **forças** e outras três equações para a rotação que relacionam as **acelerações angulares** e a **distribuição de massa** ao torque.

É importante salientar que a **força** e a **aceleração** devem ser descritas em relação ao **mesmo** SC². A força ${}^A f$ relativa ao SC {A}³ é responsável pela aceleração ${}^B a$ em termos do SC {B}⁴. A matriz direção dos co-senos pode ser usada para encontrar as componentes da força no SC {A} relativas ao SC {B} [CRA89].

A seguir, será apresentada uma forma mais geral de representar estas equações. **Todos** os valores são

²Sistema de Coordenadas

³Sistema de Coordenadas Mundo

⁴Sistema de Coordenadas Local

descritos em relação a um único SC (que pode ser o {A}) e que, no nosso caso refere-se à **posição** e à **orientação** instantânea de um SC {B} fixo no corpo.

A forma vetorial das duas equações que descrevem o movimento de um único corpo rígido é [WIL90]:

$$\begin{aligned} f &= ma - mc \otimes \dot{w} + mw \otimes (w \otimes c) \\ \tau &= J\dot{w} + mc \otimes a + w \otimes Jw \end{aligned} \quad (1)$$

onde

f = força total aplicada no corpo,

τ = torque total atuando no corpo,

m = massa do corpo,

J = matriz do tensor inercial,

c = vetor 3D do centro de massa,

w = vetor 3D da velocidade angular,

a = aceleração linear,

\dot{w} = aceleração angular.

Como estes termos foram descritos relativamente a um SC {B} fixo no corpo, os valores do **centro de massa** e da **matriz do tensor inercial** não se alteram durante a animação, podendo ser fornecidos inicialmente pelo usuário ou calculados pelo sistema.

3 Etapas da animação utilizando o ANIMA_DO

O ANIMA_DO tem como filosofia, a flexibilidade e modularidade de ferramentas que podem ser agrupadas e/ou acrescentadas em bibliotecas, permitindo uma boa estruturação do sistema. O ANIMA_DO é composto por cinco módulos independentes, como ilustra a figura 2, com funções específicas, todas elas relacionadas ao processo de Animação Modelada por Dinâmica. Nesta figura, pode-se observar como o ANIMA_DO está organizado internamente, como os módulos se comunicam entre si (através do envio de mensagens e sinais de controle), e onde se enquadraria um módulo que permite uma visualização realista da animação, por exemplo, o TOOKIMA [HOU92]. Estes módulos serão detalhados a seguir, incluindo a fase de inicialização do sistema que diz respeito ao processo de modelagem da Dinâmica, utilizando a formulação de Euler, como detalhado no item 2. Sendo assim, no ANIMA_DO, a Animação Modelada por Dinâmica será produzida de acordo com os seguintes estágios distintos [SEL89]:

- Derivação das equações de movimento,
- Resolução das equações de movimento,
- Integração das equações de movimento,
- Detecção de Colisões e conseqüente estudo da Dinâmica de Impacto,

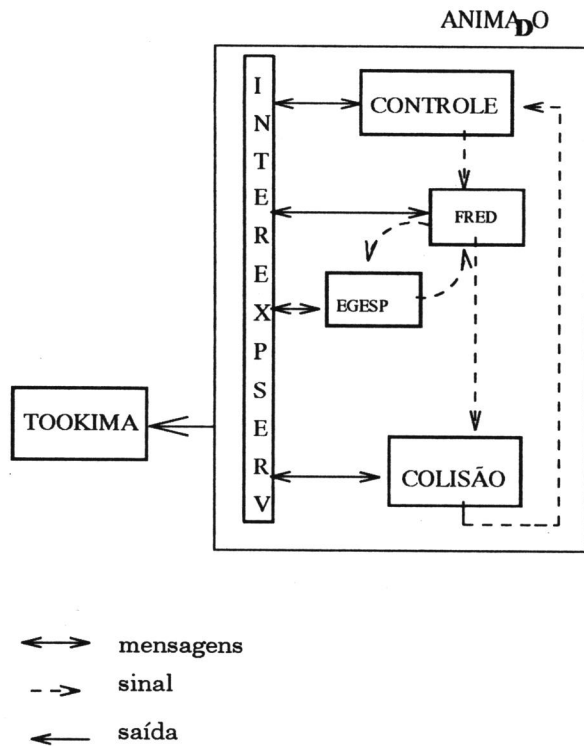


Figura 2: Um protótipo de um Sistema de Animação Modelada por Dinâmica.

