UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Visualização e Simulação de Sinais Cardiológicos Utilizando Técnicas de Realidade Aumentada

Orientador: Edgard Lamounier Junior, PhD

Co-Orientador: Alexandre Cardoso, Dsc.

Orientando: Arthur Augusto Bastos Bucioli

OUTUBRO

2008

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Visualização e Simulação de Sinais Cardiológicos Utilizando Técnicas de Realidade Aumentada

Dissertação apresentada ao Curso de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como exigência parcial para obtenção do título de Mestre em Ciências.

Edgard Lamounier Júnior, PhD (UFU) - Orientador
Alexandre Cardoso, Dsc (UFU) - Co-Orientador
Adriano de Oliveira Andrade, PhD (UFU)
Luciana Porcher Nedel, PhD (UFRGS)

Área de Concentração: Processamento da Informação
Linha de Pesquisa: Computação Gráfica/Realidade Virtual

UBERLÂNDIA
2008

Visualização e Simulação de Sinais Cardiológicos Utilizando Técnicas de Realidade Aumentada

Arthur Augusto Bastos Bucioli

Dissertação apresentada ao Curso de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia como exigência parcial para obtenção do título de Mestre em Ciências.

_____

Edgard A. Lamounier Júnior, Ph.D.

Orientador

_____

Darizon Alves de Andrade, Ph.D.

Coordenador do Curso.

FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

B921v Bucioli, Arthur Augusto Bastos, 1983-
Visualização e simulação de sinais cardiológicos utilizando técnicas de realidade aumentada / Arthur Augusto Bastos Bucioli. - 2008.
158 f. : il.

Orientador:.Edgard Afonso Lamounier Jr.

Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Computação gráfica - Teses. I. Lamounier Júnior, Edgard, 1964- II.Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.332.3

Elaborado pelo Sistema de Bibliotecas da UFU / Setor de Catalogação e Classificação

Este trabalho é dedicado à minha
família, aos meus amigos e à minha namorada,
amo todos vocês.

“O temor do Senhor é o princípio da sabedoria”

(Provérbios 1:7)

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, pois Ele me criou, me sustenta e me dá a força para prosseguir a cada dia, sem Ele eu nada seria.

Ao meu orientador Edgard Lamounier por sua amizade, paciência, conselhos e todo apoio que me deu durante esta pesquisa.

Ao professor Cláudio Kirner pelo seu apoio, incentivo e paciência e pelo voto de confiança em mim depositado.

Ao professor Alexandre Cardoso, que com seu jeito brincalhão sempre nos contagia de entusiasmo para prosseguir e alcançar resultados.

Ao amigo Alexandre Vaz por sua valiosa ajuda com o sensor.

À minha namorada Karine por seu amor e carinho durante o tempo que estamos juntos, te amo minha “minina” !!!

Agradeço também a toda minha família, pelo amor e carinho que sempre tive de todos eles.

Quero agradecer especialmente aos meus pais, porque um dia eles acreditaram em mim e não pouparam esforços para que eu chegasse até onde estou hoje, sou muito agradecido a vocês pai e mãe, tenho certeza que a sua recompensa está guardada por Deus, amo muito vocês.

Finalmente, agradeço aos amigos do núcleo de Computação Gráfica da UFU que sempre foram dispostos a me ajudar quando deles precisei.

RESUMO

Atualmente, a presença de computadores é notada, mesmo que sutilmente, em praticamente todas as áreas do conhecimento e atividade humana.

As vantagens trazidas pelos computadores são muitas, porém existem ainda deficiências que precisam de soluções eficientes. Os dispositivos tecnológicos e formas audiovisuais de comunicação entre o usuário e os computadores ainda são um grande desafio, pois para utilizar esses dispositivos, é necessário que o usuário aprenda as novas maneiras não naturais de interação, como clicar e digitar. Existem algumas propostas de interfaces com o usuário mais naturais, dentre elas a Realidade Aumentada.

A Realidade Aumentada tem como objetivo fornecer às pessoas, formas intuitivas de interação e visualização, semelhantes às formas de interação que acontecem no mundo real.

Assim, pode-se interagir com um objeto usando as mãos como principal forma de interação, inspecioná-lo e visualizá-lo em sua forma natural. Os elementos de interação são trazidos para o mundo real, ao invés da interação clássica com os computadores, onde o usuário tem que entrar no mundo do computador e a ele se adaptar.

São inúmeras as aplicações para a Realidade Aumentada, e já podem ser observados resultados em diversas áreas como Educação, Treinamento, Entretenimento e Medicina.

O foco deste trabalho está na integração entre Realidade Aumentada e Ciência da Saúde. Tradicionalmente, os sujeitos têm pouco ou nenhum conhecimento de como interpretar informações como raios-x e eletrocardiogramas. Assim, muitas vezes mesmo após esclarecimentos com os profissionais da área restam dúvidas quanto ao funcionamento de órgãos, particularidades de doenças e mesmo a interpretação das informações de exames. Inclui-se ainda, a necessidade de ferramentas para suporte ao treinamento de profissionais na área de saúde.

O sistema proposto nesta dissertação tem como objetivo fornecer uma interface amigável e intuitiva em Realidade Aumentada para visualização do batimento cardíaco. As informações necessárias a essa visualização são recebidas de fontes bem heterogêneas, que vão desde simples valores fixos de pulsação até a captura e visualização em tempo real de sensores de pulsação. Através de marcadores e um dispositivo de visualização, o usuário consegue visualizar no próprio corpo, o coração virtual animado e pulsando em tempo real, respondendo ao seu próprio batimento cardíaco.

O sistema desenvolvido poderá servir como base para estudos em escolas na área de saúde, bem como para visualização por sujeitos, proporcionando assim uma forma de auxiliar o aprendizado e a interpretação de informações médicas.

