FUNDAÇÃO DE ENSINO EURÍPIDES SOARES DA ROCHA
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LARISSA PAVARINI

ESTUDO E IMPLEMENTAÇÃO DO MÉTODO MASSA-MOLA PARA DEFORMAÇÃO EM AMBIENTES VIRTUAIS DE TREINAMENTO MÉDICO USANDO A API JAVA 3D

Marília
2006

LARISSA PAVARINI

ESTUDO E IMPLEMENTAÇÃO DO MÉTODO MASSA-MOLA PARA DEFORMAÇÃO EM AMBIENTES VIRTUAIS DE TREINAMENTO MÉDICO USANDO A API JAVA 3D

Dissertação apresentada ao Programa de Mestrado do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do Título de Mestre em Ciência da Computação (Área de Concentração: Realidade Virtual).

Orientador:
Prof[a]. Dr[a]. Fátima L. S. Nunes Marques

Marília
2006

PAVARINI, Larissa

Estudo e Implementação do Método Massa-Mola para Deformação em Ambientes Virtuais de Treinamento Médico usando a API Java 3D / Larissa Pavarini; Orientador: Fátima L. dos Santos Nunes Marques. Marília, SP: [s.n.], 2006.

146 f.

Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha.

1. Deformação 2. Modelagem de Objetos 3D 3.Realidade Virtual 4. Treinamento Médico

CDD: 006

LARISSA PAVARINI

ESTUDO E IMPLEMENTAÇÃO DO MÉTODO MASSA-MOLA PARA DEFORMAÇÃO EM AMBIENTES VIRTUAIS DE TREINAMENTO MÉDICO USANDO A API JAVA 3D

Banca examinadora da dissertação apresentada ao Programa de Mestrado da UNIVEM/F.E.E.S.R., para obtenção do Título de Mestre em Ciência da Computação. Área de Concentração: Realidade Virtual.

Resultado: Aprovada

ORIENTADOR: Prof[a]. Dr[a]. Fátima de Lourdes dos Santos Nunes Marques

1º EXAMINADOR: Fátima de Lourdes dos Santos Nunes Marques

2º EXAMINADOR: Márcio Eduardo Delamaro

3º EXAMINADOR: Alcimar Barbosa Soares

Marília, 17 de Abril de 2006.

Dedico este trabalho a minha família e ao meu amado André, pelo apoio, incentivo e, em especial, ao amor e carinho que me proporcionaram nesta jornada.

AGRADECIMENTOS

Escrever uma dissertação de mestrado é uma etapa muito trabalhosa, porém muito gratificante. E poder agradecer publicamente às pessoas que fizeram parte desta etapa da minha vida uma realidade, é melhor ainda. Pois é aqui que será expresso todo amor e carinho que tenho a cada um.

Primeiramente agradeço a ***Deus****, por estar comigo em todos os momentos, participando de cada etapa, de cada conquista que obtive neste período, de ser meu amigo incondicional, pelo privilégio de estar estudando e de me conceder sabedoria e inteligência não só na pesquisa científica, mas na vida.*

A minha querida orientadora Professora Doutora ***Fátima Nunes****, pela amizade, conselhos, pelo profissionalismo e pelas excelentes orientações que tem me proporcionado que me trouxeram amadurecimento acadêmico, profissional e principalmente humano. Por acreditar no meu trabalho e no meu esforço. Obrigado por ser acima de tudo minha grande e preciosa amiga.*

A minha família que amo tanto, em especial a meus pais ***Carlos*** *e* ***Juraci*** *e minhas irmãs* ***Carla*** *e* ***Fernanda*** *pelo carinho, amor e paciência. Por entender minha ausência em momentos importantes. Por não terem medidos esforços, por me apoiarem e sempre estarem comigo em todos os momentos. Pelas orações, pelo amor, carinho e confiança e por sempre acreditarem que eu seria capaz de chegar aqui.*

Ao meu querido e amado namorado ***André*** *por me apoiar em tudo, pelo seu amor, carinho, respeito e dedicação, por fazer da minha vida mais bela, especial e completa. Pelas intermináveis traduções de artigos, pela compreensão quando estive ausente, por sempre ter uma palavra de incentivo e por tornar tudo simples, mesmo quando eram complexas. Meu amor, meu companheiro, minha vida, meu tudo obrigado por simplesmente ser você na minha vida.*

Aos meus segundos pais ***Ananias*** *e* ***Édna*** *pelo amor incondicional e orações, acreditando no meu esforço e trabalho. Aos meus cunhados* ***Daniel*** *e* ***Fabiane*** *que mesmo longe fisicamente não se esqueceram um minuto de mim, sempre me apoiando e orando a Deus para que Ele me desse sabedoria e força;* ***Priscila*** *e* ***Widmer*** *pelas conversas, sonhos planejados juntos e dedicação quando o assunto era me ajudar; e ao* ***Matheus*** *e a* ***Suellen*** *por estarem do meu lado, em todos os momentos. Amo todos vocês.*

A todos os meus amigos e companheiros do mestrado, que muito têm me ajudado. Tenho aprendido com eles e tenho crescido muito pessoalmente e academicamente, em especial aos meus amigos ***Júnior, Claudete, Fábião e Itararé*** *e integrantes do laboratório de pesquisa LApIS (Laboratório de Aplicações da Informática na Saúde),* ***Montanha, Danilo, Adriano, Bárbara, Ana Cláudia, Leonardo****.*

Ao Prof. Doutor ***José Carlos Gomes*** *e* ***Ana Cláudia*** *por serem meus grandes amigos, e estarem tão presentes na minha vida, me dando conselhos sábios e sempre me recebendo de braços abertos em sua casa para tirar dúvidas e ensinar as intermináveis fórmulas matemáticas.*

Aos meus grandes amigos ***Paulo*** *e* ***Rodolfo****. Não tenho palavras pra expressar o quão grande é minha amizade, carinho e respeito por vocês. Talvez não consiga expressar tudo o que deveria, mas vocês representam na minha vida companheiros maravilhosos de uma grande jornada vencida.* ***Paulo*** *obrigado pelas ricas conversas, pela criatividade em todas as situações, pelo companheirismo. Por ser meu amigo em tudo e por tudo, companheiro mesmo, incentivador... Um irmão. Ao querido amigo* ***Rodolfo*** *por estar sempre do meu lado, independente de qualquer coisa. Pela ajuda incondicional, não medindo esforços. Pelas muitas conversas nos laboratórios, pelas descobertas, pelos gritos de conquista de conseguir fazer tudo funcionar. Obrigado por existir e por ser meu amigo.*

Ana Paula*, amiga, irmã e companheira de laboratório. Pessoa formidável que me ensina coisas maravilhosas a cada dia me trazendo alegria, descontração nos momentos tensos e amor em tudo.*

Leninha, Karina e Gislaine *por serem minhas queridas amigas que fazem com que meus dias sejam mais felizes,ou melhor, todos os dias.*

Doutorzinho Fábio *pela constante diversão nos laboratórios, pela amizade, pelo exemplo de dedicação, garra e conquista.*

Ao meu grande amigo e querido ***César*** *por ser tão especial, um exemplo de vida e força de vontade. Pela companhia, pelo apoio e pela dedicação. Por me ajudar nas infinitas correções de português, pelas escritas de artigos, mas acima de tudo pelo incentivo e apoio em momentos difíceis me mostrando que tudo na vida é simples e belo... Enfim obrigado por existir!*

Silvia *por me ajudar com as traduções, por ser minha "miga" e irmã do coração.*

Paulo Renato *que além de ser uma pessoa maravilhosa não tem medido esforços pra me ajudar.*

Davi *e* ***Eduardo*** *pelas longas e agradáveis conversas, pelas trocas de experiências que estamos passando juntos no mestrado, acreditando sempre em mim... Sempre. Por serem tão constantes na minha vida mesmo estando tão longe... Meus amigos queridos e especiais!*

Mychele *e* ***Tatiana*** *amigas que sempre me apoiaram e acreditaram que eu chegaria aqui e também porque entenderam – ou pelo menos tentaram entender – a minha ausência. Amo vocês.*

Aos funcionários do UNIVEM...

À ***CAPES****, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior e ao* ***CNPq****, Entidade Governamental Brasileira promotora do Desenvolvimento Científico e Tecnológico, (Processo nº 472348/04-3), pelo apoio financeiro.*

"O essencial é invisível aos olhos... só se vê bem com o coração!"
Saint Exupery

“Para ser o que sou hoje,
fui vários homens.
E se volto a encontrar-me com os homens que fui,
não me envergonho deles.
Foram etapas do que sou.
Tudo que sei custou as dores das experiências.
Tenho respeito pelos que procuram, pelos que tateiam, pelos que erram.
E o que é mais importante, estou persuadido de que minha luz se
extinguiria se eu fosse o único a possuí-la".
Goethe

PAVARINI, Larissa. **Estudo e Implementação do Método massa-mola para Deformação em Ambientes Virtuais de Treinamento Médico usando a API Java 3D.** 2006. 147f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

RESUMO

O treinamento de procedimentos médicos pode ser beneficiado com a construção de ferramentas de Realidade Virtual, que possibilitam a execução de tarefas com características de interação, imersão e engajamento. Para que essas aplicações sejam realistas, a deformação de objetos tridimensionais na cena é um fator muito importante, visto que deve haver reações a ações do usuário com o ambiente.

Esta dissertação apresenta um estudo relacionado a algumas técnicas existentes na literatura que permitem deformação em ferramentas de Realidade Virtual e a implementação de uma delas, o método massa-mola, usando a tecnologia Java. Esta implementação adapta o método citado para possibilitar a deformação de objetos tridimensionais em ambientes virtuais para treinamento médico. O sistema desenvolvido, denominado *DefApliMed* (Sistema de Deformação em Aplicações Médicas), executa a deformação através do reposicionamento de vértices e arestas do objeto no Ambiente Virtual.

Palavras-chave: Deformação, Massa-Mola, Realidade Virtual, API Java 3D, Treinamento Médico.

PAVARINI, Larissa. **Estudo e Implementação do Método massa-mola para Deformação em Ambientes Virtuais de Treinamento Médico usando a API Java 3D.** 2006. 147f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

ABSTRACT

The medical training procedures can be improved with the construction of Virtual Reality's tools, which allow the tasks performance with interaction, immersion and engagement characteristics. The deformation of three-dimensional objects in the scene is a very important factor, in order to become applications more realistic, since it must have reactions for the actions that users execute in the environment. This dissertation presents a study related to some techniques existent in the literature that allow deformation and implementation in of the mass spring method for Virtual Reality's tools, , by using the Java technology.This implementation adapts the mentioned method to enable the deformation of three-dimensional objects in virtual environments to medical training. The developed system is named *DefApliMed* (Deformation's System in Medical Applications) and executes the deformation by replacing the position of vertexes and edges of the object in the Virtual Environment.

Keywords: Deformation, Mass Spring, Virtual Reality, API Java 3D, Medical Training.

PAVARINI, Larissa. **Estudo e Implementação do Método massa-mola para Deformação em Ambientes Virtuais de Treinamento Médico usando a API Java 3D.** 2006. 147f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

RESUMEN

La práctica de procedimientos médicos puede beneficiarse con la construcción de herramientas de Realidad Virtual, que posibilita la ejecución de tareas con características de interacción e inmersión.
Para que estas aplicaciones sean realistas, la deformación de objetos tridimensionales en la escena es un factor muy importante, ya que debe haber reacciones a las acciones del usuario con el ambiente.
Esta disertación presenta un estudio relacionado a algunas técnicas que hay en la literatura que permiten la deformación en las herramientas de Realidad Virtual y la aplicación de una de ellas, dónde el método escogido fue el mass spring, usando la tecnología Java. Esta implementación adapta el método citado para posibilitar la deformación de objetos tridimensionales en ambientes virtuales para el entrenamiento médico.
El sistema desarrollado, llamado DefApliMed (Sistema de Deformación en Aplicaciones Médicas), ejecuta la deformación por el reemplazo de nuevos vértices y bordes del objeto en el Ambiente Virtual.

Palabra-llave: Deformación, Mass Spring, Realidad Virtual, API Java 3D, Entrenamiento Médico.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2D – Bidimensional

3D – Tridimensional

API – *Application Programming Interface*

AV – Ambiente Virtual

DefApliMed – Sistema de Deformação em Aplicações Médicas

E/S – Entrada e Saída

FEM – *Finit Element Method*

FFD – *Free Form Deformation*

J3D – Java 3D

JDK – *Java Development Kit*

JHGT – *Java 3D Head Generation Tool*

KGF – Quilograma Força

MI – Minimamente Invasiva

MB – *Mega Bytes*

MS – *Mass Spring*

RAM – *Randomic Access Memory*

RV – Realidade Virtual

UML – *Unified Modeling Language*

VRML – *Virtual Reality Modelling Language*

X3D – *Extensible 3D*

SUMÁRIO

1. INTRODUÇÃO

Segundo Hancock (1995) a Realidade Virtual (RV) é a forma mais avançada de interface do usuário com o computador. É a área da informática que permite visualizar, manipular, explorar, interagir e modificar dados complexos pelo computador envolvendo software, hardware, interface com o usuário e permitindo a criação de aplicações.

A utilização da RV para aplicações na área médica tem crescido em quantidade e qualidade, principalmente devido ao avanço de hardware e software observado nas últimas décadas (SZÉKELY, 1999).

Este avanço mostra-se um grande aliado à área médica na medida em que auxilia no desenvolvimento de ferramentas automatizadas para o treinamento médico. As aplicações de RV nesta área têm abrangido desde o estudo da anatomia, passando pela educação e chegando aos treinamentos de cirurgias, os quais possibilitam aos estudantes o exercício da atividade prática por meio da utilização de simuladores de procedimentos (RODRIGUES *et. al,* 2002). Esses simuladores de procedimentos constituem hoje as principais aplicações de RV em Medicina e têm o objetivo de ajudar estudantes de Medicina na fase de residência no treinamento de procedimentos médicos antes de executá-los efetivamente em pacientes reais. Desta forma, podem auxiliar esses profissionais a adquirir habilidades básicas para manipulação de instrumentos, podendo oferecer maior confiabilidade aos exames e demais procedimentos (FREITAS *et al.,* 2003).

Todo procedimento cirúrgico gera conseqüências que vão desde os dias de internação indesejados até seqüelas físicas no paciente. Para minimizar as conseqüências indesejadas, grande ênfase tem sido atribuída às técnicas minimamente invasivas (MI), que consistem em

procedimentos cirúrgicos com a mínima invasão no corpo do paciente. Assim, em geral, os orifícios necessários têm tamanho diminuto, na ordem de milímetros ou poucos centímetros.

Com a difusão da tecnologia, as técnicas MI vêm sendo aplicadas em uma série de procedimentos, desde exames de biópsia até remoção de estruturas malignas do corpo humano. Em geral, esses procedimentos empregam objetos de pequeno porte (seringas, por exemplo) que são introduzidas no corpo do paciente através de pequenas incisões.

Por se tratar de um procedimento que envolve objetos pequenos de difícil manipulação e incisões mínimas, pode haver uma dificuldade geral em relação ao treinamento. Sendo assim, a RV pode ajudar nesse tipo de treinamento, visto que permite a execução do procedimento virtual quantas vezes forem necessárias até que o aluno ou o profissional esteja pronto para praticá-lo em um paciente real.

Para que os simuladores sejam realísticos, é necessária a implementação de técnicas de detecção de colisão em tempo real e que forneçam precisão. Uma conseqüência imediata de uma colisão entre um objeto rígido e um objeto flexível é a deformação do objeto flexível, considerando -as características da ação executada e dos objetos envolvidos.

Na Ciência da Computação, a expressão tempo real, em geral, refere-se à interação simultânea entre sistemas ou pessoas, com intervalos muito curtos ou com implicações sérias ao nível do atraso. É, portanto, uma operação em que o par ação/reação deve demorar menos tempo que o atraso máximo permitido pelo sistema (SHAW, 2003).

Sistemas em tempo real são sistemas computacionais submetidos a requisitos de natureza temporal, ou seja, que se utilizam do tempo para permitir uma ação e uma reação necessária para trazer a realidade do desempenho do sistema, trazendo, assim, resultados que devem estar corretos lógica e temporalmente sendo requisitos definidos pelo ambiente físico trabalhado (FARINES *et al.,* 2000).

Esses aspectos temporais não estão limitados a uma questão de maior ou menor desempenho, mas estão diretamente relacionados à funcionalidade de um sistema em si (SABBINENI e CHAKRABARTY, 2005).

Os sistemas em geral desenvolvem o trabalho usando o tempo necessário, já sistemas de tempo real fazem o trabalho usando o tempo disponível e determinado logicamente a cada sistemas pré-determinado (LIPARI *et al.,* 2004).

A Figura 1 apresenta a caracterização do sistema de tempo real tendo uma interface intermediaria para transmitir as ações e reações feitas pelo usuário, tendo resposta dentro e fora do sistema.

Essa conceituação de tempo real é importante em sistemas de treinamento médico que utilizam a RV, visto que tais aplicações devem atender ao requisito de fornecer uma resposta dentro de um tempo máximo a fim de que usuário tenha a sensação de que a reação do sistema é imediata à ação que executou no Ambiente Virtual (AV).

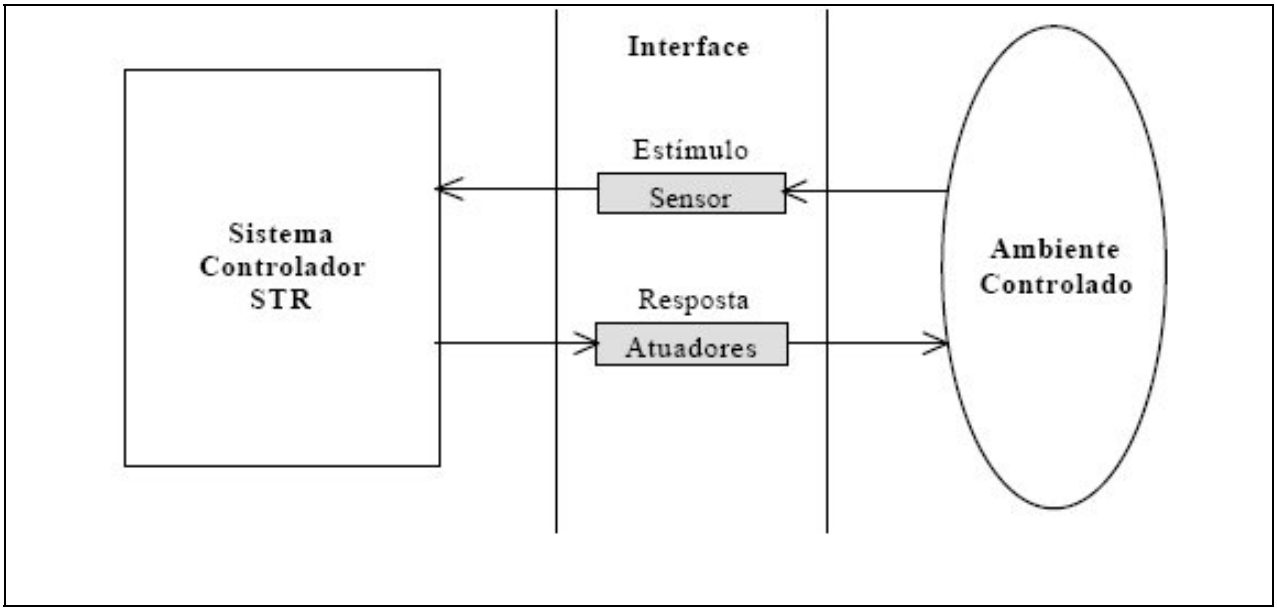


Figura 1 – Caracterização de um sistema de tempo real (FARINES *et al.*, 2000)

Assim, objetos virtuais deformáveis começam a aparecer em aplicações de RV para proporcionar realismo aos ambientes que os contêm (CHOI *et al.*, 2002). Para se obter efeitos realísticos, cálculos matemáticos complexos precisam ser desenvolvidos para que os objetos sejam remodelados pela própria aplicação, geralmente em tempo real (SCHNEIDER, 1998).

Este trabalho apresenta os estudos de técnicas de deformação e a implementação do método massa-mola, visando à aplicação em ferramentas virtuais para treinamento médico. Dentro desse contexto, o estudo foi focado nos procedimentos de treinamento médico para a execução das técnicas minimamente invasivas, considerando que essas possuem características comuns entre si.

1.1 Objetivo do Trabalho

O objetivo geral deste trabalho é a implementação de um sistema de deformação para ser implantado em aplicações de treinamentos médicos.

Como objetivos específicos, têm-se:

- implementação do método massa-mola;
- desenvolvimento do Ambiente Virtual que permita a importação dos objetos, interação e *feedback*;
- disponibilização do método para sua utilização em um *Framework* de treinamento médico que está em fase de construção;
- simulação da deformação de forma que o usuário tenha a sensação de que os objetos estão executando a tarefa em tempo real.

1.2 Justificativa do Trabalho

Em aplicações de RV, para treinamento médico, quando um objeto rígido toca a superfície de um órgão humano, deve-se oferecer um *feedback* de deformação, contribuindo para o realismo do procedimento.

Como contribuição para solução desta necessidade, o projeto *DefApliMed* (Sistema de Deformação em Aplicações Médicas), possibilita a implementação de deformação em ferramentas de Realidade Virtual, focalizando as aplicações para treinamento médico. Foi utilizado o método massa-mola, implementado na linguagem Java, utilizando a *Application Programming Interface* (API) Java 3D.

O método massa–mola foi escolhido por sua facilidade de implementação, embora a definição de parâmetros para o funcionamento deste método seja uma tarefa que envolva muito esforço, ou seja, por não ser trivial. Como o projeto visa a fornecer a base de funcionalidade para qualquer aplicação que envolva deformação, a facilidade de implementação foi um fator determinante para essa escolha.

A abordagem de programação orientada a objetos permite uma melhor estruturação das unidades (métodos e classes) envolvidas para o funcionamento do sistema, possibilitando a implementação de outros métodos de deformação sem que sejam necessárias grandes alterações, na construção do AV.

Em relação à linguagem Java, uma das características para sua utilização é que além de ser uma ferramenta que oferece recursos extremamente poderosos para a criação de aplicações é uma linguagem gratuita que pode ser expandida e inserida em locais onde recursos financeiros são escassos. Essa preocupação é devido ao fato de muitas linguagens de programação e as bibliotecas gráficas relacionadas a ela serem pagas, e como o intuito do

projeto é inserir esse procedimento para a sociedade, foi proposto a utilização dessa linguagem para inserir em hospitais públicos de forma que vários usuários possam ter acesso à ferramenta sem muito esforço financeiro.

O hardware utilizado para o desenvolvimento e os testes desta aplicação consistiu em um computador com processador Pentium4 de 3.2GHz, com 512 MB de memória RAM e placa de vídeo Inno3D Gforce FX 5200. Mesmo sendo uma máquina potente, devido ao grande crescimento e avanço da tecnologia seu custo não é tão alto, comparado com o custo das licenças dos softwares para disponibilização da ferramenta para a sociedade.

1.3 Organização do Trabalho

Para melhor compreensão dos tópicos apresentados, a dissertação está organizada em sete capítulos, além desta introdução.

O Capítulo 2 aborda o estudo sobre Realidade Virtual na Medicina, dando ênfase a procedimentos minimamente invasivos. Este capítulo também apresenta várias aplicações de simulações para treinamento médico utilizando a técnica descrita, ou seja, projetos relacionados que envolvem a deformação utilizando Realidade Virtual.

O Capítulo 3 relaciona os conceitos sobre deformação em RV, citando as técnicas mais empregadas, suas vantagens e desvantagens. Também apresenta aplicações de simulação em RV específicas da área médica que proporcionam deformação em tempo real.

O Capítulo 4 descreve a tecnologia de software escolhida para desenvolver o projeto, justificando a escolha realizada.

O Capítulo 5 apresenta o projeto *DefApliMed* explicando detalhadamente seu funcionamento e implementação.

No Capítulo 6 são apresentados os resultados obtidos e a discussão acerca desses resultados. As conclusões obtidas e a citação de trabalhos que podem contribuir para o desenvolvimento de projetos futuros relacionados ao *DefApiMed*, são apresentadas no Capítulo 7.

Por fim, no Capítulo 8 são apresentadas as referências bibliográficas que forneceram a base teórica do projeto.

Umapêndice sobre técnicas minimamente invasivas também foi incluído ao final da dissertação a fim de fornecer detalhes de assuntos relacionados ao projeto.

2. PROCEDIMENTOS MINIMAMENTE INVASIVOS E REALIDADE VIRTUAL NA MEDICINA

Para Hancock (1995) a Realidade Virtual (RV) pode ser definida, de uma maneira simplificada, como sendo a forma mais avançada de interface do usuário com o computador. Complementando, Burdea e Coiffet (1994) afirmaram que é uma nova área da Ciência da Computação, fruto do desenvolvimento da ciência e da tecnologia, permitindo que o usuário visualize, manipule e interaja com computadores e dados extremamente complexos em um ambiente sintético tridimensional gerado por computador.

Nos últimos anos a RV vem sendo amplamente explorada na área médica e aplicações vêm sendo desenvolvidas objetivando o planejamento, treinamento e a assistência de procedimentos neste campo de conhecimento.

A construção dessas aplicações traz benefícios aos estudantes, profissionais e usuários da área, permitindo assim a visualização de dados complexos, particularmente para o planejamento e realização de cirurgias e treinamento de procedimentos (LIMA e NUNES 2004). Machado *et al.* (2001) lembra que fatores como custo, disponibilidade de materiais e segurança, aliados ao realismo, são algumas das vantagens relacionadas ao uso destas aplicações.

Atualmente, os projetos de aplicações de RV na Medicina estão, na sua maioria, direcionados à construção de ferramentas de simulação para procedimentos cirúrgicos, cujo objetivo é permitir a prática de técnicas em ambientes que imitam a realidade de um procedimento, permitindo que o usuário adquira habilidade com a prática.

Com o objetivo de contextualizar o presente trabalho, este capítulo apresenta uma visão geral dos tópicos relacionados à área de RV em Medicina. Primeiramente serão

apresentados conceitos sobre a técnica Minimamente Invasiva (MI), que constitui o foco deste projeto. Em seguida são apresentados projetos desenvolvidos ferramentas de RV aplicadas à Medicina.

2.1 Técnica Minimamente Invasiva (MI)

Segundo Fernandes (2003), a partir da segunda metade do século XX, novas tecnologias ocasionaram o desenvolvimento de técnicas cirúrgicas inovadoras, proporcionando menor lesão tecidual e diminuição no tempo da operação e também a diminuição da recuperação do paciente no pós-operatório, sendo menos agressivas e trazendo grande vantagem para a recuperação precoce do bem–estar do paciente.

Essa técnica, conhecida como procedimento minimamente invasivo (MI), consiste em métodos com manuseio cirúrgico restrito ao máximo possível à área afetada, preservando as estruturas não envolvidas no procedimento e oferecendo um tempo menor de recuperação para os pacientes, com a conseqüente diminuição da permanência em ambiente hospitalar (MACHADO, 2003).

A técnica MI é aquela formada por um conjunto de procedimentos necessários para realizar uma intervenção cirúrgica mediante pequenas incisões, graças à introdução de uma câmera ou mediante mecanismos de tração, visualizando o campo por meio de dispositivos de vídeo-endoscopia e instrumental cirúrgico adequado (GARCIA, 2002).

Cohen (1997) afirma que os avanços na tecnologia óptica e transmissão da luz aumentaram a precisão dessa técnica, porém, foi a introdução das imagens de vídeo que possibilitou aos cirurgiões realizarem as intervenções cirúrgicas adaptadas aos instrumentos e tecnologias modernas e sofisticadas.

A cirurgia MI tornou-se um conceito global utilizado em diversas especialidades como cirurgias torácicas, pediátricas, ginecológicas, urológicas, traumatológicas, plásticas e otorrinolaringológicas.

De uma forma resumida, Cohen (1997) relata as vantagens desse tipo de cirurgia apoiando-se nos seguintes pontos: redução da resposta inflamatória sistemática associada à cirurgia, ou seja, a diminuição da inflamação e a melhoria na resposta imunológica apresentadas pelos pacientes; mínimo risco possível na cirurgia; diminuição da dor pós-operatório devido, principalmente, à ausência de grandes incisões cirúrgicas; seqüelas mínimas e boa cicatrização das incisões cirúrgicas.

Conceitos simplificados sobre procedimentos MI e aplicações da técnica nas diversas áreas da Medicina são apresentados no apêndice desta dissertação.

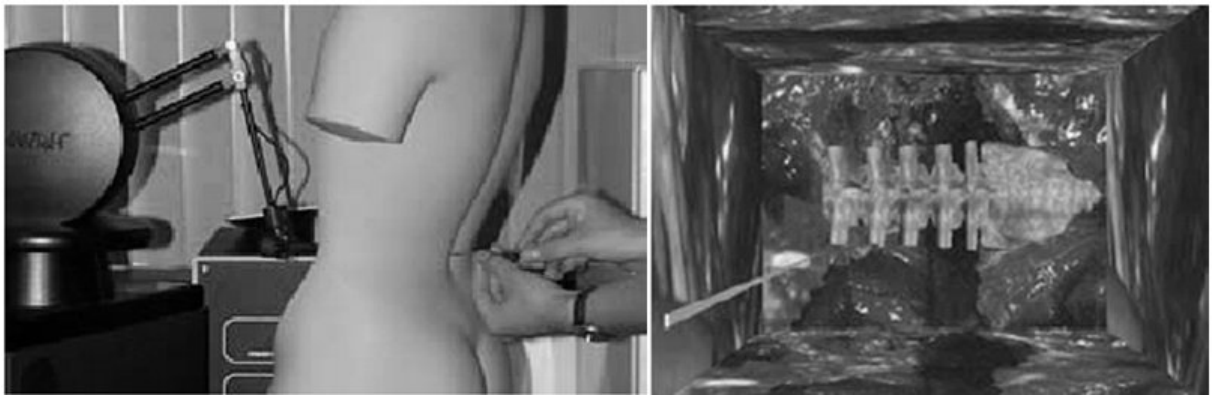
2.2 Trabalhos Correlatos - RV na Medicina

Uma grande variedade de trabalhos pode ser encontrada na literatura na área de RV aplicada à Medicina. A maioria deles oferece sistemas interativos, em tempo real, principalmente voltado para treinamento de exames e cirurgias, dando enfoque ao estímulo visual e tátil. A seguir serão citados alguns desses trabalhos, obedecendo a uma ordem cronológica crescente na apresentação.

Haluck *et al.* (2000) apresentam um simulador para treinamento em acupuntura lombar que tem o objetivo de auxiliar alunos de Medicina a evitar erros que possam causar lesões e dores desnecessárias. O treinamento em acupuntura lombar permite registrar todas as ações do usuário para posterior análise das ações internas ocorridas durante o procedimento. Consiste de uma agulha para acupuntura lombar que penetra o dorso de um manequim

humano, unida a um dispositivo háptico *Phantom Desktop* da *Sensable Technologies*, como mostra a Figura 2(a). Esse dispositivo háptico permite ao estudante sentir forças resistentes ao longo de uma trajetória. Se a trajetória é incorreta ele pode sentir, por exemplo, uma colisão com um osso.

Quanto à implementação, modelos 3D foram armazenados como arquivos VRML (*Virtual Reality Markup Language*) permitindo ilustrar como a agulha está sendo inserida, Figura 2(b). Para o desenvolvimento dos módulos físico e gráfico foram utilizadas a biblioteca GHOST e a ferramenta *WorldToolKit* da *Sense8* (RAHN,1998), além da utilização da biblioteca OpenGL (COHEN e MANSSOUR, 2006).



(a) **(b)**

Figura 2 – Simulador de Acupuntura Lombar – (a) Dispositivo háptico para treinamento, (b) Modelo 3D da espinha dorsal em VRML (HALUCK *et al.*, 2000)

Wagner *et al.* (2002) apresentam outra aplicação interessante que é o *EyeSi,* um simulador de RV para cirurgia intra-ocular, que permite variar as tarefas a serem executadas na simulação da cirurgia. Essas tarefas estão relacionadas a treinamentos como manipulação de instrumentos e cálculo de distâncias estimadas por sombras.

O projeto é baseado em uma instalação mecânica modelada para fornecer todas as percepções sensoriais do olho ao cirurgião em treinamento, como apresentado na Figura 3. O olho mecânico tem o mesmo grau de liberdade de rotação do olho humano, sendo que o efeito

dos músculos é modelado por um conjunto de molas sobre cada eixo de rotação. Além disso, a visualização realista do cenário de operação inclui efeitos de iluminação e sombras.

(a) **(b)**

Figura 3– (a) Cirurgia de olho real e (b) simulação de cirurgia de olho virtual (WAGNER *et al.*, 2002)

A Figura 4 apresenta a simulação da cirurgia vítreo-retinal na cirurgia real e na virtual, a qual apresenta características importantes para o desenvolvimento do projeto descrito no capítulo 5, visto que a partir de uma aplicação real consegue-se criar os passos para permitir a aplicação virtual do treinamento médico. A Figura 5 apresenta o treinamento virtual da cirurgia de retina.

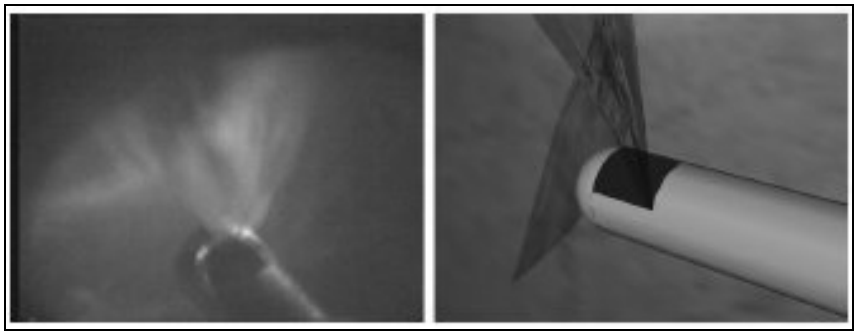
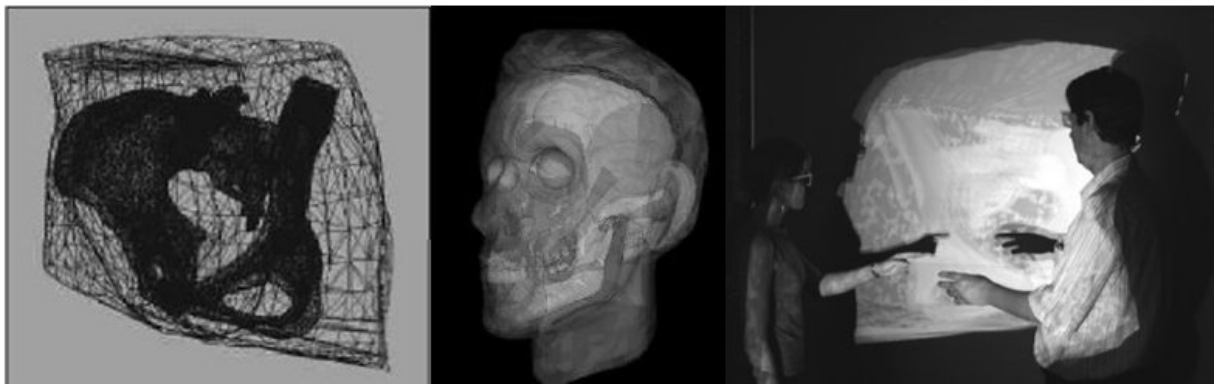


Figura 4 - Imagem real capturada (esquerda) e a visão da simulação do treinamento da cirurgia (direita) (WAGNER *et al.*, 2002).

Figura 5 – Simulação de treinamento de cirurgia de retina (WAGNER *et al.*, 2002)

Em Machado *et al.* (2004) é apresentado um sistema denominado *CyberMed*, cujo objetivo é apoiar o ensino e treinamento médico através de explorações interativas do corpo humano e da simulação realista de procedimentos médicos em um ambiente virtual imersivo. Suas principais características são: a visualização tridimensional, o uso de modelos realistas, interação espacial com sensação de toque, deformação interativa das estruturas tocadas, compartilhamento visual e supervisão e avaliação das ações do usuário.

Segundo os autores, o sistema *CyberMed* tem sido utilizado com sucesso na visualização interativa de estruturas do corpo humano, de uma forma que permite ao usuário verificar os detalhes do objeto mais internamente, conforme apresentado na Figura 6(a). Também permite que o usuário observe melhor o objeto por meio do uso de semi-transparência, como apresenta a Figura 6(b), e dá suporte à educação e à prática médica tendo como principal motivação a redução de custos de sistemas para simulação em medicina, aliado ao uso de componentes convencionais, tornando essa tecnologia acessível. A Figura 6(c) mostra pessoas estudando uma região específica, que é a pélvica, utilizando o sistema *CyberMed.*



(a) **(b)** **(c)**

Figura 6 – Sistema CyberMed – (a) Bacia modelada (b) modelo do crânio (c) Estudo da região pélvica (MACHADO *et al.*, 2004)

Outros exemplos importantes de aplicações de RV envolvendo a técnica de cirurgia MI são os sistemas: KISMET (*Kinematic Simulation, Monitoring and Off-Line Programming*

Environment for Telerobotics) e LapSim, que são dispositivos disponíveis comercialmente, projetados para treinamento e simulação de cirurgias laparoscópicas. A Figura 7 apresenta uma simulação de uma cirurgia laparoscópica usando o dispositivo KISMET como suporte.