- Controle das equações de movimento,
- Reprodução da Cinemática.

3.1 A fase de inicialização

Considerando os corpos rígidos 3D como massas estendidas e assumindo que estas massas não mudam de forma ou massa durante a animação, as informações necessárias para a fase de inicialização do ANIMADO serão:

- **Informações Invariáveis**
 1. m = massa do objeto
 2. c = centro de massa do objeto
 3. I = distribuição de massa ao redor do centro de massa

- **Informações Variáveis**

Os corpos rígidos têm seis graus de liberdade (três de translação, três de rotação). Assumindo um SC local fixo no objeto, os graus de liberdade translacionais podem representar um deslocamento relativo a um SC fixo e inercial do espaço mundo, ou ao longo do SC local atual (ou qualquer outro eixo). Similarmente, os graus de

liberdade de rotação descrevem a orientação do objeto em relação a algum SC referencial.

A ordem das rotações serão fixadas como:

1. rotação em x
2. rotação em y
3. rotação em z ,

conhecidas como **rotações de Euler** [CRA89]. A ordem das rotações de Euler podem variar. Outras informações variantes são as forças f e os torques τ que causam o movimento.

3.2 Resolução das equações de movimento: o EGESP

Utilizando a formulação matricial da equação 1, a resolução deste sistema de equações permite encontrar as acelerações (linear e angular), dadas as forças e os torques (assumindo que a posição e velocidade iniciais são conhecidas). Portanto, no máximo, o sistema possuirá seis equações e seis incógnitas ($a_x, a_y, a_z, \dot{w}_x, \dot{w}_y, \dot{w}_z$). Para resolver o sistema de equações, encontrando as acelerações, será utilizado o EGESP⁵, com pivoteamento parcial (parâmetro T) [HOR75].

3.3 Integração das equações de movimento: o FRED

De posse dos valores obtidos para as acelerações do(s) objeto(s) pelo EGESP, a próxima etapa do processo será encontrar as velocidades (linear e angular) utilizando-se o FRED⁶ como Método de Integração Numérica para Resolução de edo⁷ com m variáveis independentes. O FRED encontrará os valores destas velocidades e será novamente utilizado para encontrar os valores das posições e orientações ocupadas pelo(s) objeto(s) durante o decorrer da animação.

3.4 Detecção de Colisão e a Dinâmica de Impacto: o COLISÃO

Durante uma Animação por Computador, existe a possibilidade de dois ou mais objetos se chocarem. No caso da Animação Modelada por Dinâmica, a interpenetração de dois ou mais objetos não é um efeito desejável porque não representa uma realidade física, causando um efeito visual incorreto.

É comum a existência de sistemas de animação que não contém módulos para a Detecção de Colisão, forçando o animador a forjar visualmente um ponto

⁵método numérico de Eliminação de Gauss para resolução de sistemas lineares ESParsos

⁶Ferramenta para Resolução de Equações Diferenciais

⁷equações diferenciais ordinárias

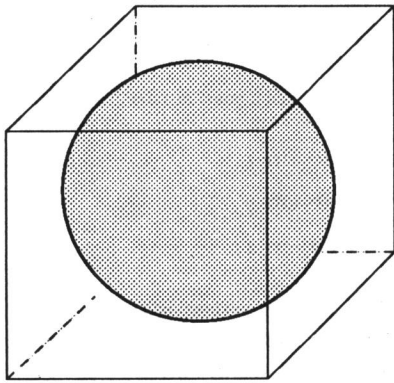


Figura 3: Um exemplo de *bounding box*.

aproximado de intersecção entre os objetos, simulando uma resposta aproximada de impacto. Isto consome muito tempo de CPU, é uma tarefa não-amigável e pouco inteligente. Sendo assim, serão discutidas algumas técnicas existentes para Detecção e Resposta à Colisão, bem como a estratégia usada para tal abordagem no ANIMADO.

3.4.1 Detecção de colisão

A Detecção da Colisão está diretamente relacionada ao momento em que se deu o choque entre os objetos. É um processo caro de ser resolvido, especialmente nos casos em que muitos objetos estão envolvidos na animação ou quando possuem formas geométricas mais complexas. Existem inúmeros algoritmos para a Detecção de Colisão e, em geral, eles são repletos de casos específicos.