Palavras-chave:

Realidade Aumentada, Cardiologia, Biofeedback, Simulação, Visualização, ARToolKit, Processamento de Sinais, Processamento de Imagens, Aplicações em Tempo Real.

ABSTRACT

Nowadays, the presence of computers is noted, even subtle, in almost every area of human knowledge and activity.

The advantages emerged by computers are many, but there are still drawbacks that need efficient solutions. The tech devices and audiovisual ways of communication between the user and computers are still a great challenge, because to use successfully these devices, is needed to the user to learn the new unnatural ways of interaction, like clicking and typing. There are some more natural approaches of interfaces, and One of them is known as Augmented Reality.

Augmented Reality has as goal to provide people, with more intuitive ways of interaction and visualization, close to those that happen in real world.

So, it's possible to interact with an object using the hands as main way of interaction, inspect it and view it in it's natural shape. The interaction elements are brought to the real world, instead of classic interaction with computers, where the user has to get inside the computer's world and adapt himself to it.

The applications for Augmented Reality are countless, and results can be already seen in several areas like Education, Training, Entertainment and Medicine.

The focus of this work is on the integration between Augmented Reality and Health Science. Traditionally, the patients have limited or none knowledge about Medicine, which by the way stops him to interpret information like x-ray and electrocardiograms. So, very often even after clarifying with the specific professionals, remain doubts about organ function, disease particularities and even interpretation of tests.

The system proposed in this dissertation, has as goal to provide a friendly and intuitive interface in Augmented Reality for heart beat viewing. The data needed to visualization are received from very heterogenic fonts, which goes from simple fixed values up to real time capture and viewing from heart beat sensors. Trough markers and a viewing device, the users are allowed to view in its own chest the virtual animated heart and it's beating in real time, answering to his own heart beating.

The developed system should be used as bases to studies on health schools, as well to view by patients, providing a way of help learn and interpret of medical data.

Keywords:

Augmented Reality, Cardiology, Biofeedback, Simulation, Viewing, ARToolKit, Signal Processing, Image Processing, Real-time Applications

SUMÁRIO

1. Introdução 17

1.1. Contextualização ---------- 17

1.2. Motivação ---------- 19

1.3. Objetivo e Metas ---------- 21

1.4. Estrutura da Dissertação ---------- 22

2. Fundamentos Teóricos 23

2.1. Introdução ---------- 23

2.2. Realidade Virtual ---------- 23

2.3. Realidade Aumentada ---------- 29

2.4. Captura e Tratamento de Sinais Cardiológicos ---------- 35

2.5. Conclusões ---------- 38

3. Trabalhos Relacionados 39

3.1. Introdução ---------- 39

3.2. Project on Image Guided Surgery ---------- 39

3.3. VIDA: Visualizador Interactivo de Dados biomecânicos Animados ---------- 43

3.4. Die-Heart ---------- 47

3.5. Mitral Valve Surgery Using Pre-Operative Cardiac Models ---------- 51

3.6. Sumário e Conclusões ---------- 54

4. Arquitetura do Sistema 58

4.1. Introdução ---------- 58

4.2. Arquitetura ---------- 58

4.3. Elementos físicos do sistema ---------- 61

4.4. Elementos de software do sistema ---------- 63

4.5. Conclusões ---------- 65

5. Detalhes de Implementação 66

5.1. Introdução ---------- 66

5.2. Interface de operação ---------- 66
5.3. Gerador da visualização em Realidade Aumentada (GVRA) ---------- 91
5.4. Conclusões ---------- 105
6. Resultados e Limitações 106
6.1. Introdução ---------- 106
6.2. Preparação do ambiente de Realidade Aumentada ---------- 106
6.3. Escala automática do coração virtual ---------- 107
6.4. Opção Simular Pulso Constante ---------- 108
6.5. Opção Buscar ECG em Arquivo ---------- 112
6.6. Opção Capturar ECG de Imagem ---------- 114
6.7. Opção Captar Dados do Sensor. ---------- 122
6.8. Testes adicionais ---------- 124
6.9. Limitações Detectadas ---------- 125
6.10. Sumário e Conclusões ---------- 126
7. Conclusões e trabalhos futuros 127
7.1. Introdução ---------- 127
7.2. Conclusões ---------- 127
7.3. Trabalhos Futuros ---------- 129
7.4. Considerações Finais ---------- 131
Referências Bibliográficas 132
Apêndice 1 - PUBLICAÇÕES 140
Anexo A – Tarefas Relacionadas à Realidade Virtual 141