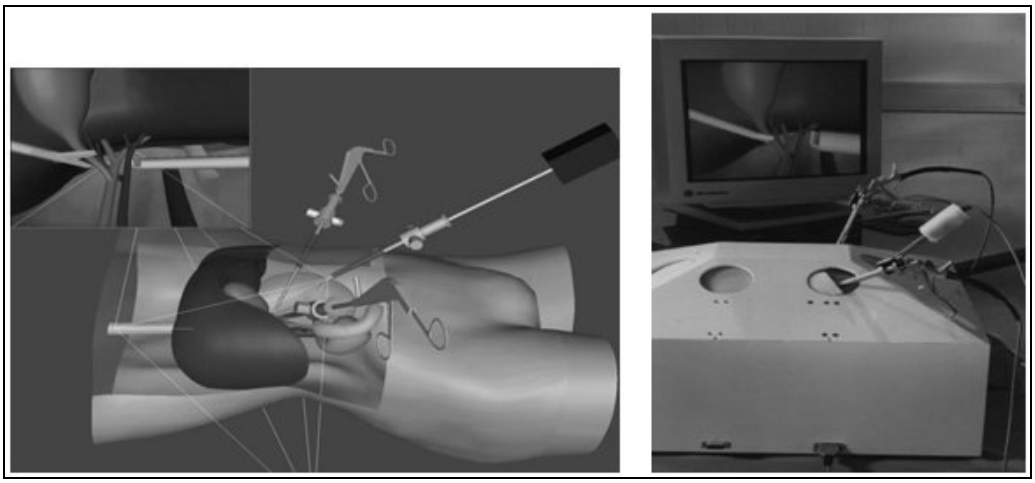


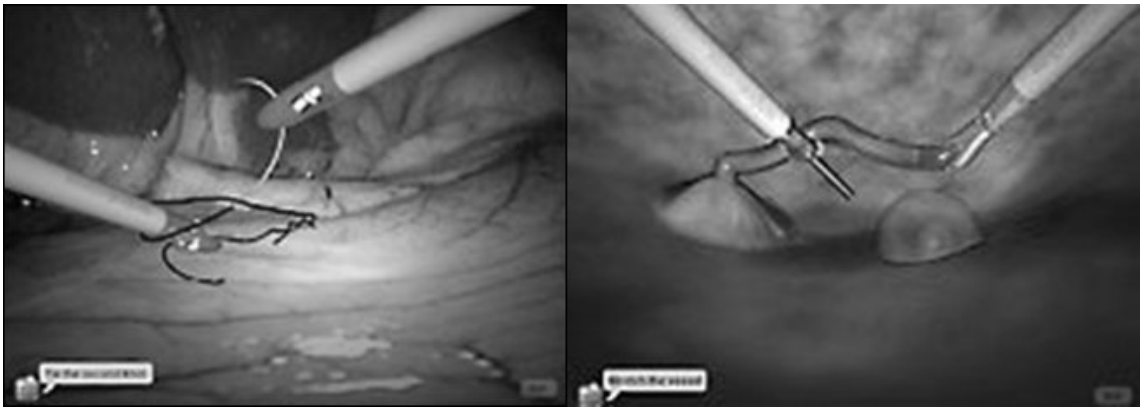
Figura 7 – Simulação médica com suporte do dispositivo KISMET (KISMET, 2003)

O *KISMET* planeja, simula, programa e monitora tarefas tele-operacionais visualizando diversas cenas ao mesmo tempo e permite interação com riqueza de detalhes (KISMET, 2006).

Já o sistema *LapSim* ajuda no treinamento de cirurgias laparoscópicas que substituem o paciente real por um virtual. Implementando os procedimentos e o ambiente de cirurgia, o sistema provê uma experiência de aprendizagem efetiva ao usuário.

O sistema inclui interatividade de vídeo ao vivo, provendo, assim, um ambiente virtual realístico para estudante. A interface do sistema é simples, mas é possível variar a complexidade conforme a necessidade de treinamento de cirurgia. Outra característica interessante é que qualquer tipo de treinamento pode ser criado e modificado por professores para ajustar às necessidades específicas do aluno.

O sistema é dividido em três módulos: habilidades básicas como corte e sutura, procedimento de dissecação e cirurgias ginecológicas, apresentados nas Figuras 6, 7 e 8 respectivamente.



(a) **(b)**

Figura 8 – a) Sutura e b) Corte (LAPSIM SYSTEM, 2006)

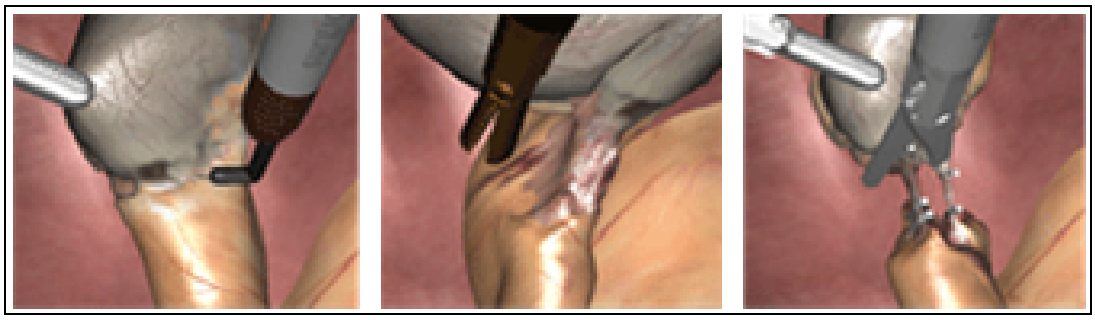


Figura 9 – Etapas da dissecação (LAPSIM SYSTEM, 2006)

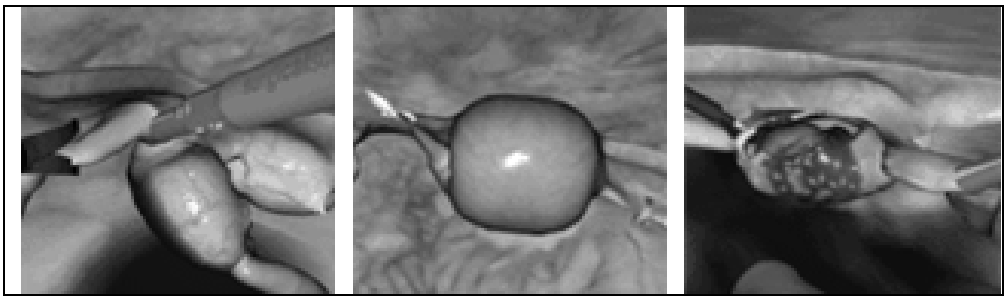


Figura 10 – Cirurgia Ginecológica (LAPSIM SYSTEM, 2006)

2.3 Projetos Desenvolvidos no LApIS

O LApIS (Laboratório de Aplicações de Informática na Saúde) do UNIVEM – Centro Universitário Eurípides de Marília envolve pesquisas relacionadas com a área médica, tendo como um dos objetivos o desenvolvimento de aplicações para treinamento médico de

baixo custo. Algumas pesquisas já foram concluídas e publicadas em forma de dissertações de mestrado ou de artigos sendo descritas a seguir.

Lima e Nunes (2004) desenvolveram um protótipo para punção de mama, com a finalidade de oferecer aos usuários a possibilidade de treinar as tarefas relativas a este exame por meio de técnicas de RV. A Figura 11 mostra a interface da ferramenta, que disponibiliza duas imagens obtidas através de exame mamográfico e objetos 3D representando uma mama e uma seringa, compondo a cena para o treinamento. O protótipo informa ao usuário quando houve a colisão, ou seja, quando a agulha atingiu o eventual nódulo e as imagens são armazenadas em um banco de dados, cujo conteúdo foi fornecido por especialistas em mastologia.

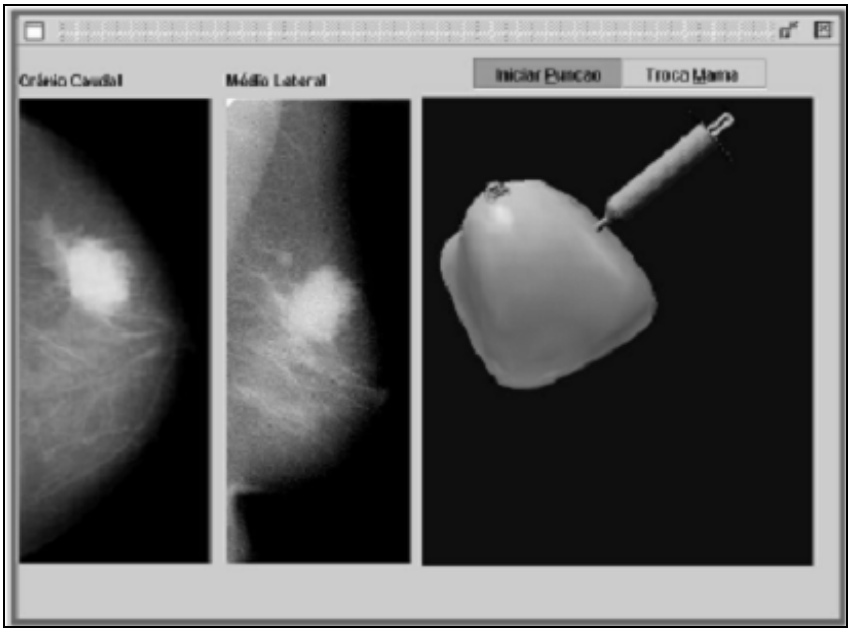


Figura 11 – Protótipo para punção de mama (LIMA e NUNES, 2004)

Riquelme (2004), a partir da proposta de Lima e Nunes (2004), desenvolveu estudos relacionados com a detecção de colisão entre os objetos 3D compostos na cena. O trabalho apresenta uma análise comparativa dos métodos disponíveis na API Java 3D para detecção de colisão e propõe um novo algoritmo que proporciona precisão e rapidez, características necessárias em ferramentas para treinamento médico.

Já Berti e Nunes (2004) desenvolveram a ferramenta *VRVis* que possibilita a visualização de bases de imagens médicas utilizando grandes volumes de informações abstratas a partir de técnicas de RV. Foram estudados os conceitos de Visualização de Informação e suas aplicações em várias áreas, a fim de apresentar as possibilidades de representação da informação da forma mais intuitiva possível para o usuário.

O projeto de Hermosilla e Nunes (2004) foi relacionado à implementação de uma ferramenta para construção dinâmica de estruturas de feto. Para a realização do projeto, pesquisas relacionadas a características das estruturas fetais foram feitas, a fim obter os dados necessários (medidas de estruturas fetais) para proporcionar a geração dinâmica dos objetos. A Figura 12 apresenta a interface e uma caixa de diálogo com o código em VRML gerado a partir dos parâmetros fornecidos. A Figura 13 mostra um comparativo entre as imagens do ultra-som convencional (2D), de um ultra-som 3D e a imagem gerada pela ferramenta.

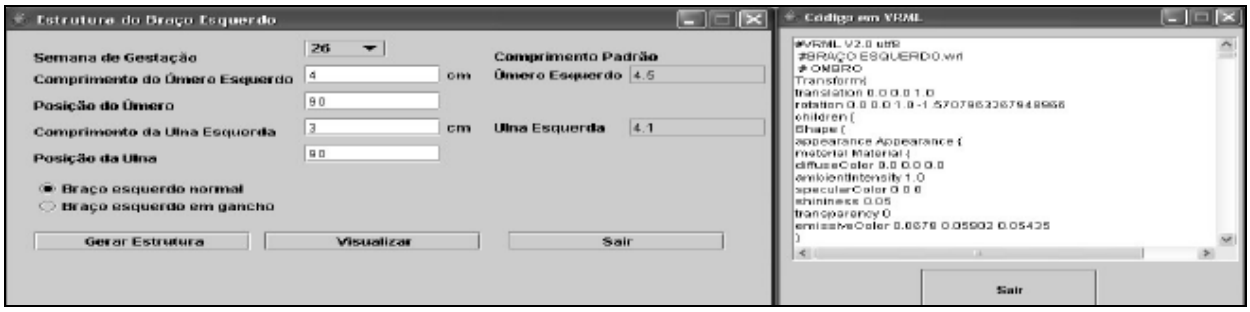
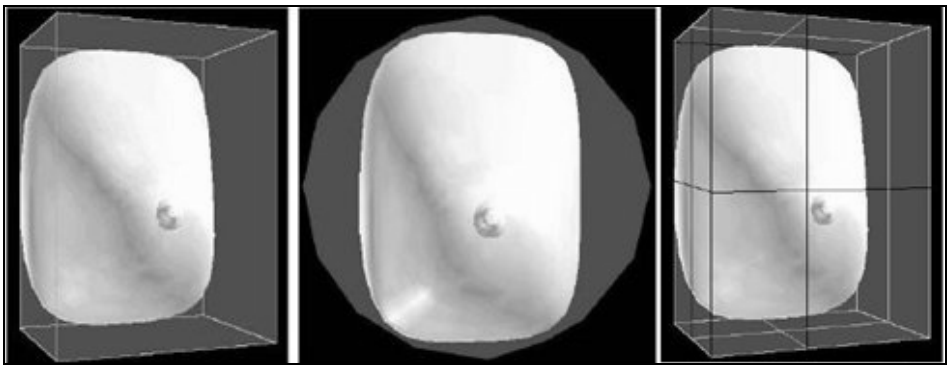


Figura 12 – Telas da interface indicando passagem dos parâmetros e criação do código em VRML (HERMOSILLA, 2004)

Figura 13 – Comparativo das imagens 2D e geração do feto 3D. (HEMORSILLA, 2004)

Em aplicações que envolvam a detecção de colisão são de extrema importância a precisão e desempenho em tempo real para simular com realismo os procedimentos do usuário. Kera *et al.* (2005) apresentaram uma análise dos algoritmos de detecção de colisão nativos da API Java3D e a construção de um método novo que proporciona detecção de colisão, utilizando também o protótipo de Lima e Nunes (2004) como estudo de caso. Foi desenvolvido um método que usa o conceito de *Octrees* para refinar os resultados dos métodos *BoundingBox* (BB) e *BoundingSphere* (BS), fornecidos pela API Java 3D, conforme mostra a Figura 14.



(a) **(b)** **(c)**

Figura 14 – Representação de bordas para detecção de colisão: (a) método *BoundingBox* (b) método *BoundingSphere* (c) conceito de *Octress* (KERA *et al.*, 2005)

Botega *et al.* (2005) lembram que a estereoscopia faz com que uma pessoa tenha sensação de profundidade ao observar uma cena. Usando o mecanismo de visão binocular, o cérebro humano trata imagens bidimensionais para conseguir tal efeito, sendo que cada olho vê uma imagem, definindo, assim, a visão estéreo.

A partir dessas definições, os autores desenvolveram um método utilizando anaglifos, para possibilitar a sensação de que o modelo 3D está em relevo em relação ao plano inicial no Ambiente Virtual.

A interação do usuário com o objeto permite acessar as visões do olho esquerdo e direito e a junção das duas visões, conforme apresenta o exemplo da Figura 15.

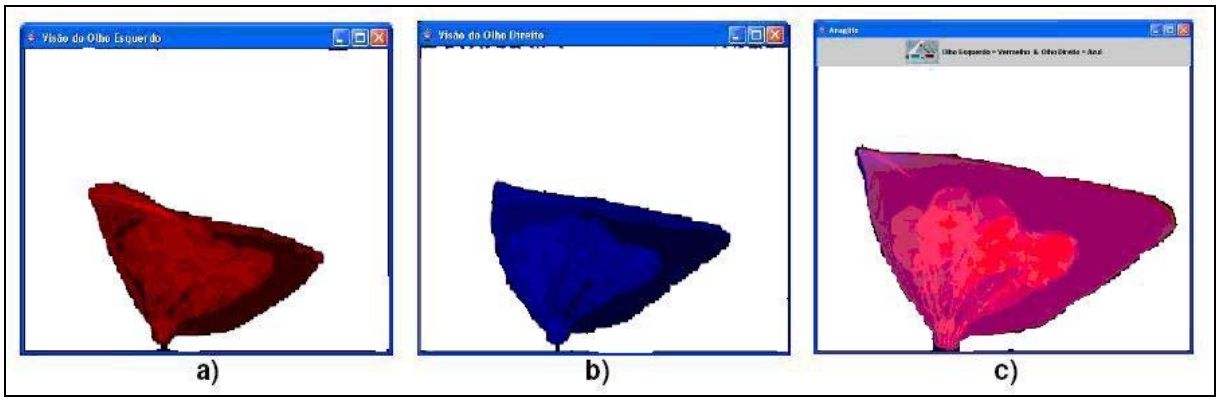


Figura 15 – (a) Representação da visão do olho esquerdo, (b) Representação da visão do olho direito e (c) Exemplo de representação do Anaglifo com visão frontal (BOTEGA *et al.* 2005)

Foi construído, ainda, um Atlas de anatomia e fisiopatologia do câncer de mama (RAMOS, 2005), com o objetivo de contribuir para o ensino de anatomia e fisiologia da mama para estudantes de Medicina que, convencionalmente, é realizando através de livros e cadáveres.

A Figura 16 e Figura 17 mostram, respectivamente, a interface do projeto e um exemplo de desenvolvimento de um câncer em diferentes estágios.

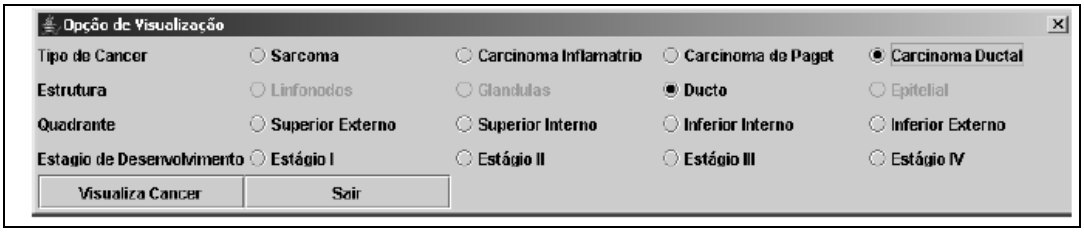


Figura 16 – Interface do Atlas (RAMOS, 2005)

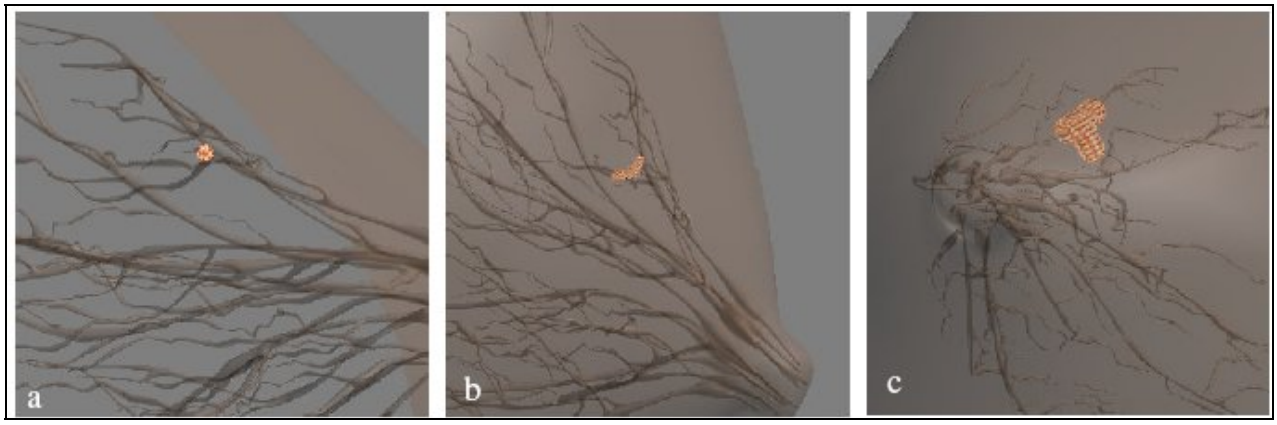


Figura 17 – Exemplo de desenvolvimento do câncer em diferentes estágios: a) Estágio I, b) Estágio II e c) estágio III (RAMOS, 2005)

2.4 Conclusão

Este capítulo traz conceitos relacionados à RV aplicados à Medicina e procedimentos MI, com o objetivo de apresentar projetos desenvolvidos relacionados com o objetivo do presente trabalho. A RV tem estado presente em treinamento de cirurgias e exames para ajudar alunos e profissionais da área médica na aquisição de experiência antes de realizar procedimentos de intervenção em pacientes reais. O próximo capítulo abordará conceitos relacionados a deformações de modelos 3D, fornecendo a base teórica do método implementado neste trabalho.

3. DEFORMAÇÃO – CONCEITOS E APLICAÇÕES

Este capítulo apresenta conceitos sobre a criação de um AV e técnicas de deformação, ressaltando principalmente como estas funcionam e onde se aplicam. Serão apresentadas também algumas aplicações de RV que utilizam essas técnicas.

3.1 Ambiente Virtual e Deformação

O termo mundo virtual é usado para denotar o mundo digital criado a partir de técnicas de computação gráfica. Uma vez sendo possível interagir e explorar esse mundo por meio de dispositivos de Entrada e Saída (E/S), ele se transforma em um AV ou ambiente de Realidade Virtual (VINCE, 1995).

Segundo Bishop (1992), para o desenvolvimento de um AV é necessário um estudo detalhado, a fim de definir percepção sensorial, hardware, software, interface com o usuário, fatores humanos e os requisitos para a aplicação a ser desenvolvida. Kirner (1996) lembra que é necessário, ainda, ter certo domínio sobre dispositivos não convencionais de E/S, computadores de alto desempenho, sistemas paralelos e distribuídos, modelagem geométrica tridimensional, simulação em tempo real, navegação, detecção de colisão, deformação, avaliação, impacto social e projeto de interfaces.

Um dos objetivos da RV é possibilitar a realização de experiências, treinamentos e o desenvolvimento de habilidades que sejam transferíveis para o mundo real. Neste sentido, o uso de AVs com alto grau de realismo desempenha um papel importante, tornando as

aplicações mais próximas da experiência cotidiana de seus usuários (HUFF *et al.* 2004 *apud* DEBEVEC, 1998).

Dependendo do tipo de aplicação desenvolvida, é necessário maior grau de realismo para permitir ao usuário a imersão e maior interação com o ambiente. Este realismo está estreitamente relacionado com os objetos modelados.

Usualmente, um objeto é representado tridimensionalmente por sistemas de coordenadas espaciais. Para que se tenha um determinado grau de realismo, vários pontos devem ser observados. Um dos principais é o uso da iluminação natural no processo de geração das cenas sintéticas por meio de técnicas de computação gráfica (HUFF *et al.* 2004).

Além da luz, outros aspectos importantes são a detecção de colisão e a deformação. Machado (2003) define a detecção de colisão como a etapa da modelagem física que permite registrar as interações entre objetos no mundo virtual, sendo que na simulação uma resposta deve ser emitida ao encontro dos objetos.

Em simulação de aplicações de procedimentos médicos, além da detecção de colisão, é importante saber onde ocorreu o impacto entre os objetos modelados e se houve ou não uma interpenetração entre eles, podendo causar uma possível deformação. Os conceitos e utilização da deformação serão mais detalhados a seguir.

3.2 Conceitos de Deformação

Schneider (1998) afirma que uma cena de um AV é composta de vários objetos, podendo estes ser estáticos ou dinâmicos.

Os objetos estáticos mantêm a mesma forma durante toda sua existência no AV, ou seja, não têm interação com outros objetos em cena. Os objetos dinâmicos são os que têm

interação dentro da cena, ou seja, são aqueles que mudam de posição, forma ou aparência entre um quadro e outro (SILVA *et al.* 2004).

No mundo real, a maioria dos objetos tem característica dinâmica. Para que um AV seja realístico, é necessário modelar esses objetos da forma mais próxima ao cotidiano, o que envolve definir suas ações e reações.

Uma das reações a serem analisadas é a deformação de objetos, pois estes podem ter sua forma alterada em conseqüência de estímulos diversos que são capazes de modificar sua aparência. Objetos que compõem uma cena virtual e que sofrem a deformação são muito estudados em computação gráfica, animação e RV (COLAVITTI, 2004; COSTA e BALANIUK, 2001). Áreas como engenharia, entretenimento e *design* de ambientes têm utilizado objetos deformáveis para criar e editar superfícies de sólidos complexos. As aplicações variam desde simulação de prospecção e extração de petróleo, simulação de desenvolvimento de carros, simulação de visitas a museus virtuais até simuladores de procedimentos médicos.

Sendo assim, a deformação pode ocorrer em qualquer tipo de aplicação simuladora podendo ser desenvolvida por animação 2D e 3D ou por interação do usuário em tempo real (SCHNEIDER, 1998).

Manssour (2003) lembra que a animação é o ato de induzir a ilusão a partir do seqüenciamento rápido de imagens estáticas. As etapas da animação estão divididas em modelagem tridimensional do objeto, aplicação de textura para definição dos movimentos que vão indicar qual o trajeto do objeto, qual sua deformação, sua velocidade e todos os demais dados necessários para criar a ilusão de um movimento natural, no entanto é importante observar que a animação é pré-determinada no programa, ou seja, os valores relacionados a estes atributos citados acima são estáticos.

A deformação por interação possui as mesmas características da deformação por animação, sendo diferente apenas na sua geração, pois nesse caso ocorre devido à intervenção de um usuário em tempo real e não é pré-determinada, como ocorre na animação.

Como tem crescido a capacidade de processamento do hardware, objetos deformáveis começam a aparecer com maior freqüência em aplicações de RV, devido à necessidade para simular a reação ao encontro entre dois ou mais objetos (GLADILIN, 2003).

Na literatura três técnicas são as mais citadas para se obter deformação de objetos em aplicações de RV: *Free Form Deformation* (FFD) ou Deformação de Livre Forma (GIBSON E MIRTCH, 1997), *Mass Spring* (MS) ou Massa Mola (ZILL, 2003) e *Finite Element* (FE) ou Elemento Finito (GLADILIN, 2003). As próximas seções apresentam a explicação de cada uma dessas técnicas.

3.3 Técnicas de Deformação

A deformação de objetos 3D pode ser implementada através de modelos que permitem a manipulação desses objetos, sendo que o comportamento dos mesmos é representado por uma simulação que pode ser definida como geométrica, física ou híbrida (SCHNEIDER, 1998).

De acordo Choi *e*t al. (2002), a simulação geométrica, é baseada em manipulações de dados geométricos como vértices e pontos, que representam dobras e ondulações através de equações geométricas. Usa artifícios matemáticos para obter resultados visualmente convincentes que permitem mudança na estrutura do objeto. Utiliza a cinemática, ramo da física que procura descrever os movimentos sem se preocupar com as forças que originam estes movimentos (RAMALHO *et al.*, 1993).

A simulação física baseia-se em leis da física para obter a forma do objeto sob certas condições, envolvendo massa, aceleração e o comportamento dos objetos sob efeito de alguma força física, seja ela interna ou externa (MACHADO, 2003). Utiliza a dinâmica, ramo da física que estuda as relações entre as forças e os movimentos que são produzidos por estas, ou seja, se preocupa com a causa e a conseqüência dos estímulos causados (RAMALHO *et al.*, 1993).

Segundo Schneider (1998), a simulação física exige um grande processamento e seu principal objetivo é que o objeto deformável simule o mesmo comportamento de um objeto real. A simulação híbrida é um comportamento que mistura as técnicas geométrica e física.

3.3.1 *Free Form Deformation* (FFD)

Free Form Deformation (FFD), também conhecida como Deformação de Forma Livre, é uma técnica geométrica que provê um alto nível de ajuste e controle individual dos pontos do objeto. A técnica muda a estrutura do objeto deformando o espaço em que ele se encontra (GIBSON e MIRTCH, 1997). Dado um objeto carregado na cena, o mesmo é envolvido por uma grade 3D que sofre deformação; por conseqüência, o objeto deformará dependendo da deformação ocorrida na grade. A Figura 18 apresenta como a deformação é feita com este método.

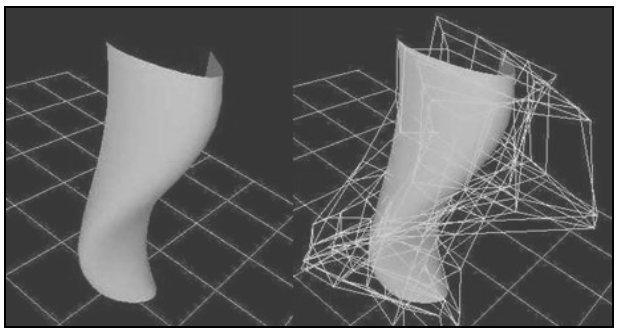


Figura 18 – Exemplo de deformação com o método *Free Form Deformation* (GIBSON e MIRTCH, 1997)

Sederberg (1986) afirma que esta técnica não é intuitiva, pois consiste em envolver os objetos deformados em um simples paralelepípedo para se ter o controle dos seus pontos. Estes pontos são manipulados para deformar o espaço circunvizinho e induzir deformações nos objetos.

Choi *et al.* (2002) lembram que os movimentos de controle dos pontos e as correspondentes mudanças na estrutura dos objetos devem estar ligados a fim de se obter as deformações desejadas.

Na modelagem geométrica, freqüentemente, deseja-se deformar um modelo aplicando força nos nós através de dobras, torção, comprimindo e esticando alguma parte do modelo. Isso pode ser feito modificando-se a própria representação geométrica. Uma abordagem diferente é a transformação espacial. É possível aplicar uma transformação espacial para o sistema de coordenadas, mapeando-se cada posição antes da deformação (x,y,z) para a posição obtida depois da deformação das coordenadas da função φ(x,y,z) (HIROTA *et al.*, 1999). A função φ é definida como uma combinação linear de pontos de nós de largura X, indicada na equação 1 (HIROTA *et al.*, 1999).

$$\varphi(x,y,z) = \sum_{I=1}^{n} b_1(x,y,z) X_I$$

(1)

Mesmo sendo versátil, Choi *et al.* (2002) afirmam que neste método é difícil limitar as deformações para pequenas regiões porque somente os objetos embutidos dentro do paralelepípedo são deformados globalmente. Afirmam ainda, que o processo é considerado tedioso, pois sempre que são necessárias modificações subseqüentes é preciso especificações explícitas para se aplicar deformações, ou seja, caso a deformação for requerida em objetos pequenos, é necessário se estudar como será deformada a parte desejada.

Esta abordagem permite que o método se torne versátil e a deformação seja definida independente da representação de geometria, como mostra a Figura 19.

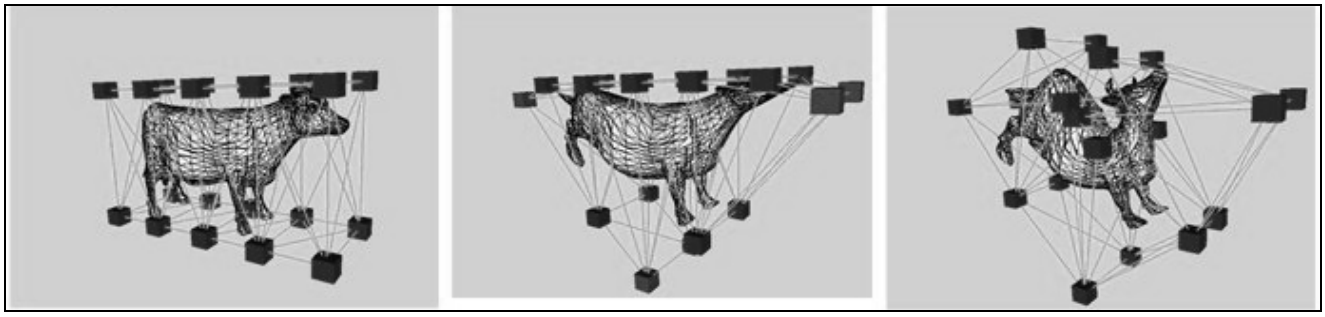


Figura 19 – Exemplo de objeto deformado utilizando o método FFD (HIROTA *et al.*, 1999).

Como o método FFD desconsidera propriedades físicas para deformação de objetos, essa técnica acaba sendo limitada e custosa para o desenvolvedor, pois não permite a manipulação dos objetos que compõem a cena, não sendo então uma técnica satisfatória para aplicações de RV. De acordo com Choi *et al.* (2003) e Gibson e Mirtch (1997) esse método não traria realismo à cena quando empregado em um treinamento cirúrgico.

Sendo assim esse método de deformação não seria bem aplicado ao projeto em questão, pois as regiões para deformação são pequenas e minuciosas e uma grade 3D envolvendo o objeto não traria resultados precisos que é um dos objetivos do projeto.

3.3.2 *Finite Element Method* (FEM)

O método dos elementos finitos ou *Finite Element Method* (FEM), originou-se da necessidade de resolver elasticidade complexa e análise de estruturas. Sua característica principal é a discretização da malha de um domínio contínuo em um conjunto de subdomínios discretos. Matematicamente é um método usado para descobrir a solução da aproximação de

equações diferenciais parciais, como também de equações de integrais e equações de transporte de calor (WIKPÉDIA, 2006).

A discretização do domínio é o primeiro e talvez o passo mais importante em qualquer análise de elementos finitos, pois a maneira na qual o domínio é discretizado afetará as exigências de armazenamento no computador, o tempo de processamento e a precisão dos resultados numéricos.

Para um domínio 1D, que é de fato uma reta de linha curvada, são freqüentemente os elementos de segmentos de linha curtos que se interconectaram para formar a linha original, Figura 20(a). Para um domínio 2D domínio, os elementos são normalmente pequenos triângulos ou pequenos retângulos Figura 20(b). Os elementos retangulares são melhores para serem discretizados em regiões regulares, enquanto os triangulares podem ser usados para regiões irregulares. Em uma solução 3D, o domínio pode ser subdividido em tetraedros, prismas triangulares, ou cubos apresentados na Figura 20(c), onde os tetraedros são os mais simples e de melhor visualização de domínios arbitrários (HIROTA *et al.*, 1999).

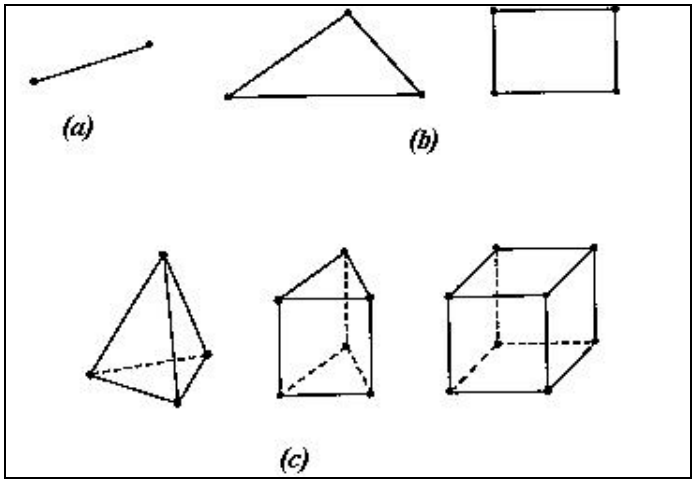


Figura 20 – Elementos Finitos básicos - (a) 1D, (b) 2D e (c) 3D (HIROTA *et al.*, 1999)

É uma técnica baseada na física usada para encontrar uma aproximação para que uma função contínua satisfaça a deformação. Essa função contínua é dividida em elementos que se aproximam em uma equação de equilíbrio para cada elemento, solucionando assim a restrição

dos pontos do objeto e as bordas de cada elemento do objeto, de forma que a continuidade dos elementos seja alcançada (HIROTA *et al.,* 1999).

O FEM utiliza a teoria da elasticidade que afirma que a deformação de um corpo é proporcionar a tensão aplicada desde que o limite elástico do material não seja excedido. Normalmente, o comportamento elástico dos materiais segue o regime elástico na lei de *Hooke*[1] (ZILL, 2003) apenas até um determinado valor a partir do qual a relação de proporcionalidade deixa de ser definida. Posteriormente, se essa força continuar a aumentar atinge-se um outro patamar em que o corpo perde a sua elasticidade e a deformação passa a ser permanente, chegando-se finalmente ao ponto em que o material não resiste e rompe (WIKIPÉDIA, 2006).

Segundo Zienkiewicz (1977), que constitui o instrumento básico da análise estrutural. Efetivamente, a análise de qualquer tipo de estrutura pode ser feita pela integração das equações da teoria da elasticidade. As dificuldades conhecidas nessa integração levam, porém, à formulação de modelos matemáticos aproximados, adequados ao tratamento de tipos particulares de estruturas. Tais modelos matemáticos, ainda contínuos, são formulados em termos de variáveis generalizadas e referem-se geralmente a domínios com menor número de dimensões do que o domínio tridimensional da teoria da elasticidade.

Essa técnica está relacionada ao estudo discretizado de objetos deformados dentro de um conjunto de elementos, onde um objeto deformado é representado pelo deslocamento dos nós e é realizada uma soma finita de interpolação para cada elemento. O discretização do domínio é normalmente considerado como uma tarefa de pré-processamento, pois pode ser separado completamente de outros passos. Foram desenvolvidos pacotes utilizando o método de elemento finito que têm a capacidade de subdividir uma linha, uma superfície e o volume

[1] A lei de Hooke é a lei da física relacionada à elasticidade de corpos, que serve para calcular a deformação causada pela força exercida sobre um corpo, fazendo a força igual o deslocamento da massa a partir do seu ponto de equilíbrio multiplicando pela característica constante da mola ou do corpo que sofrera deformação, ou seja: F=X.K. (ZILL, 2003)

dos elementos correspondentes e também provêem o numerando global discretizado de elemento de domínios 2D e 3D. A Figura 21 apresenta exemplos de elementos finitos discretizados (CHOI *et al.*, 2003).