O algoritmo implementado no ANIMADO, basicamente verifica a existência de intersecção entre dois objetos, otimizada utilizando-se *bounding boxes* como mostra a figura 3. Com isto, o número de testes de polígonos de intersecção é minimizado, encontrando-se uma “não-colisão” entre os objetos com muita rapidez. Este fato é simples de ser observado: quando um ponto é testado contra um poliedro, ele deverá ser primeiramente testado contra a *bounding box* do poliedro. Na maioria dos casos, elimina-se a necessidade de comparar este ponto com todas as faces do poliedro, agilizando o algoritmo.

Contudo, como já foi dito, os algoritmos para a Detecção de Colisão são complexos, dependentes da geometria do objeto e na maioria das vezes, não são totalmente abrangentes. Este fato também pode ser verificado no algoritmo utilizado no ANIMADO, que limita-se ao tratamento do seguinte caso:

- Cada aresta do objeto é testada contra cada face do outro objeto,

- Se a aresta interceptar a face em algum ponto,
- Calcule o ponto de colisão [MAD93].

Por enquanto, as rotinas que analisam a existência de intersecção entre dois sólidos não tratam os seguintes casos degenerados: aresta pertencente à face, ponto pertencente à face, objetos coincidentes, intersecção de vértices, etc. Atualmente, [MAD93] está sendo reavaliado para uma classe mais abrangente de objetos [FER93] para ser adaptado e integrado ao Sistema de Modelagem Phoenix [LT91].

3.4.2 Resposta à colisão

No momento em que ocorre um choque, as forças internas envolvidas são muito intensas e, portanto, as eventuais forças externas (como por exemplo, a gravidade) são desprezíveis durante o breve intervalo de tempo do choque. O sistema é considerado um sistema isolado. Estas forças internas interagem entre si obedecendo a Lei da Ação e Reação que diz: “Os impulsos das forças internas se neutralizam dois a dois e, como no sistema isolado a resultante externa é nula, o impulso total no sistema isolado é nulo” [SYM71]. Desta forma, a variação da quantidade de movimento do sistema permanece constante.

O tipo de algoritmo empregado para a modelagem da resposta à colisão dependerá do tipo de objetos envolvidos no choque. Entre os mais utilizados estão os algoritmos baseados na inserção de “molas” (força elástica) e os algoritmos baseados diretamente na resolução analítica do problema. Soluções analíticas são tipicamente mais rápidas em colisões bruscas porque a solução do sistema deverá ser encontrada uma única vez. Entretanto, para colisões amenas, a inserção de “molas” é mais aconselhável. Neste caso, a gravidade pode causar alguma interpenetração entre os dois objetos, e a solução analítica deverá ser aplicada diversas vezes. Uma simples mola que neutralize a gravidade é mais rápida e mais estável, sendo mais apropriada neste caso [MW88]. Estas duas abordagens serão detalhadas a seguir.

- Força Elástica

Dadas as posições atuais e os graus de liberdade dos objetos, as forças de interação são assim calculadas:

$$\begin{aligned} F_{elastica} &= k_{elastica} \rho \\ F_{amortecimento} &= k_{amortecimento} v \end{aligned} \quad (2)$$

onde,

$k_{elastica}$ e $k_{amortecimento}$ são constantes do modelo,

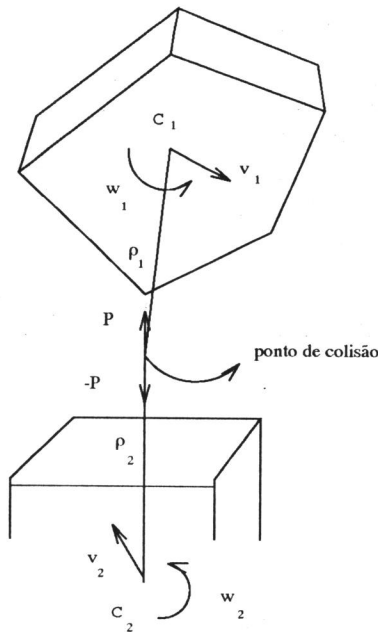


Figura 4: Um modelo de colisão entre dois corpos rígidos [MW88].

ρ corresponde à distância do ponto de colisão ao centro de massa do objeto,

e v é a velocidade do objeto.