LISTA DE FIGURAS

Figura 1.1 – Ambiente em Realidade Virtual. 18
Figura 1.2 – Ambiente em Realidade Aumentada [9]. 19
Figura 2.1 – Ambiente de Realidade Virtual Não-Imersiva [22]. 26
Figura 2.2 – Ambiente com Isolamento Acústico e Visual [23]. 26
Figura 2.3 – 2 Exemplos de dispositivos visuais HMD [30]. 28
Figura 2.4 – Exemplos de Dispositivos Táteis do tipo Luva [31,32]. 28
Figura 2.5 - Diagrama adaptado de Realidade/Virtualidade Contínua [35]. 29
Figura 2.6 - Método de visão por vídeo baseada em monitor e webcam [37]. 31
Figura 2.7 – Estrutura de um marcador no ARToolKit. 32
Figura 2.8 – Imagem Transformada com Diferentes valores de limiar/threshold. 32
Figura 2.9 – Reconhecimento de Marcadores. 33
Figura 2.10 – Alinhamento entre marcadores e objetos virtuais. 34
Figura 2.11 – Estrutura de Funcionamento do software ARToolKit. 35
Figura 2.12 – Máquina para Exame de ECG com saída em papel milimetrado [49]. .. 36
Figura 2.13 – ECG com as divisões das fases do ciclo cardíaco [50]. 36
Figura 2.14 – Posicionamento dos LEDs e Fotosensor em um oxímetro[84]. 38
Figura 3.1 – Exemplo de Ressonância Magnética [10]. 40
Figura 3.2 – modelos Tridimensionais gerados a partir de imagens de ressonância magnética [10]. 41
Figura 3.3 – Equipamento para Rastreamento dos órgãos do paciente [10]. 42
Figura 3.4 – Alinhamento entre pacientes e modelos tridimensionais [10]. 42
Figura 3.5 – Arquitetura do projeto VIDA [18]. 44
Figura 3.6 – Sincronização do vídeo com boneco virtual [18]. 46
Figura 3.7 – Visualização de esforço muscular [18]. 46
Figura 3.8 – Arquitetura do sistema Die-Heart [19]. 47
Figura 3.9 – Ecocardiograma e imagens do processo de segmentação [19]. 48
Figura 3.10 – Aperfeiçoamento dos Contornos [19]. 49
Figura 3.11 – Contornos nos 6 ângulos e criação da geometria tridimensional [19]. .. 49
Figura 3.12 – Diversos modelos-chave do ciclo cardíaco [19]. 50
Figura 3.13 – Visualizações do Die-Heart em Realidade Aumentada [19]. 51
Figura 3.14 – Diagrama com a Estrutura do sistema [20]. 52
Figura 3.15 – Corte de ressonância Magnética e Modelos tridimensionais [20]. 53
Figura 3.16 – Exemplos de Visualizações obtidas com o sistema proposto [20]. 54

Figura 4.1 – Arquitetura do sistema. 58
Figura 5.1 – Ambiente do C++ Builder. 67
Figura 5.2 – Componentes padrão do C++ Builder x Componentes Suipack. 68
Figura 5.3 – Tela Inicial do sistema. 69
Figura 5.4 – Elementos de Interface da opção Simular Pulso Constante. 70
Figura 5.5 – Situação de Erro na Opção Simular Pulso Constante. 71
Figura 5.6 – Cenário resultante do clique sobre o botão Simular. 72
Figura 5.7 – Arquivo config.ini gerado pela opção Simular Pulso Constante. 72
Figura 5.8 – Elementos de Interface da opção Buscar ECG de Arquivo. 74
Figura 5.9 – Pré-visualização de um arquivo de ECG e ajuste de Amostras/seg. 75
Figura 5.10 – Interface da Opção Capturar ECG de Imagem. 77
Figura 5.11 – Alocação Dinâmica de uma matriz tridimensional em C++. 79
Figura 5.12– Interface após abertura da imagem. 79
Figura 5.13 – Seleção de cor pelo botão “Escolher Cor do Sinal”. 82
Figura 5.14 – Estado da Janela após pressionamento do botão “Obter Sinal”. 83
Figura 5.15 – Processo de Apagar Ruídos do gráfico ECG. 85
Figura 5.16 – Janela com as informações atualizadas. 86
Figura 5.17 - Interface final da opção Capturar ECG de Imagem. 87
Figura 5.18 – Interface Inicial da opção Captar Dados do Sensor. 88
Figura 5.19 – Sensor utilizado no sistema. 89
Figura 5.20 – Esquema de funcionamento do circuito de amplificação desenvolvido. 89
Figura 5.21 – Interface com sensor em funcionamento. 90
Figura 5.22 – Modelo Original do Coração Virtual. 92
Figura 5.23 – Comparação entre o modelo original do coração virtual com o editado. 93
Figura 5.24 – Textura editada e corações virtuais com e sem textura. 94
Figura 5.25 – Poses de uma animação. 95
Figura 5.26 – Modelos virtuais adicionais utilizados no sistema. 96
Figura 5.27 – Marcadores utilizados no sistema. 97
Figura 5.28 – Redimensionamento do coração virtual. 98
Figura 6.1 – Teste da Funcionalidade de escala automática do coração virtual. 108
Figura 6.2 – Configurações para teste da opção Simular Pulso Constante. 109
Figura 6.3 – Quadros da animação simples com os 2 sujeitos. 109
Figura 6.4 – Arquivo animstyle.ini padrão. 110

Figura 6.5 – Quadros da animação complexa para os 2 sujeitos. 111
Figura 6.6 – Novas Configurações inseridas na interface de operação. 111
Figura 6.7 – Tratamento de erro na abertura de arquivo ECG. 112
Figura 6.8 – Gráfico aberto com sucesso na opção Buscar ECG em Arquivo. 113
Figura 6.9 – Visualização da animação simples a partir de gráfico de controle..... 113
Figura 6.10 – Configurações utilizadas no teste final da opção Buscar ECG em Arquivo. 114
Figura 6.11 – Imagens utilizadas nos testes..... 115
Figura 6.12 – Conversão de Imagem pelo Irfan View..... 116
Figura 6.13 – Escolha semi automatizada da cor do sinal. 117
Figura 6.14 – Escolha manual da cor do sinal. 117
Figura 6.15 – Reconhecimento do Sinal com tolerâncias de 15% e 35%. 118
Figura 6.16 – Prévia atual do sinal reconhecido e gráfico mostrando seu armazenamento. 119
Figura 6.17 – Imagem de controle, gráfico com falhas e correção de falhas resultante. 120
Figura 6.18 – Interface Atualizada e arquivo ECG gerado a partir de Imagem. 121
Figura 6.19 – Interface da Opção Captar Dados do Sensor para os 2 sujeitos. 123
Figura 6.20 – Simulação com o sensor para os 2 sujeitos, sendo 65 o valor de bpm do sujeito à esquerda e 75 o valor de bpm do sujeito à direita. 124
Figura A.1 – Sistema de Modelagem Poligonal. 141
Figura A.2 – Primitivas do 3ds Max®..... 142
Figura A.3 – Primitivas do VRML. 143
Figura A.4 – Criação de Objetos Partindo de Primitivas..... 143
Figura A.5 – Criação de um Modelo Utilizando Modelagem Poligonal. 144
Figura A.6 – Utilização da Ferramenta MeshSmooth para suavização de modelos. . 145
Figura A.7 – Diferentes configurações de texturização sobre um modelo. 146
Figura A.8 – Resultado Visual para o Código da Tabela 1. 147
Figura A.9 - Textura Fractal criada em 3dsMax..... 148
Figura A.10 – Algumas Formas de Mapeamento do 3dsMax. 149
Figura A.11 – Luzes Virtuais e suas Correspondentes no Mundo Real [44,45,46]. .. 152