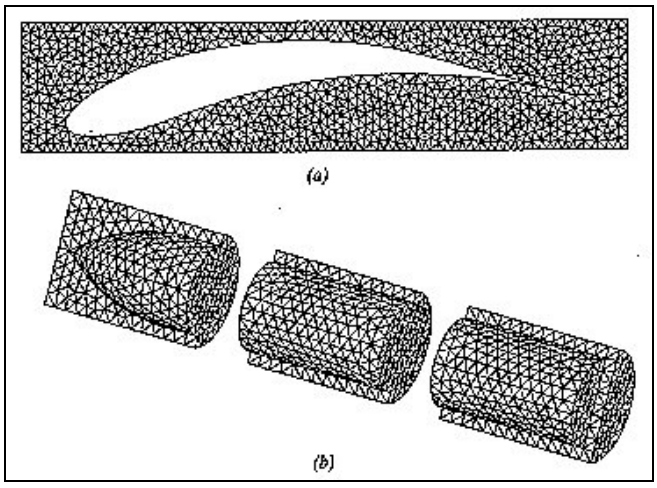


Figura 21 – Exemplo de discretização de elementos finitos – (a) 2D com elementos triangularizados e (b) 3D com elementos tetraedros. (HIROTA *et al.*, 1999)

Segundo Bro-Nielsen (1988), a interpolação da estrutura da função é usada para modelar a variação contínua do deslocamento dentro de cada nó do elemento. Para se obter precisão, a deformação do modelo é realizada com a utilização de cálculos matemáticos para acelerar o processamento.

Hirota *et al.* (1999) citam que um exemplo de cálculo utilizando a técnica FEM é dado quando se deseja achar a função escalar $\phi(x,y,z)$ de um objeto. O valor de ϕ no ponto (x,y,z) é aproximado pela equação 2.

$$\boxed{\phi(x,y,z) \cong \sum h_i(x,y,z)\phi_i} \qquad (2)$$

A variável h_i é a função de interpolação para elementos contendo (x,y,z) e ϕ_i é referente aos valores de $\phi(x,y,z)$ dos pontos do elemento do nó. Resolvendo-se a equação de equilíbrio, determina-se o conjunto finito dos valores ϕ_i dos nós i que minimiza a energia potencial total do objeto.

Para se resolver um FEM existe cinco passos básicos, segundo Hirota *et al.* (1999):

1) deriva-se uma equação de equilíbrio da equação de energia potencial em termos de deslocamento de material;

2) selecionam-se elementos finitos apropriados e funções de interpolação correspondentes. Subdivide-se o objeto em elementos;

3) para cada elemento, reescrevem-se os componentes da equação de equilíbrio em termos de funções de interpolação e os deslocamentos do nó do elemento;

4) combina-se O conjunto das equações de equilíbrio para todos os elementos em um único sistema e resolve-se o sistema para os deslocamentos do nó para o objeto inteiro;

5) usam-se os deslocamentos de nó e a função de interpolação de um elemento particular para calcular deslocamentos para pontos no interior do elemento.

A vantagem em se utilizar este método é para resolver equações diferenciais parciais com domínios complexos ou quando se deseja grande precisão no domínio da aplicação

Essas vantagens fazem com que o método geralmente tenha um custo de processamento alto para projetar um protótipo. Embora uma prova empírica para a análise modal que testa e medidas de resposta dinâmicas, por exemplo, tem que ser executada para verificar os resultados, utilizando o método de elemento finito provê uma conta bastante segura de como uma estrutura ou um componente se comportará sob uma tensão. Cálculos grandes como a estrutura de edifício e tão pequeno quanto um componente de um dispositivo de medida pode ser modelada com precisão suficiente e com suposições razoáveis utilizando esse método (HIROTA *et al.*, 1999).

Apesar de ser uma técnica que possibilita cálculos minuciosos para gerar deformação de um objeto o método FEM, tem suas limitações e desvantagens. Uma delas está ligada às exigências computacional. Em particular, a técnica tem apresentado dificuldade de ser aplicada em sistemas de tempo real, pois os vetores que guardam valores referentes à força e à massa e a matriz que armazena valor de rigidez são calculados integrando-se com objeto,

sendo assim esses parâmetros citados devem, teoricamente, ser reavaliados para saber como o objeto se deforma. Esta reavaliação é muito cara, em termos de processamento, e é evitada quando os objetos sofrem deformações pequenas (GLADILIN e MIRTCH, 2003).

3.3.3 *Mass Spring* (MS)

Mass Spring (MS), ou também conhecido como massa-mola, é uma técnica baseada na física que permite a remodelagem de objetos deformados através de nós de massa conectados por *springs* (molas), como mostra a Figura 22 (GIBSON e MIRTCH, 1997).

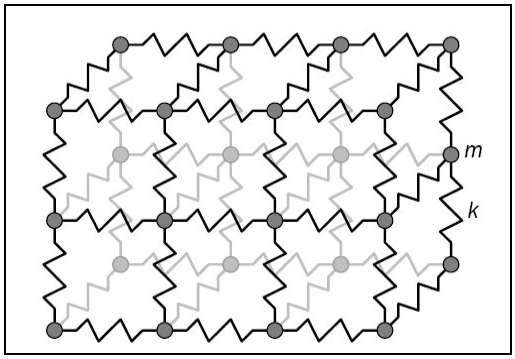


Figura 22 – Molas conectadas a um ponto de massa que exercem forças nos pontos vizinhos quando a massa é deslocada do resto das posições (GIBSON E MIRTCH, 1997)

Choi *et al.* (2002) afirmam que a técnica discretiza objetos deformados dentro de nós que são conectados de modo elástico ao objeto a ser deformado. É uma técnica que envolve formulação de matrizes e sistemas de equações diferenciais governadas por modelos dinâmicos para prevenir a instabilidade numérica (KÜHNAPFEL *et al.* 2000; GÜDÜKBAY, 1997).

Segundo Hornby (1995), a mola pode ser considerada como uma estrutura que possui elasticidade, que permite a deformação quando uma pressão é exercida nela. A força de uma mola é normalmente linear, baseada na lei de *Hooke* (ZILL, 2003), mas molas não lineares

podem ser usadas para simularem tecidos como pele humana, que exibem um comportamento não elástico.

Para se obter a formulação da força da mola é preciso considerar a deformação do objeto sob uma rede de nós de massa conectados por molas (CHOI *et al.,* 2002), apresentados na Figura 23 . Cada mola assumida obedece à lei de *Hooke* (ZILL, 2003) e cada movimento é contido por um amortecimento da força proporcional a sua velocidade.

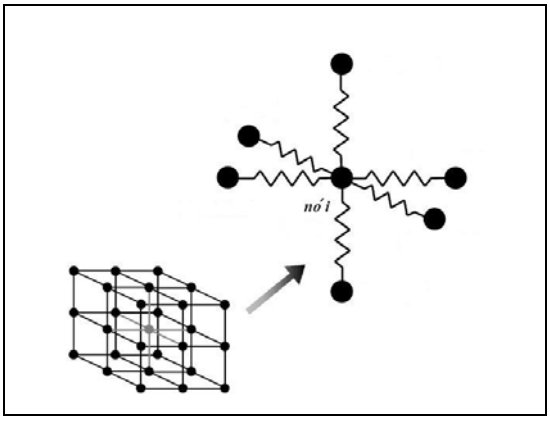


Figura 23 – Nó central (*i*) conectado aos seus vizinhos através de molas (CHOI *et al.*, 2002)

Em sistemas dinâmicos, a segunda lei de Newton governa o movimento de um único ponto de massa na rede. O funcionamento geral da técnica MS é dado pela equação 3 explicada a seguir.

$$m_i \ddot{x}_i = -\gamma_i \dot{x}_i + \sum_j g_{ij} + f_i \tag{3}$$

Segundo Gibson e Mirtch (1997), o lado esquerdo da equação 3 representa respectivamente massa do ponto (m_i) e sua posição ($\ddot{x}_i$), que é a segunda derivada da equação da aceleração de Newton. O lado direito é referente à atuação da força sobre o ponto da massa, onde $-\gamma_i \dot{x}_i$, é a derivada da velocidade dependente do amortecimento da força, g_{ij} é a força exercida na massa *i* pela mola entre massas *i* e *j* e f_i é o somatório das outras forças externas agindo na massa m_i.

As equações de movimento são agrupadas por um sistema para permitir a movimentação de todos os pontos de massa da rede. Concatenando os vetores de posição de massa individual dentro de uma simples posição 3D do vetor x, é obtida a equação 4:

$$M(\frac{d^2X}{dt^2}) + C(\frac{dX}{dt}) + Kx = f \tag{4}$$

Neste caso *M*, *C* e *K* são matrizes de massa, amortecimento e constante da mola. O vetor *f* é um vetor tridimensional que representa as forças externas totais no ponto de massa. Desenvolvendo a equação 4 como um sistema de equações diferenciais de primeira-ordem, obtém-se o resultado apresentado nas equações 5 e 6.

$$(\frac{dV}{dt}) = M^{-1}(-Cv - Kx + f) \tag{5}$$

$$(\frac{dX}{dt}) = v \tag{6}$$

Hirota *et al.* (1999) lembram que a técnica é de fácil compreensão e implementação permitindo interatividade e simulação em tempo real, mas sua utilização fica difícil quando se deseja modelar propriedades muito complexas com grandes quantidades de vértices, pois traz uma sobrecarga na memória fazendo com que o processamento e a manipulação desses objetos na cena fiquem lerdos e não traga resultados satisfatórios.

Exemplos de utilização do método são apresentados em Waters (1987; 1992), que descreve um modelo de animação 3D para expressões faciais, utilizando modelagem física para tecidos da face; Delingette *et al.* (1999) e Keeve (1996), que apresentam aplicações de simulações de cirurgias crânio-faciais e Kühnapfel *et al.* (2000), que descrevem um treinamento de cirurgia endoscópica usando simulação de deformação de tecidos.

O método massa-mola possui algumas desvantagens devido aos valores determinados para a constante da mola que compõem o objeto. Não é trivial derivar esses valores devido às propriedades medidas do material e também por existirem em alguns

obstáculos, como o caso de incompreensão de objetos volumétricos ou superfícies finas, devido à dificuldade na modelagem desses tipos de objetos (GIBSON E MIRTCH, 1997).

3.4 Aplicações na Área Médica Utilizando Técnicas de Deformação

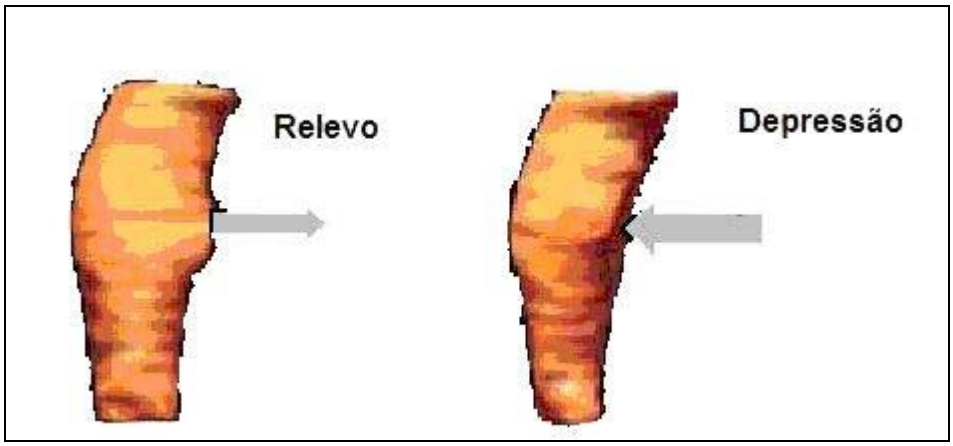
A seguir são relacionadas algumas aplicações, para treinamento médico, que utilizam as técnicas de deformação já citadas, em ordem cronológica crescente.

3.4.1 Deformação e Cortes em Cirurgias Virtuais

Shi e Xia (2001) que desenvolveram técnicas para deformação de objetos e cortes em aplicações de cirurgia virtual, apresentaram um algoritmo de reconstrução de geometria, a partir de um modelo tetraédrico com base no volume. Este modelo é baseado em propriedades físicas reais de tecidos moles, e utiliza o método de elementos finitos.

Para simulação da operação cirúrgica foi criado um algoritmo global de cortes. A modelagem das geometrias foi definida a partir de conjuntos de imagens médicas, dos quais foram extraídos os contornos da imagem. Foi reconstruída a topologia do órgão e, por último, a geometria foi gerada.

A Figura 24 mostra a deformação da parte inferior da perna humana, sendo que a Figura 24(a) representa uma deformação de relevo e a Figura 24(b) mostra uma depressão.



(a) **(b)**

Figura 24 – Deformação: (a) relevo e (b) depressão da perna (SHI e XIA, 2001)

A Figura 25 apresenta a simulação de uma cirurgia em três fases: no início, no meio e no final da incisão.

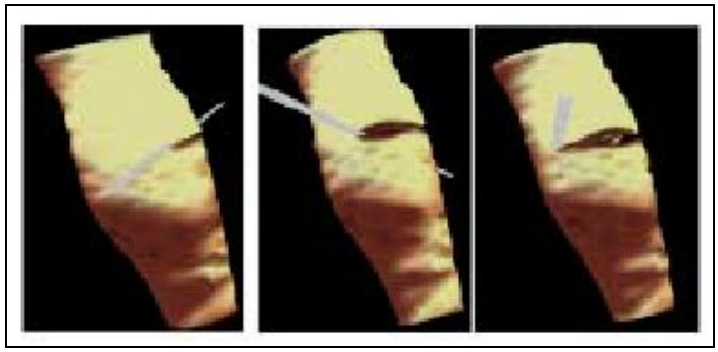


Figura 25 – Três passos da simulação cirúrgica de incisão em uma perna virtual (SHI e XIA, 2001)

3.4.2 LEM – Simulação de Tecidos Flexíveis em Tempo Real

O método LEM (*Long Elements Method*) foi desenvolvido por Costa e Balaniuk (2002) tendo o propósito de ser um novo método físico, baseado em simulações de deformação de objetos, satisfatório para animações em tempo real e interação com ambientes virtuais.

A abordagem de uma solução de implementação estática para deformações globais elásticas de objetos foi baseada no princípio de Pascal para conservação de volume do objeto,

utilizando o modelo de deformação de elementos finitos. Os volumes são discretizados em elementos, que definem uma malha em uma ordem de magnitude tetraédrica menor ou malhas cúbicas.

Foram modeladas as partes físicas dos objetos usando variáveis como: pressão, densidade, volume e tensão. A abordagem descrita foi usada para implementar um simulador genérico de tecidos flexíveis.

O simulador foi desenvolvido utilizando a linguagem de programação C++ em uma plataforma Windows NT, possuindo três módulos principais. O primeiro módulo é responsável pela definição do modelo. A geometria e a parte física do objeto são simuladas usando informações que podem ser derivadas de um segmento de dado real ou de um modelo intermediário. O segundo módulo é responsável pela repetição da simulação que resolve as equações matemáticas do sistema para obter a deformação da estrutura do objeto e o último módulo é responsável pela renderização dos objetos. A Figura 26 mostra a deformação de objetos flexíveis sendo tocados por um objeto rígido, ou seja, um objeto não deformável.

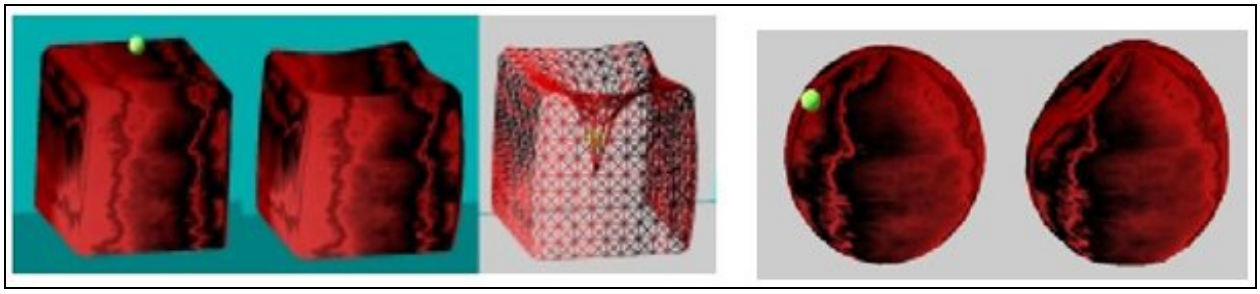


Figura 26 – Tecido flexível virtual tocado por uma esfera rígida (COSTA E BALANIUK, 2001)

De acordo com os pesquisadores, o método criado permite deformações elásticas em modelos físicos de forma que seja preservado o volume do objeto, permitindo mudanças em tempo real na sua topologia e fazendo com que os cálculos sejam rápidos. Costa e Balaniuk (2001) afirmam que o método tem duas vantagens: (a) o número de elementos que preenche um objeto é em ordem de magnitude, só não é quando há uma discretização baseada em tetraedro ou em elementos cúbicos, e (b) a interface gráfica e o feedback do dispositivo

podem ser derivados diretamente dos elementos e não é preciso intermediário geométrico (COSTA e BALANIUK, 2001).

3.4.3 Simulação de Deformação em Tecidos Flexíveis

O trabalho de Choi *et al.* (2002) apresenta um simulador de tecidos flexíveis desenvolvido com a linguagem Java, utilizando a API Java 3D, para permitir a simulação da propagação da força para simulação de deformação usando o modelo massa-mola (MS).

Objetos deformados foram discretizados em um sistema onde cada nó está conectado a nós adjacentes com *springs* (molas). Quando um nó estiver sujeito a uma força externa, essa força desloca os nós da superfície que estão próximos do ponto de contato. Este contato com o objeto, causado por uma força externa, é propagado sucessivamente em todos os nós do objeto.

Para implementar este tipo deformação, foram definidos três passos: a formulação de força da mola, a escala sucessiva de propagação da força e a seqüência da propagação da força.

A Figura 27(a) mostra a simulação da deformação em uma malha, na qual foi escolhido um vértice para aplicar uma determinada força, fornecendo como resultado a deformação de relevo. A Figura 27(b) mostra a mesma malha aplicando-se a força em dois pontos específicos, ou seja, em duas superfícies de nós. A Figura 27(c) é o resultado da deformação da força externa aplicada a um conjunto de nós da superfície, e já contendo a textura.

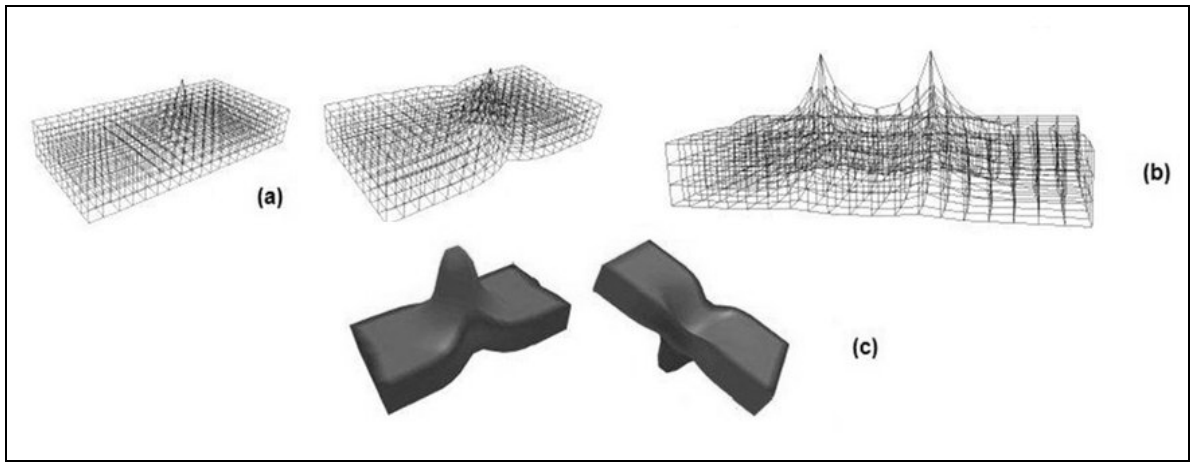


Figura 27 – (a) Deformação em um ponto na malha (b) Forças aplicadas na superfície em dois pontos específicos (c) Deformação resultante com textura (CHOI *et al.*, 2002)

A Figura 28 exemplifica a técnica aplicada em um órgão do corpo humano, apresentando a deformação final.

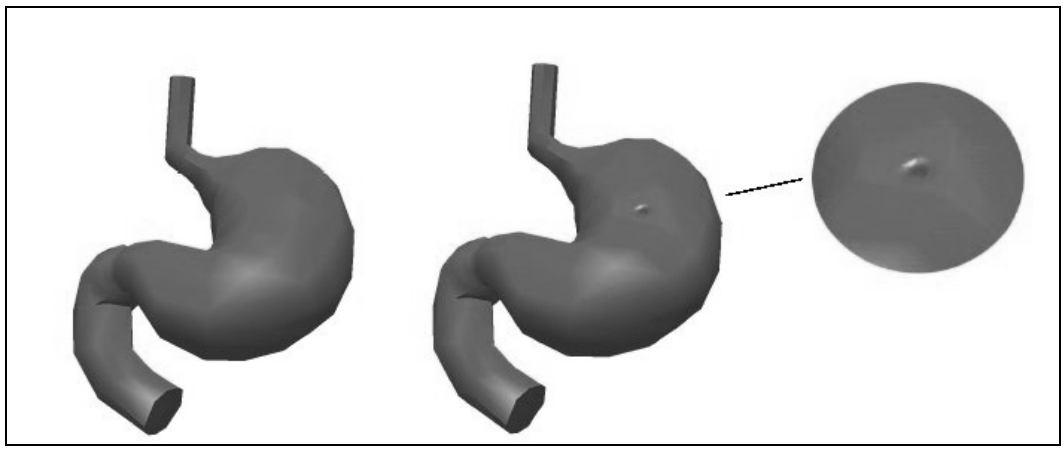


Figura 28 – Deformação localizada no objeto (CHOI *et al.*, 2002)

Choi *et al.* (2002) ainda descrevem que para visualizar o comportamento dinâmico a coordenada da superfície do modelo geométrico é atualizada com novas posições (dos nós), redefinidos após cada simulação.

O modelo foi aplicado para materiais homogêneos, nos quais os objetos são compostos de nós de treliças com a mesma massa e quantidade de molas. Choi *et al.* (2002) afirmam que os resultados obtidos pelo modelo podem ser usados para tarefas colaborativas, tal como manipulação virtual de objetos deformados.

3.4.4 Deformação Elástica para Simulação de Cirurgia Háptica

Webster *et al.* (2002) desenvolveram um simulador háptico de sutura, que consiste no uso do método massa-mola em tempo real para deformação de órgãos 3D em simuladores de cirurgias hápticas. Como descrito, o sistema é baseado no modelo de deformação massa-mola permitindo assim a velocidade no processamento dos cálculos durante cada passo realizado em uma cirurgia e para o funcionamento dos cálculos matemáticos, foi utilizada a 2ª lei de Newton para gerar o movimento dos objetos dentro da cena.

O simulador provê precisão e interatividade na deformação do órgão em simulações de cirurgia háptica e possui componentes que modelam e deformam a pele, o tecido e o material de sutura em tempo real, além de registrar todos os estados de atividade durante a tarefa.

O sistema possui quatro etapas: (1) gravidade, onde todas as massas são aceleradas; (2) a detecção de colisão, usada quando o usuário toca no órgão, causando depressão ou relevo No objeto virtual; (3) as molas, que a partir do tamanho inicial, são esticadas ou comprimidas; (4) análise de instabilidades no coeficiente numérico da força, para prevenir o movimento que as massas receberão quando uma força pequena ocorrer.

A força de contato calculada pelo simulador gera deslocamentos de tecido. Os cálculos das forças de resistentes variam de acordo a profundidade e o ângulo de inserção da agulha. As forças podem variar, conforme a necessidade de utilização da agulha na pele virtual. As Figura 29 e Figura 30 mostram o órgão no ambiente virtual, a deformação e o funcionamento da aplicação.

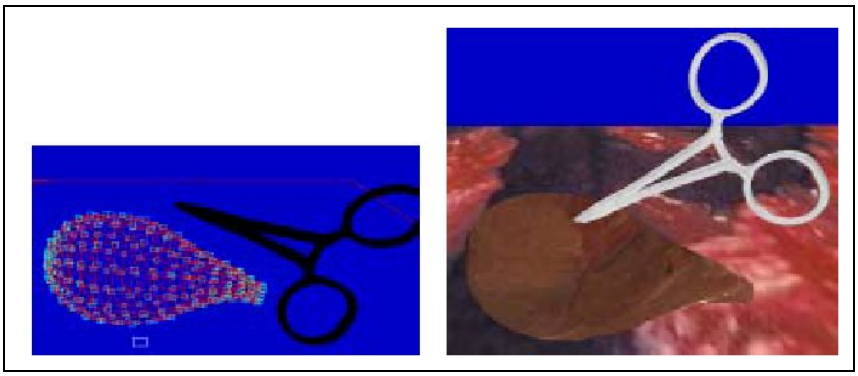


Figura 29 – (a) órgão no modelo *wireframe* e (b) possível deformação (WEBSTER *et al.*, 2002)

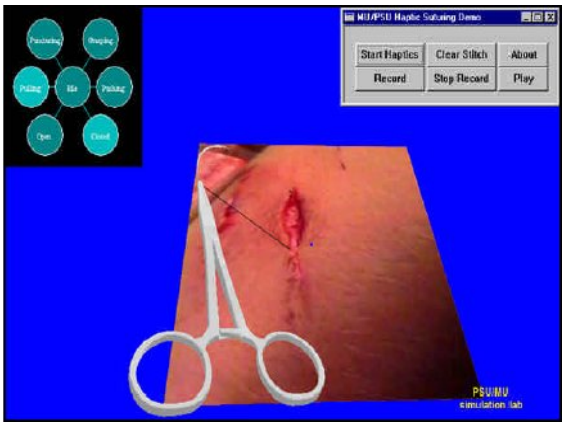


Figura 30 – Imagem de treinamento em prática de sutura sendo executada no simulador (WEBSTER *et al.*, 2002)

Webster *et al.* (2001) afirmam que com o uso do simulador, estudantes e profissionais da saúde podem cada vez mais melhorar a técnica, pois o treinamento pode ser realizado a qualquer hora sem nenhuma restrição.

3.4.5 Treinamento de Procedimentos Percutâneos

Dimaio e Salcudean (2002) fizeram pesquisas em relação a métodos que quantificam a força aplicada em um determinado objeto e qual a deformação resultante em diversos meios. Com base no método de elementos finitos, criou-se um sistema interativo que simula a inserção de uma agulha virtual em tecidos flexíveis com três graus de liberdade de

movimento, permitindo o planejamento, o treinamento de terapias e procedimentos percutâneos. A Figura 31 e a Figura 32 mostram o procedimento de inserção de uma agulha virtual.

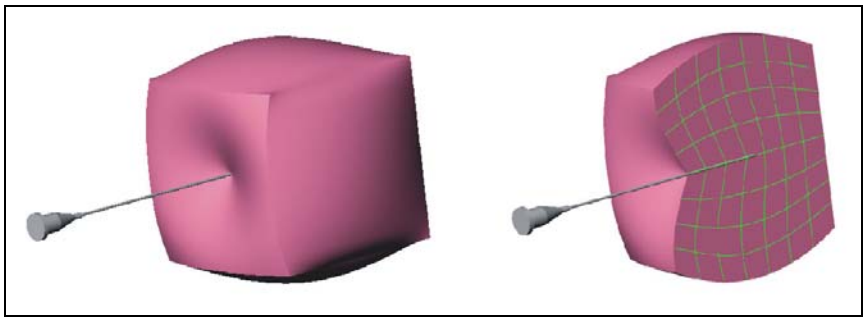
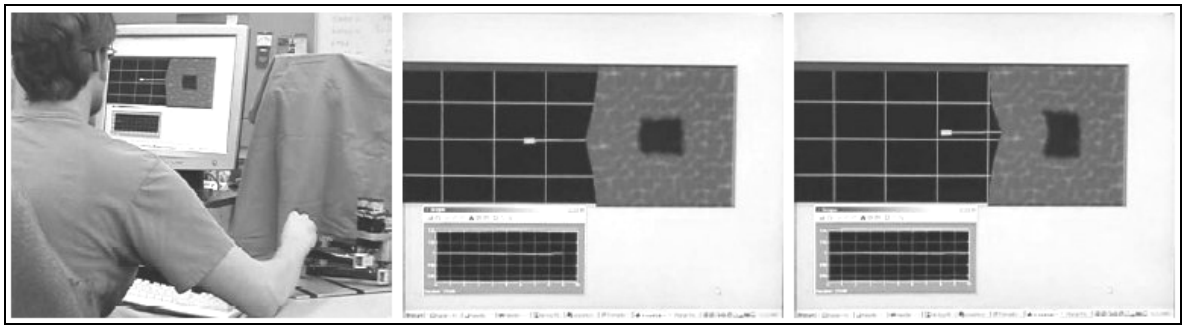


Figura 31 – Inserção de agulha em tecidos moles (DIMAIO e SALCUDEAN, 2002)



(a) **(b)** **(c)**

Figura 32 – (a) Dispositivo Háptico responsável pela interação; (b) Interação de Relevo e (c) Interação de Profundidade (DIMAIO e SALCUDEAN, 2002)

O sistema calcula a deformação do tecido durante a penetração da agulha, e um algoritmo em tempo real permite que os usuários manipulem a agulha virtual e verifiquem como esta penetra no objeto.

3.4.6 Modelagem e Análise Volumétrica do Coração

Park *et al.* (2005) desenvolveram um Atlas virtual do coração humano a fim de reconhecer a forma e os batimentos cardíacos de um paciente e determinar se seu coração está normal, já que várias doenças do coração estão relacionadas a estes dois fatores.

A geração das imagens 3D do Atlas Virtual foi realizada a partir de reconstrução de imagens provenientes de ressonância magnética. Após a digitalização das imagens do coração do paciente, é feito o processamento das imagens para a reconstrução das imagens 3D a partir das imagens 2D. É possível calcular o ritmo cardíaco e o tamanho do coração do paciente e fazer a comparação com o modelo sadio disponibilizado pelo Atlas.

Os pesquisadores utilizaram o método de elementos finitos para simular a distribuição do sangue ao longo do coração durante a contração. Nas análises, trabalharam com métodos de deformação para a segmentação automática e análise do volume do coração representado nas imagens. A modelagem 3D pode prover a visualização do formato do coração, como mostra a Figura 33.

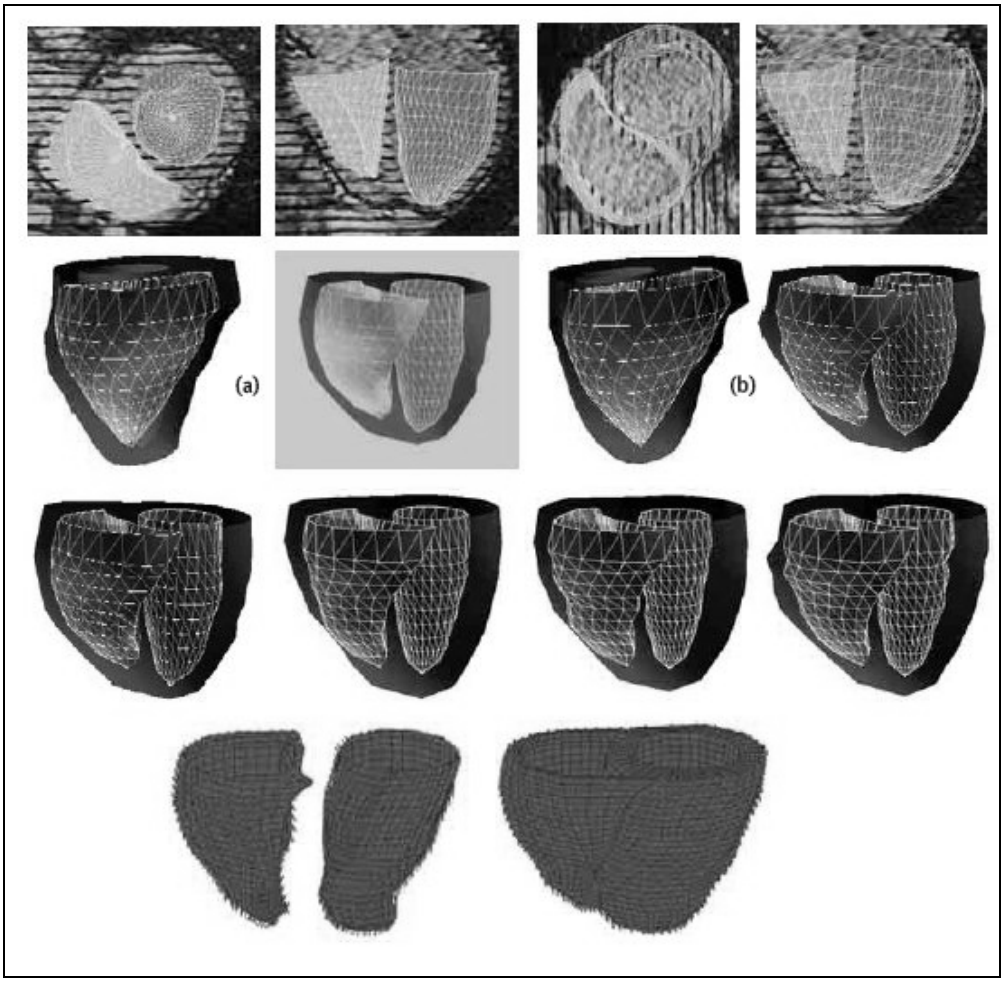


Figura 33 – Modelagem de um coração: (a) captura da imagem 2D, (b) tratamento da imagem através dos métodos desenvolvidos, gerando o objeto 3D e aplicando a devida deformação (Park *et al.*, 2005)

3.4.7 Simulação de Cirurgia Hepática

Dellingete e Ayache (2005) criaram um simulador para treinar profissionais da área de saúde a executar procedimentos cirúrgicos MI, tendo como estudo de caso a cirurgia hepática.

O planejamento de cirurgia hepática envolve a determinação da funcionalidade dos segmentos que correspondem ao fígado. Uma vez determinados os passos da cirurgia, a simulação é introduzida para reproduzir os procedimentos cirúrgicos com realismo. Foram desenvolvidas duas ferramentas: a primeira se preocupa com o planejamento da cirurgia hepática e a segunda é relacionada à simulação dos gestos e movimentos.

Foram propostos dois algoritmos complementares: um para aperfeiçoar os cálculos da deformação e outro com base no método de elementos finitos.

O primeiro algoritmo explora as linearidades do modelo durante um estágio do pré-cálculo. Uma conseqüência direta do modelo linear é que a deformação sob a aplicação é obtida através das diversas forças aplicadas. Conseqüentemente, é possível calcular o estágio da deformação obtida pela força aplicada a cada nó e armazenar esta informação. Durante a simulação, basta calcular uma combinação linear destas deformações armazenadas para recuperar as deformações reais do modelo. A Figura 34 apresenta os passos da simulação da para o treinamento de cirurgia hepática.

O segundo algoritmo envolve o cálculo referente ao movimento dos objetos rígidos dentro da cena em cada interação com o objeto flexível. Dependendo do desempenho do computador e da rigidez do material, é necessário limitar o número dos nós no objeto a fim de satisfazer uma execução em tempo real.

Figura 34 – Objetos sintéticos 3D que representam o fígado e os objetos que compõe a cirurgia no AV (DELLINGETE e AYACHE, 2005)

3.5 Conclusão

Este capítulo abordou os principais conceitos relacionados à aplicação de deformação em objetos 3D, apresentando aplicações na área médica que executam deformações em tempo real.

Foram apresentados três métodos disponíveis na literatura, permitindo um estudo aprofundado sobre cada um deles, verificando suas vantagens e desvantagens.

Algumas aplicações mostradas neste capítulo foram desenvolvidas utilizando os métodos citados e, principalmente, permitindo uma análise mais ampla para verificar o funcionamento dos métodos e escolher o método para o desenvolvimento da aplicação deste trabalho.

O próximo capítulo descreve as tecnologias utilizadas no projeto desenvolvido, incluindo características de linguagem e bibliotecas importantes para a compreensão da implementação realizada.

4. TECNOLOGIAS UTILIZADAS

Este capítulo introduzirá conceitos sobre a linguagem de programação Java com ênfase nas características de uma das suas bibliotecas, a API Java 3D e também uma visão geral de seu funcionamento em relação à disponibilização de objetos 3D em uma cena.

4.1 A Linguagem Java

Java é uma linguagem de programação orientada a objetos e multiplataforma (LEMAY & CADENHEAD, 1999).

Segundo Deitel e Deitel (2003), a linguagem Java possui muitas vantagens e toda a segurança de uma linguagem compilada. Não possui ponteiros, tem a coleta automática de lixo, é multitarefa e permite o controle de *threads*. Oferece grande suporte para aplicações de pequeno, médio e grande porte para Internet e Intranet. Citam ainda como desvantagens da linguagem: lentidão comparada com outras linguagens pelo fato de ser interpretada e não compilada e certa complexidade no uso das bibliotecas.

Portanto mesmo com algumas características negativas a linguagem Java possui diversas vantagens para desenvolvimento de aplicações, como a portabilidade para outras plataformas e sistemas operacionais (Windows, Linux, Unix e MacOS), estabilidade e um conjunto de bibliotecas com recursos de comunicação e gráficos, como, por exemplo, a API Java 3D, que foi utilizada no desenvolvimento do método.

A documentação de Java fornecida pela Sun (2006) afirma que devido às características de orientação a objetos, sua interface contribui para reutilização de código,

utilização de bibliotecas de terceiros com proteção e encapsulamento e, além disso, oferece robustez e segurança.

A escolha da linguagem Java é devido ao fato da sua gratuidade uma vez que, como citado anteriormente, pode ser utilizado em locais onde recursos financeiros são escassos, visto que a idéia do projeto é permitir treinamento de procedimentos médicos para qualquer público.