• Solução Analítica

Neste caso, o tipo de colisão está diretamente relacionado à forma do objeto envolvido no instante do choque. A solução analítica para o problema depende da conservação do momento durante a colisão, e tem como produto final a velocidade linear e angular finais de cada objeto. Nesta seção será apresentada uma solução analítica geral para a colisão de dois corpos rígidos [MW88].

Seguindo a figura 4, a cada objeto é associada uma velocidade linear (v_i), uma velocidade angular (w_i), uma massa (m_i), um centro de massa (c_i) e uma matriz do tensor inercial (I_i) relativa ao centro de massa do corpo. Todos estes valores (para ambos os objetos) devem ser expressos em relação ao mesmo SC $\{A\}$. Tão logo seja detectada uma colisão, cada objeto terá um vetor associado ρ_i , que aponta do seu centro de massa ao ponto de colisão. Além destes valores, serão necessários três vetores unitários ortogonais (i, j, k) que definirão o "SC de colisão", k será perpendicular ao plano de colisão, i e j pertencerão a este plano.

A definição do plano de colisão é arbitrária. No caso do ANIMADO define-se o seguinte: Caso um vértice do objeto colida com a face do outro objeto, então, a face definirá o plano de colisão. Se uma aresta de um objeto colidir com uma aresta de outro objeto, estas duas arestas definem o plano de colisão. Se os dois vértices colidirem, a componente k do vetor normal ao plano de colisão apontará na direção da reta que os une.

Assumindo a existência de um único ponto de colisão, a solução para a dinâmica da colisão dependerá da resolução de quinze equações lineares, com quinze incógnitas: nova velocidade linear para cada objeto (v'_1, v'_2), nova velocidade angular para cada objeto (w'_1, w'_2) e um vetor impulso P . Por convenção, o impulso terá direção dada do objeto 2 para o objeto 1 [MW88].

As doze equações lineares que representam a mudança do momento angular e linear de cada objeto, devido ao impulso P , são derivadas das seguintes equações vetoriais:

$$\begin{aligned} m_1 v'_1 &= m_1 v_1 + P \\ m_2 v'_2 &= m_2 v_2 - P \\ I_1 w'_1 &= I_1 w_1 + \rho_1 \otimes P \\ I_2 w'_2 &= I_2 w_2 - \rho_2 \otimes P \end{aligned} \quad (3)$$

O coeficiente de elasticidade dos objetos envolvidos também deve ser levado em consideração e ele determina quanto de energia cinética é gasta neste processo. Assim, para um choque inelástico ($\epsilon = 0$), onde pode ocorrer deslize entre as superfícies, o impulso deverá ser perpendicular ao plano de colisão. Neste caso, considera-se que o produto interno de P com i e j será zero e a diferença de velocidade do ponto de colisão (quando vista de um dos dois objetos) será zero na direção k . Desta forma, outras três equações lineares devem ser consideradas:

$$\begin{aligned} P \cdot i &= 0 \\ P \cdot j &= 0 \\ (v'_2 + w'_2 \otimes \rho_2 - v'_1 - w'_1 \otimes \rho_1) \cdot k &= 0 \end{aligned} \quad (4)$$

No ANIMADO, o próprio sistema responderá à colisão, determinando a nova velocidade linear e angular dos objetos colidentes.

3.5 Controle das equações de movimento e dos módulos: o CONTROLE

O módulo CONTROLE se responsabilizará pelo envio de mensagens e sinais entre os módulos que compõem

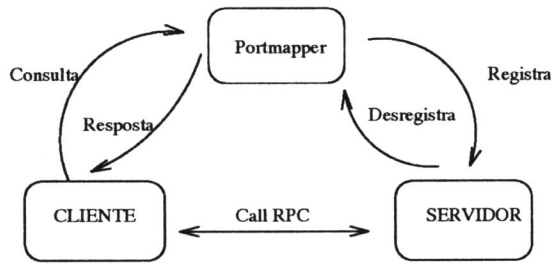


Figura 5: Esquema geral de comunicação entre portmapper, cliente e servidor.

o sistema. Caso ocorram colisões, ele determinará qual será a próxima etapa a ser executada pelo sistema. Depois da análise de colisão, se a velocidade relativa dos objetos no ponto de colisão (na direção da normal ao ponto de contato local) for **menor** do que uma certa taxa, pode-se considerar que ocorreu contato contínuo entre os objetos. Se os objetos estão aplicando uma força mutuamente (se um dos objetos for o chão), a dinâmica ainda pode ser simulada usando as equações de impacto citadas no item 3.4.2.