LISTA DE TABELAS

Tabela 3.1 – Sistemas Estudados e Características. 55
Tabela 5.1 – Exemplos de situações possíveis para a animação simples 101
Tabela 7.1 – Características dos Sistemas Estudados e do sistema desenvolvido..... 128
Tabela A.1 – Código em VRML para aplicação de Texturas em Objetos. 147

Capítulo 1

1. INTRODUÇÃO

1.1. Contextualização

Quando os computadores começaram a ser desenvolvidos, não havia grandes preocupações com as formas e dispositivos de comunicação com o usuário, pois eram máquinas sofisticadas e exclusivamente operadas por pessoas altamente qualificadas. Com o tempo, porém, os computadores foram diminuindo de tamanho, aumentando seu desempenho e reduzindo o custo, o que contribuiu grandemente para a popularização dos mesmos. A partir de então, as interfaces de operação, tanto em termos de dispositivos de entrada quanto de visualização das informações, passaram por um longo processo de desenvolvimento, que até hoje continua a ser aprimorado [1,2,3].

A busca por uma interface de operação ideal ainda é uma preocupação atual, pois as mais modernas interfaces presentes nos atuais sistemas não satisfazem plenamente os requisitos de usabilidade e naturalidade de operação.

Já existem tecnologias visando esse tipo de interface com o usuário, dentre elas pode-se destacar a Realidade Virtual e a Realidade Aumentada [4]. Essas duas tecnologias trazem como proposta interfaces de visualização mais naturais, pois ao invés de botões e janelas, a interação nesse caso é feita com objetos tridimensionais que mimetizam com boa qualidade seus correspondentes no mundo real, sendo capazes de apresentar ao usuário detalhes de forma, cor, textura e mesmo comportamentos dos objetos. Essas características propiciam uma interação mais fácil, com menos necessidade de adaptação. A grande diferença entre essas duas tecnologias está na forma e no ambiente de interação.

A Realidade Virtual utiliza técnicas de Computação Gráfica e multimídia para reproduzir objetos tridimensionais bastante realistas, com proporções geométricas, cor, textura, animações e comportamentos semelhantes aos correspondentes no mundo real. Dentre as maiores aplicações de Realidade Virtual, podem-se destacar os jogos de videogames como Playstation [5], Nintendo Wii [6] e Xbox [7]. O mercado de jogos cresceu tanto que há algum

tempo seu faturamento ultrapassou o das produções cinematográficas [8]. A Figura 1.1 apresenta um ambiente em Realidade Virtual.

Figura 1.1 – Ambiente em Realidade Virtual.

Nas aplicações de Realidade Virtual, pouco ou nenhum elemento do mundo real está presente, praticamente tudo é reconstruído no ambiente virtual, e o usuário interage com esses elementos através de dispositivos como o mouse, teclado, joysticks, entre outros. Algo que pode se visto como uma deficiência nesses dispositivos, é que eles demandam treinamento por parte dos usuários para que haja um bom aproveitamento das aplicações. Em muitas situações este treinamento acontece sem maiores complicações, porém algumas classes de usuários podem ter problemas na utilização destes dispositivos, sendo que alguns até se assustam ao se deparar com dezenas de botões e formas novas de interação. Embora existam situações onde ainda é imprescindível a presença de dispositivos convencionais, podem ser identificadas muitas outras que se beneficiariam de formas mais naturais de interação. É nesse ponto que se insere a Realidade Aumentada. A Figura 1.2 apresenta um ambiente em Realidade Aumentada.

Figura 1.2 – Ambiente em Realidade Aumentada [9].

A Realidade Aumentada utiliza diversos elementos presentes na Realidade Virtual como base de visualização, porém o ambiente predominante nas aplicações é o mundo real. Os elementos virtuais são inseridos de alguma forma no mundo real e o usuário interage com esses elementos utilizando ações comuns do mundo real, como pegar, empurrar e manipular. É possível segurar objetos virtuais com as mãos, inspecioná-los e mesmo executar ações mais complexas de interação entre 2 ou mais objetos, como por exemplo, a montagem de um quebra-cabeças virtual unindo as peças virtuais através de manipulação no mundo real.

Assim como na Realidade Virtual, as aplicações possíveis para Realidade Aumentada são inúmeras, nas mais diversas áreas do conhecimento. Uma das áreas que tem demonstrado grande potencial para desenvolvimento de aplicações e estudos é a Ciência da Saúde. Já existem aplicações para auxiliar cirurgias, visualizar próteses, órgãos virtuais, estudo de anatomia, entre outras [10, 11, 12, 13].

1.2. Motivação

Os sinais do corpo utilizados em diagnóstico médico são tradicionalmente capturados e armazenados em mídias e formas que demandam

grande treinamento técnico para que seja possível a interpretação e compreensão dos dados ali contidos [14,15,16,17]. Assim, estudantes e pessoas leigas, ao observarem um exame ou diagnóstico médico, frequentemente não conseguem entender o significado de símbolos, gráficos ou imagens de raios X.