No contexto do projeto a linguagem foi escolhida devido à quantidade de bibliotecas prontas e também gratuitas que permitem a inclusão de recursos de uma maneira fácil e orientada a objetos.

Uma das características da API Java 3D é permitir a importação e manipulação de objetos modelados, o que é essencial para o trabalho, uma vez que quando ocorre deformação é necessário mudar as posições dos vértices e arestas do objeto modelado.

4.2 API Java 3D

A API Java 3D é uma extensão padrão ao *Java Development Kit* (JDK) de Java 2 constituindo uma interface de programação voltada para aplicações gráficas e *applets*[2], com recursos para construir ambientes tridimensionais. Possui mecanismos de alto nível para criar e manipular geometrias tridimensionais e construir estruturas usadas na renderização dessas geometrias.

Segundo a Sun (2006), com esses mecanismos, o programador pode descrever mundos virtuais grandes e com riqueza de detalhes. Os programas escritos podem interagir

[2] *Applets* são programas Java que podem ser embutidos em documentos de *Hypertext Markup Language* (HTML), ou seja, em página web. Quando um navegador carrega uma página da web que contém uma applet, esta é copiada no navegador e começa a ser executada. (DEITEL & DEITEL, 2003)

com gráficos tridimensionais. A API Java 3D pode ser executada no topo de bibliotecas gráficas de mais baixo nível, como OPENGL e Direct3D, conforme mostra a Figura 35.

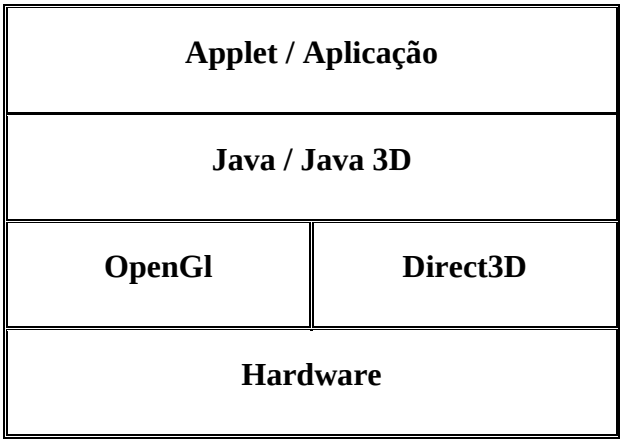


Figura 35 – Camadas de software relacionadas à API Java 3D (SUN, 2005)

A hierarquia das classes Java 3D é apresentada na Figura 36, com a finalidade de mostrar que os projetos desenvolvidos são de alto nível para criar e manipular os objetos tridimensionais geométricos.

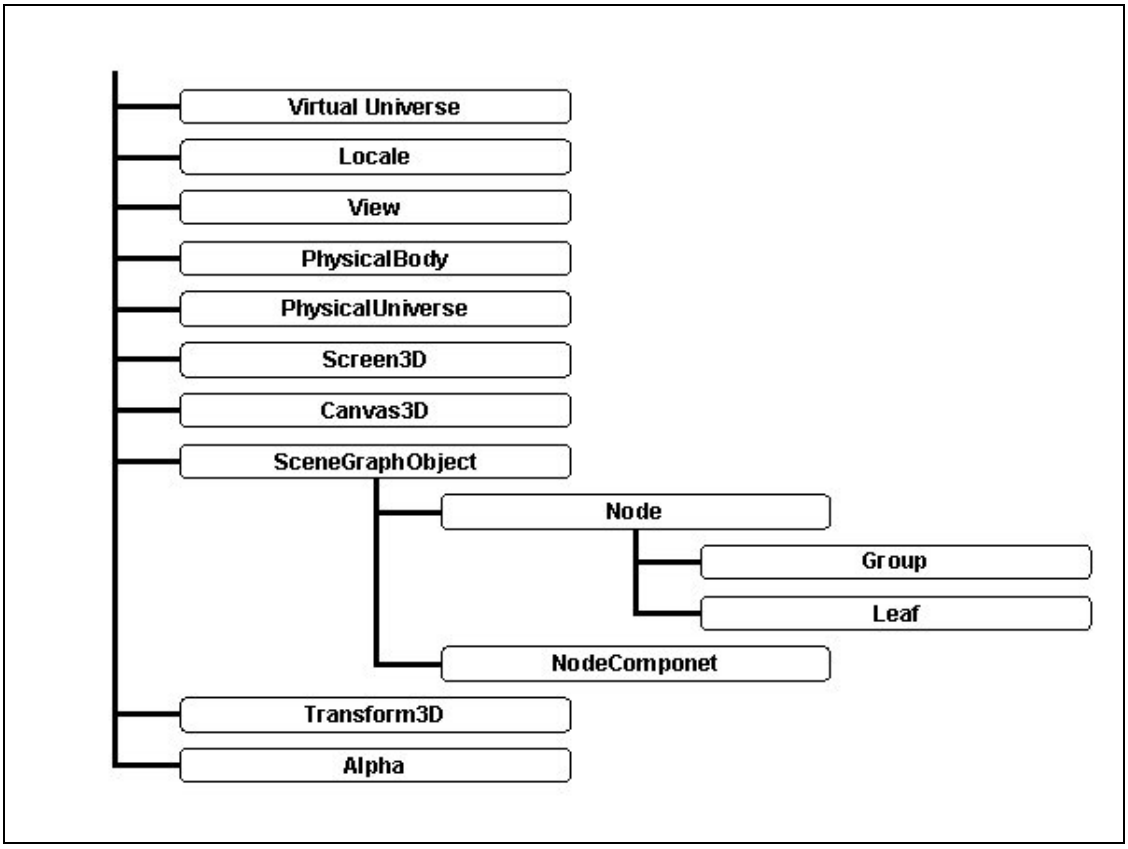


Figura 36 – Hierarquia das classes Java 3D (SUN, 2006)

A API Java 3D permite a manipulação do grafo de cena, uma estrutura do tipo árvore cujos nós são objetos instanciados das classes Java 3D. Os arcos que ligam esses objetos representam o tipo de relação estabelecida entre dois nós, de acordo com Bicho *et al.* (2003).

A Figura 37 mostra um grafo de cena de uma cena simples feito no Java 3D.

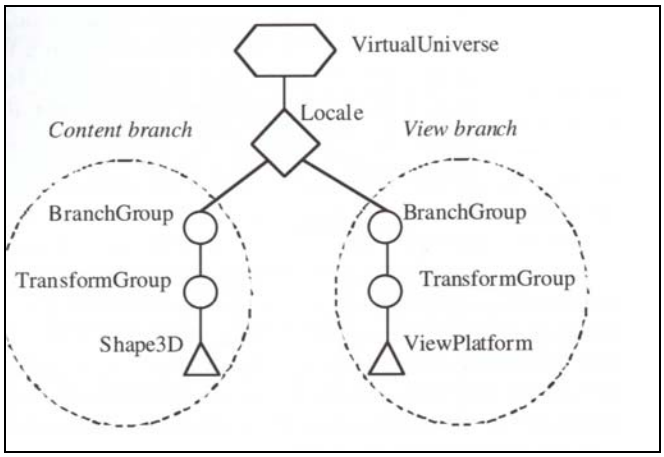


Figura 37 – Um grafo de cena simples da API Java 3D (PALMER, 2001)

O nó raiz de um grafo é instanciado a partir de um objeto do nó da classe *VirtualUniverse*. Este pode possuir um ou mais objetos *Locale*, que contêm as demais informações da cena. Sua função é fornecer um ponto de referência dentro do mundo virtual. Podem existir mais de um nó *Locale* dentro de um programa. Um *VirtualUniverse* mantém uma lista de objetos *Locale* que são ativados em um programa corrente (PALMER, 2001).

Cada *Locale* possui um número de nós filhos do tipo *BranchGroup*. O nó raiz de cada ramificação de um grafo de cena é sempre representado pelo objeto *BranchGroup* (SUN, 2006).

O nó *TransformGroup* é utilizado para especificar a posição do objeto relativo ao nó *Locale*, bem como a orientação e a escala dos objetos geométricos no universo virtual (MANSSOUR, 2003).

Tipicamente existem dois tipos de nós *BranchGroup:* o *Content branch* e o *View branch*, indicados na Figura 37.

O *Content branch,* ou grafo de cena de conteúdo, descreve os objetos a serem renderizados, ou seja, especificam a geometria, textura, som, objetos de interação, luz e a localização desses objetos no espaço do universo virtual. Já o *View branch,* ou grafo de cena de visão, especifica as atividades e os parâmetros relacionados com o controle da vista da cena, tais como orientação e localização do usuário. Sendo assim, ambos determinam o que será renderizado na cena, sem se preocupar com a ordem dos objetos (PALMER, 2001; BICHO *et al.*, 2002).

O *Content branch* é composto pelos nós *BranchGroup*, *TransformGroup* e pelo nó *Shape3D*.

O nó *Shape3D* é responsável por armazenar informações da geometria que compõe a cena. Podem existir vários nós *Shape3D* em um grafo de cena e cada um deles com o seu *TransformGroup,* representando, assim, objetos distintos dentro da cena (PALMER, 2001). A Figura 38 mostra o nó Shape3D e seus componentes: *Geometry,* que fornece a forma geométrica do objeto e o *Appearance,* que especifica a textura, transparência, material, estilo de linha e outras propriedades (PALMER, 2001).

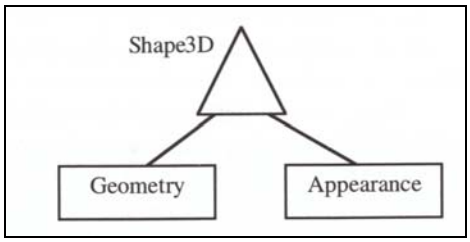


Figura 38 – Os componentes do nó *Shape3D* (PALMER, 2001)

O grafo de cena de visão é composto pelos nós *BranchGroup*, *TransformGroup* e *ViewPlatform*.

O nó *ViewPlatform* é responsável pela visão final do usuário dentro do universo. Bicho *et al.* (2002) afirmam que este é o único nó presente no grafo de cena que faz

referências aos objetos que definem este mundo virtual. Os objetos que compõem o nó são: *View,* com duas ramificações (*PhysicalBody e PhysicalEnvironment), Canvas 3D* e *Screen 3D,* como apresenta a Figura 39.

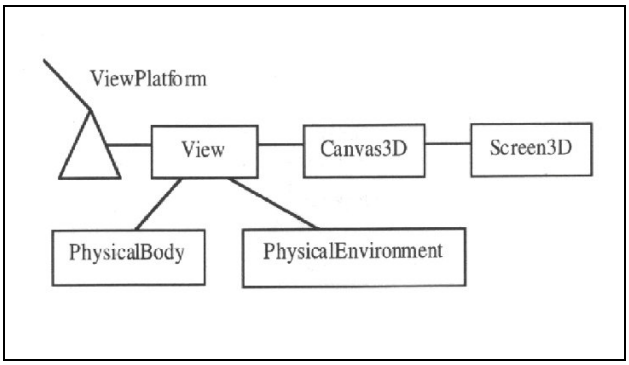


Figura 39 – Objetos referenciados pelo nó *ViewPlatform* (PALMER, 2001)

O objeto *View* armazena todas as informações necessárias para renderizar o grafo de cena, além de coordenar todo o processo de renderização. Este nó está associado às classes: *Physical Body,* que possui informações sobre características físicas do usuário, como a localização dos olhos e a distância entre as pupilas e *Physical Environment,* que gerencia informações sobre o local do ambiente físico do usuário final, dispositivos de áudio e sensores de movimento. Já o objeto *Canvas3D* garante a apresentação gráfica da cena enquadrada em uma janela de visualização, pois agrupa todos os parâmetros relacionados a essa janela e, finalmente, *Screen 3D* possui as propriedades físicas da tela (SUN, 2006 e PALMER, 2001).

4.3 Criação de Geometrias em Java 3D

Para definir a modelagem de um objeto é preciso conhecer o processo de descrição do mesmo. A representação de um objeto, segundo Manssour (2003), deve ser feita de uma forma que seja fácil de usar e analisar.

Em J3D existem três formas possíveis de se criar uma geometria. A primeira é por meio do uso de classes prontas que fornecem primitivas gráficas, como cubo, esfera, cilindro e cone; a segunda é usando uma lista de vértices e uma lista de arestas ou faces poligonais e a terceira forma é importando um arquivo de um objeto modelado, através da classe *Loader* (SUN, 2005). Nas próximas seções cada uma dessas maneiras será explicada detalhadamente.

4.3.1 Primitivas Geométricas na Biblioteca

Segundo Manssour (2003), a maneira mais simples de definir uma geometria de um modelo em Java 3D, é através das primitivas gráficas *Box, Cylinder, Sphere* e *Cone,* que estão disponíveis no pacote *com.sun. j3d. utils.geometry.* A utilização destas primitivas no código é feita por meio da instanciação de classes que possuem o mesmo nome (SUN, 2005; MANSSOUR, 2003).

A Figura 40 mostra o trecho de código de alguns construtores disponíveis nesta classe e a Figura 41 mostra a representação destas geometrias no mundo tridimensional.

```
1. Box();       // Box default com todas as dimensões = 1.0
2. Box (float xdim, float ydim, float zdim, Appearance ap);
3.
4. Cone();      // Cone default com raio = 1.0 e altura = 2.0
5. Cone (float radius, float height);
6. Cone (float radius, float height, Appearance  ap);
7.
8. Cylinder (); // Cilindro default com raio = 1.0 e altura = 2.0
9. Cylinder(float radius, float height);
10.Cylinder(float radius, float height, Appearance  ap);
11.
12.Sphere();    // Esfera default com raio = 1.0
13.Sphere(float radius);
14.Sphere(float radius, Appearance  ap);
```

Figura 40 – Código dos construtores das classes das primitivas (MANSSOUR, 2003)

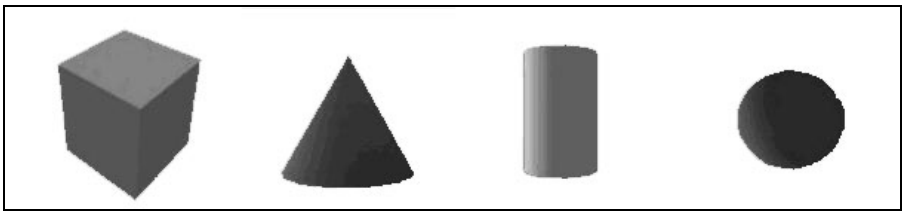


Figura 41 – Resultado da criação das instâncias de *Cone, Cylinder, Box* e *Sphere* (MANSOUR, 2003)

Além da geometria, a API também possui um nó específico em seu grafo de cena, responsável pela aparência do objeto criado por meio de definições das propriedades do material que o compõe como, por exemplo, cor, textura e transparência.

A Figura 42 mostra as classes que permitem a criação das geometrias primitivas oferecidas pela API, sua descrição e um exemplo.

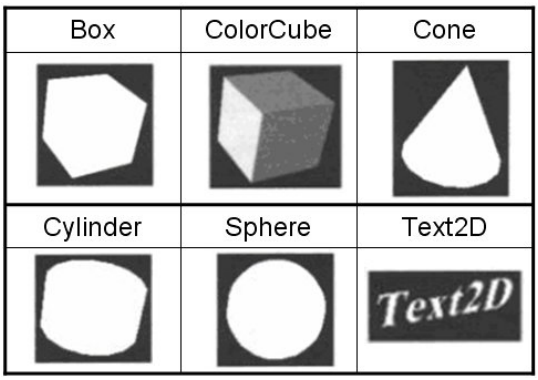


Figura 42 – As primitivas da classe *Geometry* (PALMER, 2001)

4.3.2 Classe *GeometryArray*

A classe *GeometryArray* permite que sejam construídos modelos geométricos 3D mais complexos (BICHO *et al.*, 2002).

A forma de se criar um objeto utilizando esta classe é parecida com a estrutura de dados de qualquer sistema gráfico. Consiste em um conjunto de vetores que contêm coordenadas, coloração e valores de textura que são combinadas com diferentes tipos de geometrias.

Através de uma lista de vértices, arestas ou faces poligonais é possível estabelecer uma malha de polígonos que representa uma superfície composta por faces planas, triangulares ou quadráticas, que permitem representar objetos 3D.

Esta forma permite a modelagem de objetos irregulares e mais complexos com certo nível de realismo, pois as arestas fornecem uma maior uniformidade.

A Figura 43 apresenta a hierarquia da classe *GeometryArray* e as classe e subclasses que compõem a estrutura.

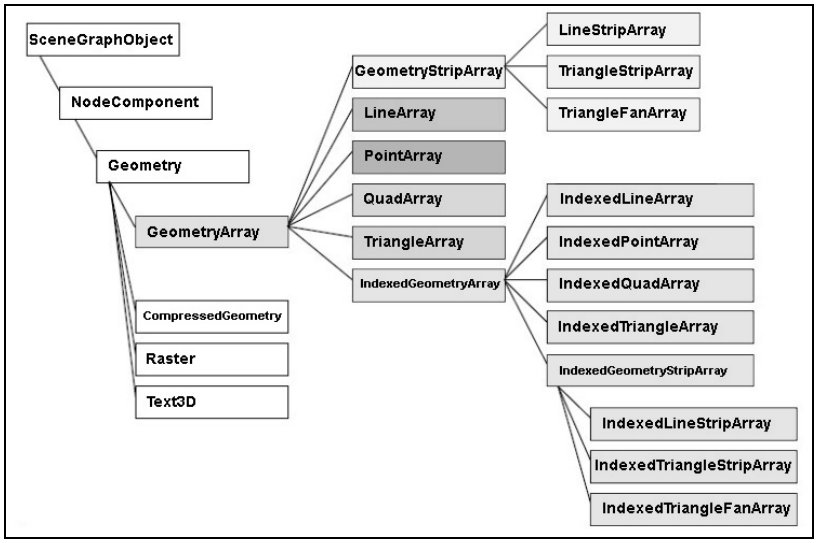


Figura 43 – Hierarquia da Classe *GeometryArray* (Sun, 2006)

A Figura 44 (PALMER, 2001) mostra as classes relacionadas à classe *GeometryArray*.

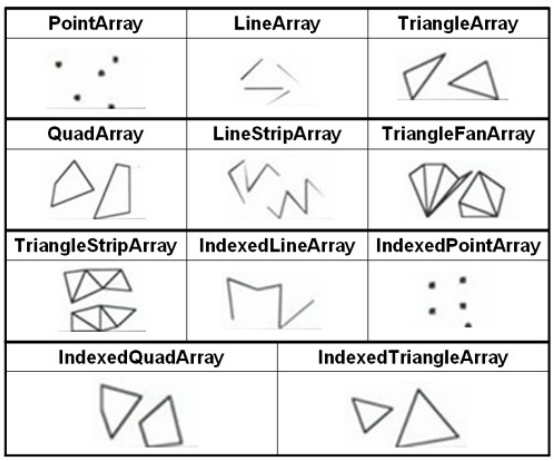


Figura 44 – Tipos de classe da classe *GeometryArray* (PALMER, 2001)

É importante ressaltar que o número total de vértices indica quais componentes estão presentes em cada vértice. As subclasses *PointArray*, *LineArray*, *TriangleArray* e *QuadArray* definem respectivamente: um conjunto de vértices, um vetor de segmentos, um vetor de triângulos individuais e um conjunto de vértices onde cada quatro vértices correspondem aos quatro lados de um quadrado individual (PALMER, 2001).

As demais classes que compõem a hierarquia são subclasses das classes citadas acima que oferecem recursos específicos.

4.3.3 Carregando Modelos Geométricos Prontos

Barrilleaux (2001) lembra que a API Java 3D não possui um formato de arquivo próprio, mas permite desenvolver classes chamadas *Loaders*, ou carregadores. Uma classe *Loader* lê gráficos particulares, ou seja, formatos de arquivos que são instanciados como um objeto no grafo da cena. Nos seus utilitários, a SUN proveu a classe *LoaderBaser* que permite instanciar a classe *Loader* e uma interface para carregar os arquivos.

A classe *Loader* definida no pacote *com.sun.j3d.loaders* permite a importação de uma grande variedade de formatos de arquivos, apresentados na Tabela 1.

O código do objeto modelado não faz parte da API Java 3D ou do pacote *Loader*, o pacote é responsável apenas pelo carregamento do arquivo à cena. As classes *Loader* e *Scene* da API J3D definem interfaces comuns que são responsáveis pelo carregamento de arquivos e retornam um objeto na cena. O arquivo *wavefront*, também denominado aramado, é um deles. Sua extensão é denominada como *.obj* como apresenta a Tabela 1, e foi o formato escolhido para importar os modelos 3D na aplicação.

Tabela 1 – Possíveis formatos de arquivos carregados pela API Java 3D (SUN, 2006)

Formato do Arquivo	Software que Gera o Arquivo
3DS	*3D – Studio*
COB	*Caligari trueSpace*
DEM	*Digital Elevation Map*
DXF	*AutoCAD Drawing interchange File*
IOB	*Imagine*
LWS	*LightWave Scene Format*
NFF	*WorldToolKit NFF Format*
OBJ	*Wavefront*
PDB	*Protein Data Bank*
PLAY	*PLAY*
SLD	*Solid Works (*arquivos *.prt* e *.asm)*
VRT	*Superscape*
VTK	*Visual Toolkit*
WRL	*Virtual Reality Modeling Language*
X3D	*Extended 3D*

O arquivo *wavefront* foi escolhido porque tem uma estrutura relativamente simples permitindo a definição de malhas poligonais com associação de propriedades do material do objeto, segundo Palmer (2001).

A Figura 45 mostra um trecho do arquivo *wavefront*. As linhas que começam com a letra "*v*" indicam a definição dos vértices e os três números que precedem a letra, são as coordenadas x, y, z. A letra "*f*" define as faces e são índices dentro da lista de vértices (PALMER, 2001).

```
# Max2Obj Version 4.0 Mar 10th, 2001
vertices
v  19.052755 32.943306 6.281933
v  18.801495 33.893066 6.281933
v  19.092958 34.209656 6.281933
v  19.685932 33.893066 6.281933
v  20.339209 32.943306 6.281933
v  17.656956 32.943306 13.350986
v  17.425169 33.893066 13.252367
v  17.694042 34.209656 13.366765
v  18.24106 33.893066 13.599508
v  18.843706 32.943306 13.855919
….
faces
f -530/-800 -525/-795 -524/-794
f -524/-794 -529/-799 -530/-800
f -529/-799 -524/-794 -523/-793
f -523/-793 -528/-798 -529/-799
```

Figura 45 – Exemplo de trecho de código do arquivo *wavefront* de um objeto carregado

Para importar arquivo apresentado para a API Java 3D é preciso utilizar as bibliotecas referentes à classe *Loader* por meio do código apresentado na Figura 46.

```
import com.sun.j3d.loaders.*;
import com.sun.j3d.loaders.objectfile.*;
```

Figura 46 – Código para importação de bibliotecas para permitir o carregamento de objetos dentro da API Java 3D (SUN, 2006)

A partir do momento que se usam as bibliotecas são necessárias linhas de códigos que permitam a importação do arquivo. A Figura 47 apresenta um trecho de código responsável por importar um arquivo de extensão *wavefront* em uma aplicação implementada utilizando a API Java 3D.

```
protected Node importaArquivo()
{
        ObjectFile f = new ObjectFile();
        Scene s = null;
        Try
        {
                s = f.load(filename);
        }catch(Exception e)
        {
                System.exit(1);
        }
        return s.getSceneGroup();

}
```

Figura 47 – Trecho de código para inserção de um arquivo utilizando a API Java 3D (PALMER, 2001)

O resultado da importação é mostrado na Figura 48. O objeto *wavefront* está carregado em uma cena utilizando a API J3D.

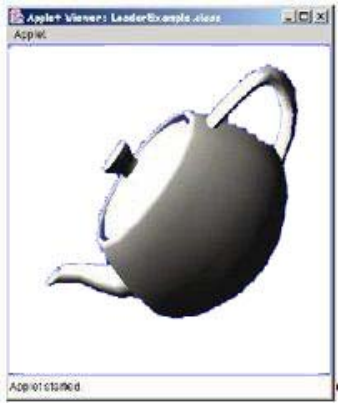


Figura 48 – Objeto do tipo *wavefront* carregado em um ambiente construído usando a API Java 3D (MANSSOUR, 2003)

É importante ressaltar que para poder carregar os arquivos especificados na Tabela 1, é necessário utilizar classes prontas em pacotes específicos ou desenvolver novas classes para o funcionamento correto das aplicações, como foi explicado sobre o arquivo *wavefront*.

Selman (2002), por exemplo, disponibilizou o pacote *j3dTree* contendo exemplos de programas que carregam objetos modelados usando VRML na API Java 3D, conforme é apresentado na Figura 49.

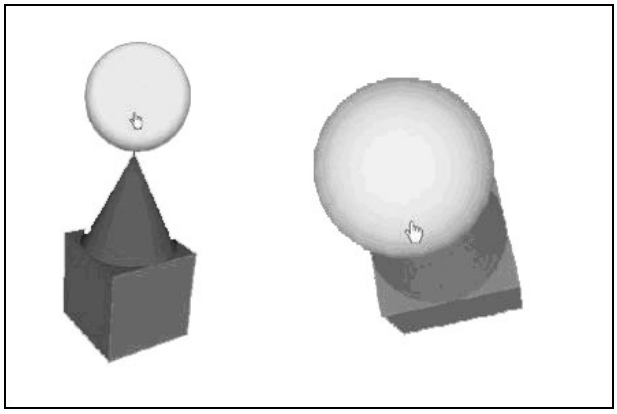


Figura 49 – Programa *VrmlPickingTest,* objeto modelado no VRML carregado em um ambiente construído usando a API Java 3D (SELMAN, 2002)

Moss (2001) desenvolveu o JHGT (Java 3D *Head Generation Tool*) para permitir a reconstrução de semblantes com características da face de um indivíduo usando dados de imagens de crânio. As faces foram modeladas em VRML 2.0 e convertidas para o formato *Wavefront* para serem carregadas. A Figura 50 mostra a interface da aplicação.

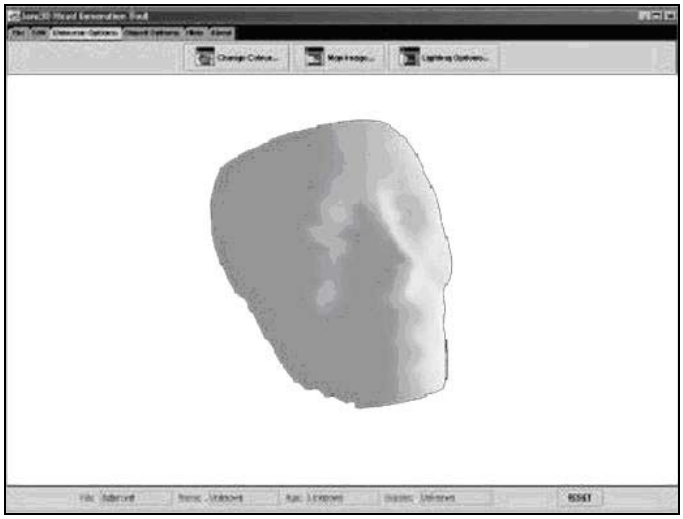


Figura 50 – A interface do usuário do sistema JHGT (MOSS, 2001)

Outro exemplo importante a ser citado é o projeto Konno (1998), que apresenta uma aplicação que importa arquivos do tipo VRML e X3D (*Extensible 3D*). Como no exemplo de Selman (2002), Konno (1998) utilizou um pacote para o carregamento dos objetos. A Figura 51 mostra a interface o funcionamento da ferramenta.

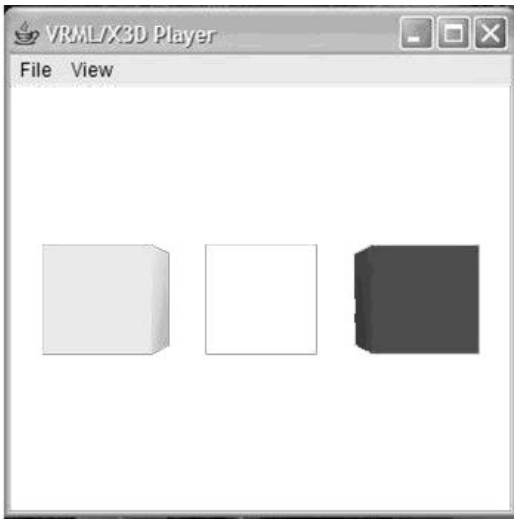


Figura 51 – Interface do programa *VrmlPlayer*: arquivo.*wrl* importado (KONNO, 1998)

Outro exemplo interessante disponibilizado por Konno (1998) é o *VrmlViewer*, que mostra as informações do grafo de cena em forma de árvore. Um exemplo de objeto carregado e o respectivo grafo de cena são mostrados na Figura 52.

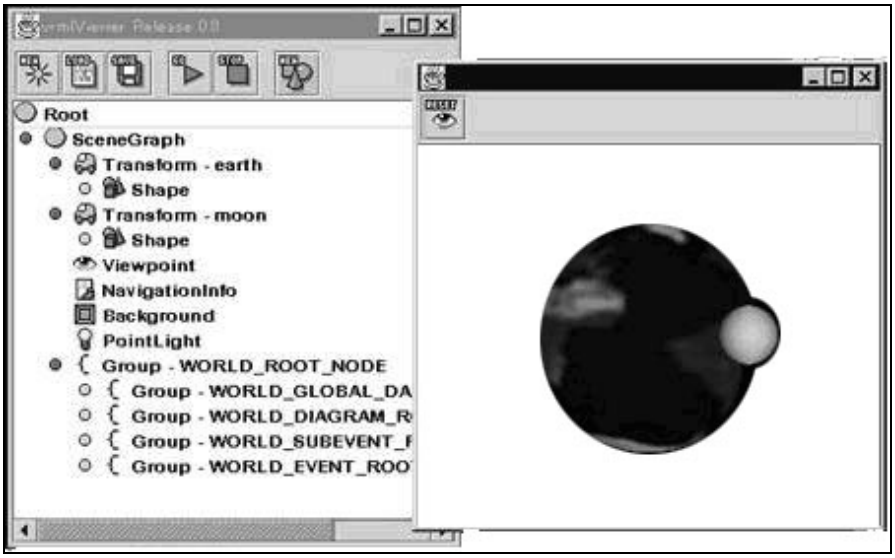


Figura 52 – Interface do programa *VrmlViewer* (KONNO, 1998)

4.4 Conclusão

Neste capítulo foram abordados conceitos sobre a criação e manipulação de geometrias usando a API J3D. Esses conceitos, juntamente com o estado da arte apresentado nos capítulos 2 e 3 , têm o objetivo de fornecer a base teórica para a compreensão do sistema desenvolvido, apresentado no próximo capítulo.

5. *DefApliMed* - SISTEMA DE DEFORMAÇÃO EM APLICAÇÕES MÉDICAS

Como já mencionado, a construção de ferramentas para treinamentos médicos virtuais exige realismo na composição da cena e na interação do usuário com o AV. Muitos pesquisadores vêm estudando funcionalidades importantes para propiciar esse realismo. Entre elas, cita-se a detecção de colisão e a conseqüente deformação, necessária quando há colisões envolvendo objetos flexíveis. A deformação permite que o usuário perceba o tipo de objeto que está manipulando na cena.

O *DefApliMed*- Sistema de Deformação em Aplicações Médicas é um conjunto de classes que simula a deformação em objetos flexíveis quando tocados por objetos rígidos, utilizando o método massa-mola. Um diagrama com os passos executados pelo sistema está apresentado na Figura 53.

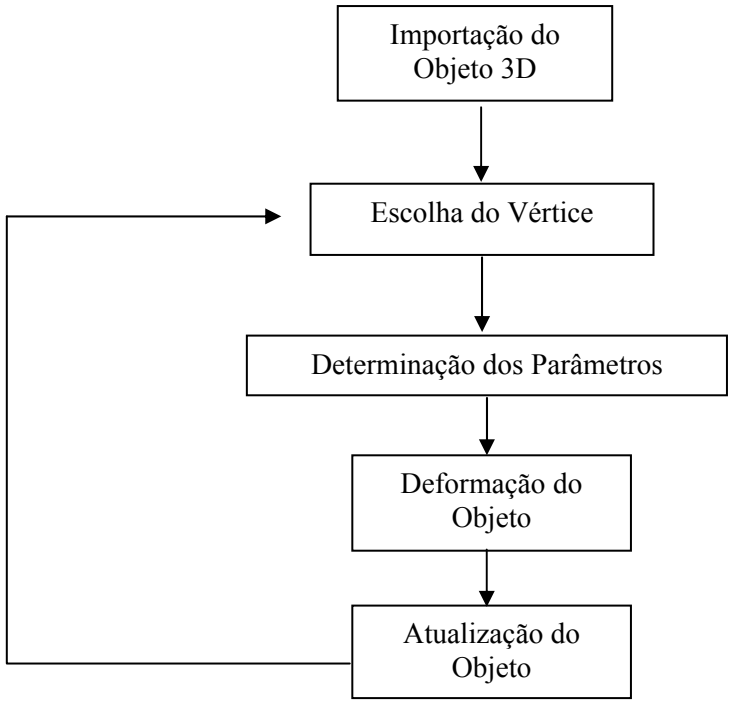


Figura 53 – Esquema do sistema computacional para permitir deformação em objetos 3D

O sistema foi implementado utilizando a linguagem Java (versão j2sdk1. 4.2_05) e a biblioteca Java 3D (versão 1.3_01), executados inicialmente na plataforma Windows. O usuário seleciona o objeto a ser carregado e esse objeto é ajustado na cena através da aplicação de uma escala. Após a importação do objeto, o usuário pode escolher o local a ser deformado tendo a possibilidade de alterar parâmetros de acordo com as características particulares de cada objeto. A seguir será detalhado cada um dos passos necessários até atingir o resultado final.

5.1 Implementação Orientada a Objetos

O paradigma de Orientação a Objetos oferece a vantagem de reaproveitamento de código e a facilidade na estruturação do sistema, viabilizando a inserção de novos recursos sem muitas modificações (DEITEL e DEITEL, 2003).

A UML (*Unified Modelling Language*) é uma linguagem de modelagem unificada que auxilia a visualização das classes que compõem o projeto e a comunicação entre objetos existentes nas classes.

Basicamente, a UML permite que desenvolvedores visualizem o projeto em diagramas padronizados, especifique significados, isto é, a semântica, utilizando uma notação independente de processos.

É muito utilizada em sistemas orientados a objetos, permitindo especificar, visualizar, construir e documentar os artefatos de um sistema de software orientado a objetos (SILVA, 2002).

A Figura 54 apresenta o diagrama de classes do *DefApliMed* e as próximas seções apresentam explicações detalhadas de cada uma das classes envolvidas no projeto.

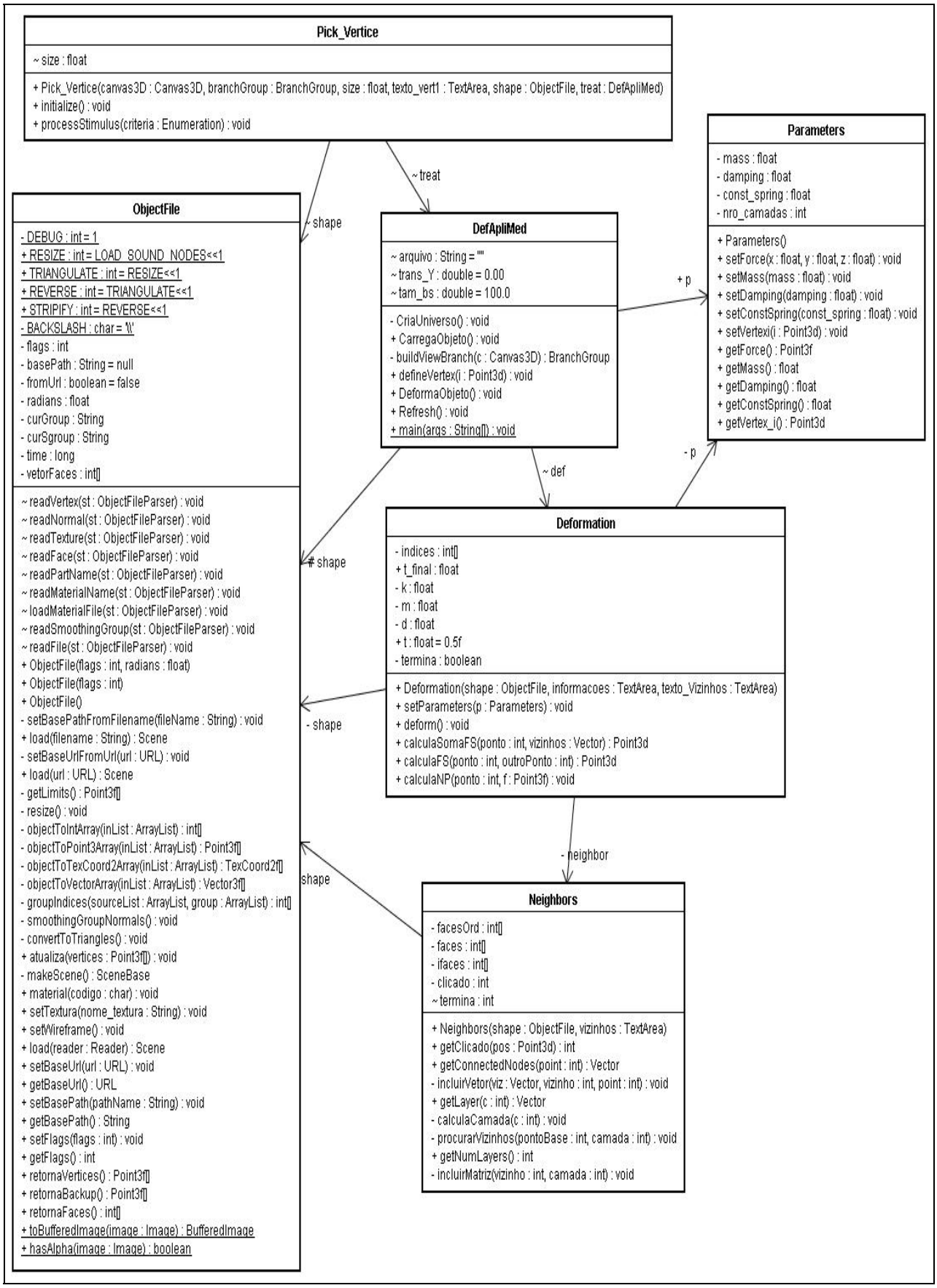


Figura 54 – Diagrama de classes do *DefApliMed*

5.1.1 Classe *DefApliMed*

A classe *DefApliMed* é a classe principal responsável pela interface com o usuário e pela criação da cena. Ela instancia as classes *Pick_Vertice, ObjectFile e Deformation* e cria as estruturas necessárias para criação da cena que são: o *canvas, VirtualUniverse* e *Locale,* descritas no Capítulo 4.