3.6 Comunicação entre os módulos: o INTEREXPSERV

Tendo como plataforma de trabalho o ambiente Unix, possuidor de recursos para a criação e manipulação de *sockets*⁸, o INTEREXPSERV⁹ foi criado para capacitar a comunicação entre os diversos módulos do ANIMADO, como mostra a figura 2. Ele executa as tarefas de comunicação entre todas as partes envolvidas no sistema (faz o esquema do servidor), interpreta expressões e armazena dados.

Os módulos que compõem o ANIMADO correspondem a processos que se comunicam através de chamadas de procedimentos remotos¹⁰. A comunicação entre os processos é baseada no modelo cliente/servidor, ou seja, na comunicação entre processos que necessitam de algum serviço e outros que realizam estes serviços. As requisições de serviços por parte dos clientes e o cadastramento destes serviços por parte dos servidores são administradas por um processo chamado *portmapper*¹¹ [NE92], como ilustra a figura 5.

No ANIMADO, espera-se que um dos módulos inicie o processo de animação. O INTEREXPSERV será o processo servidor que esperará que qualquer um dos outros módulos requirite um serviço para ele. Quando requisitado, ele dará atenção es-

⁸ *socket* é o bloco básico de comunicação em rede através de mecanismos de transporte baseados em fluxos[Mic88].

⁹ INTERpretador de EXPressões e SERvidor

¹⁰ RPC = Remote Procedure Call

¹¹ Um mapeador de portas

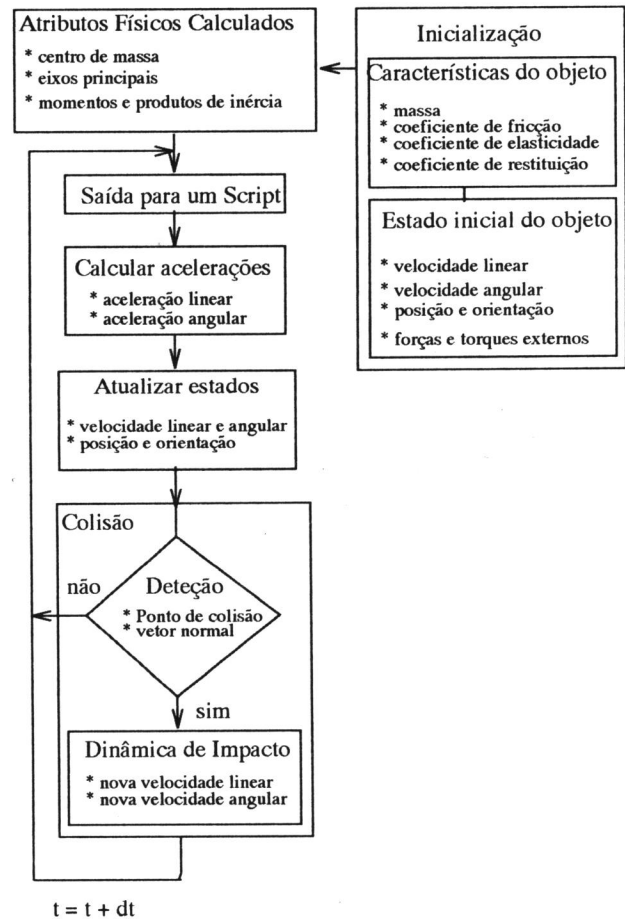


Figura 6: Esquema de funcionamento do ANIMADO.

pecial a um cliente de cada vez. O CONTROLE será o cliente responsável pela iniciação da comunicação, sua tarefa será a de definir o tempo inicial e final da animação e cuidar da discretização do tempo na integração, quando necessário. Também será responsável pelas mudanças relativas às novas forças que atuarão nos objetos que estão sendo animados quando houver uma colisão.

4 O funcionamento do ANIMADO

No ANIMADO, as etapas para a produção da Animação Modelada por Dinâmica interagem seguindo o esquema referenciado pela figura 6.

Na fase de inicialização do ANIMADO, cada objeto poderá apresentar características físicas, tais como: coeficiente de fricção, coeficiente de restituição, massa, etc. Em seguida, são calculadas outras propriedades físicas: centro de massa, tensor momento de inércia, eixos principais, etc.