A maioria dos sistemas de Realidade Aumentada estudados até a compilação desta dissertação, tratam os sinais de interesse de forma estática, utilizando apenas sinais pré-capturados, sem um biofeedback em tempo real [10,18,19,20]. Isto pode incorrer em problemas na interpretação dos dados estudados, exigindo um esforço mais significativo por parte do profissional da área, na avaliação dos dados referentes ao seu sujeito. Além disso, não foi detectado nenhum trabalho em Realidade Aumentada que tivesse como foco de estudo avaliações em tempo real dos batimentos cardíacos através da análise de sinais cardiológicos de sujeitos.

A captação de sinais cardiológicos tem como objetivo monitorar e visualizar o funcionamento do coração, extraindo informações tais como a pulsação, potência dos músculos cardíacos e periodicidade das batidas. Essas informações permitem ao médico cardiologista diagnosticar diversas anomalias e prover tratamentos para as mesmas. Porém, assim como os outros sinais mencionados, os sinais eletrocardiográficos são armazenados em mídias nada intuitivas, tornando o entendimento e compreensão dos dados ali contidos um desafio. Desta forma, pessoas sem conhecimentos técnicos em medicina são impossibilitadas de interpretar mesmo que parcialmente as informações, incluindo-se aí os sujeitos portadores de problemas cardiológicos e estudantes de Medicina.

Utilizando técnicas de Realidade Aumentada, é possível criar novas formas para visualização e interpretação de sinais cardiológicos, formas essas que podem facilitar o entendimento das informações por pessoas leigas e mesmo estudantes de Medicina.

Houve outros pontos de importância decisiva na motivação desta pesquisa. São eles:

- Necessidade de Desenvolvimento de aplicações em Realidade Aumentada no campo da Cardiologia, especificamente para o tratamento de sinais eletrocardiográficos;

- Existem diversas formas de entradas de dados cardiológicos, motivando a criação de um sistema em Realidade Aumentada capaz de interpretá-las uniformemente e gerar saídas compatíveis;
- Ausência de sistemas em Realidade Aumentada para captação e visualização em tempo real de sinais eletrocardiográficos;
- Mostrar a possibilidade de integração entre sistemas de Realidade Aumentada e sensores para captura de dados biológicos

1.3. Objetivo e Metas

O objetivo deste trabalho é investigar técnicas em Sistemas de Realidade Aumentada que permitam, de forma efetiva, visualizar sinais cardiológicos advindos de diversas fontes de informação. Para chegar aos objetivos propostos, foram definidas algumas metas a serem cumpridas. São elas:

- Pesquisar softwares de visualização médica que utilizem técnicas de Realidade Aumentada;
- Estudar o funcionamento do coração humano, bem como a interpretação e extração de informações dos sinais cardiológicos capturados;
- Investigar e classificar todas as possíveis formas de armazenamento de sinais cardiológicos;
- Pesquisar os diversos tipos de sensores existentes para captura de sinais cardiológicos, as tecnologias envolvidas e seu funcionamento;
- Escolher um sensor de baixo custo e fácil integração para utilização no sistema e comprovação da possibilidade de integração com Realidade Aumentada;
- Desenvolver módulos em sistema de computação, responsáveis pelo processamento das diversas formas de entradas de sinais, dentre elas os sinais advindos do sensor;
- Gerar um coração virtual com características de geometria, textura e comportamento que simulem com relativa precisão um coração humano real;
- Desenvolver animações para o coração virtual que se assemelhem aos movimentos cardíacos naturais.
- Integrar todos os módulos do sistema desenvolvido em uma interface com o usuário rica em usabilidade e funcionalidade.

- Testar o sistema desenvolvido com dados reais de eletrocardiogramas, bem como voluntários para teste da integração do sistema com o sensor proposto;
- Avaliar a usabilidade do sistema junto a potenciais usuários.

1.4. Estrutura da Dissertação

O Capítulo 2 investiga diversos aspectos relevantes a esta pesquisa, de modo a fornecer uma base teórica sólida para os tópicos subseqüentes.

O Capítulo 3 apresenta algumas aplicações já existentes em sistemas com temas relacionados de alguma forma ao conteúdo desta pesquisa.

A arquitetura do sistema proposto, seus requisitos e funcionalidades, bem como as ferramentas a serem utilizadas na implementação são apresentados no Capítulo 4.

Os detalhes implementação do sistema proposto são discutidos no Capítulo 5, onde é descrita a criação de cada um dos módulos e as técnicas utilizadas.

O Capítulo 6 descreve os resultados conseguidos com esta pesquisa, bem como as limitações detectadas no sistema desenvolvido.

O Capítulo 7 é destinado a expor as conclusões obtidas com esta pesquisa, discutindo as contribuições deste trabalho, bem como trabalhos futuros possíveis que possam aprimorar os resultados já existentes.

As referências bibliográficas utilizadas na composição deste trabalho são apresentadas ao final desta dissertação.

Capítulo 2

2. FUNDAMENTOS TEÓRICOS

2.1. Introdução

A presente pesquisa apresenta um caráter bastante multidisciplinar, abrangendo diversas áreas de estudo. Sendo assim, torna-se adequada uma explanação ainda que breve dos diversos conceitos e tecnologias utilizadas. Os conceitos gerais das tecnologias serão apresentados de forma sucinta, enquanto os aspectos mais específicos e diretamente relacionados com este trabalho poderão receber maior destaque, de acordo com a necessidade deste detalhamento.