Após a criação dessas estruturas são criados os *BranchGroups* que são adicionados ao nó *Locale*, como mostra o grafo de cena do sistema apresentado na Figura 55.

Esta classe agrupa todas as funcionalidades definidas nas demais classes para que o funcionamento da deformação ocorra de forma coerente.

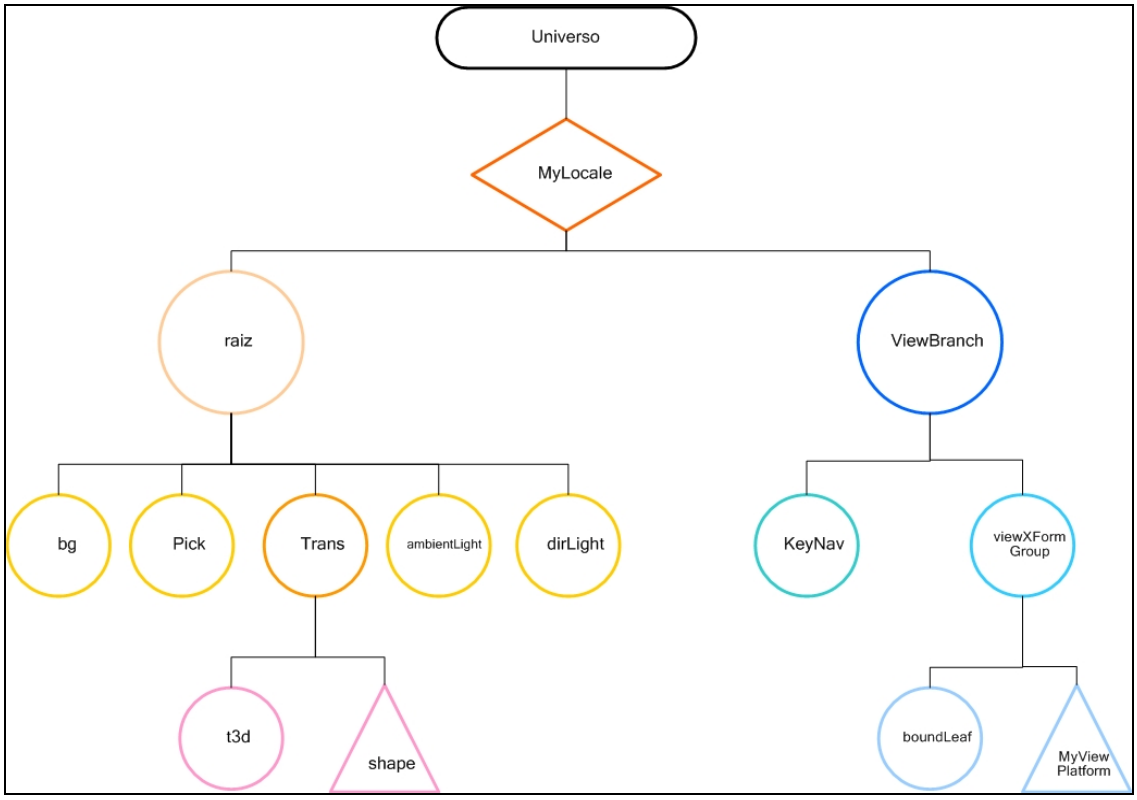


Figura 55 – Grafo de cena do Sistema *DefApliMed*

A Tabela 2 lista os nós presentes no grafo de cena e uma breve descrição de sua funcionalidade dentro do AV.

Tabela 2 – Nós do grafo de cena e suas respectivas descrições

Nó	Descrição
Universo	Responsável pela criação do ambiente virtual.
MyLocale	Nó onde são anexados os *BranchGroups*.
Raiz	*BranchGroup* onde são incluídos o objeto importado, as luzes, as transformações necessárias do objeto e o comportamento que permite verificar o ponto selecionado.
bg	*Background* do ambiente Virtual.
Pick	Comportamento que permite verificar o ponto selecionado.
Trans	Transformações aplicadas no objeto importado.
T3d	Rotação, translação e escala.
shape	Objeto importado.
ambientLight	Luz ambiente do AV.
dirLight	Luz direcional apontada para o objeto importado.
ViewBranch	Nó de visualização onde são anexados os nós referentes à navegação do AV.
KeyNav	Nó responsável por controlar a navegação pelo teclado.
viewXFormGroup	Nó responsável por controlar a navegação pelo mouse.
boundLeaf	Define a região influenciada pela navegação com o mouse.
MyViewPlataform	Controla a posição, orientação e escala do campo de visão.

Para propiciar uma visualização adequada do objeto carregado foi necessária a inserção de luz e elementos de navegação no contexto do sistema.

A luz foi inserida dentro do nó *BranchGroup*, que são *ambientLight* e *dirLight*, juntamente com o objeto a ser deformado e as características do comportamento (*behavior*) necessário para verificar qual local foi selecionado no objeto. A navegação dentro do sistema foi inserida no nó *BranchGroup* de visualização, sendo realizada com a utilização do teclado e do mouse.

A interface pode ser dividida em quatro partes, como é apresentada na Figura 56.

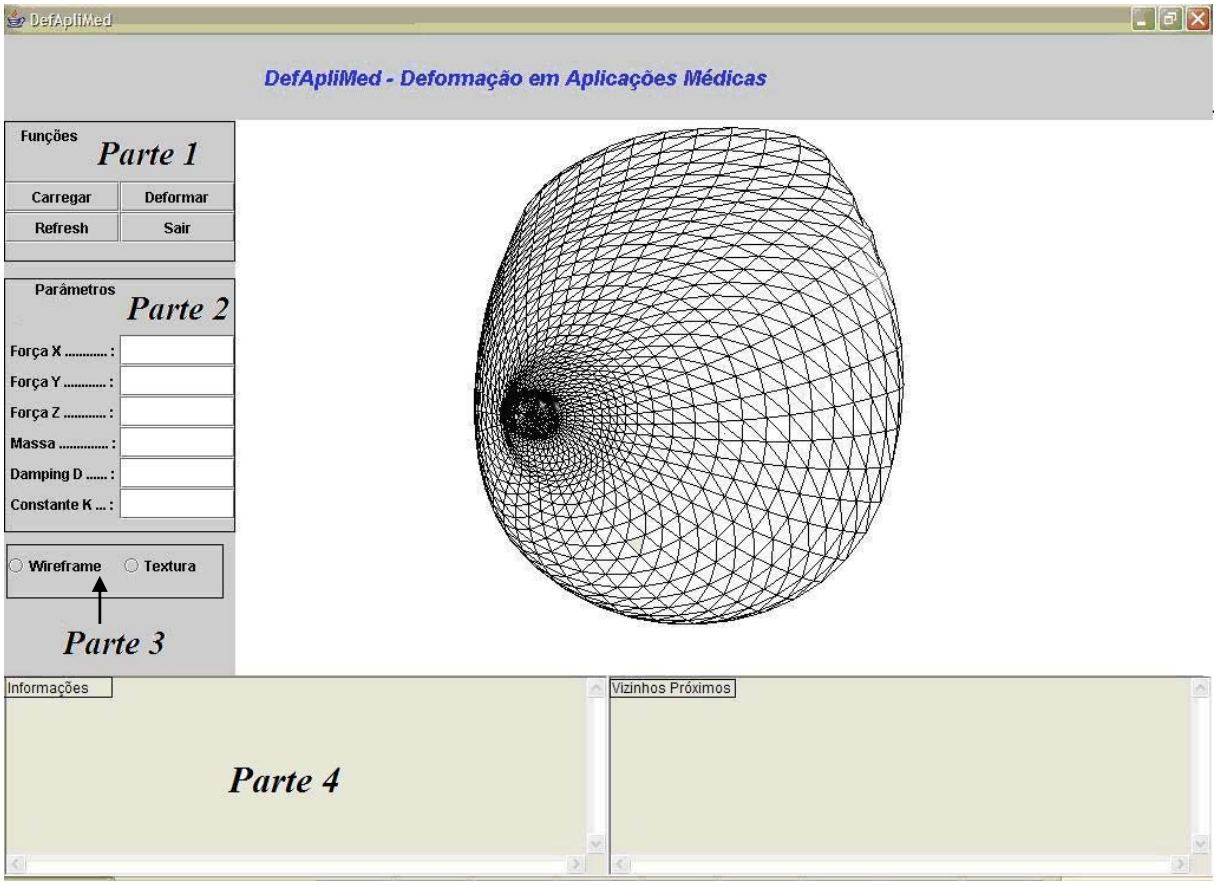


Figura 56 – Partes específicas da interface e seus respectivos funcionamentos

A classe *DefApliMed* possui métodos que permitem o tratamento de parâmetros e também o recarregamento do objeto para que este retorne a sua forma original. Esses controles podem ser visualizados na Figura 56 – parte 1. Já a parte referente aos parâmetros do objeto pode ser visualizada na Figura 56 – parte 2. Seus campos permitem que o usuário insira os valores dos respectivos parâmetros para que a deformação ocorra.

A parte inferior da interface é constituída de dois campos que informam ao usuário as coordenadas do ponto selecionado, o vértice vizinho mais próximo a este ponto e quais as camadas afetadas ao longo da simulação de deformação. Na Figura 56 - parte 4 mostra-se a exibição destes valores nos campos *Informações* e *Vizinhos Próximos*.

A classe *DefApliMed* permite, ainda, a visualização do objeto de duas maneiras: por *wireframe*, também conhecido como aramado, e com textura. A Figura 56 – parte 3 apresenta

a seleção desta característica. A Figura 57 apresenta um objeto incluído na cena sendo visualizado com e sem textura.

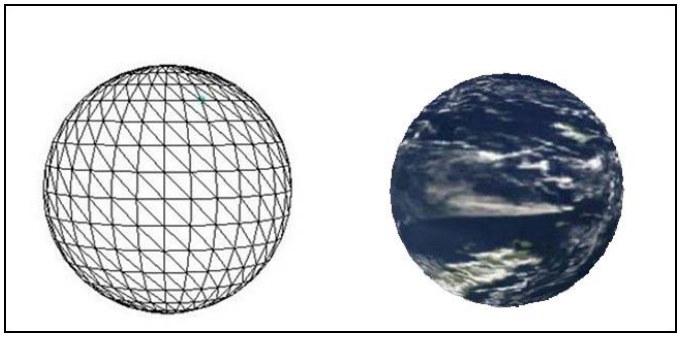


Figura 57 – Objeto visualizado com e sem textura

5.1.2 Classe *ObjectFile*

Como descrito no Capítulo 4, a API Java 3D fornece três formas de construção de geometrias: primitivas, vértices e arestas e importação de arquivos. A abordagem utilizada neste projeto é a importação de arquivos, visto que o objeto de estudo é o treinamento médico, o qual exige modelagens que simulem órgãos humanos com um nível satisfatório de realismo.

Sendo assim, foram feitos estudos relacionados à importação de arquivos e formas possíveis de se realizar a manipulação dos vértices a fim de gerar a deformação. Para que o sistema proposto atendesse a esses requisitos necessários, foram implementadas algumas adaptações para tornar possível acessar as estruturas internas do objeto (vértices e faces) e, a partir delas, gerar a deformação requerida.

Na Figura 54 a classe *ObjectFile* apresenta métodos que são responsáveis por carregar arquivos no formato *wavefront* (.obj) (SUN, 2005), que podem ser modelados, por exemplo, em ferramentas como: *Blender* (BLENDER,2006), *3D Studio Max* (3D Studio, 2005) e *Maya* (CHRISTOV *et al.* 1999).

A classe *ObjectFile* do sistema *DefApliMed* é uma adaptação da classe *ObjectFile* da API Java 3D (SUN, 2005). Esta adaptação permitiu prover as faces e os vértices que compõem o objeto, que são informações necessárias para o processo de deformação. Outra adaptação realizada foi permitir a atualização do objeto após uma deformação.

A funcionalidade da classe *ObjectFile* é atuar como um *parser*[1], que transforma os dados contidos no arquivo "*ObjectFile.java"* em estruturas que podem ser apresentadas no sistema utilizando a API Java 3D, permitindo a importação correta do arquivo.

O funcionamento da classe *ObjectFile* possui três etapas essenciais:

- *Parser* do arquivo *wavefront*;
- Criação da estrutura *GeometryInfo;*
- Criação do objeto *shape* que é apresentado na cena.

A classe *ObjectFileParser* realiza a análise do arquivo *wavefront*, dividindo o código do arquivo em átomos. Esses átomos são interpretados pela classe *ObjectFile,* criando uma estrutura de vetores de vértices e de faces.

A partir das estruturas geradas, é possível criar a estrutura *GeometryInfo*, que é responsável por unir essas informações e, conseqüentemente, permitir a criação do *Shape3D* para ser adicionado ao nó *BranchGroup*.

A adaptação realizada no *DefApliMed* faz com que a classe *ObjectFile* herde todas as características da classe *Shape3D*. Isso permite que a classe *ObjectFile* atue com as características da classe *Shape3D* facilitando a importação dos objetos na cena, como é apresentado no trecho de código da Figura 58.

```
public class ObjectFile extends Shape3D implements Loader
```

Figura 58 – Trecho de código referente à herança das características da classe *Shape 3D*

[1] *Parser* é um programa de computador, ou apenas um componente de um programa, que serve para analisar a estrutura gramatical de uma entrada, manipulando os átomos, que são segmentos de texto ou símbolos que podem ser manipulados (Aho *et al.,* 1995).

A linha de código apresentada na Figura 59 cria o vetor de faces que é fornecido para outras classes permitindo o detalhamento da estrutura do objeto.

```
vetorFaces = groupIndices(coordIdxList, triList);
```

Figura 59 – Trecho de código referente à criação do vetor de faces

Como mencionado anteriormente, a partir da estrutura *GeometryInfo* é possível criar um objeto do tipo *Shape3D*, mas para tanto, é necessário que o método *MakeScene* crie a estrutura *GeometryInfo*.

Como a classe *ObjectFile* herda as características da classe *Shape3D* foi possível atribuir essa geometria ao *ObjectFile*, por meio do código apresentado na Figura 60.

```
this.setGeometry(gi.getGeometryArray(true, true, false));
```

Figura 60 – Trecho de código referente à atribuição da geometria ao *ObjectFile*

Foram criados ainda seis métodos para a manipulação dos dados disponibilizados, sendo quatro responsáveis por retornar dados referentes à estrutura do objeto e dois referentes à textura do mesmo.

Os quatro primeiros métodos criados foram: *retornaVertices(), retornaBackup(), retornaFaces()* e *Atualiza()*. Esses métodos possuem as seguintes funcionalidades, respectivamente: retorna a posição atual das coordenadas de cada vértice, ou seja, o vetor de vértices; retorna a posição inicial das coordenadas do vetor de vértices; retorna as informações de quais vértices compõem uma face; e, por último, da mesma forma que o método *MakeScene()*, o método *Atualiza()* cria um objeto do tipo *GeometryInfo*, utilizando as informações do objeto deformado para permitir a atualização desse objeto dentro da cena.

Na Figura 61 são apresentados três objetos importados pela classe *ObjectFile* utilizando a biblioteca Java 3D. A classe provê o carregamento de objetos simples como de objetos mais complexos.

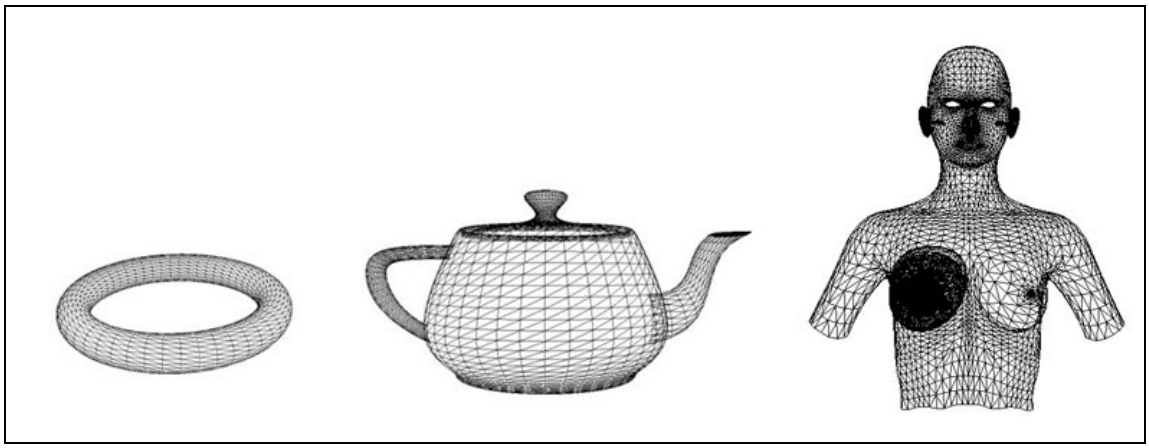


Figura 61 – Objetos mais simples e complexos importados e carregados na API Java 3D através da classe *ObjectFile.*

O arquivo *ObjectFile.java* utilizado no projeto foi todo reestruturado para importar arquivos *wavefront* (especificamente com a extensão *.obj*) e permitir a sua manipulação.

5.1.3 Classe *Pick_Vertice*

A funcionalidade de escolha do local a ser deformado é determinada no sistema pela classe *Pick_Vertice*, que permite ao usuário escolher qual local do objeto carregado será deformado e que foi uma classe adaptada.

Quando o usuário interage com o AV, o sistema informa em qual posição do objeto (coordenadas x, y, z) ocorreu esta interação. A classe *Pick_Vertice* define qual o vértice mais próximo à posição selecionada. Esta classe foi criada com base em um exemplo da biblioteca Java 3D chamado "*IntersectInfoBehavior.java*." (SUN, 2006).

Após realizar as adaptações necessárias no exemplo citado, a classe *Pick_vertice* verifica as coordenadas selecionadas pelo usuário no objeto e retorna as coordenadas do vértice mais próximo ao ponto selecionado.

As adaptações realizadas na classe *Pick_Vertice* são relacionadas à inclusão de chamadas a métodos presentes no sistema *DefApliMed* para informar quais as coordenadas do local que sofre a deformação, como mostra o trecho de código na Figura 62 .

```
treat.defineVertex(closestVertLocal);
```

Figura 62 – Código referente à definição das coordenadas do local da deformação.

O vértice mais próximo foi obtido pela função *getClosestIntersection*, própria da API Java 3D, presente na classe *PickIntersection*.

A Figura 63 apresenta uma posição selecionada a partir de uma interação do usuário com o objeto. A esfera vermelha indica a posição selecionada pelo usuário e a esfera verde indica o vértice mais próximo a essa interação. Na parte inferior da interface aparecem os valores das coordenadas do local selecionado e do vértice mais próximo.

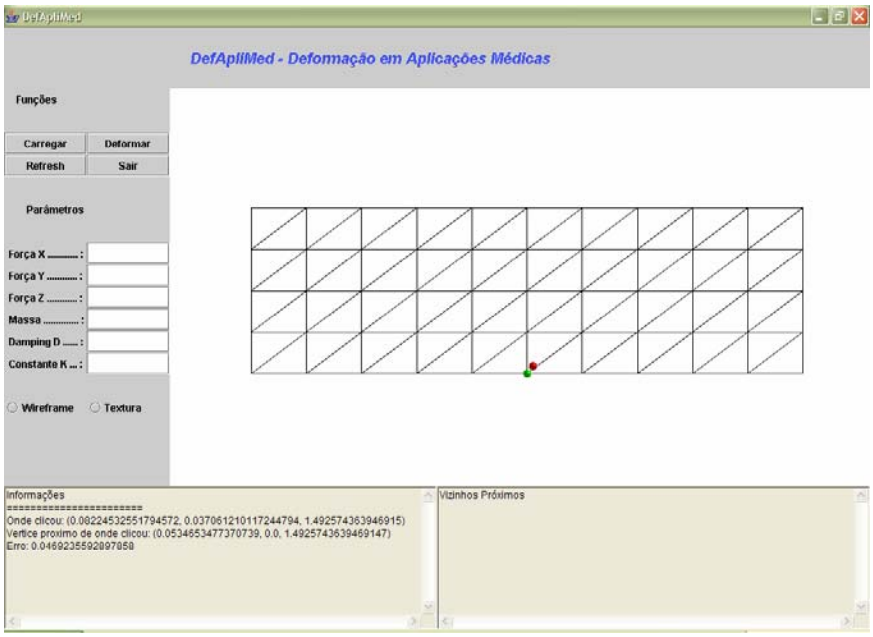


Figura 63 – Valores das coordenadas selecionadas pelo usuário e do vértice próximo ao clicado.

A esfera vermelha e a esfera verde inseridas na cena servem para orientar o usuário em relação ao local selecionado e o vértice mais próximo ao ponto. Essas coordenadas servirão de base para o funcionamento da classe *Neighbors*.

Como a deformação é realizada com base no vértice mais próximo ao vértice clicado, o efeito visual causado por esta diferença deve ser medido. A classe *Pick_Vertice* calcula a distância entre o vértice mais próximo e o local clicado, ou seja o erro envolvido nesta parte da escolha do vértice.

É necessário o cálculo deste erro, pois como são estudos iniciais de deformação para sistemas de tempo real de deformação é necessário à precisão.

5.1.4 Classe *Parameters*

O método massa-mola tem parâmetros específicos para gerar a deformação que são: massa, constante de amortecimento, constante da mola e força envolvida na deformação. Estes parâmetros possuem valores específicos dependendo de cada estudo de caso de deformação que pode vir ocorrer em um objeto carregado na cena.

A classe *Parameters* possui métodos responsáveis por armazenar e recuperar valores nos parâmetros citados para que os métodos presentes na classe *Deformation* consigam acessá-los para gerar a deformação.

A Tabela 3 apresenta os métodos pertencentes à classe *Parameters* e suas funcionalidades no sistema. Esses métodos são responsáveis pelo armazenamento e atribuição dos valores nos parâmetros para o funcionamento correto do sistema.

Tabela 3 – Métodos e suas funcionalidades da classe *Parameters*

Métodos da Classe *Parameters*	**Funcionalidade**
public void setForce()	Fixa os valores (x, y, z) para a força.
public void setMass()	Fixa o valor da massa.
public void setDamping()	Fixa o valor da constante de amortecimento.
public void setConstSpring()	Fixa o valor da constante da mola.
public void setVertexi()	Fixa o valor do vértice escolhido.
public Point3f getForce()	Atribui os valores (x, y, z) da força ao sistema.
public float getMass()	Atribui o valor da massa ao sistema.
public float getDamping()	Atribui o valor da constante de amortecimento ao sistema.
public float getConstSpring()	Atribui o valor da constante da mola ao sistema.
public Point3d getVertex_i()	Atribui o valor do vértice escolhido ao sistema.

5.1.5 Classe *Neighbors*

A classe *Neighbors* possui basicamente duas funções: a primeira é construir a estrutura de camadas e armazená-la e a segunda é encontrar os vértices vizinhos conectados de um determinado ponto selecionado do objeto.

Inicialmente foi preciso encontrar a posição do vértice selecionado dentro do vetor de vértices, ou seja, o índice desse vértice. Isso é necessário para verificar quais vértices estão conectados ao vértice escolhido, uma vez que essa verificação é realizada com base nos índices dos pontos presentes no vetor de faces.

Para facilitar essa busca os vértices contidos no vetor de faces foram ordenados e os índices de suas posições originais foram mantidos. Essa ordenação foi implementada utilizando o método *QuickSort*.

Como o objeto importado possui faces triangulares, o vetor de faces é composto por conjuntos de três valores. Seu formato pode ser representado como uma matriz, conforme mostrado na Figura 64.

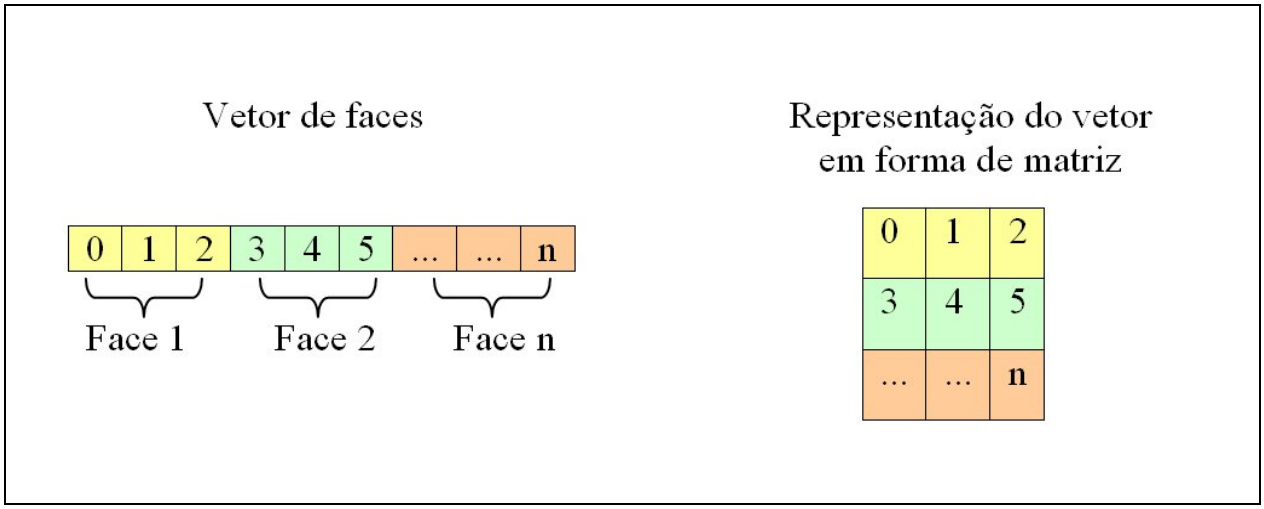


Figura 64 – Formato do vetor de Faces

Nesta estrutura, considerando que cada face é representada por uma linha da matriz, e que todos os vértices de uma determinada face estão conectados, é possível identificar os vértices conectados, verificando em quais faces eles aparecem.

A partir dessas informações foi desenvolvido um algoritmo para criar uma estrutura de camadas. Essas informações foram armazenadas utilizando a classe *Vector* da API Java. Essa mesma classe foi utilizada para armazenar os dados da estrutura que indica os vizinhos de um determinado vértice.

Inicialmente o algoritmo busca todos os vértices conectados ao ponto selecionado através da classe *Pick_Vertice*, sendo os vértices encontrados pertencentes à primeira camada. Posteriormente, para formar as próximas camadas, o algoritmo deve buscar todos os vértices

conectados a cada vértice da camada anterior, desprezando os vértices que já constam na estrutura.

Para isso existe o método *getLayer()* que possui a função de definir os vértices pertencentes a uma determinada camada à medida que as camadas são solicitadas pela classe *Deformation.*

A estrutura apresentada na Figura 65 ilustra como são formadas as camadas de um determinado objeto. O ponto central em vermelho indica o vértice inicial para a deformação. São verificados quais vértices estão diretamente conectados ao vértice escolhido. Esses vértices foram destacados com a cor verde formando a primeira camada. Para formar a segunda camada são verificados quais os vértices estão conectados aos vértices da primeira camada, obtendo os pontos destacados em azul. Os pontos destacados em rosa compõem a terceira camada.

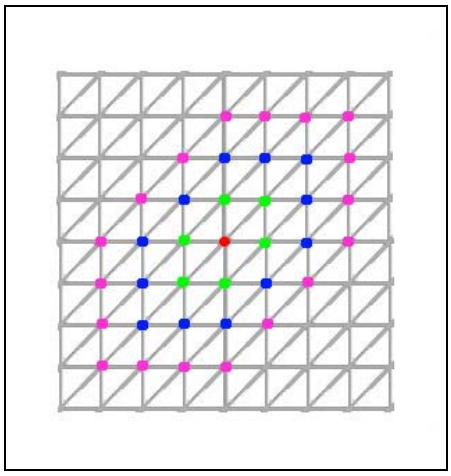


Figura 65 – Determinação das camadas de um objeto

5.1.6 Classe *Deformation*

A classe *Deformation* permite o reposicionamento de vértices e demais estruturas de objetos 3D na cena, de forma a simular uma deformação ocorrida após uma interação.

Para se obter a formulação da força da mola é preciso considerar a deformação do objeto sob uma rede de nós de massa conectados por molas (CHOI *et al.,* 2002), apresentados na Figura 66. Cada mola assumida obedece à lei de *Hooke* (ZILL, 2003) e cada movimento é contido por um amortecimento da força proporcional a sua velocidade.

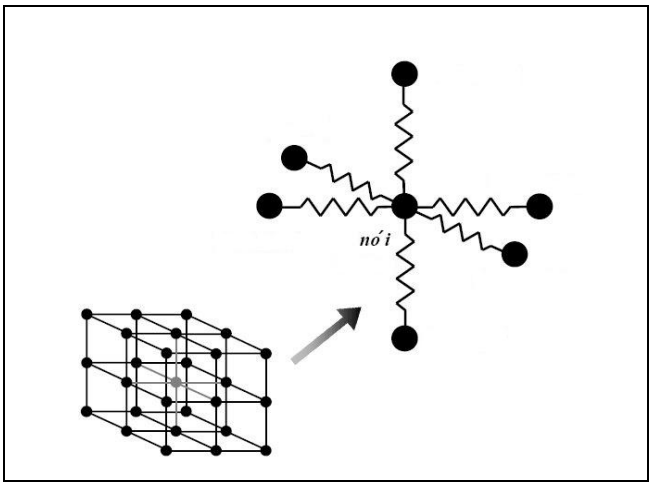


Figura 66 – Nó central (*i*) conectado aos seus vizinhos através de molas (CHOI *et al.*, 2002)

O nó *i*, que representa o vértice que será modificado, é conectado a seis nós adjacentes através das molas. A força incidente no nó *i* é dada pela resultante da força das seis molas e da força externa (F_i) aplicada no nó, como mostra a equação 1. O comportamento dinâmico do nó *i* é regido pelas equações de Newton, aceleração e velocidade mostrados nas equações 2 e 3, respectivamente que foram utilizadas para a implementação do projeto.

Para gerar a deformação de um objeto, é utilizada a equação 7 (Choi *et al*., 2002), onde m_i, u_i e α_i são, respectivamente, *massa, posição* e a *constante de amortecimento* do nó *i*. O vetor que representa a *distância* entre o nó *i* e o nó *j* é dado por $\boldsymbol{r}_{ij} = \boldsymbol{u}_j - \boldsymbol{u}_i$, e l_{ij} é o *tamanho natural da mola* que conecta o nó *i* ao nó *j*. A mudança de posição do nó *i* é obtida por meio da resolução da equação 7, usando o método de diferenças finitas e Δt indica uma etapa de *variação de tempo*. A posição do nó *i* no instante $t + \Delta t$ é dada por $u_i(t + \Delta t)$, e é calculada tomando-se a posição do nó *i* no instante corrente *t* e a sua posição no instante anterior $t - \Delta t$.

As equações 8 e 9 representam a velocidade e a aceleração de Newton, respectivamente. Ambas as equações permitem que se determine a variação de distância sobre o tempo, implicando assim a posição final da mola em uma determinada variação do tempo (CHOI *et al.*, 2002).

$$m_i \frac{d^2 u_i}{dt^2} + \alpha_i \frac{du_i}{dt} + \sum_{j \in nós\_conectados} \frac{k_{ij}(|r_{ij}| - l_{ij})}{|r_{ij}|} r_{ij} = F_i \tag{7}$$

$$\frac{du}{dt} \approx \frac{\Delta u}{\Delta t} = \frac{u(t+\Delta t) - u(t-\Delta t)}{\Delta t} \tag{8}$$

$$\frac{d^2 u}{dt^2} \approx \frac{\Delta^2 u}{\Delta t^2} = \frac{u(t+\Delta t) - 2u(t) + u(t-\Delta t)}{(\Delta t)^2} \tag{9}$$

A classe *Neighbors* possui a tarefa de verificar, durante os cálculos da deformação, como será propagada a deformação para os pontos vizinhos, sendo os vizinhos ao nó *i* os nós que estão conectados a ele. A classe *Neighbors* será mais bem explicada na seção 5.1.6.

O vetor de faces triangularizadas, determinado pela classe *ObjetcFile*, é composto por conjuntos de três valores inteiros que representam as faces, sendo que cada conjunto é referente a uma face. Para verificar quais vértices estão conectados ao vértice escolhido, basta encontrar quais conjuntos contêm este vértice, sendo que os vértices remanescentes estão conectados a ele, ou seja, pertencem à mesma face. Dados os vizinhos é necessário definir em quantas camadas do objeto a força exerce influência.

A classe *Deformation* considera que os dados referentes ao objeto estão organizados em dois vetores: um vetor de vértices que indica as coordenadas (x, y, z) para cada vértice; e um vetor de faces que indica as ligações entre os vértices que formam as faces.

Um exemplo é apresentado na Figura 67. Pode-se visualizar uma estrutura com duas faces, sendo face 1 formada pelos vértices 0, 1 e 2 e a face 2 formada pelos vértices 1,3 e 2.

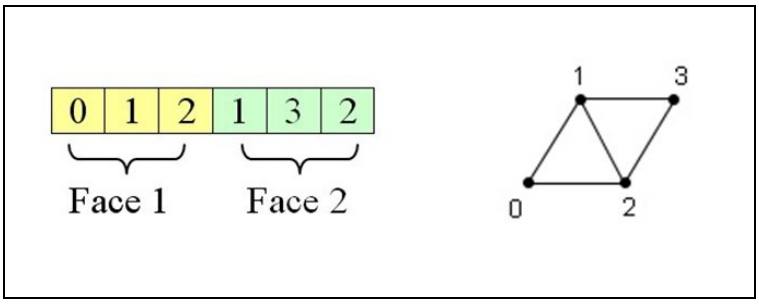


Figura 67 – Exemplo da estrutura do vetor de face

Os vértices que estão diretamente conectados ao vértice escolhido pertencem à primeira camada de deformação. A segunda camada é composta pelos vértices que estão conectados aos vértices da primeira camada, ou seja, vértices secundários ao vértice escolhido, e assim por diante. A partir disso, as informações sobre quais vértices pertencem a quais camadas são armazenadas em estruturas de dados definidas na classe *Neighbors*.

O sistema permite que o usuário estipule valores para um vetor de força que é determinado como um parâmetro. Como está sendo considerado um espaço 3D, essa força pode ser aplicada em um ou mais dos eixos (x, y, z).

Outro fator a ser considerado durante o cálculo da deformação é o número de camadas afetadas por esta.

À medida em que os vértices vão sendo deslocados pela força externa, a força das molas que conectam os demais vértices também os afetam e, assim, a deformação é propagada para as camadas, ou seja, quando um vértice se desloca da posição anterior ele carrega junto com ele os demais vértices com os quais ele possui ligações

O critério de implementação para conseguir o efeito de deformação das camadas está relacionado com a quantidade de força aplicada e também aos parâmetros de constante da mola e amortecimento uma vez que são eles que permitem uma modelagem matemática da estruturas físicas do objeto em questão.

Desta forma quando um ponto é deslocado a força da mola correspondente a este ponto deve ser recalculada através da operação 10, que calcula a força de uma única mola,

tendo como componentes específicos o *k* que corresponde a constante da mola, o *r*.que representa o vetor de distancia entre o ponto *i* e *j*, e o *l* que é o tamanho natural da mola.

Após feitos os cálculos esse valor se reflete no restante da fórmula descrita anteriormente deslocando os vértices vizinhos, gerando assim uma deformação do ponto.

$$\frac{k_{ij}(|r_{ij}| - l_{ij})}{|r_{ij}|} r_{ij} \qquad (10)$$

Quando o resultado deste cálculo é um valor maior que zero significa que esta força deve ser propagada para as próximas camadas, caso contrário significa que a camada atual é a última camada a ser processada.

Em resumo, o número de camadas afetadas pela deformação depende da força externa aplicada e como essa força é propagada pelas molas, sendo que o ponto onde a força das molas se aproxima de zero indica a última camada da deformação.

Após considerar todos os parâmetros envolvidos é, então, realizado o cálculo da deformação sendo que os resultados e a atualização do objeto na cena são apresentados em tempo de execução.

5.1.7 Exemplo de Execução do Sistema DefApliMed

Para exemplificar o funcionamento da implementação do sistema *DefApliMed* foi realizado um teste específico para se obter os valores das coordenadas dos vértices envolvidos na deformação em cada intervalo de tempo, ou seja, desde o momento anterior à deformação até a posição final do vértice após a deformação. Os testes foram aplicados em um objeto importado, simulando uma mama.