O estado dinâmico de cada objeto descreve a velocidade linear, a velocidade angular, a posição e a orientação atual do objeto. O estado dinâmico cor-

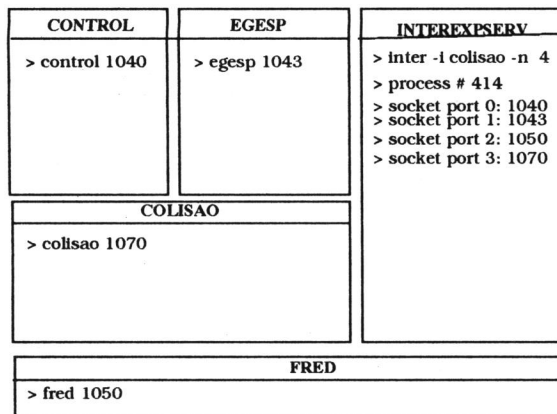


Figura 7: Comunicação entre os módulos do ANIMADO.

rente é usado para a obtenção do próximo estado dinâmico num intervalo de tempo próximo ($t + dt$). A atualização é feita em três níveis:

1. As acelerações angular e linear dos objetos são encontradas usando um Método para Resolução de Sistemas Lineares Esparsos (no caso, o EGESP) de acordo com a Formulação de Euler. O estado dinâmico corrente é utilizado para a obtenção destas acelerações.
2. Os objetos são movidos usando o estado dinâmico corrente. Isto envolve a Integração de edo utilizando um Método Numérico (no caso, o FRED) para a obtenção das posições e orientações dos objetos no próximo intervalo de tempo. Se o movimento do objeto for por meio de um arquivo de descrição (*script*), então, o estado pode ser lido ali.
3. Finalmente, com o auxílio do COLISÃO testa-se a existência de intersecção entre os objetos. Se o contato ocorreu, o novo estado dinâmico dos objetos que foram afetados pela colisão é calculado usando a Dinâmica de Impacto.

5 Descrição Funcional do ANIMADO

Os módulos que compõem o ANIMADO são ativados pelo comando `processos`, que abrirá os seus cinco módulos principais (EGESP, FRED, COLISÃO, CONTROL e o INTEREXPSERV), como mostra a figura 7.

Depois de escolhida a opção desejada, o ambiente de execução do ANIMADO estará pronto para começar a execução da animação, bastando ativar cada um dos módulos, via uma porta de comunicação, dada pelo INTEREXPSERV. O processo é desencade-

ado e, através das janelas, pode-se visualizar a comunicação entre todos os módulos.

6 Exemplo

Este arquivo de descrição corresponde a uma apresentação informal de um dos exemplos criados com o objetivo de ilustrar [ROD93] e produzir a simulação de uma colisão entre dois objetos (esfera e cubo), ilustrada na figura 8.

```
#####
#          COLISAO
#####

##### OBJETOS ENVOLVIDOS: esfera e cubo

##### CARACTERISTICAS DA ESFERA

massa                2.0
coeficiente de elasticidade 0.5
centro de massa      (0., 0., 0.)
momento de inercia   (4., 4., 4.)
produtos de inercia  (0., 0., 0.)

##### CONDICoes INICIAIS DA ESFERA

posicao inicial       (0., 3., 0.)
inclinacao           (0., 0., 0.)
velocidade linear inicial (0., -0.5, 0.)
velocidade angular inicial (5., 5., 5.)

##### CARACTERISTICAS DO CUBO

massa do cubo        1.7
coeficiente de elasticidade 1.0
centro de massa      (0., 0., 0.)
momento de inercia   (1., 1., 1.)
produtos de inercia  (0., 0., 0.)

##### CONDICoes INICIAIS DO CUBO

posicao inicial       (0., 0., 0.)
inclinacao           (0., 0., 0.)
velocidade linear inicial (0., .7, 0.)
velocidade angular inicial (10., 0., 0.)

##### TEMPO NA ANIMACAO

tempo inicial        0.
tempo final          3.3
incremento           0.3
```