2.2. Realidade Virtual

Quando os computadores começaram a ser desenvolvidos, não havia grandes preocupações com a facilidade de utilização das interfaces de operação, pois eram máquinas sofisticadas e exclusivamente operadas por pessoas altamente qualificadas. Neste contexto, interfaces baseadas em alguns poucos símbolos visuais como LEDs intermitentes e dígitos eram plenamente aceitas pela comunidade científica. Com o tempo porém, os computadores foram diminuindo de tamanho, aumentando seu desempenho e reduzindo o custo, o que contribuiu grandemente para a popularização dos mesmos. A partir de então, as interfaces de operação, tanto em termos de dispositivos de entrada quanto de visualização das informações, passaram por um longo processo de desenvolvimento, que até hoje continua a ser aprimorado. O surgimento de dispositivos como teclado e mouse aliado a novas interfaces com mais componentes gráficos, possibilidades mais amplas de cores e efeitos de animação e som, tornaram possível ao usuário comum realizar diversas tarefas cotidianas no computador sem precisar de auxílio técnico ou grandes períodos de treinamento. A exploração de muitos dos novos recursos foi inclusive possibilitada em muitos casos através da experimentação e intuição, dispensando manuais de utilização dispendiosos e geralmente cansativos de se ler e seguir.

As interfaces visuais baseadas em janelas ou outras abordagens, dominaram o mercado por muitos anos, porém os defeitos e dificuldades trazidos por essas interfaces já começam a aparecer e a prejudicar a produtividade em diversas áreas do

conhecimento, o que mostra a demanda por novas formas de interação entre os computadores e as pessoas, formas essas que sejam de fácil adaptação e operação. A Realidade Virtual é uma das tentativas de criar uma melhor interação com os computadores [4,21].

É possível encontrar várias definições para a Realidade Virtual, uma delas é de que ela é uma interface avançada para aplicações computacionais, que permite ao usuário navegar e interagir, em tempo real, em um ambiente tridimensional, podendo fazer uso de dispositivos multisensoriais, para atuação ou feedback [4,21].

Uma das atribuições dos ambientes tridimensionais em sistemas de Realidade Virtual é a busca por reproduzir objetos e cenários reais na maioria de suas características, como cores, textura, formato geométrico e reações a ações. Em alguns casos também é possível definir cenários e objetos fictícios ou microscópicos, como por exemplo uma viajem pelas artérias até o coração ou um jogo onde os principais cenários são de guerras entre naves espaciais. Uma boa reprodução de objetos, cenários e interações pode iludir o usuário ao ponto de ele acreditar que está, de fato, dentro do cenário, participando dos eventos que ali ocorrem.

Muitas vezes não é possível gerar ambientes altamente detalhados, com texturas de alta resolução e qualidade visual fotorrealista, devido a limitações do sistema, como por exemplo, dispositivos de hardware. Apesar dessas limitações, a tecnologia de Realidade Virtual pode ser utilizada para inúmeros fins, como treinamento, aprendizado e entretenimento.

Além das limitações visuais impostas pelo hardware atual, há também limitações relativas às formas de interação com os ambientes virtuais, pois toda comunicação entre usuários e ambientes ocorre por intermédio de dispositivos especiais, como teclado, mouse, joysticks e outros. De acordo com o tipo de dispositivos utilizados na interação e a intensidade da sensação de realidade presenciada pelo usuário, os sistemas de Realidade Virtual podem ser divididos em Imersivos e Não-Imersivos.

Sistemas de Realidade Virtual Não-Imersivos são os mais populares, visto que os dispositivos necessários para sua utilização são bastante acessíveis e conhecidos. Como dispositivos de interação pode-se destacar o teclado, mouse e joysticks, e dentre os dispositivos de visualização pode-se destacar os televisores e monitores de computador. Além da visualização e interação, muitos sistemas de Realidade Virtual contam também com efeitos sonoros para enriquecer a experiência com a aplicação.

Sistemas de Realidade Virtual Não-Imersivos são assim chamados pela forma com que a interação e intercâmbio de informações com o usuário acontece. Primeiramente, os dispositivos utilizados para impressionar os sentidos visual e auditivo do usuário e propiciar a este a sensação de presença no ambiente, são na maioria dos casos uma pequena parte do ambiente real, possibilitando a inserção de ruídos na apresentação das aplicações, tanto nos aspectos visuais como nos sonoros. Por exemplo, um jogo de computador utilizando um monitor e caixas de som para apresentar ao usuário os elementos do jogo pode sofrer interferência do meio, visto que o monitor é apenas uma pequena janela no vasto campo de visão presenciado pelo usuário e, da mesma forma, as caixas de som são apenas algumas das fontes sonoras que podem estar presentes no ambiente. Poderão acontecer distrações por conta da iluminação, objetos em movimento e sons externos como conversas e ruídos em geral. Dessa forma, para que o usuário tenha uma boa experiência com as aplicações, ele deve tentar filtrar toda espécie de ruídos do mundo real, prendendo-se firmemente àquela pequena fração de informações representada pela aplicação, em detrimento de todas as outras que possam estar acontecendo em paralelo, o que exige bastante concentração. Uma forma encontrada para se maximizar a experiência do usuário com os sistemas é utilizar ambientes isolados ao máximo do mundo real. Ambientes desse tipo são utilizados não apenas para auxiliar sistemas de Realidade Virtual, mas também outras aplicações, como nos cinemas, onde o ambiente é escuro para concentrar a informação visual na tela e isolado acusticamente, fazendo a informação sonora da aplicação predominar sobre os sons do ambiente real. As Figuras 2.1 e 2.2 ilustram, respectivamente, um sistema de Realidade Virtual Não-Imersivo e um ambiente isolado visual e acusticamente.

Figura 2.1 – Ambiente de Realidade Virtual Não-Imersiva [22].

Figura 2.2 – Ambiente com Isolamento Acústico e Visual [23].

Como foi apresentado, os sistemas de Realidade Virtual Não-Imersivos exigem bastante concentração por parte dos usuários para que haja uma interação satisfatória. Muitas pessoas não têm disposição nem tampouco motivação para se submeterem a esse

nível de concentração, o que por vezes impossibilita o bom aproveitamento da tecnologia. Para tentar solucionar os maiores problemas presentes na Realidade Virtual Não-Imersiva, existem abordagens com diferentes formas de interação com o usuário, presentes nos sistemas de Realidade Virtual Imersiva.