Como explicado anteriormente, quando o usuário interage com o objeto 3D, o local selecionado pode não ser um vértice específico e sim algum lugar referente à face do objeto 3D. Neste exemplo específico o local da interação foi referente às coordenadas (-0.12154512698399841, 0.14277891415463606, 0.5363563711998138) e o vértice mais próximo a estas coordenadas é o ponto (-0.12829181738197803, 0.13247904367744923, 0.5383584648370743). A distância entre o local onde ocorreu a interação para o vértice mais próximo é de (0.01247451570624584), denominado como erro.

O erro é o cálculo da distância entre o vértice mais próximo e o local selecionado. Seu cálculo é necessário a fim de determinar a precisão do método,visto que esta é uma característica extremamente desejável em aplicações de RV para treinamento médico.

A Tabela 4 apresenta os resultados do cálculo do reposicionamento do vértice escolhido em cada intervalo de tempo.

É importante destacar na Tabela 4 que "Início" é o ponto escolhido pelo usuário sendo os cálculos iniciados somente a partir do instante 0.0s.

Tabela 4 – Cálculo das posições do vértice escolhido

Tempo	Coordenadas
Início	(-0.121545127, 0.142778914, 0.536356371)
0.0	(-0.012799282, 0.013217028, 0.053460665)
0.5	(-0.012739592, 0.013155381, 0.052711163)
1.0	(-0.012620428, 0.013032295, 0.051213868)
1.5	(-0.012382467, 0.012786489, 0.048222575)
2.0	(-0.011907091, 0.012295454, 0.04224571)
2.5	(-0.010957355, 0.01131449, 0.03030116)

Os valores apresentados na Tabela 4 são referentes ao vértice 323, ou seja, o índice deste ponto no vetor de vértices é 323. Nesta posição estão armazenadas as coordenadas *x*, *y* e *z* deste vértice no espaço, que correspondem aos valores -0.121545127, 0.142778914, 0.536356371, respectivamente.

Esses valores descritos equivalem a um instante inicial em que ainda não ocorreu nenhuma deformação e são apresentados na Tabela 4 na linha *"Início*". As linhas seguintes

da tabela apresentam os valores do cálculo da deformação para instantes de tempo entre 0 e 2.5 segundos. Este tempo é referente ao tempo de aplicação da força e é pré-estipulado devido ao fato da ausência de dispositivos hápticos, uma vez que estes obtêm estes valores por meio da interação com o usuário.

Os vértices que fazem parte das camadas também devem ser recalculados para se obter uma posição final relacionada com o cálculo da deformação. Os resultados desses cálculos são apresentados nas tabelas a seguir, as quais apresentam as coordenadas de um momento inicial sem deformação e no momento final, após a deformação. Na Tabela 5 observa-se os vértices pertencentes à primeira camada, com suas respectivas coordenadas no momento inicial e final.

Tabela 5 – Reposicionamento dos vértices na camada 1.

Vértice	Inicio – sem deformação	Final da deformação
644	(-0.009149485, 0.015102698, 0.05373898)	(-0.007833861, 0.012923225, 0.04598423)
645	(-0.01604858, 0.011591789, 0.05354823)	(-0.013731719, 0.009922247, 0.045820337)
1069	(-0.011700768, 0.010573569, 0.05451884)	(-0.0100157065, 0.00905421, 0.046647537)
1075	(-0.014155584, 0.015551468, 0.05278442)	(-0.0121137975, 0.013308155, 0.04516751)
1509	(-0.007807069, 0.0120730605, 0.05465527)	(-0.006686539, 0.010333444, 0.046768464)
1515	(-0.017194351, 0.0136435535, 0.052723568)	(-0.014714153, 0.011677092, 0.04511725)

De maneira similar, a Tabela 6 e a

Tabela 7 apresentam os vértices pertencentes à segunda e à terceira camadas, respectivamente. Percebe-se que aumentou a quantidade de vértices nessas camadas, pois essas são formadas por vértices conectados aos vértices da camada imediatamente anterior.

Tabela 6 – Reposicionamento dos vértices na camada 2.

Vértice	Início	Final da deformação
42	(-0.0052172756, 0.017188616, 0.053488255)	(-0.005047997, 0.016627107, 0.051742677)
53	(-0.018779216, 0.010287124, 0.0533984)	(-0.018164346, 0.009952008, 0.051654752)
158	(-0.0108827595, 0.0077402843, 0.05519138)	(-0.010527819, 0.007488971, 0.05338903)
159	(-0.015808798, 0.01757241, 0.051697586)	(-0.0152917, 0.016997522, 0.050010126)
289	(-0.0023614513, 0.010023883, 0.055381052)	(-0.0022851895, 0.0096971365, 0.053573947)
312	(-0.021013837, 0.013716551, 0.051780384)	(-0.020326173, 0.013268796, 0.050090376)
577	(-0.003613816, 0.013755554, 0.05452055)	(-0.0034976942, 0.013306219, 0.05274031)
622	(-0.019763544, 0.012099503, 0.052656136)	(-0.019114379, 0.011704758, 0.050935302)
1068	(-0.0067710807, 0.008802988, 0.055420574)	(-0.0065512094, 0.008517079, 0.05360991)
1074	(-0.018620972, 0.015447725, 0.051752858)	(-0.018009955, 0.0149418, 0.05006233)
1510	(-0.015160271, 0.009297268, 0.054261662)	(-0.01466362, 0.008995275, 0.05248756)

1512	(-0.010762056, 0.017746601, 0.052696113)	(-0.010410628, 0.017164899, 0.050974842)

Tabela 7 – Reposicionamento dos vértices na camada 3.

Vértice	Início	Final da deformação
576	(-0.0015034903, 0.006164217, 0.056037948)	(-0.0014824584, 0.0060780654, 0.05525397)
578	(-0.007128564, 0.020152535, 0.052316025)	(-0.00702881, 0.019870492, 0.051584143)
601	(-0.0011207655, 0.01941458, 0.052812394)	(-0.0011051015, 0.019142974, 0.05207358)
621	(-0.01805108, 0.008284388, 0.054034162)	(-0.017798329, 0.00816868, 0.053278055)
623	(-0.02253972, 0.015133277, 0.050744098)	(-0.02222405, 0.014921483, 0.050034203)
688	(-0.020873219, 0.009363166, 0.05340669)	(-0.020580877, 0.009232274, 0.052659422)
1026	(0.002186807, 0.011383928, 0.054916337)	(0.0021562297, 0.011224661, 0.054148097)
1070	(-0.014537155, 0.0068549966, 0.05484994)	(-0.014333651, 0.006759267, 0.054082472)
1071	(-0.010368052, 0.0048365192, 0.055709496)	(-0.010223003, 0.0047689746, 0.054930083)
1072	(-0.012695258, 0.02000251, 0.051476493)	(-0.012517603, 0.019722564, 0.050756365)
1073	(-0.0177982, 0.01927565, 0.050406404)	(-0.017549112, 0.019005926, 0.049701255)
1116	(-0.02286936, 0.012466794, 0.051918257)	(-0.022548871, 0.012292479, 0.05119177)
1466	(7.444207E-4, 0.01558068, 0.053964276)	(7.340186E-4, 0.015362705, 0.053209353)
1467	(0.0031674649, 0.007017339, 0.05562769)	(0.0031231965, 0.006919156, 0.05484949)
1508	(-0.0060778037, 0.005437563, 0.05601051)	(-0.0059927986, 0.0053615975, 0.055226896)
1514	(-0.020352058, 0.016999708, 0.050601263)	(-0.020067118, 0.016761791, 0.049893383)
1556	(-0.02174865, 0.010995848, 0.052724767)	(-0.021443842, 0.010842083, 0.051986992)
1557	(-0.024239823, 0.013769494, 0.050964538)	(-0.023900436, 0.013576859, 0.05025156)

Na Figura 68 é possível visualizar a evolução dos cálculos de reposicionamento dos vértices no objeto 3D, considerando os valores apresentados nas tabelas anteriores. Nota-se que, na Figura 68 a partir do instante de tempo 0.0s até o instante de tempo 1.0s, visualmente não é possível perceber a ocorrência da deformação no objeto, sendo esta comprovada apenas por meio dos valores obtidos nas tabelas anteriores.

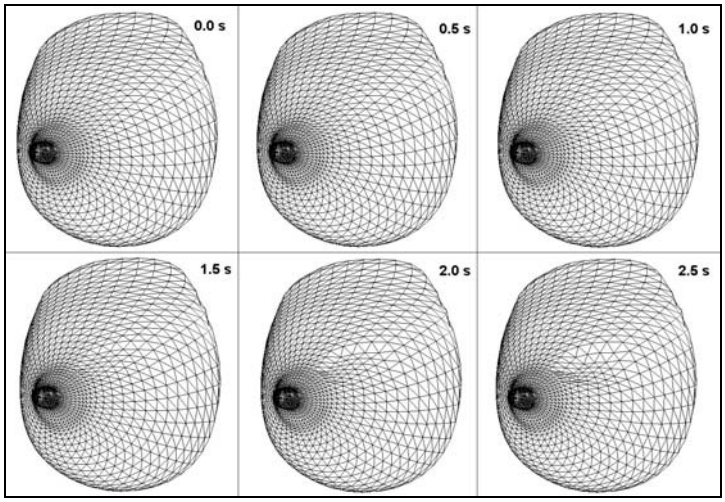


Figura 68 – Evolução da deformação nos tempos determinados

5.2 Conclusão

Neste capítulo, foi descrito o projeto e a implementação de um conjunto de classes que permite a deformação em objetos 3D, na qual utilizou-se implementação orientada a objetos objetivando o reuso de código em futuros projetos.

As classes descritas se comunicam por meio de interfaces bem definidas, sendo possível dessa forma à inclusão de outros métodos de deformação além do método massa-mola, sem que sejam necessárias profundas alterações nos códigos que implementam as classes desenvolvidas.

Foram executados diversos testes para a verificação do sistema, mesmo após sua conclusão. Estes testes serão apresentados no próximo capítulo descrevendo seus resultados, a determinação dos valores dos parâmetros relacionados à fórmula da deformação, bem como as dificuldades encontradas no decorrer do desenvolvimento do sistema.

6. RESULTADOS E DISCUSSÕES

Os testes realizados no sistema permitiram verificar a funcionalidade e também observar algumas dificuldades relacionadas à implementação.

As próximas seções apresentam os critérios utilizados para a realização dos testes, bem como os resultados obtidos e as discussões em relação a esses resultados.

6.1 Critérios para Realização dos Testes

O sistema *DefApliMed* permite a execução da tarefa de deformação de forma simplificada, ou seja, fornece os métodos e cálculos para que o processo de deformação possa ser desenvolvido por meio de estudos de casos específicos, permitindo alcançar valores realísticos de acordo com o objeto modelado em questão.

Para execução dos testes, valores empíricos foram atribuídos aos parâmetros relacionados à força e o número de vértices do objeto modelado.

A força é um parâmetro que varia conforme a intensidade da interação do usuário. Os testes com a variação deste parâmetro visam simular diferentes intensidades de interação que podem ser obtidos, por exemplo, por meio de equipamentos não convencionais, como dispositivos hápticos ou luvas.

A quantidade de vértices de um objeto permite a visualização de maior ou menor nível de detalhe envolvendo as deformações. Este valor é um fator crítico quando consideradas deformações em tempo real, uma vez que se não houver cuidados para definição

do número de vértices utilizados na modelagem do objeto, o sistema perde em desempenho podendo não obter resultados aplicáveis em tempo real.

A definição automática do número de camadas deve levar em consideração a força envolvida, os parâmetros que definem o comportamento da estrutura do objeto (constante da mola, constante de amortecimento e a massa) e a distância entre os vértices. Essa distância está relacionada com a quantidade de vértices existentes no objeto modelado, ou seja, se os vértices de um objeto estiverem muito próximos, o número de camadas deve ser maior do que quando considerado o mesmo objeto com menos vértices envolvidos, para obter a mesma área de deformação.

Isto é determinado através do cálculo da equação 10, citado anteriormente, que define através da força envolvida nas molas e as características do objeto, definidas pelos parâmetros, a área em que a deformação ocorreu.

Existem alguns parâmetros que são definidos e permanecem fixos durante todos os testes: a constante da mola (K), a constante de amortecimento (D) e a massa (M), que são valores específicos de cada objeto. Para se obter uma deformação mais realística é necessária a realização de estudos de caso para obtenção desses valores.

Um exemplo é o estudo necessário para a definição dos valores da constante da mola, que podem ser definidos por meio de um estudo com equipamentos apropriados para a medição de pequenas forças sobre um objeto real.

Assim, para a realização dos testes foi estabelecida uma padronização para fornecer os valores referentes a estes últimos parâmetros. O padrão de unidades adotado foi o sistema internacional MKS, que adota como medidas padrões: Metros, Quilogramas, Segundos e Newtons.

Os testes apresentados foram obtidos utilizando um computador com processador Pentium4 de 3.2GHz, com 512 MB de memória RAM e placa de vídeo Inno3D Gforce FX 5200.

Para permitir a realização dos testes é importante destacar que a interação do usuário com a ferramenta permite a deformação em um dos vértices do objeto, simulando deformação de profundidade na superfície do mesmo, ou seja, uma depressão, por meio do deslocamento das posições dos vértices envolvidos no procedimento de deformação.

A primeira etapa dos testes realizados consistiu na modelagem de objetos primitivos disponibilizados na ferramenta *3D Studio Max* e exportados para a API Java 3D. Em uma segunda etapa foram aplicados testes em objetos modelados que representam partes do corpo humano, tais como a mama e as nádegas.

6.2 Testes com Objetos Primitivos

Para testar o sistema, inicialmente foi implementada uma aplicação que possui uma malha poligonal formada por vértices, não considerando volume ou massa real.

A função de deformação foi implementada com base nos valores inseridos pelo usuário. Um exemplo da função de deformação é apresentado na Figura 69 que podem ser visualizados dois testes realizados sobre a malha poligonal, sendo que, a Figura 69 (A) apresenta esta malha sem nenhuma deformação. Neste contexto a deformação pode ocorrer em duas situações, deformação em profundidade, apresentada na Figura 69 (B) e deformação de relevo apresentada na Figura 69 (C).

É importante salientar que para obter uma deformação de profundidade foi realizada a aplicação de uma força com valor negativo, já para uma deformação de relevo a força aplicada possui um valor positivo.

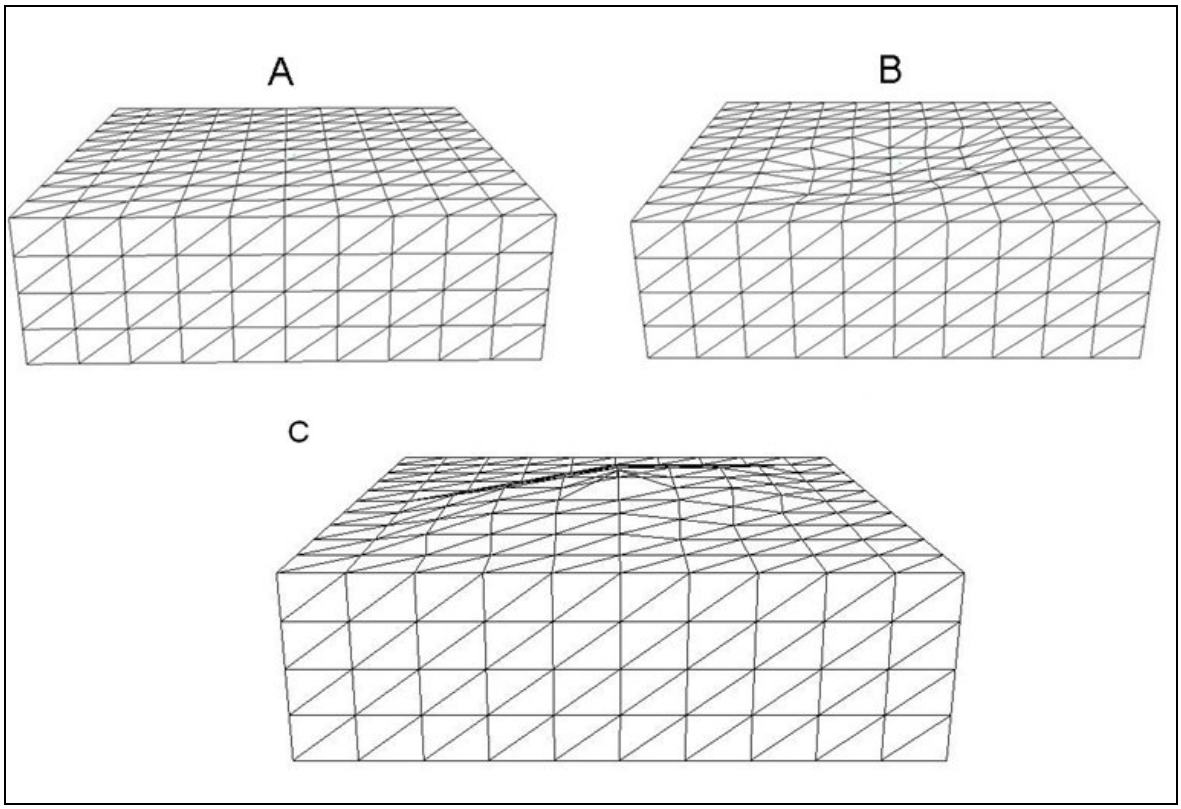


Figura 69 – Exemplo de deformação em um malha poligonal: (a) malha sem deformação; (b) e (c) malha com deformação de profundidade e relevo.

Para esta aplicação utilizando uma malha poligonal, o resultado apresentado foi visualmente satisfatório, indicando que o sistema poderia ser submetido a novos testes em primitivas mais complexas até que se chegasse ao nível de testes utilizando objetos representando partes do corpo humano, modeladas para ser empregada em aplicações de treinamento médico em tempo real.

Após os testes realizados com a malha poligonal foram realizados testes em primitivas básicas com quantidades de vértices variados para testar o desempenho do sistema e a variação de tempo de processamento considerando estruturas diferentes. Sendo assim testes aplicando diferentes valores para a força foram realizados.

A Figura 70 apresenta o resultado dos testes aplicados em uma primitiva do tipo cone. A força foi empregada no eixo *z* (seguindo o sistema de coordenadas tridimensionais) para obter uma deformação em profundidade, sendo o valor da força igual a -9.81 Newtons. Este valor foi escolhido empiricamente e equivale a 1KGF (quilograma força) (WIKIPEDIA, 2006). Foram realizadas variações na quantidade de vértices e faces do objeto para verificar os resultados com níveis diferentes de detalhamento

Na Figura 70 (A) o número de vértices do objeto é de 481 e de faces é 912. Na Figura 70 (B) o número de vértices e faces do cone são 2401 e 4752, respectivamente, e, por fim, a Figura 70 (C) apresenta um objeto constituindo de 4801 vértices e 9552 faces.

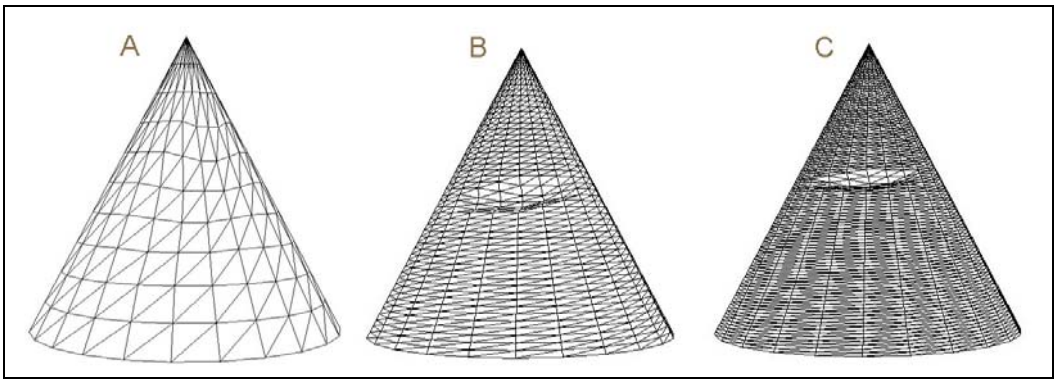


Figura 70 – Deformação aplicada em um cone com quantidade de vértices e faces variadas.

Estes testes foram realizados utilizando uma força que resultou em um número de camadas igual a dois, ou seja, a área atingida com a força empírica aplicada atingiu apenas 2 camadas para obter a deformação. Como pode ser visualizado na Figura 70 a área onde ocorre a deformação foi influenciada pela quantidade de vértices que compõem o objeto, ou seja, quanto maior a quantidade de vértices para representar o mesmo objeto, menor será a área afetada pela deformação, pois os vértices estão a uma distância muito menor em relação a objetos que possuem poucos vértices.

É interessante notar que a deformação ocorrida no cone tende a se propagar mais horizontalmente do que verticalmente. Isso ocorre, porque a forma de modelagem do objeto

influencia no comportamento das molas. Este cone possui vértices mais espaçados na horizontal do que na vertical influenciando, assim, no resultado da deformação. Para a modelagem de objetos do corpo humano é necessário um cuidado especial para que os vértices estejam igualmente distribuídos, evitando, assim, este efeito.

Da mesma forma, foram realizados testes no mesmo objeto variando o número de camadas, considerando o mesmo valor anterior para o parâmetro força. A Figura 71 apresenta a seqüência dos resultados variando o número de camadas.

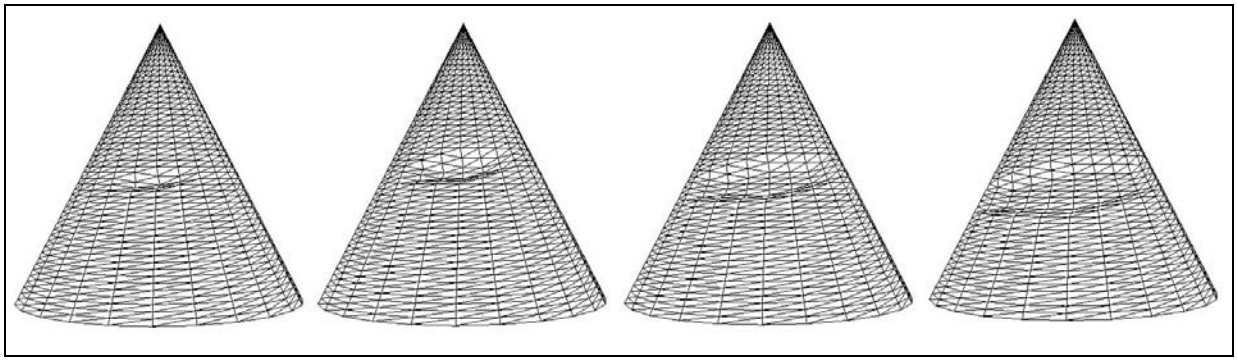


Figura 71 – Deformação no cone variando-se o número de camadas.

Outra possibilidade para alterar resultados de deformação é a aplicação de diferentes valores para a força. A Figura 72 apresenta a diferença da deformação no cone utilizando diferentes valores para a força, sendo que na Figura 72 (A) foi utilizada a força de -9.81 Newtons e na Figura 72 (B) foi utilizada a força de -19.62 Newtons.

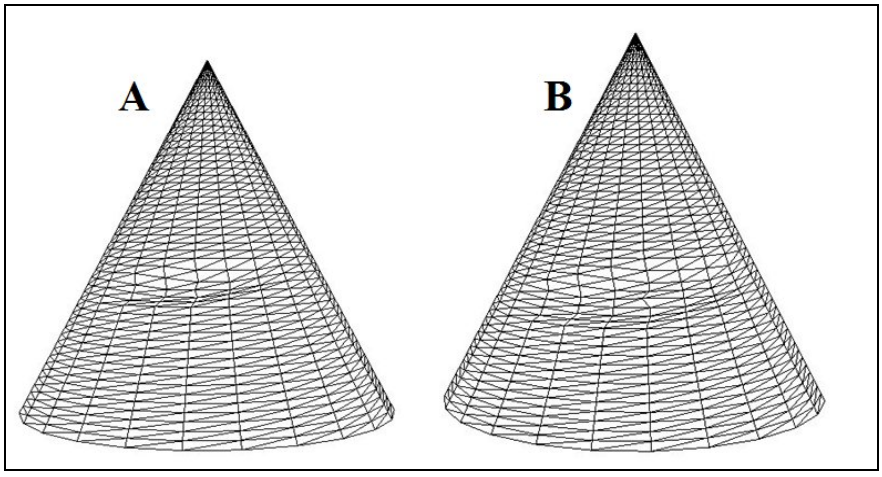


Figura 72 – Diferença na deformação dos objetos variando o valor da intensidade da força

Percebe-se na Figura 72 que foi utilizando a força de -19,62 Newtons obteve-se uma deformação mais profunda do que quando aplicada à força de -9.81 Newtons, comprovando assim a influência da força para a deformação.

Os testes anteriores foram repetidos com a primitiva esfera, também fornecida pela API Java 3D, utilizando os mesmos parâmetros empregados para a primitiva cone, ou seja, parâmetros de força, massa, constante de amortecimento e constante da mola.

Da mesma forma que a intensidade da força varia para a primitiva cone este fato ocorreu na primitiva esfera, diferenciando apenas os eixos de aplicação da força, uma vez que na esfera foi mesclada à intensidade das forças nos eixos X e Z, permitindo deformação em outras direções.

Por exemplo, para deformar determinada posição de uma esfera a fim de formar uma depressão na superfície é necessário combinar forças em dois eixos. A Figura 73 apresenta os resultados utilizando a força de -9.81 Newtons nos eixos X e Z.

Foram utilizados níveis de detalhamento na criação do objeto, isto é, foram variadas quantidades de vértices e faces. Sendo assim os valores das vértices e das faces da primitiva esfera foram respectivamente: 482 vértices e 960 faces, 1202 vértices e 2400 faces e 4902 vértices e 9800 faces, apresentados nas Figura 73 (A), Figura 73 (B) e na Figura 73 (C).

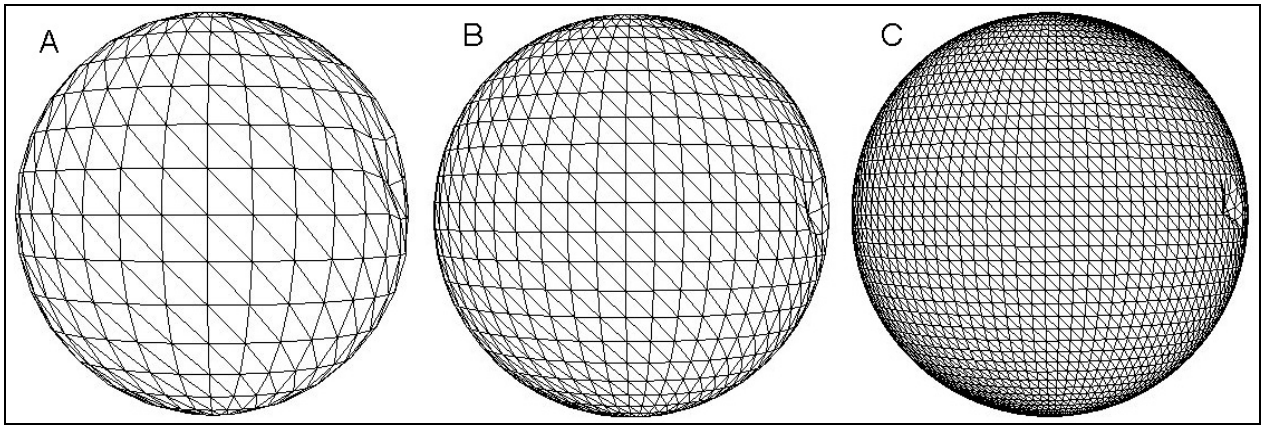


Figura 73 – Deformação da primitiva esfera variando a quantidade de vértices e faces.

Testes para verificar a influência da força na quantidade de camadas na deformação também foram realizados em relação à primitiva esfera, ou seja, para estes testes foram utilizadas forças que resultassem nas deformações apresentadas na Figura 74. Como pode ser observado que o número de camadas varia em função da força aplicada.

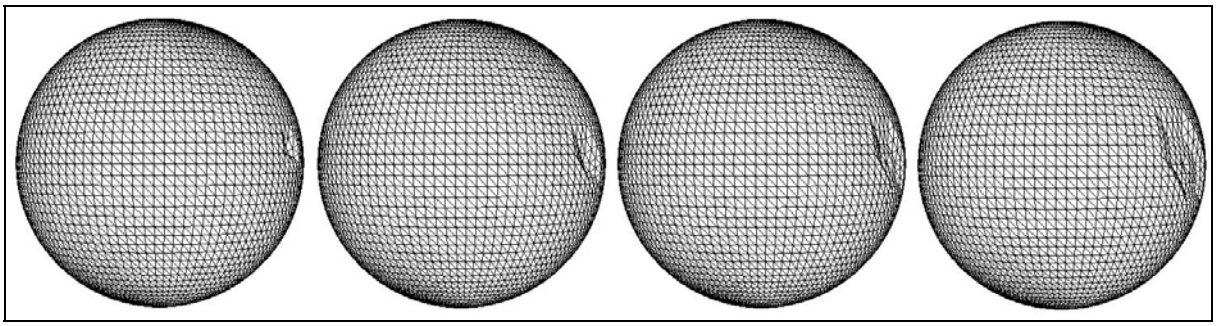


Figura 74 – Deformação nos eixos X e Z da primitiva esfera alterando o número de camadas.

Através dos testes realizados foi possível verificar que o sistema utilizando o método massa-mola obteve resultados satisfatórios para as primitivas cone e esfera, visto que foi notória a variação dos resultados em função da variação dos parâmetros.

6.3 Testes com Objetos Modelados com Formato do Corpo Humano

Nesta fase foram realizados testes com objetos considerados mais complexos e que estivesse no contexto de treinamento médico, uma vez que este é o propósito deste trabalho.

Os objetos que serviram de testes relacionados ao corpo humano foram modelados na ferramenta de modelagem *3D Studio Max* (3D Studio, 2006).

Seguindo este propósito, os objetos modelados escolhidos para testes iniciais foram: as nádegas e a mama, pois como o trabalho está relacionado com procedimentos MI, são duas regiões do corpo humano que permitem procedimentos que envolvem inserção de agulha.

Esses objetos estão apresentados na Figura 75 e os testes foram executados com a mesma configuração de máquina utilizada nos testes com primitivas, descritos anteriormente.

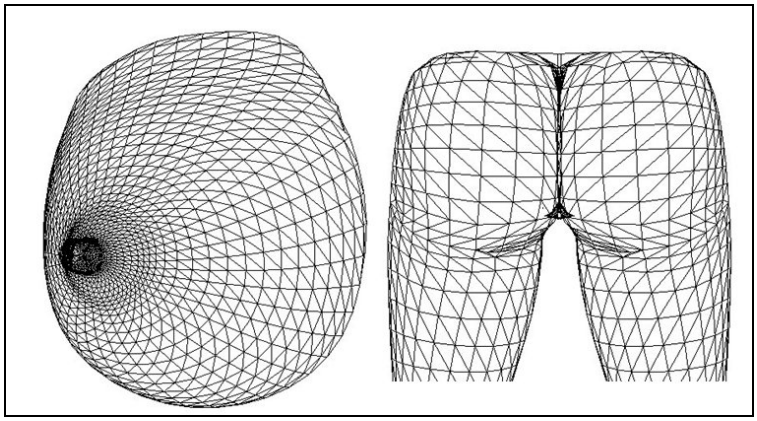


Figura 75 – Mama e Nádegas modeladas, importadas para o sistema *DefApliMed*

A metodologia para a execução dos testes obedece ao mesmo padrão dos testes com primitivas, variando-se o valor referente à força, e conseqüentemente, o número de camadas envolvidas e a quantidade de vértices e arestas.

No caso de objetos que representam o corpo humano é necessário um cuidado especial quanto às medidas do objeto modelado. Os objetos utilizados nestes testes não foram modelados seguindo proporções reais. Assim, tornou-se necessário alguns ajustes na escala do objeto, no momento da importação, a fim de permitir que estes se enquadrassem em medidas próximas às reais.

Os primeiros testes foram realizados para verificar a influência da força no procedimento de deformação. Os valores utilizados foram: -3, -1.5, -0.7 e -0.3 Newtons. Esses valores forma escolhidos de forma empírica até chegar a um resultado visual semelhante ao fenômeno de deformação que ocorre em objetos reais. O primeiro teste foi realizado como uma força de grande intensidade e esta força foi sendo reduzida gradativamente até se obter um resultado próximo ao esperado.

É importante destacar que os valores são negativos, pois, como citado anteriormente, pretende-se obter uma deformação de profundidade. Como no sistema de coordenadas 3D o

objeto está na posição central do AV, ou seja, na posição (0,0,0) o eixo que permite a sensação de profundidade é o eixo z. Os eixos x e y fazem com que a deformação seja direcionada horizontalmente e verticalmente, respectivamente, o que não resulta em uma deformação realística visível.

Os parâmetros massa, constante de amortecimento e constante da mola são valores fixados, porque como dito anteriormente, para que se tenha a variação é necessário um estudo aprofundando das características dos tecidos que compõem os órgãos humanos.

Para realizar testes no objeto nádegas os parâmetros massa, constante de amortecimento e constante da mola foram pré-estipulados com valores fixos equivalentes a 500g (gramas), 0.9N/m/s (Newtons por metro por segundo) e 0.1 N/m (Newtons por metro), respectivamente.

Foram pré-estipulados alguns valores empíricos para o parâmetro força que correspondem a: -0.3, -0.7 e -1.5 Newtons. Como nos testes das primitivas pode ser observado que à medida que o valor da força aplicada à área a ser deformada vai aumentando o número de camadas atingidas vai sendo incrementado.

A Figura 76 e a Figura 77 mostram testes variando a força aplicada e, conseqüentemente a quantidade de camadas atingidas e a deformação dependendo da quantidade de vértices existentes no objeto modelado.

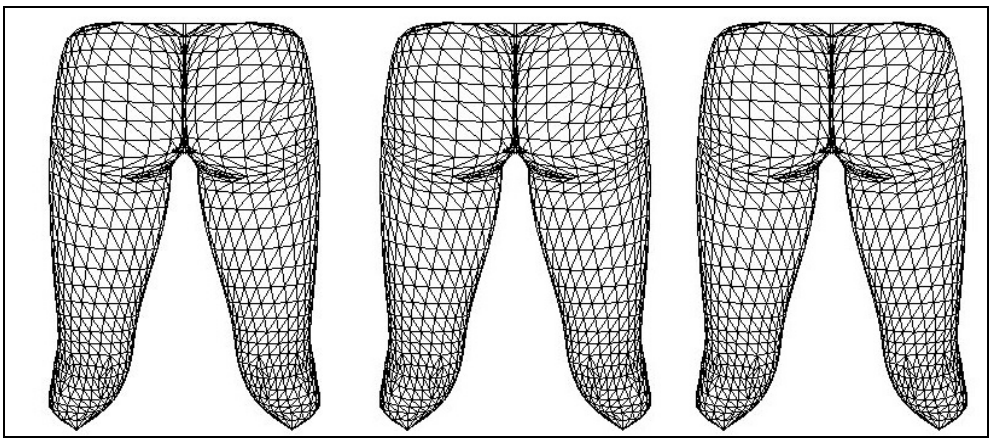


Figura 76 – Deformação obtida devido à variação da força aplicada no objeto nádegas.

Outro teste realizado foi verificar a quantidade de vértices envolvido no objeto para gerar a deformação. A Figura 77(A) apresenta um objeto nádegas que possui 472 vértices e 922 faces e na Figura 77(B), o objeto possui 1890 vértices e 3732 faces.

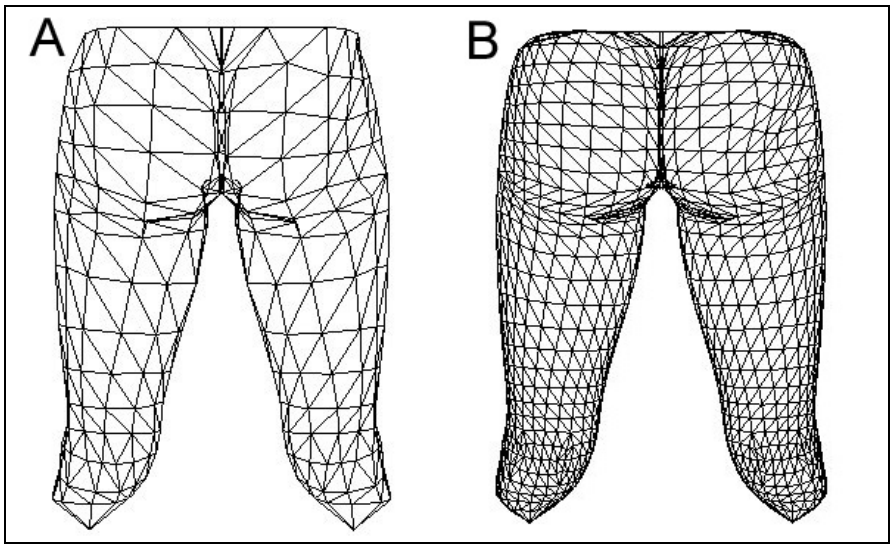


Figura 77 – Variação da quantidade de vértices e faces do objeto nádegas.

O próximo objeto a passar por alguns testes, como os apresentados anteriormente, é o objeto com o formato de uma mama. Neste caso, os valores determinados para os parâmetros da fórmula são diferentes dos aplicados para o objeto nádegas, porém continuam sendo empregados valores empíricos. Os parâmetros massa, constante de amortecimento e constante da mola foram pré-estipulados com valores fixos equivalentes a 300g (gramas), 0.7N/m/s (Newtons por metro por segundo) e 0.3 N/m (Newtons por metro), respectivamente. Considerando o objeto mama, por exemplo, dever-se-ia verificar a massa envolvida na sua deformação, a tonicidade da mama usada no treinamento médico e o amortecimento necessário para gerar uma deformação realística.