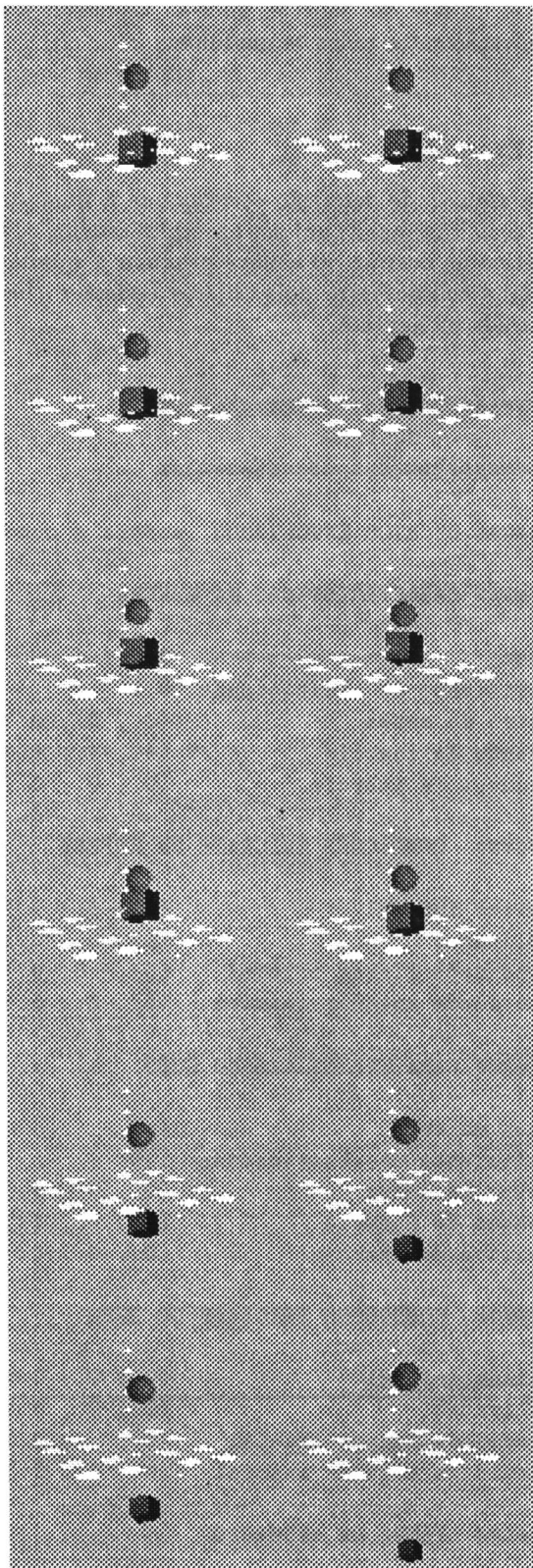


Figura 8: Um exemplo de colisão utilizando o ANIMADO, que deve ser visualizado da esquerda para a direita, de cima para baixo.

7 Conclusão

Sabe-se que a Animação Modelada por Dinâmica consome muito tempo de CPU e que necessita de ferramentas capazes de suportar cálculos intensivos, com boa precisão numérica. No ANIMADO, tanto o EGESP quanto o FRED possuem desempenho bastante adequado para contornar este problema.

A Animação Modelada por Dinâmica também requer um conhecimento das leis físicas que regem o nosso universo para efeitos da modelagem da animação. Caso contrário, o processo físico não será modelado com fidelidade, prejudicando o realismo da animação. Neste contexto, seria interessante substituir a formulação dinâmica de Euler pela formulação recursiva de Armstrong para avaliar o desempenho do sistema (velocidade, robustez, generalidade, etc.).

Três grandes problemas que os pesquisadores de animação por computador [TT91] têm enfrentado na abordagem por Dinâmica são: **difíceis de serem implementados**, pois os gastos para a construção de um modelo físico específico geralmente é elevado e, na maioria das vezes, o modelo não é reaproveitável, **difíceis de serem controlados**, pois o comportamento do modelo é frequentemente determinado indiretamente a partir de parâmetros não muito intuitivos (especificação de forças ou torques que são grandezas newtonianas difíceis de serem estimadas no modelo), além de estarem frequentemente sujeitos a problemas de instabilidade numérica, **podem ser lentos e custosos**, devido ao fato de que alguns modelos são computacionalmente complexos, levando o sistema a cálculos intensivos.