Tal como nos sistemas Não-Imersivos, as formas de comunicação com o usuário nos sistemas de Realidade Virtual Imersiva baseiam-se nos sentidos visual e auditivo, bem como na interação com as aplicações por meio de dispositivos especiais. A grande diferença está justamente na forma de implementação dos dispositivos. Ao invés de competirem com os estímulos visuais e auditivos presentes no mundo real como é comum nos sistemas Não-Imersivos, os dispositivos tanto a nível visual, auditivo e de interação são projetados de forma a isolar os usuários do mundo real, inserindo-os completamente nas realidades virtuais fornecidas pelas aplicações e evitando interferências externas. Assim, os dispositivos nesta abordagem tendem a ser mais customizados para o usuário, muitas vezes acoplados ao próprio corpo do mesmo. Por esse motivo, surgem diversas exigências no sentido de proporcionar ao mesmo tempo conforto, confiabilidade, tamanho/peso reduzidos e respostas às ações do usuário em tempo real ou muito próximo disto. Para satisfazer estas exigências, os custos inerentes à construção e projeto dos dispositivos sofrem aumentos significativos na maioria dos casos, o que também acaba sendo refletido nos valores finais dos produtos.

Por serem produtos de aplicações específicas, pouco conhecidos pelas pessoas em geral, de uso individual e preço muitas vezes elevado, os dispositivos de Realidade Virtual Imersiva enfrentam bastante dificuldades para se popularizarem e serem produzidos em massa.

Apesar dos problemas apresentados, a experiência de utilização de aplicações de Realidade Virtual com dispositivos Imersivos se torna muito mais real para o usuário, por reproduzir com relativa precisão algumas características percebidas pelos sentidos humanos. Por exemplo, um dispositivo de visualização chamado HMD (Head Mounted Display) [24] utilizado em Realidade Virtual Imersiva, se bem implementado pode reproduzir características ópticas como estereoscopia [25,26], profundidade de campo [27,28] e rastreamento do movimento ocular. Existem hoje dispositivos para simular sensações auditivas, visuais e mesmo táteis. A Figura 2.3 apresenta 2 exemplos de HMD e a Figura 2.4 mostra 2 exemplos de um dispositivo tátil do tipo luva [29].

Figura 2.3 – 2 Exemplos de dispositivos visuais HMD [30].

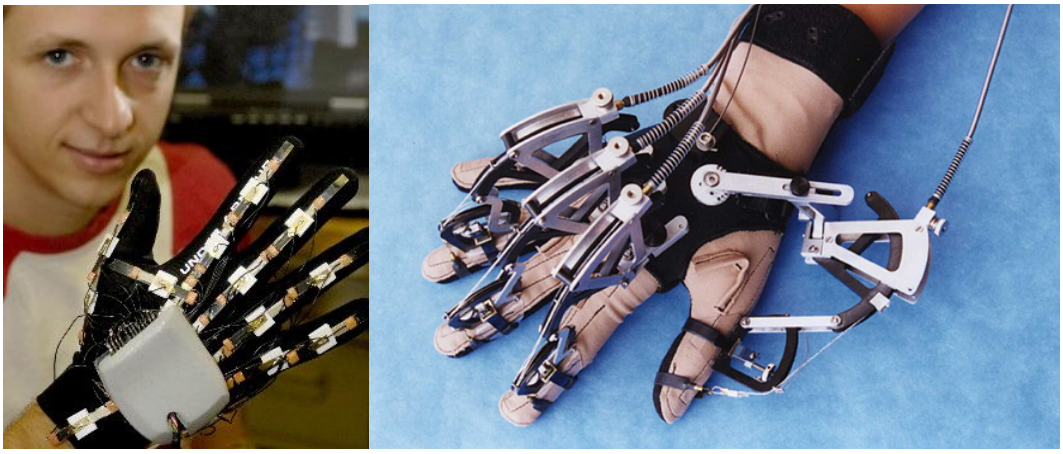

Figura 2.4 – Exemplos de Dispositivos Táteis do tipo Luva [31,32].

Não restam dúvidas quanto aos avanços de interatividade conseguidos através da Realidade Virtual em diversas áreas de aplicação. Novos dispositivos e técnicas de visualização são desenvolvidos a cada dia aproximando cada vez mais o usuário das aplicações, tornando a imersão nos ambientes virtuais muito mais realista.

A abordagem de interatividade utilizada pela Realidade Virtual no sentido de isolar o usuário do mundo real, fazendo-o interagir através de dispositivos especiais, não é uma solução adequada para toda espécie de problemas e aplicações. Os próprios dispositivos especiais, em particular os de sistemas Não-Imersivos, trazem desconforto a alguns usuários, resistência de outros e mesmo receio na utilização, pelo fato de nem sempre terem utilização facilitada e intuitiva, exigindo por vezes um período de adaptação ou treinamento. Essa restrição diminui a quantidade de aplicações possíveis para a Realidade Virtual e ao mesmo causa um certo isolamento de alguns grupos de usuários.

Para solucionar as deficiências da Realidade Virtual existem algumas abordagens diferenciadas, como a Projeção Interativa [33] onde através de projetores

multimídia e câmeras de vídeo é possível criar aplicações que utilizam apenas as mãos como meio de interação com os usuários e fornecem informações visuais em tempo real projetadas sobre alguma superfície. Outra abordagem bastante interessante é a Realidade Aumentada que foi utilizada neste projeto como base da interação e informação visual e será detalhada a seguir.