A Figura 78 apresenta uma comparação da deformação aplicando diferentes valores para a força como descrito. Pode-se perceber que a Figura 78(A), em que a força aplicada foi de -3 Newtons, obteve uma deformação muito profunda, ou seja, fora do contexto de uma aplicação de treinamento médico, pois uma força de intensidade 3N gera uma deformação

visto que em aplicações médicas geralmente utilizam-se pequenas forças para obter deformações específicas no local desejado.

Outros testes foram realizados diminuindo-se a intensidade da força e obtendo-se deformações com menores dimensões, como pode ser visualizado na Figura 78(B), Figura 78(C) e Figura 78(D), considerando a aplicação das forças de -1,5, -0,7 e -0,3 Newtons, respectivamente.

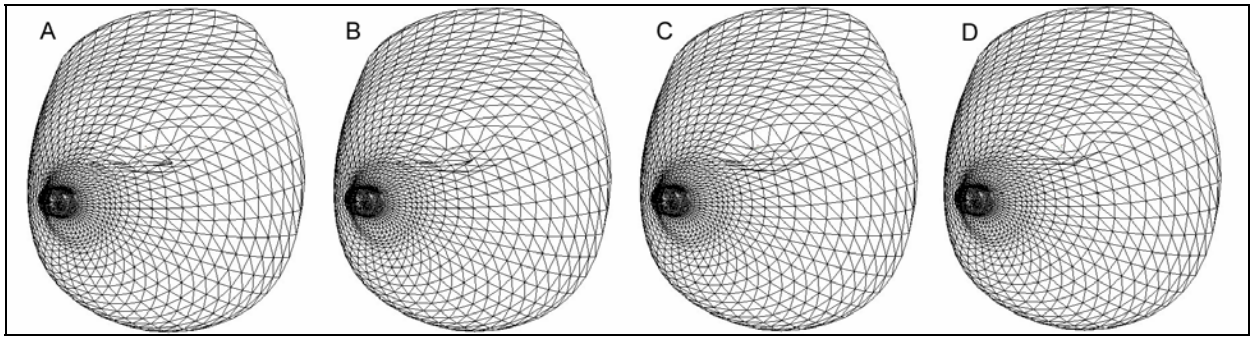


Figura 78 – Exemplo de teste com variação da força

A Figura 78 mostra a variação da quantidade de camadas em função da força aplicada. Na Figura 78(A) houve aplicação de um força de -3N e o número de camadas atingidos foi cinco; a Figura 78(B) utilizou -1.5N para a força e o número de camadas foi de 4, a Figura 78(C) utilizou -0.7N para força e o número de camadas foi de 4. Nota-se que a quantidade de camadas atingidas na Figura 78(B) e Figura 78(C) são iguais, isso devido ao fato da quantidade de força aplicada ser muito próximo não variando assim a área atingida. A Figura 78(D) utilizou a força -0.3N e o número de camadas envolvido foi de 3. Isso comprova que a força aplicada influência diretamente no número de camadas que são afetadas pela deformação.

Em seguida foram realizados testes com a mama modelada variando o número de vértices e faces. A Figura 79 apresenta a deformação ocorrendo na mama com 442 vértices e

880 faces e a Figura 79(B) apresenta a deformação ocorrendo na mama com 1762 vértices e 3520 faces.

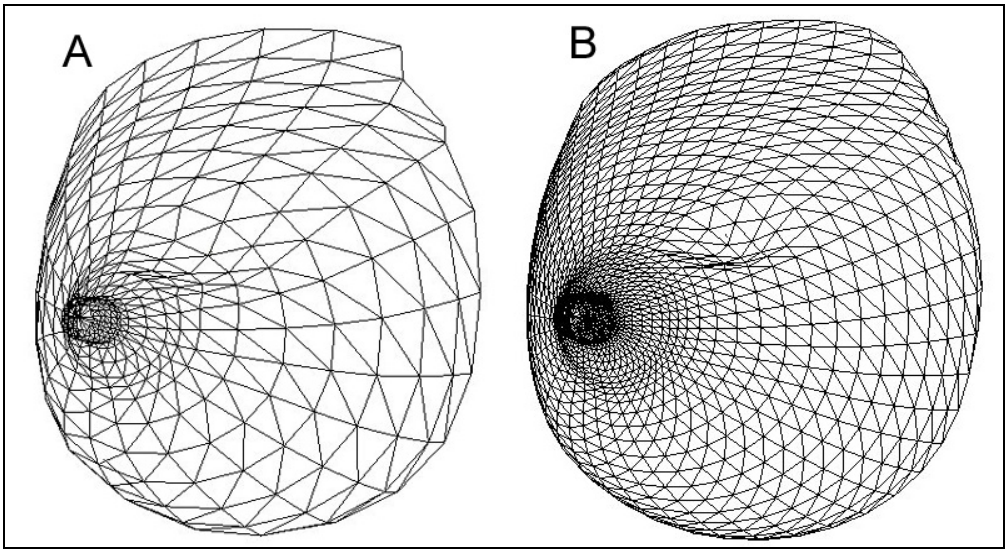


Figura 79 – Deformação na mama modelada com variação de vértices.

Pode-se notar que quanto maior o número vértices, mais realista é a deformação. É importante destacar, ainda, o aumento que a quantidade de vértices e faces causa na área de deformação envolvida, representada pelas camadas. A deformação em objetos com maior quantidade de vértices fica mais restrita resultando em uma deformação com área menor do que quando é utilizado o objeto com menos vértices.

Um ponto importante a ser verificado no sistema é que para todos os parâmetros foram utilizados valores empíricos. Essa dificuldade foi verificada porque são necessários estudos aprofundados sobre a composição e a resistência de cada órgão modelado para se obter resultados reais, fazendo com que a massa varie no sistema com valores corretos. No entanto, por meio dos exemplos anteriores, foi possível perceber que o método implementado permite a variação de parâmetros. Assim, uma vez obtidos valores reais, basta fornecê-los ao sistema e, se necessário, adequá-los também à modelagem do objeto.

6.3.1 Desempenho do Sistema

A Tabela 8 e a Tabela 9 apresentam os valores obtidos de camadas e tempo em relação aos objetos modelados aplicando os parâmetros empíricos determinados anteriormente e traz resultados relacionados ao desempenho do sistema e quantas camadas foram atingidas em função da força aplicada.

Tabela 8 – Resultados da deformação com valores empíricos realizados nas nádegas

	nº	nº				Força			nº		
Objeto	**vértices**	**faces**	**Massa**	**Amort/to**	**K**	**X**	**Y**	**Z**	**Camadas**	**Tempo**	**Desvio padrão**
Nádegas	474	922	500	0,9	0,1	0	0	-0,3	4	**0,25**	**0,01270214**
Nádegas	474	922	500	0,9	0,1	0	0	-0,7	4	**0,28**	**0,00868012**
Nádegas	474	922	500	0,9	0,1	0	0	-1,5	4	**0,25**	**0,00984322**
Nádegas	474	922	500	0,9	0,1	0	0	-3	5	**0,25**	**0,01078116**
Nádegas	1890	3732	500	0,9	0,1	0	0	-0,3	4	**2,6**	**0,05883914**
Nádegas	1890	3732	500	0,9	0,1	0	0	-0,7	5	**2,6**	**0,12823542**
Nádegas	1890	3732	500	0,9	0,1	0	0	-1,5	5	**2,6**	**0,10242613**
Nádegas	1890	3732	500	0,9	0,1	0	0	-3	6	**2,7**	**0,02630357**
Nádegas	2366	4676	500	0,9	0,1	0	0	-0,3	4	**1,55**	**0,01547938**
Nádegas	2366	4676	500	0,9	0,1	0	0	-0,7	4	**1,58**	**0,15327147**
Nádegas	2366	4676	500	0,9	0,1	0	0	-1,5	5	**1,26**	**0,18573784**
Nádegas	2366	4676	500	0,9	0,1	0	0	-3	6	**2,14**	**0,03863346**

Tabela 9 – Resultados da deformação com valores empíricos realizados na mama

	nº	nº				Força			nº		
Objeto	**vértices**	**faces**	**Massa**	**Amort/to**	**K**	**X**	**Y**	**Z**	**Camadas**	**Tempo**	**Desvio padrão**
Mama	1762	3520	300	0,7	0,3	0	0	-0,3	5	**1,069**	**0,01318248**
Mama	1762	3520	300	0,7	0,3	0	0	-0,7	4	**1,0826**	**0,042361408**
Mama	1762	3520	300	0,7	0,3	0	0	-1,5	4	**1,0796**	**0,009082339**
Mama	1762	3520	300	0,7	0,3	0	0	-3	4	**1,0532**	**0,01100303**
Mama	2537	5070	300	0,7	0,3	0	0	-0,3	4	**1,4828**	**0,076024557**
Mama	2537	5070	300	0,7	0,3	0	0	-0,7	4	**1,997**	**0,167429587**
Mama	2537	5070	300	0,7	0,3	0	0	-1,5	4	**1,5499**	**0,184498991**
Mama	2537	5070	300	0,7	0,3	0	0	-3	5	**1,2281**	**0,013370365**
Mama	3441	6878	300	0,7	0,3	0	0	-0,3	4	**1,8391**	**0,076024557**
Mama	3441	6878	300	0,7	0,3	0	0	-0,7	4	**2,0689**	**0,167429587**
Mama	3441	6878	300	0,7	0,3	0	0	-1,5	5	**3,1685**	**0,184498991**
Mama	3441	6878	300	0,7	0,3	0	0	-3	6	**4,8374**	**0,013370365**

Estes testes foram realizados utilizando valores empíricos, onde o principal objetivo era gerar a deformação nos objetos em questão (mama e nádegas). Foi possível analisar dois parâmetros importantes: o número de camadas atingidas pela força aplicada no objeto e o tempo de processamento para gerar a deformação.

Como se pode observar na Tabela 8 e na Tabela 9, conforme a força aumenta, uma possível conseqüência é o aumento do número de camadas e esse aumento resulta em um maior tempo de processamento devido a uma maior quantidade de vértices afetados.

A Figura 80 apresenta o gráfico de desempenho do objeto nádegas referente aos resultados da Tabela 8, destacando a influência da força no tempo de processamento no sistema. Como esperado, pode-se perceber que com o aumento da força aplicada pelo usuário e o aumento da quantidade de vértices do objeto ocorre o aumento do tempo de processamento, ou seja, com a variação da força a área de deformação aumenta fazendo com que o número de camadas afetadas também sofra variação, e essa variação tem como conseqüência um aumento do número de vértices afetados pela deformação. Este fator afeta o tempo de processamento, apresentado no gráfico, podendo perceber que quanto maior a força aplicada, maior o tempo de processamento.

No objeto nádegas com 474 vértices e 922 faces a representação do desempenho é dada por uma linha quase reta na horizontal, isso ocorre devido ao fato de com um objeto com um menor número de vértices apresenta menor influência em relação à quantidade de camadas a deformação não se propaga para um grande número de camadas, como pode ser observado na Tabela 8. Por isso, mesmo com a variação da força aplicada não há efeitos significativos sobre o número de camadas.

Ainda observando a Tabela 8 pode-se verificar que conforme a quantidade de vértices aumenta como no caso dos dois objetos nádegas (o primeiro com 1890 vértices e 3732 faces e o segundo com 2366 vértices e com 4676 faces), isto é refletido no número de

camadas atingidas. Isto ocorre devido às características físicas (tonicidade, elasticidade, entre outros) dos objetos modelados definidas através dos parâmetros de deformação. Como resultado, pode-se perceber a variação do tempo de processamento apresentado na Figura 80.

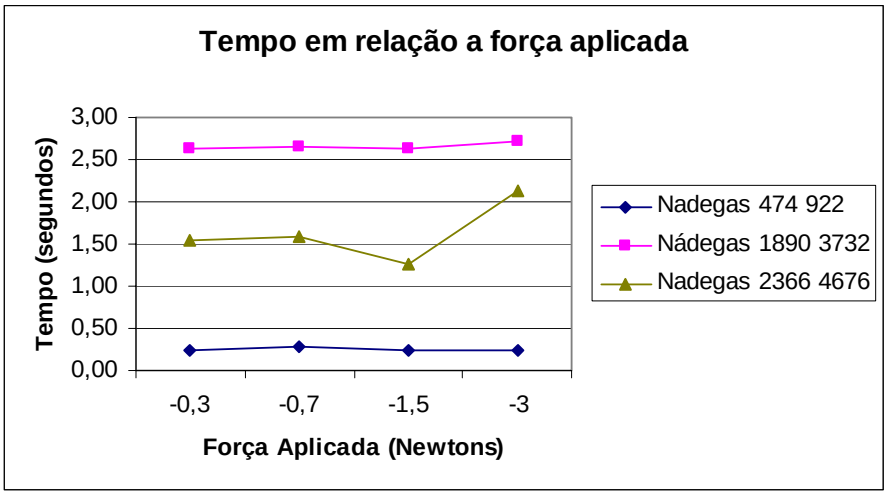


Figura 80 – Impacto da Força no desempenho da deformação no objeto Nádegas

Para o objeto mama os resultados obtidos, que são apresentados na Figura 81, diferem do objeto nádegas, uma vez que as características de estrutura dos objetos e as características físicas modeladas através dos parâmetros de deformação são diferentes, ou seja, os valores utilizados para os parâmetros são diferentes devido às características de estrutura dos objetos e as características físicas modeladas através dos parâmetros de deformação, resultando assim em um efeito de deformação diferenciado.

Estas características físicas e da estrutura do objeto exercem influência no número de camadas e no número de vértices afetados, resultando em tempos de processamento também diferentes. No entanto, é importante observar que uma tendência que ocorre no objeto nádegas se repete para o objeto mama, ou seja, conforme aumenta o número de vértices, mais evidente fica o resultado do número de camadas em relação à força.

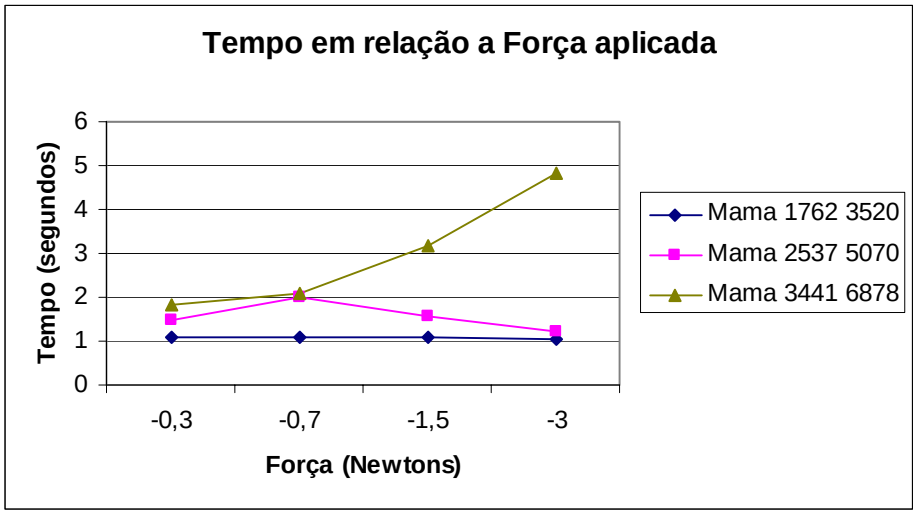


Figura 81 – Impacto da Força no desempenho da deformação no objeto Mama

É importante destacar que, apesar das mesmas forças serem aplicadas aos dois objetos, o número de camadas afetadas não foi o mesmo entre eles. Isto se deve à característica de modelagem dos objetos, sendo que em diferentes regiões a distância entre os vértices e, conseqüentemente, o tamanho das arestas é diferente, resultando em alterações da força da mola envolvida.

6.4 Aplicação de Textura

Foram também realizados testes para a inclusão de textura para a obtenção de um maior realismo para o sistema desenvolvido. Com isto foram estudadas formas para a aplicação de uma determinada textura em um objeto tridimensional, bem como o controle da parte de iluminação do ambiente virtual.

Para prover textura no objeto foram utilizados dois métodos que tratam da imagem, provenientes de Chan (2002).

Como descrito e apresentado no grafo de cena na Figura 55, existe no ambiente virtual duas luzes, sendo uma luz ambiente e uma luz direcional apontada para o objeto. O controle da iluminação é importante, pois a partir da combinação das propriedades do material aplicado no objeto e as condições de luz presentes no ambiente é que será possível uma melhor visualização de detalhes da textura.

Para a aplicação da textura é necessário realizar um mapeamento de uma imagem sobre o objeto tridimensional. Para esta etapa houve dificuldades para realizar este mapeamento no objeto “mama” devido à forma de modelagem empregada no objeto. Já para o objeto “nádegas” não houve este tipo de dificuldade sendo que a textura pôde ser mapeada com sucesso. Entretanto, houve dificuldades na definição de um material que trabalhasse em conjunto com as luzes presentes no ambiente para se gerar um efeito que permitisse a visualização da deformação.

Na Figura 82 é possível perceber que houve a deformação de depressão e o relevo no objeto 3D.

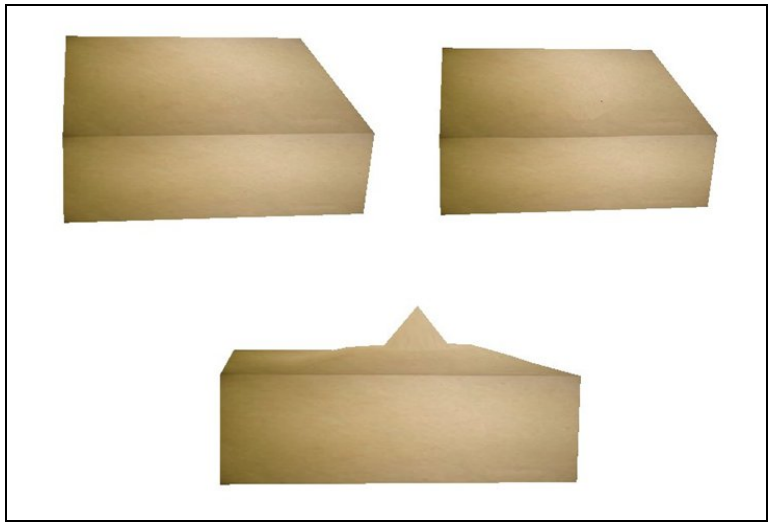


Figura 82 – Textura aplicada em uma malha simples

A Figura 83 apresenta os resultados referentes à utilização de textura nos objetos descritos.

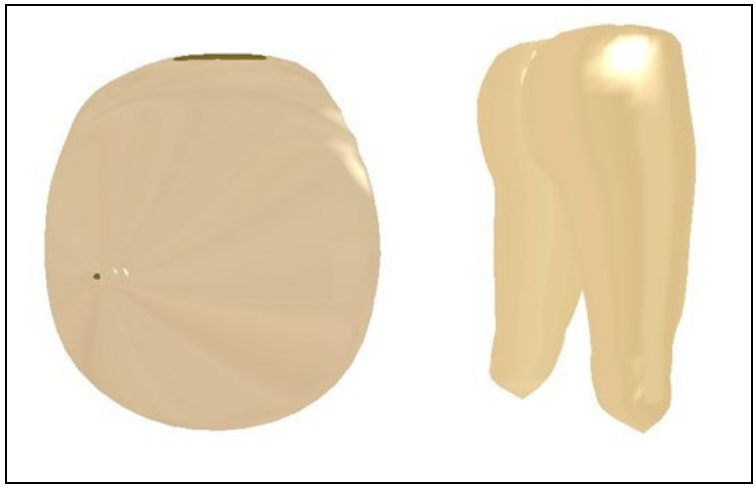


Figura 83 – Resultados obtidos com a utilização de textura

6.5 Aplicação do sistema desenvolvido

O sistema *DefApliMed* foi desenvolvido inicialmente para implementar o método massa-mola e verificar o comportamento deste método em objetos modelados com realismo, a fim de avaliar a sua adequação para aplicações de RV para treinamento médico.

A implementação orientada a objetos, utilizando a linguagem Java faz com que o código possa ser reutilizado em uma proposta de criação de um *framework* para treinamento médico. Este *framework* englobar funcionalidades como detecção de colisão, estereoscopia e a inserção da deformação desenvolvida neste trabalho. Essas funcionalidades são imprescindíveis para que a simulação de procedimentos médicos contenha o maior grau de realismo possível.

A partir deste *framework* será possível a construção de ferramentas para diversos tipos de treinamentos médicos que envolvam técnicas minimamente invasivas, concentrados inicialmente naquelas que podem simular exames de punção.

Outro ponto relevante é permitir que a deformação dos objetos seja feita com uma interação mais adequada, utilizando equipamentos hápticos. Como o foco principal do sistema *DefApliMed* é permitir a deformação do objeto, a interação foi implementada apenas com o mouse e com o teclado, mas devido à estruturação dos códigos utilizando o paradigma orientado a objetos, é possível criar uma classe para essa interação e permitir o uso de dispositivos não convencionais para esse treinamento.

7. CONCLUSÕES E TRABALHOS FUTUROS

A deformação aparece em aplicações de RV para proporcionar maior realismo aos ambientes que os contêm. Sendo assim, é necessário que os objetos se modifiquem em resposta a uma colisão entre objetos, muitas vezes proporcionada a partir de uma interação com o usuário.

A aplicação de deformação tem um campo muito vasto, tanto para pesquisa como para a implementação. O trabalho apresentado é uma das possíveis formas de implementar a deformação em objetos 3D, utilizando o método massa-mola.

Neste trabalho a deformação ocorre através do reposicionamento dos vértices dos objetos modelados na cena virtual, obedecendo ao equacionamento do método citado.

O sistema apresentado forneceu, além do procedimento de deformação, a implementação de uma estrutura de classes que permite a inserção de novos modelos de deformação, ou seja, a estruturação das classes permite o reuso das classes criadas para poder criar outras classes de deformação relacionadas aos métodos citados anteriormente, que são: massa-mola, elemento finito e forma livre de deformação.

Os testes realizados permitiram verificar a adequação do sistema ao objetivo proposto, ou seja, implementar um procedimento de deformação usando o método massa-mola e o desenvolvimento do AV que permite a importação dos objetos modelados. Visto que esses dois tópicos foram descritos no objetivo deste trabalho foi possível implementar uma forma ainda genérica a deformação em objetos que representam partes do corpo humano.

Para que se tenha um treinamento de procedimentos médicos é necessário que se tenha precisão, ou seja, detalhes minuciosos para que o sistema forneça o realismo necessário. Com

os testes foi possível perceber que para se ter precisão é necessário que a modelagem do objeto seja feita com quantidade suficiente de vértices para que se obtenha um realismo.

Outro ponto importante é que o treinamento médico precisa fornecer resposta em tempo real. Isto significa que o sistema deve prover uma resposta rápida à interação do usuário para que seja possível incrementar a sensação de envolvimento do AV.

A principal contribuição deste trabalho é a criação de uma estrutura de classes genérica que permite a inserção de qualquer método de deformação. No caso deste projeto o método escolhido foi o massa-mola, mas caso seja necessária a inserção de outro método de deformação, a estrutura desenvolvida já oferece um suporte para a inclusão de classes relacionadas ao novo método.

Outra contribuição importante está relacionada à importação e manipulação de arquivos 3D na cena virtual, uma vez que a API J3D consegue importar objetos, mas não permite manipular os vértices desse objeto importado na cena. A modificação da classe trará benefícios aos projetos relacionados que estão sendo desenvolvidos no laboratório LApIS, uma vez que, para se alcançar o treinamento médico em alguma parte do corpo humano, é necessário a importação do objeto modelado 3D e a manipulação das suas estruturas no mundo virtual.

Da mesma forma, o método de deformação implementado no trabalho permitirá incluir esta característica em projetos diversos possibilitando treinamentos mais realísticos.

O sistema desenvolvido executa a deformação através do reposicionamento de vértices e arestas do objeto no Ambiente Virtual. A partir de testes realizados em objetos modelados que representam estruturas do corpo humano, foi possível perceber que a implementação realizada é capaz de fornecer a deformação nos objetos, e também permitindo que essa deformação ocorra em um tempo estipulado que permite ao usuário ter a sensação de que a

deformação ocorre em tempo real, características importantes em aplicações de treinamento médico.

7.1 Trabalhos Futuros

Várias diretrizes de pesquisa podem ser delineadas a partir dos resultados obtidos neste trabalho. A primeira diretriz aponta para o estudo da aplicação de textura nos objetos, pois uma vez importados, é necessário aplicar a textura correta e de forma realística para o estudo. Como se pode notar, os primeiros passos para a inserção da textura já foram estudados, porém não foram obtidos resultados satisfatórios para aplicação em treinamentos reais.

Outra diretriz importante é o estudo dos parâmetros do método massa-mola, visando a obter valores reais a partir do estudo de caso de tecidos que compõem os órgãos humanos.

Ainda neste contexto, pesquisas podem ser realizadas para criação de métodos que facilitem a modelagem de objetos com os parâmetros necessários para o funcionamento do método implementado.

Outro trabalho relacionado a este é a inclusão de equipamentos não convencionais, uma vez que no trabalho presente foram utilizados somente dispositivos convencionais. Estes dispositivos permitirão a interação mais adequada com o usuário e também a obtenção de parâmetros para fornecer parâmetros mais reais para a aplicação.

Como citado anteriormente, o trabalho possui uma estrutura genérica. Desta forma, sendo assim a implementação de outros métodos de deformação, já citados na literatura, consiste em possibilidades a serem desenvolvidos a fim de que possam ser realizados estudos

comparativos entre os métodos citados na literatura focando essencialmente o treinamento médico.

8. REFERÊNCIAS

3D Studio Max – Autodesk 3ds Max - Site Oficial, Disponível em: <http://www4.discreet.com/3dsmax>, acesso em: 03de abril 2006.

AHO A. V., SATHI R., ULLMAN J.D. **Compiladores – Princípios, Técnicas e Ferramentas**. São Paulo: Guanabara Koogan S.A.,1995. p344.

BARRILLEAUX, J. **3D user Interfaces with Java 3D**. 1.ed. Greenwich: Manning Publication, 2001. 499p.

BERTI, C. B., NUNES, F. L. S. **Visualização de informações de bases de imagens médicas utilizando realidade virtual**. In: V Workshop de Informática Médica, 2005, Porto Alegre, Anais do V Workshop de Informática Médica, Porto Alegre, 2005.

BICHO, A. L., et al. **Programação Gráfica 3D com OpenGL, Open Inventor e Java 3D**. Revista Eletrônica de Iniciação Cientifica (REIC 2002), v. 2, n. 2, p. 1-43, mar. 2002.

BOTEGA, L. C., NUNES, F. L S., MONTANHA, F. **Implementação da Estereoscopia de baixo custo para aplicações em ferramentas de Realidade Virtual para treinamento médico**. In: XVIII SIBGRAPI 2005, Outubro 2005, Natal, Anais do XVIII SIBGRAPI, Natal 2005.

BISHOP, F., et al. **Research Directions in VR Environments**. In: Computer Graphics – ACM v.26, n.3, p.153-177, Aug, 1992.

BURDEA, G., et al. **Virtual Reality - based Training for Diagnosis of Prostate Cancer**. In: IEEE Transactions of Biomedical Engineering, vol. 46, n.10, p. 1253-1260, 1999.

BRO-NIELSEN, M. **Finite Element Modeling in Surgery Simulation.** In: Proceedings of IEEE, vol. 86, n. 3, p. 490-503, 1988.

BLENDER, Ferramenta de Modelagem 3D - Site Oficial. Disponível em: <http://blender.org/cms/Home.2.0.html>. Acesso em: 27 de março 2006.

COHEN M., MANSSOUR, I.H. **OpenGl – Uma abordagem Prática e Objetiva**. 1º ed. São Paulo: Novatec, 2006. 486 p.

COHEN, R. V. **Laparoscopia Intervencionista - Conseqüências metabólicas sistêmicas e Imunológicas**. 2º ed. Rio de Janeiro: Interlivros, 1997. 127p.

COLAVITTI, F. **A explosão da Realidade Virtual.** Revista Galileu, São Paulo vol. 13, n. 158, p. 62-67, 2004.

COSTA, I. F, BALANIUK, R. **LEM - An Approach for Real Time Physically based Soft Tissue Simulation**. In: International Conference in Automation and robotics (ICRA 2001), Seul–Korea do Sul, 2001.

CHAN, P. **The Java(TM) Developers Almanac 1.4, Volume 1 – Example and Quick Reference.** 1 ºed, Boston: Addison-Wesley – Pearson Education, 2002. 897p.

CHOI, K.S *et al.* **A Scalable Force Propagation Approach for Web – Based Deformable Simulation of Soft Tissues**. In: Web3D'02, 2002, Arizona, ACM Proceedings, 2002. p.185-193.

CHRISTOV S. et al. **Maya™ Learning Maya 2**. Canadá: Alias Wavefront Education, 1999, 516p.

DELINGETTE, H., COTIM, S., AYACHE, N. **A Hibrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation**. In: Computer Animation 1999. Anais IEEE Computer Society, 1999. p.70-81

DELLINGETTE, H.; AYACHE, N. **Hepatic Surgery Simulation.** In: Communications of the ACM. vol. 48, n.2, p. 43-48, 2005

DEITEL, H.M. e DEITEL, P.J. **Java Como Programar**. 4º ed. Porto Alegre: Bookman, 2003. 1110p.

DIMAIO, S.P.; SALCUDEAN, S.E. **Needle Insertion Modelling and Simulation**. In: Proceedings of The IEEE International Conference on Robotics and Automation, 2002. Anais IEEE Computer Society, 2002. p.2098-2105

FARINES, J.M. et al. **Sistemas de Tempo real.** Escola de Computação'2000-IME-USP..Disponível eletronicamente: <http://www.lcmi.ufsc.br/~romulo/discipli/cad-cbtisa/escola-1.pdf>.Acesso em 10 de julho de 2006.

FERNANDES, Y. B. **Neurocirurgia Minimamente Invasiva.** Jornal da Unicamp, ed. 224 - 11 a 17 de agosto de 2003. Disponível eletronicamente: <http://www.unicamp.br/unicamp/unicamp_hoje/ju/agosto2003/ju224pg02a.html.> Acesso em 27 de mar. de 2006.

FREITAS, C., NEDEL L. **Framework para Construção de Pacientes Virtuais: Uma Aplicação em Laparoscopia Virtual.** In: Proceedings of VI Symposium on Virtual Reality (SVR), 2003, Ribeirão Preto. SBC Brazilian Computer Society. p.283-293.

GARCIA, V. **Cirurgia laparoscópica: Principios: Basicos en cirurgia minimamente invasiva**. 2002. Disponível eletronicamente em: <http://utreia.uninet.edu/cirugia/manual/Fundamentos/pdf/laparoscopia.pdf/>. Acesso em 27 de março de 2006.

GIBSON S.F.F., MIRTCH B., **A Survey of Deformable Modeling in Computer Graphics**, In: Mitsubishi Electric Information Technology Center America, vol. 1, Outubro 1997. p1-22. Disponível eletronicamente em: <http://www.merl.com/people/frisken/deformationSurvey.pdf>. Acesso em 03 de abril de 2006.

GÜDÜKBAY, U., ÖZGÜÇ, B., e TOKAD, Y. **A Spring Force Formulation for Elastically Deformable Models.** In: Computer & Graphics, vol. 21, n .3, 1997. p335-346.

GLADILIN E., **A Biomechanical Modeling of Soft Tissue and Facial Expressions for Craniofacial Surgery Planning**. Tese de Doutorado. FUBerlin, pp.8-15, 2003.

HALUCK, R.S. *et al.* **A Haptic Lumbar Puncture Simulator**. In: Proceedings of Medicine Meets Virtual Reality (MMVR'2000), 2000, Medicine Meets Virtual Reality(MMVR'2000) Disponível eletronicamente: <http://cs.millersville.edu/~webster/haptics/lumbar/index.html>. Acesso em 27 de março de 2006.

HANCOCK, D. **Viewpoint: Virtual Reality in Search of Middle Ground**. In: IEEE Spectrum, vol. 1, n. 32, 1995.

HERMOSILLA, L. G., NUNES, F. L. S. **Proposta para construção de biótipo brasileiro para representar fetos virtuais.** In: V Workshop de Informática Médica, 2005, Porto Alegre, Anais do V Workshop de Informática Médica, Porto Alegre, 2005.

HIROTA G., MAHESHWARI R., Lin M.C. **Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization**. In: SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques, 1999, Michigan, Anais ACM Digital Library, Michigan, 1999. p.234 - 245

HORNBY, A S. **Oxford Advanced Learner's Dictionary of Current English.** 15 ed. Oxford: Oxford University Press, 1995. p. 1152.

HUFF, R., et al. **Usando Iluminação Baseada em Imagens na Geração de Ambientes de Realidade Mista.** In: Proceedings of VII Symposium on Virtual Reality (SVR), 2004, Anais VII Symposium on Virtual Reality (SVR), São Paulo, 2004. p137-148.

KEEVE, E., GIROD, S., AND GIROD, B. **Craniofacial Surgical Simulation**. In: Proceedings of MICCAI 2004, Springer-Verlag Lecture Notes in Computer Science, 2004, Anais do 4° Conferência Internacional de Visualização biomedical em Computação. p541-546.

KERA, M. **Detecção de colisão utilizando hierarquias em ferramentas de realidade virtual para treinamento médico**. 2005 92 f. Grau: Monografia (Bacharelado em Computação) Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

KISMET - 3D SIMULATION SOFTWARE. Disponível em: <http://www-kismet.iai.fzk.de> Acesso em: 30 de janeiro de 2006.

KIRNER, C. **Apostila do Ciclo de Palestras de Realidade Virtual**. In: Atividade do Projeto AVVIC – CNPQ (Protem – CC – Fase III) – DC/UFSCAR, São Carlos, Outubro, 1996. p1-10.

KONNO, S. **CyberVrml97 - Virtual Modeling Language Development Library.** Disponível eletronicamente em: < http://www.cybergarage.org/vrml/cv97/cv97java/> Acesso em: 27 de março de 2006.

KÜHHANAPFEL, U., ÇAKMAK, H.K., e MAAB, H. **Endoscopic surgery training using virtual reality and deformable tissue simulation**, In: Computers & Graphics vol. 24, 2000. p671-682.

LAPSIM SYSTEM. Disponível em: <http://www.surgical-science.com>. Acesso em: 27 de março de 2006.

LEMAY, L., CADENHEAD, R. **Aprenda em 21 dias JAVA ™** . 1º ed, São Paulo:Campus, 1999.576p.

LIMA, L.; NUNES, F. L. S. **Utilização de Realidade Virtual em treinamento médico: um protótipo de ferramenta para exame de punção de mama**. In: III Congresso Latino-Americano de Engenharia Biomédica, 2004, João Pessoa. Congresso Latino-Americano de Engenharia Biomédica, 2004. v. 1. p1695-1698.

LIPARI, G., et al. **Real Time Scheduling - Task Syncronyzation in Reservation - Based Real Time Systems**. In: IEEE Transations on Computers, vol. 53 n.12, p. 1591 - 1602, 2004.

MACHADO, L. S., *et al.* **Modelagem Tátil, Visualização Estereoscópica e Aspectos de Avaliação em um Simulador de Coleta de Medula Óssea**. In: Proceedings of IV Symposium on Virtual Reality (SVR). Florianópolis. SBC Brazilian Computer Society, 2001, p23-31.

MACHADO, L. S. **A Realidade Virtual no Modelamento e Simulação de Procedimentos Invasivos em Oncologia Pediátrica: Um Estudo de Caso no Transplante de Medula Óssea**. 2003, 130 f. Grau: Tese(Doutorado em Engenharia) Departamento de Engenharia de Sistema Eletrônicos da Escola Politécnica da Universidade de São Paulo, 2003.

MACHADO, L. S. *et al.* **Cybermed: Realidade Virtual Para Ensino Médico**. In: III CONGRESSO LATINO-AMERICANO DE ENGENHARIA BIOMÉDICA, 2004, João Pessoa. Proceedings of the International Federation for Medical and Biological Engineering. p573-576.

MANSSOUR. I. H. **Introdução a Java 3D ™**. In: Revista Rita, v.10, n.1, 2003. p71-96

MOSS, A. **Java 3D Head Generation Tool.** 2001. Grau: Dissertação de Mestrado Department of Computer Science, University of Sheffield, Inglaterra, 2001.

PALMER, I. **Essential Java 3D Fast - Developing 3D Graphics Applications in Java**. 1º ed. SPRINGER VERLAG NY, 2001, 279p.

PARK, K., et al. **Volumetric Heart Modeling and Analysis**. In: Communications of the ACM. vol. 48, n.2, 2005. p43-48.

RAHN, S. **WorldToolKit ™ Release 8 – Technical Overview – High-Performance Visual Simulation Development Software.** Mill Valley: Sense 8 CORPORATION, 1998.78p.

RIQUELME, F. **Estudo comparativo de tecnologias de *software* para Realidade Virtual.** 2005. Grau: Dissertação (Mestrado em Ciência da Computação) - Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

RAMALHO Jr., F., FERRARO, N. G., SOARES, P. A. de T. **Os fundamentos da Física**. 6°ed. São Paulo: Moderna, 1993.

RAMOS, F. M.. **Aplicação de Realidade Virtual para construção de Atlas de Anatomia e Fisiopatologia do Câncer de Mama.** 2005. 83 f. Grau: Dissertação (Mestrado em Ciências da Computação) – Centro Universitário Eurípides de Marília, Marília, 2005.