O ANIMADO foi estruturado ciente destas dificuldades, tendo como plataforma de trabalho estações SUN, e utilizando o sistema operacional Unix. Descrito em linguagem C e C++, com aproximadamente 3200 linhas de código, sua ideologia é direcionada à Programação Orientada a Objetos com utilização de *sockets* para a comunicação entre os diferentes módulos que compõem o sistema, baseado no modelo cliente/servidor [Mic88]. Sua performance é bastante satisfatória embora consuma ainda muita CPU. Entretanto, devido à grande flexibilidade e modularidade de sua arquitetura que comporta a comunicação entre seus diferentes módulos, este problema poderia ser minimizado, por exemplo, através da exploração da ociosidade de alguma máquina, distribuindo ou duplicando alguns de seus módulos na rede.

8 Trabalhos Futuros

Uma das principais metas para os próximos anos é o desenvolvimento de um sistema de animação in-

tegrado para a animação de cenas 3D envolvendo seres humanos conscientes da existência do ambiente em que estão. Tal sistema abrangeria uma interdisciplinaridade de abordagens e integraria aspectos e Métodos de Animação, Mecânica, Robótica, Fisiologia e Inteligência Artificial.

1. Produzir, automaticamente, seres humanos sintetizados com comportamento natural,
2. Melhorar a complexidade e o realismo do movimento: o realismo do movimento necessita ser melhorado não somente sob o ponto de vista das juntas (movimentos articulados), como no caso dos robôs, mas também, na deformação dos corpos, mãos e faces durante a animação,
3. Reduzir a complexidade da descrição do movimento,
4. Ampliar os casos de detecção de colisões, abordando o problema a nível da complexidade geométrica do objeto.

9 Agradecimentos

Gostaríamos de agradecer Valério Cappabianco Falcão II, pelo suporte computacional oferecido durante o desenvolvimento deste trabalho.

Referências

- [CRA89] John J. CRAIG. *Introduction to Robotics, Mechanics and Control*. Addison-Wesley, 1989.
- [FER93] Rubens Ramos Jr. FERNANDES. Relatório interno sobre o geomod (em elaboração), DCA - FEE - UNICAMP, 1993.
- [HAH88] James K. HAHN. Introduction to issues in motion control. *SIGGRAPH'88 - course notes*, (3):101-130, 1988.
- [HOR75] Robert. W. HORBECK. *Numerical Methods*. Quantum, 1975.
- [HOU92] Marcelo da Silva HOUNSELL. TOOKIMA: Uma ferramenta para animação modelada por computador. Master's thesis, DCA - FEE - UNICAMP, Abr 1992.
- [LT91] F. LOSERIES and Wu Shin TING. User manual for geometrical modeler PHOENIX. Technical report, 1991.
- [MAD93] Heraldo França MADEIRA. Geomod: Modelador geométrico (em andamento). Master's thesis, DCA - FEE - UNICAMP, 1993.
- [Mic88] SUN Microsystems. *A socket-based inter-process communications tutorial*. SUN Microsystems, 1988.
- [MW88] Matthew MOORE and Jane WILHELMS. Collision Detection and Response for Computer Animation. *Proc. SIGGRAPH'88*, (4):289-298, 1988.
- [NE92] Farid NOURANI and Marcos J. C. EUZÉBIO. Implementação de um portmapper. Relatório interno, DCA - FEE - UNICAMP, 1992.
- [ROD92] Maria Andréia Formico RODRIGUES. Animação: Mané mosquito e Corujito. Relatório Interno 02, DCA - FEE - UNICAMP, Fev 1992.
- [ROD93] Maria Andréia Formico RODRIGUES. ANIMADO: Um protótipo de um Sistema de Animação Modelada por Dinâmica. Master's thesis, DCA - FEE - UNICAMP, Jul 1993.
- [SEL89] Scott SELBIE. An introduction to the use of dynamic simulation for the animation of human movement. In *State of the Art in Computer Animation*, pages 33-45. Springer-Verlag, 1989.
- [SYM71] Keith R. SYMON. *Mechanics*. Addison - Wesley Publishing Company, 1971.
- [TT85] Nadia M. THALMANN and Daniel THALMANN. *Computer Animation - Theory and Practise*. Springer - Verlag, Tokyo - New York, 1985.
- [TT91] Nadia M. THALMANN and Daniel THALMANN. Complex models for animating synthetic actors. *IEEE Computer Graphics and Applications*, pages 32-44, Sep 1991.
- [WIL90] Jane WILHELMS. Dynamics for computer graphics: A tutorial. *SIGGRAPH'90 - course notes*, (8):85-115, Aug 1990.