RODRIGUES, M; *et al.* **Um Ambiente Interativo Distribuído para a Simulação de Procedimentos Cirúrgicos**. In: Proceedings of V Symposium on Virtual Reality (SVR), 2002. Fortaleza. SBC Brazilian Computer Society, p179 – 190.

SEDERBERG, T.W. PERRY S.R. **Free form deformation of solid geometric models", em Computer Graphics Proceedings.** In: Annual Conference Series, Proceedings of SIGGRAPH ACM, vol.86, 1986. p151-160

SELMAN, D. **Java 3D Programming** Disponível eletronicamente em: <http://www.manning.com/selman>. Data do último acesso abril de 2005.

PALMER, I. **Essential Java 3D Fast - Developing 3D Graphics Applications in Java**. 1º ed. SPRINGER VERLAG NY, 2001, 279p.

SABBINENI H., HAKRABARTY K. **Real Time and Embeded Systems- Location - Aided Flooding: An Energy- Efficient Data Dissemination Protocol for Wirelless Sensor Networks**. In: IEEE Transations on Computers, vol. 54 n.1, p. 36 - 46, 2005.

SILVA, A. **Dominando a Tecnologia de Objetos: Programação- Implementação – Soluções – Problemas. UML – Java – C++.** Petrópolis: BookExpress, 2002. p512

SILVA, R. J. M.; RAPOSO, A. B.; GATTASS, M. **Grafo de Cena e Realidade Virtual** In: Monografias em Ciência da Computação, n.11/04, Departamento de Informática, PUC-Rio, Rio de Janeiro, 2004. Disponível eletronicamente: < http://www.tecgraf.puc-rio.br/ >. Data do último acesso: abril de 2006.

SUN – **Java 3D API Tutorial**, 2000. Disponível eletronicamente em: <http://java.sun.com/developer/onlineTraining/java3d/>. Data do ultimo acesso em: 13 outubro de 2004.

SHAW A.C., **Sistemas e Software de Tempo Real**. 1º ed. São Paulo: Bookman, 2003. 240 p.

SHI, J.Y, YAN, L.X, **Deformation and Cutting in Virtual Surgery**. In: Proceedings of the International Workshop on Medical Imaging and Augmented Reality, 2001, Anais IEEE Computer Society.

SCHNEIDER, B. O., **Modelagem dinâmica de Mundos Virtuais.** 1998. 65f. Grau: Dissertação de Mestrado apresentado a Universidade Federal de São Carlos, São Carlos, 1998.

SZÉKELY, G. *et al.* **Virtual Reality Basead Surgery Simulation for Endoscopic Gynaecology.** In: Medicine Meets Virtual Reality. nº 62, 1999, Studies in health technology and Informatics - IOS Press p. 351-357.

The Java Developers Almanac 1.4. Disponível em: <http://javaalmanac.com/egs/index.html >Acesso em: 30 de janeiro de 2006.

VINCE, J. **Virtual Reality Systems**, Massachusetts: Addison – Wesley, 1995

WEBSTER, R., et al. **Elastically Deformable 3D Organs for haptic Surgical Simulation.** In: Medicine meets Virtual Reality, 2002. IOS Press. 2002.

WAGNER, C.; SCHILL, M. A.; MÄNNER, R. **Collision Detection and Tissue Modeling in a VR-Simulator for Eye Surgery**. In: Proceedings of Eigth Eurographics Workshop on Virtual Environments, 2002. Disponível eletronicamente: <http://www-li5.ti.uni-mannheim.de/publications/EletronicPublications/Keytechno14.pdf> Acesso em: 27 de março de 2006.

ZIENKIEWICZ, O.C. **The Finite Element Method**, New York: Mc Graw-Hill, 1977.

ZILL D.G. **Equações Diferenciais com aplicações em Modelagem.** 3ªed São Paulo: Thomson, 2003. p492

WIKIPEDIA. **Enciclopédia livre**. Disponível em: < http://pt.wikipedia.org>. Acesso em 09 março 2006.

APÊNDICE

Técnicas minimamente invasivas.

As áreas que mais utilizam a técnica MI são: Gastroenterologia, Ginecologia, Oncologia, Cardiologia, Urologia e Oftalmologia. Alguns exemplos são detalhados a seguir.

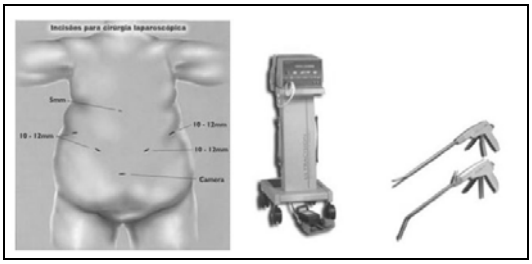
A.1 Gastroenterologia

Gastroenterologia é a especialidade da Medicina que trata das patologias do aparelho digestivo. Também leva em conta a relação do aparelho digestivo (composto pela boca, faringe, esôfago, estômago e intestinos delgado e grosso) com outros órgãos como vesícula e fígado (SABISTON, 1999).

Nesta área de gastroenterologia muitas cirurgias têm aplicado a técnica de laparoscopia, uma técnica minimamente invasiva, que segundo Cohen (1997), surgiu no Brasil em 1990. São feitas incisões abdominais de 0,5 a 1,2 cm. Através dessas cavidades abdominais é inserido gás carbônico para criar um espaço para que o cirurgião trabalhe com a videocâmara, instrumentos e grampeadores especiais.

Com isso tem-se obtido ótimos resultados, principalmente nas cirurgias de retirada de vesícula, e ultimamente, nas cirurgias barriátricas, que são cirurgias especificas para tratamento de obesidade mórbida (COHEN, 2005).

Existem diversas opções cirúrgicas para a cirurgia barriátrica que são: as restritivas, as más absortivas e as híbridas. Cohen (2005) afirma que o acesso laparoscópico é o mais benéfico para o paciente, já que são reduzidos o tempo de internação, a dor e o período de afastamento das atividades diárias, além de que as complicações do tratamento cirúrgico laparoscópico da obesidade são hoje menores que aquelas da cirurgia convencional.



a) **(b)**

Figura A1 – (a) Incisões na área abdominal; (b) aparelhos utilizados na cirurgia (HONDA, 2005).

Aplicando-se a técnica MI, são realizadas quatro ou cinco incisões, com cerca de 1cm cada uma, para acesso à cavidade abdominal. Uma minúscula câmera de vídeo e pequenos instrumentos são inseridos através das incisões e os cirurgiões acompanham o procedimento por um monitor de vídeo. A Figura A1 mostra os locais dessas incisões e o tipo de equipamento utilizado para este tipo de cirurgia.

Pode-se ainda citar a cirurgia de retirada de pedra de vesícula, que aplica a técnica MI e tem sido mais utilizada do que a cirurgia com o método convencional. Em vez de incisões de 20 a 30 cm de extensão, a operação é realizada através de quatro pequenos orifícios de 0,5 cm no abdômen, conforme mostra a Figura A2. Na primeira incisão é introduzido um trocáter, aparelho que possui uma câmera em sua ponta e funciona como um microscópio, ou seja, permite visualizar toda a cavidade. Mais três pequenas incisões são feitas para inserção de pinças que serão utilizadas no procedimento, indicadas na Figura A2.

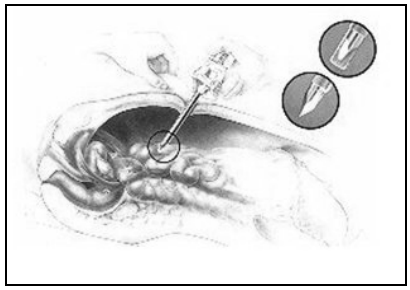
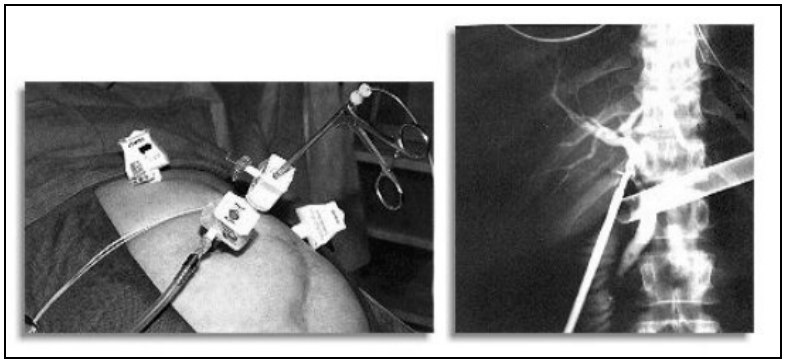


Figura A2 – Inserção do trocáter no abdômen (SOBRINHO, 2005).

Além deste procedimento, segundo Sobrinho (2005), são realizados, em alguns casos, um exame de raios-X para detectarem-se pedras no canal da bile. Se estas forem detectadas, são retiradas na própria cirurgia. A Figura A3 apresenta as incisões e o exame de raios-X para a retirada das pedras.



(a) (b)

Figura A3 – (a) Incisões para o procedimento da cirurgia de retirada de pedra na vesícula (b) imagem de raio-X para verificação de pedra no canal da bile (SOBRINHO, 2005)

A.2 Ginecologia

A ginecologia é a parte da medicina que estuda o sistema reprodutor feminino. Nessa área estão localizados os órgãos reprodutores pélvicos e as estruturas pélvicas adjacentes, como aparelho urinário e intestino (SABISTON, 1999). Existem dois tipos de aplicações de técnicas MI nessa área: o exame diagnóstico, que é um procedimento utilizado para reconhecer doenças existentes no interior do abdômen e o procedimento cirúrgico.

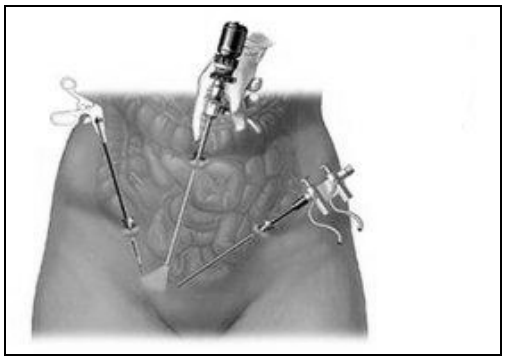


Figura A4 - Incisões para cirurgia assistida por vídeo (COHEN, 2005).

Segundo Cohen (2005), o exame diagnóstico é ambulatorial, na qual o paciente só necessita da internação durante algumas horas. Já o procedimento cirúrgico consiste em uma avaliação médica completa, que inclui exame ginecológico feito com anestesia geral. É feita uma incisão no interior da cicatriz umbilical, inserindo-se uma agulha. Através dela é feita a insuflação do abdômen com gás carbônico. O gás empurra as alças intestinais para cima, longe dos órgãos genitais, permitindo a inserção de uma microcâmera que auxilia na visualização dos órgãos em um monitor. A Figura A4 apresenta a introdução da agulha para diagnóstico.

Para saber se há necessidade de cirurgia, alguma anormalidade é observada na laparoscopia diagnóstica. A vídeo-laparoscopia cirúrgica consiste no tratamento dessa anormalidade e, para a cirurgia MI, neste caso, são necessários instrumentos adicionais para melhor manusear os órgãos (COHEN, 2005). Normalmente esta cirurgia é utilizada para tratar miomas uterinos, aderências, endometriose, cisto de ovário, gravidez nas tubas e até mesmo em cirurgia para a retirada do útero, ovário e trompas, além de incontinência urinária.

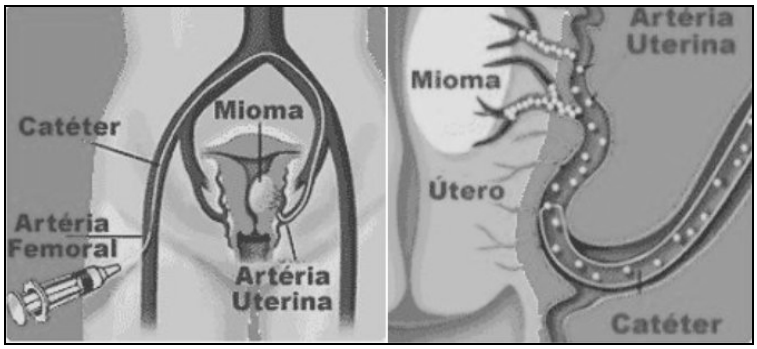
Além disso, a técnica MI é empregada na embolização, que é um procedimento de radiologia uterina para realizar a obstrução intencional de um vaso em uma determinada região, a fim de impedir que continue passando sangue naquele local (KISILEVZKY E

MARTINS, 2003). É usada em situações de hemorragias uterinas, sendo uma nova alternativa para tratar as mulheres que apresentam mioma sintomático. A incisão é feita na pele, sendo pequena e com anestesia local. (BOZZINI *et al.,* 2005).

(a) (b)

Figura A5 - a) Cirurgia de embolização Uterina b) monitor que auxilia a cirurgia (KISILEVZKY E MARTINS, 2004).

O processo consiste em fazer uma incisão de aproximadamente 2 mm na pele da virilha. Para que ocorra essa obstrução é introduzido um catéter[3] que é colocado dentro do sistema vascular, permitindo que se possa enxergar através da pele, podendo levar o catéter ao lugar onde se deseja interromper o fluxo de sangue (KISILEVZKY E MARTINS, 2003). A Figura A5 mostra que a cirurgia é muito simples e demonstra como é feita a introdução do catéter no paciente. As Figuras A6 (a) e A6 (b) apresentam como o catéter é conduzido pelas artérias uterinas[4]e como são injetados até que as artérias fiquem entupidas para que o mioma não receba mais sangue.



a) (b)

Figura A6 - (a) Introdução do catéter no corpo, (b) introdução de partículas de plástico no mioma (KISILEVZKY E MARTINS, 2004).

[3] Tubo Plástico Fino (KISILEVZKY E MARTINS, 2003).
[4] Artérias responsáveis por levarem o sangue para o útero e os miomas (SABISTON, 1999)

Outro tipo de cirurgia que usa a técnica MI é a miomatose uterina. A miomatose é ocasionada por leiomiomas, que são tumores pélvicos sólidos mais freqüentes nas mulheres em idade fértil (BOZZINI, 2005).

Segundo Guimarães *et al.* (2002), a miomatose aparece quando ocorre problema de irregularidade menstrual sendo a menorragia[5] a mais freqüente. O aumento do volume uterino leva a um aumento da pressão pélvica, relacionado à dor e compressão de outras estruturas como: reto e bexiga, causando obstipação[6] e incontinência urinária. A Figura A7 mostra um exemplo de mioma.

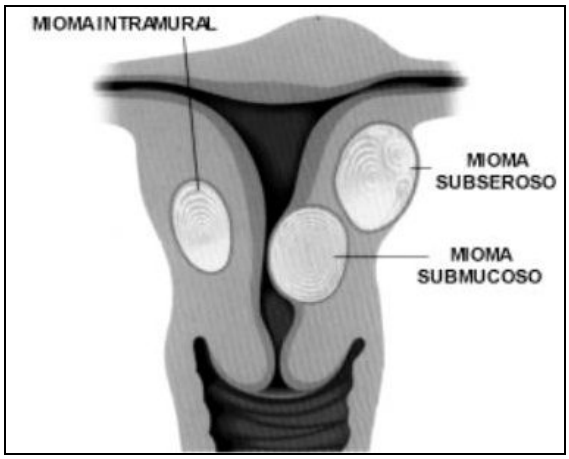


Figura A7 – Exemplo de mioma (GUIMARÃES *et al.*, 2002).

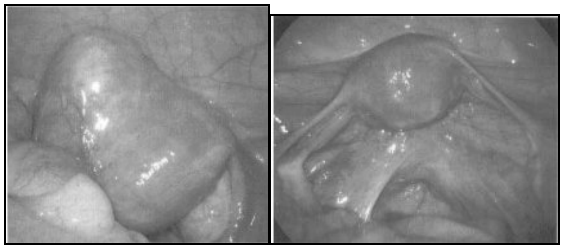
A miomectomia é um procedimento cirúrgico para a remoção dos leiomiomas. Há várias técnicas para realizar esta cirurgia: a via histeroscópica, a via laparoscópica ou a via abdominal, dependendo do número dos nódulos, da sua localização e do seu tamanho (GUIMARÃES *et al.*, 2002).

A miomectomia laparoscópica é utilizada para extrair miomas que se encontram na porção externa do útero. Pequenas incisões são realizadas na parede abdominal por onde é introduzida uma microcâmera de vídeo e instrumentos apropriados para realizar a extração do mioma (MENCAGLIA, 2002).

Este procedimento é realizado com anestesia geral com um pequeno telescópio que varia de 5 a 10 mm de diâmetro, introduzido pela cicatriz umbilical (um pequeno corte menor que 1 cm) acoplada a uma mini-câmera gerando imagens em um monitor de TV como apresenta a Figura A8.

[5] Perda uterina excessiva de sangue, ocorrendo em intervalos regulares e sendo o período de perda mais duradouro que habitual na menstruação. Excesso de fluxo menstrual (GUIMARÃES *et al*, 2002)
[6] Prisão de ventre. (GUIMARÃES *et al*, 2002)

A miomectomia por via histeroscópica é utilizada somente para extrair os miomas que se encontram debaixo da camada interna do útero e se exteriorizam para a cavidade uterina (LASMAR e BARROZO, 2002).



(a) (b)

Figura A8 - Miomectomia Laparoscópica: a) Mioma Uterino, b) laparoscopia diagnóstica (GUIMARÃES, 2002).

Não requer qualquer incisão cirúrgica. O médico introduz um tubo flexível chamado histeroscópio através da vagina e colo uterino e, com instrumentos apropriados, extrai o mioma. Este procedimento é realizado geralmente de forma ambulatorial e com anestesia.

Segundo Lasmar e Barrozo (2002), após a dilatação do colo uterino, a cirurgia é realizada com um equipamento cirúrgico chamado vela de Hegar. O conjunto completo é formado com o acoplamento do endoscópio de 4 mm, da fonte de luz, da microcâmera, do monitor de vídeo, do sistema de documentação, da unidade elétrica de alta freqüência, do irrigador e do aspirador.

A.3 Oncologia

Câncer é o nome dado a um conjunto de mais de 100 doenças que têm em comum o crescimento desordenado, chamado de maligno, de células que invadem os tecidos e órgãos, podendo espalhar–se para outras regiões do corpo, ocorrendo o processo denominado de metástase (INCA, 2006).

Estas células tendem a ser muito agressivas e incontroláveis pelo organismo, pois sua multiplicação é muito rápida e determina a formação de tumores[7] ou neoplasias malignas. Existe também o tumor benigno, que consiste em uma massa localizada de células que se multiplicam vagarosamente e se assemelham ao seu tecido original, raramente constituindo um risco de vida.

[7] Acúmulo de células cancerosas. (INCA, 2005)

As cirurgias MI são aplicadas no diagnóstico e tratamento de diversos tipos de câncer, conforme descrito a seguir.

A.3.1 Câncer de Próstata

A próstata é um dos órgãos do sistema urogenital masculino que fica situado abaixo da bexiga e, no indivíduo jovem, tem um tamanho aproximado de uma noz. No meio da próstata passa a uretra, canal que conduz a urina da bexiga até o exterior. Sua função é armazenar o líquido que compõe o sêmen sintetizado pela próstata. (BURDEA, 1999)

O câncer de próstata aparece na região mais externa da próstata podendo ser diagnosticado através do exame de palpação (toque retal). Em sua fase inicial os sintomas não são tão nítidos. Seu desenvolvimento é progressivo e, quando diagnosticado, não possui um método de tratamento (INCA, 2005).

Os cânceres de próstata usualmente são descobertos com dois exames: o antígeno prostático específico e o toque retal digital. Para a confirmação extra desses exames é feita a biópsia.

Lemos (2005) define a biópsia como um procedimento minimamente invasivo, que através de uma agulha, guiada por ultra-som, retira o material a ser examinado via trans-retal, podendo ser feito com ou sem anestesia no próprio consultório do médico. O exame dos fragmentos retirados pode definir se a doença é maligna ou benigna.

A.3.2 Câncer no Sangue - Leucemia

Segundo o INCA (2005), a leucemia é uma doença maligna dos glóbulos brancos (leucócitos). Sua origem é, na maioria das vezes, não conhecida. Tem como principal característica o acúmulo de células jovens anormais, conhecidas como blásticas, na medula óssea, que substituem as células sanguíneas normais.

Os tratamentos possuem o objetivo principal de destruir as células leucêmicas para que a medula óssea volte a produzir células normais. Segundo INCA (2005) os possíveis procedimentos médicos no tratamento da leucemia são: mielograma, punção lombar, cateter e transfusões.

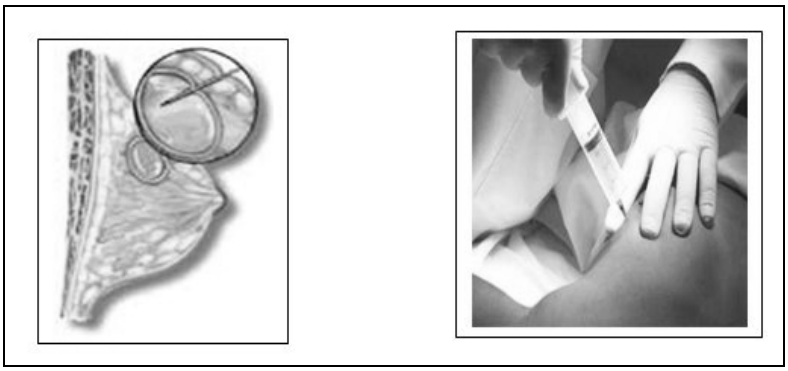
A punção é o tratamento que utiliza a técnica MI. A medula espinhal é a parte do sistema nervoso que tem a forma de cordão. É forrada por três membranas conhecidas como meninges. Através delas circula um líquido claro denominado de líquor. A punção lombar consiste na aspiração do líquor para um exame citológico e também para injeção de quimioterapia, com a finalidade de impedir a profilaxia de células leucêmicas e para destruir quando houver doença nesse local. Sua realização é feita na maioria das vezes com anestesia local inserindo-se uma seringa e retirando o líquor para exames futuros. (FURTADO, 2003)

A.3.3 Câncer de Mama

O câncer de mama é o tipo de câncer que se manifesta com maior freqüência entre mulheres brasileiras (INCA, 2006).

Para prevenção do câncer de mama, existem três métodos para rastreamento e diagnóstico, que são: auto-exame das mamas, exame clínico e mamografia (INCA, 2005).

Quando há suspeita de malignidade, o médico pode solicitar a biópsia ou punção aspirativa, conforme apresenta a Figura A9.



(a) **(b)**

Figura A9 - Biópsia da Mama por meio de Punção aspirativa por agulha (a) Diagrama esquemático, (b) Exame real (LIMA E NUNES, 2004)

Segundo Freitas Jr. (2001) esse método é um procedimento ambulatorial que utiliza a técnica MI, pois, além da agressão ao paciente ser mínima, é de baixo custo. É baseado no uso de uma agulha de fino calibre que é introduzida na pele, em direção à lesão, com o intuito de coletar células para posterior avaliação de sua morfologia. O exame é feito por movimentos de vaivém da agulha em diversas direções dentro do tumor, puxando o êmbolo da seringa para a aspiração do material celular. O material é colocado em uma lâmina de vidro e encaminhado para avaliação laboratorial (INCA, 2005).

A.4 Tórax / Cardiologia

A Cardiologia refere-se à área da Medicina que estuda o sistema circulatório que compreende coração, vasos sanguíneos, vasos linfáticos, sangue, linfa, líquido falor aquídiano e líquido intracelular. O coração é um músculo forte que funciona como uma bomba de ejeção contínua e a função principal é levar o sangue para o sistema circulatório. (D'ÂNGELO e FATTINI, 2004). O tórax, localizado na região superior do tronco é também chamado de caixa torácica, sendo constituído por vértebras torácicas, osso esterno, costelas e cartilagens costais. Segundo D'Ângelo e Fattini (2004), sua principal função é alojar os órgãos fundamentais como pulmão e coração, esôfago, traquéia e brônquios. A técnica MI é aplicada em vários procedimentos dentro desta especialidade médica, como descrito a seguir.

A.4.1 Revascularização Cardíaca

A eficiência da cirurgia de revascularização cardíaca sem a utilização de circulação extracorpórea[8] é freqüentemente questionada, principalmente, quando o acesso às artérias coronárias[9] é restrito. Por isso, Cremer *e*t al. (2000) afirmam que está havendo uma expansão, em todo o mundo, dos procedimentos com invasão mínima do coração, para o tratamento de afecções das artérias coronárias.

A cirurgia de revascularização do miocárdio é um tratamento que cuida da insuficiência coronária, principalmente com a utilização da artéria mamária. Jatene *et al.* (1997) desenvolveram um estudo para tratar esta lesão através da revascularização de miocárdio com a técnica minimamente invasiva. Foi feita uma incisão de 8 a 10 cm (na época era considerado como minimamente invasivo) e são inseridos os equipamentos de ótica, como a câmera, os instrumentos cirúrgicos necessários para a dissecação da artéria mamária.

Segundo os pesquisadores, a utilização da videotoracoscopia e o uso de eletrocautério com ponta foram empregados para que a artéria mamária pudesse ser inteiramente dissecada. A Figura A10 mostra a o tamanho da incisão e os equipamentos utilizados.

[8] Procedimento em que o sangue passa a circular fora dos vasos sanguíneos do corpo, ou seja, em equipamentos específicos (Araújo *et al*, 1996).
[9] Vasos sanguíneos que nutrem o músculo cardíaco-miocárdio. (Araújo *et al*, 1996)

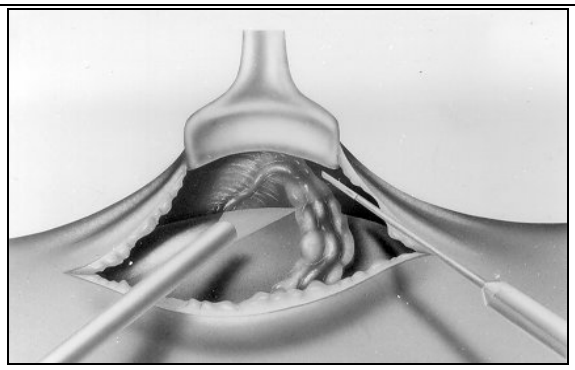


Figura A10 - Dessecação da artéria mamária, introdução da câmera à esquerda e o eletro-cautério com ponta longa à direita. (JATENE *et al.*, 1997)

A.4.2 Hérnia de Hiato

Hérnia de hiato é uma doença que tem como causa o alargamento da passagem do esôfago do tórax para o abdômen (o hiato esofágico) e a migração de parte do estômago para o tórax, levando ao refluxo do ácido gástrico para o esôfago, provocando a esofagite. (COHEN, 2005)

Segundo Sobrinho (2005), entre o tórax e o abdômen existe um músculo que separa estas duas cavidades, chamado diafragma. O orifício do diafragma, pelo qual o esôfago passa para chegar ao abdômen chama-se hiato. O hiato é pequeno e apenas tem espaço para a passagem do esôfago, mas normalmente por inúmeros motivos ele se encontra alargado fazendo com que o estômago suba para o tórax, formando uma hérnia de hiato. Só se indica a cirurgia quando a hérnia é muito grande causando compressão aos órgãos que estão próximos a ela, no caso o pulmão e o coração.

Para que ocorra a cirurgia são feitas cinco incisões que variam entre 0,5 e 1 cm de extensão para a introdução de tubos metálicos, conhecidos como trocáteres, na cavidade abdominal. Na ponta do trocáter é colocada uma videocâmara e as pinças que serão utilizadas para a cirurgia. (SOBRINHO, 2005; COHEN, 2005).

A.5 Ortopedia

Normalmente quando se indica a cirurgia para problema ortopédico é traçado um planejamento operatório cuidadoso e um preparo adequado do paciente, pois este ato cirúrgico deve ser realizado em um ambiente apropriado, com a instrumentação adequada e o pós-operatório bem conduzido para o sucesso do tratamento.

Dentro da ortopedia, a cirurgia de coluna tem utilizado a técnica MI, realizando procedimentos como microcirurgia vídeo-endoscópicas das hérnias discais abordagens da coluna dorsal por vídeo toracoscopia e realização de artrodeses vídeo-assistido. A Figura A11 mostra como são realizados estes tipos de cirurgias por vídeo-endoscopia e a introdução da câmera no paciente.

(a) **(b)**

Figura A11 - Introdução da Câmera e a realização da cirurgia minimamente invasiva através do vídeo-endoscopia (Técnicas Cirúrgicas, 2005)

Outro tipo de cirurgia que utiliza a técnica MI é a cirurgia torácico vídeo-assistido ou conhecida como vídeo-toracoscopia. São feitas incisões mínimas na musculatura do tórax e as costelas são afastadas. São associados ao local da cirurgia pequenos tubos, pelos quais passam a câmera e os equipamentos necessários para a cirurgia. Essas técnicas podem ser usadas em casos selecionados de tratamento das hérnias discais dorsais, fraturas, tumores e deformidades da coluna torácica para realização de artrodese, bem como o tratamento da hiperidrose[10].

Há ainda a artrodose lombar vídeo assistida que também pode ser realizada através de vídeo-endoscopia. A Figura A12 mostra o tamanho da incisão para este tipo de cirurgia.

Figura A12 - Tamanha da incisão para cirurgias de artrodose lombar vídeo assistido utilizando a técnica MI (Técnicas Cirúrgicas, 2005)

[10] Condição clinica caracterizada pelo excesso de suor, principalmente nas mãos e nas plantas dos pés (Araújo *et al*, 1996).

A.6 Oftalmologia

A oftalmologia é uma especialidade que estuda a visão, um dos órgãos dos sentidos de grande importância para a interação do ser humano com o ambiente. (SABISTON, 1999). As cirurgias MI também têm entrado na área de oftalmologia, principalmente em cirurgia da catarata. Segundo Iradier (2001) a catarata é uma opacificação do cristalino[11], que acontece com o passar dos anos, ou por outras causas como traumatismos e inflamações oculares. Se a opacificação já estiver bastante avançada é necessária a extração da catarata e a substituição do poder dióptricodo cristalino por uma lente intra-ocular. A Figura A13 mostra o processo da cirurgia.

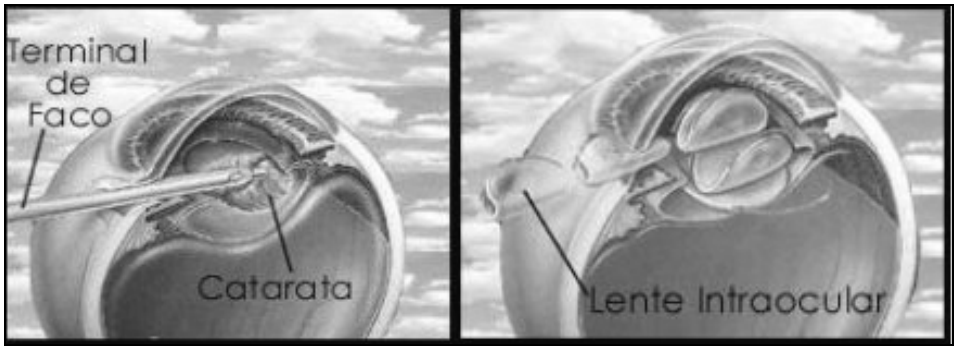


Figura A13 – Processo de cirurgia de catarata (IRAIDER, 2001).

Aplicando-se a técnica facoemulsificação[12], que é feita através de um procedimento MI, o cristalino opacificado (catarata) é pulverizado em micro partículas que são aspiradas do interior do olho. A cirurgia é realizada com a introdução da câmera através de uma agulha com uma abertura mínima que varia de 2 a 3 milímetros. Tratando-se de uma técnica MI, devido ao pequeno tamanho da abertura e a leve manipulação cirúrgica, o procedimento pode realizar-se com anestesia local com o auxílio de um colírio. (IRAIDER, 2001).

A.7 Referências

ARAÚJO, C. L. C, et al. **Stedman Dicionário Médico.** 25°ed. Rio de Janeiro: Guanabara Koogan S.A, 1996.

[11] O cristalino é uma lente biológica responsável de aproximadamente 20% do poder diópitrico do olho (IRADIER, 2001)
[12] Técnica que consiste na substituição por um cristalino artificial ou lente intra-ocular. (ARAÚJO et al, 1996)

BOZZINI N., et al. **Projeto Diretrizes - Miomatose Uterina**. Disponível eletronicamente: <http://www.amb.org.br/projeto_diretrizes/100_diretrizes/MIOMATOS.pdf>, acesso em: abril de 2005.

BURDEA, G.C. **Haptic issues in virtual environment.** In: Proceedings of the Computer Graphics International, 2000. p. 295-302.

BUCHALLA, A.P. **Olho no laser**. Revista Veja, vol. 35, n.17, p. 70, 2002.

BURDEA, G., COIFFET, P. **Virtual Reality Technology**. In: Proceedings of John Wiley & Sons, New York, 1994.

COHEN, R. V. **Laparoscopia Intervencionista - Conseqüências metabólicas sistêmicas e Imunológicas**. 2º ed. Rio de Janeiro: Interlivros, 1997.

COHEN, R. V. **A Referencia brasileira em Cirurgia Laparoscópica.** Disponível eletronicamente em <http://www.cirurgiabariatrica.com.br/outras/indexoutras.htm>. Último acesso: abril de 2005.

CREMER, J. T, et al. **Minimally invasive coronary artery revascularization on the beating heart.** In: Ann Thorac Surg, vol.69, 2000. p1787 – 1791

D'ÂNGELO, J. G., FATTINI, C. A. **Anatomia Humana Sistêmica e Segmentar (para estudante de medicina)**. 2ºed. São Paulo: Atheneu, 2004.

FERREIRA, A. B. H. **Novo Aurélio Século XXI: o dicionário da língua portuguesa**. 3º ed. Rio de Janeiro: Nova Fronteira, 2000.

FREITAS JR, R. **Punção Aspirativa por Agulha Fina: Estudo Comparativo entre dois Diferentes Dispositivos para Obtenção da amostra Citológica.** 2001. Grau: Tese de Doutorado – UNICAMP, Campinas, 2001.

FURTADO, F. **Muito perigo para poucos.** Revista Ciência Hoje, vol. 34 nº 199, 2003. p44–45.

GUIMARÃES, A. L. R, et al. **Valor Atual para Tratamento da Hidroneforse Antenatal**. Publicada Sinopse de Urologia pelo grupo editorial Moreira Junior. Vol. 6, nº 5, 2002. p107 – 113.

HEARN D., BAKER M.P. **Computer Graphics – C version**. 2º ed. New Jersey: Prentice Hall, 1997. p583-598

HEBERT, S. et al. **Ortopedia e Traumatologia – Princípios e Prática**. 2º ed. São Paulo: ArtMed, 1998.

HONDA, J.A. **Gastroweb**. Disponível eletronicamente em: <http://www.gastroweb.com.br/>. Data do último acesso: abril de 2004.

INCA - **Instituto Nacional de Câncer**. Disponível eletronicamente: <http://www.inca.gov.br/> Data do último acesso abril de 2005.

IRAIDER, M. T., **Cirurgia de Catarata – Simpósio de Cataratas.** In: Congresso da Sociedade Espanhola de Oftalmologia. Barcelona, Espanha. Disponível eletronicamente: <http://www.drairadier.com/cirugia/p-ccatarata.htm> . Data do último acesso agosto de 2004.

JATENE F. B., et al. **Cirurgia de Revascularização do Miocárdio Minimamente Invasivo com utilização de Videotoracoscopia**. In: Instituto do Coração do Hospital das Clínicas - FMUSP, São Paulo, SP. Arquivos Brasileiros de Cardiologia vol.68 nº2, 1997.

KISILEVZKY, N.H, MARTINS, M. S., **Embolização Uterina para Tratamento de Mioma Sintomático**. Experiência Inicial e Revisão de Literatura. Publicado Radiol. Bras. vol. 36 nº 3, 2004. p129-140.

LASMAR, R., BARROZO, P., **Uma abordagem Prática**. Editora Médica e Científica Ltda., 2002.

LEMOS, G. C. **Urologia – Câncer de Próstata**. Disponível eletronicamente: <http://www.gustavolemos.com.br/cancer_prostata.htm>. Data do último acesso: abril de 2005.

MENCAGLIA L., ALBUQUERQUE, L. C. N., **Histeroscopia Diagnóstica**. Rio de Janeiro: Médica e científica Ltda, 2002.

SABISTON A., DAVID C. **Tratado de Cirurgia: As Bases Biológicas da Prática Cirúrgica Moderna.** 15º ed. Rio de Janeiro: Guanabara Koogan, 1999.

SOBRINHO S. **Pedra na Vesícula: Cirurgia para pedra na Vesícula Biliar Colecistectomia videolaparoscópica.** Sociedade Portuguesa beneficente do Amazonas. Disponível eletronicamente em: <http://www.internext.com.br/videocirurgia/pedra_na_vesícula.htm>. Data do último acesso: abril de 2005.

TÉCNICAS CIRÚRGICAS. **Cirurgias Minimamente Invasivas.** Disponível eletronicamente em: <http://www.cirurgiadacoluna.com.br/tecnicas_cirurgicas.htm >. Data do último acesso abril 2005.

TENDICK, F., et al. **A Virtual Environment Testbed for training Laparoscopic Surgical Skills.** Presence, vol. 9, n.3, p. 236-255, 2000.

WATERS, K., **A muscle model for animating three-dimensional facial expression"**. In: Proceedings, Annual conference Series, proceedings of SIGGRAPH vol. 87. ACM SIGGRAPH, 1987. p17-24.

WATERS, K. A **physical model of facial tissue and muscle articulation derived from computer tomography data**. In: Proceedings of visualization in Biomedical Computing, vol. 574, 1992. p574–584.