2.3. Realidade Aumentada

A Realidade Aumentada é definida usualmente como a sobreposição de objetos virtuais tridimensionais, gerados por computador, com um ambiente real, por meio de algum dispositivo tecnológico [34]. Entretanto, esta conceituação é muito geral e só fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

A Realidade Misturada [34,35], misturando o real com o virtual, abrange duas possibilidades: a Realidade Aumentada, cujo ambiente predominante é o mundo real, e a Virtualidade Aumentada, cujo ambiente predominante é o mundo virtual. Pode-se dizer, então, que a Realidade Aumentada é uma particularização da Realidade Misturada. A Figura 2.5 apresenta o diagrama adaptado de Realidade/Virtualidade Contínua, mostrando as possibilidades gradativas de sobreposição do real com o virtual e vice-versa.

Figura 2.5 - Diagrama adaptado de Realidade/Virtualidade Contínua [35].

A Realidade Aumentada proporciona ao usuário uma interação segura, sem grandes necessidades de adaptação ou treinamento, uma vez que ele pode trazer para o seu ambiente real objetos virtuais, aumentando a visão que ele tem do mundo real. Isto é obtido, através de técnicas de Visão Computacional e de Computação

Gráfica/Realidade virtual, resultando na sobreposição de objetos virtuais com o mundo real [36,37,38,39].

Além de permitir que objetos virtuais possam ser introduzidos em ambientes reais, a Realidade Aumentada proporciona também, ao usuário, o manuseio desses objetos com as próprias mãos, possibilitando uma interação atrativa e motivadora com o ambiente [40,41,42].

No entanto, para que os objetos virtuais façam parte do ambiente real e sejam manuseados, deve-se utilizar um software com capacidade de processar a visão do ambiente real e posicionamento dos objetos virtuais, além de acionar dispositivos tecnológicos apropriados para Realidade Aumentada.

Apesar de grande parte dos dispositivos utilizados em um ambiente de Realidade Virtual poderem ser utilizados em ambientes de Realidade Aumentada, existem casos, onde é necessário que haja algumas adaptações. Por conseguinte, as principais diferenças encontradas entre esses dispositivos tecnológicos estão situadas nos displays e rastreadores [43].

Existem várias maneiras de desenvolver uma aplicação de Realidade Aumentada. Uma das mais simples é composta por um microcomputador com uma webcam instalada.

Para que se possa misturar o ambiente real ao virtual, um software específico de Realidade Aumentada captura as imagens do ambiente real, através de uma webcam, e as processa para fazer a sobreposição dos objetos virtuais no cenário real. Além disso, o software também gerencia interações dos objetos virtuais e seu posicionamento em tempo real, fazendo com que o usuário tenha a impressão de um ambiente unificado. A Figura 2.6 mostra o modelo desse método, denominado visão por vídeo baseada em monitor e webcam.

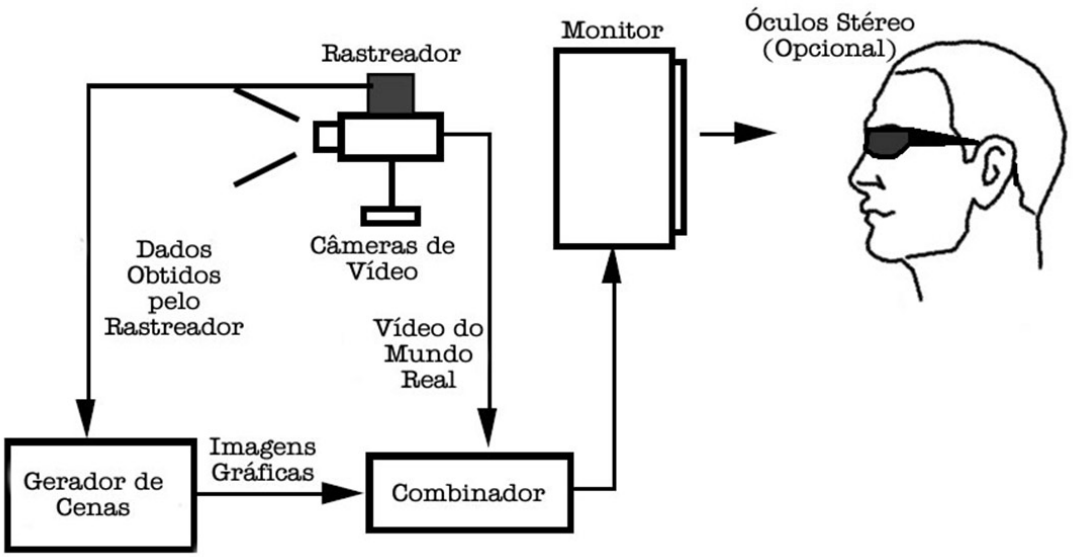

Figura 2.6 - Método de visão por vídeo baseada em monitor e webcam [37].

O método exemplificado na Figura 2.6 é utilizado por muitas aplicações de Realidade Aumentada, dentre elas o ARToolKit [45].

O ARToolKit é um software apropriado para o desenvolvimento de aplicações de Realidade Aumentada, operando através de técnicas de visão computacional, processamento de imagens e posicionamento espacial.

As maiores dificuldades que ocorrem no desenvolvimento das aplicações de Realidade Aumentada, estão em executar em tempo real uma série de atividades, envolvendo: rastreamento de objetos reais, calibração e posicionamento de objetos virtuais, combinação dos dois ambientes e realização de interação com objetos virtuais. Nesse contexto, através de técnicas de visão computacional, o software ARToolKit permite rapidamente rastrear e calcular a posição de suas placas de referência (marcadores), em relação à câmera, possibilitando a inserção de objetos virtuais sobre elas e permitindo o seu manuseio com as mãos, movimentando junto os objetos virtuais associados. Estas placas ou marcadores são objetos físicos com bordas quadradas ou retangulares geralmente de cor preta. No centro da marca é inserido um desenho também preto sobre fundo branco ou o contrário. A monocromaticidade e alto contraste das marcas é importante para o processamento de imagem realizado posteriormente pela aplicação. A Figura 2.7 exemplifica a estrutura de um marcador no ARtoolKit.

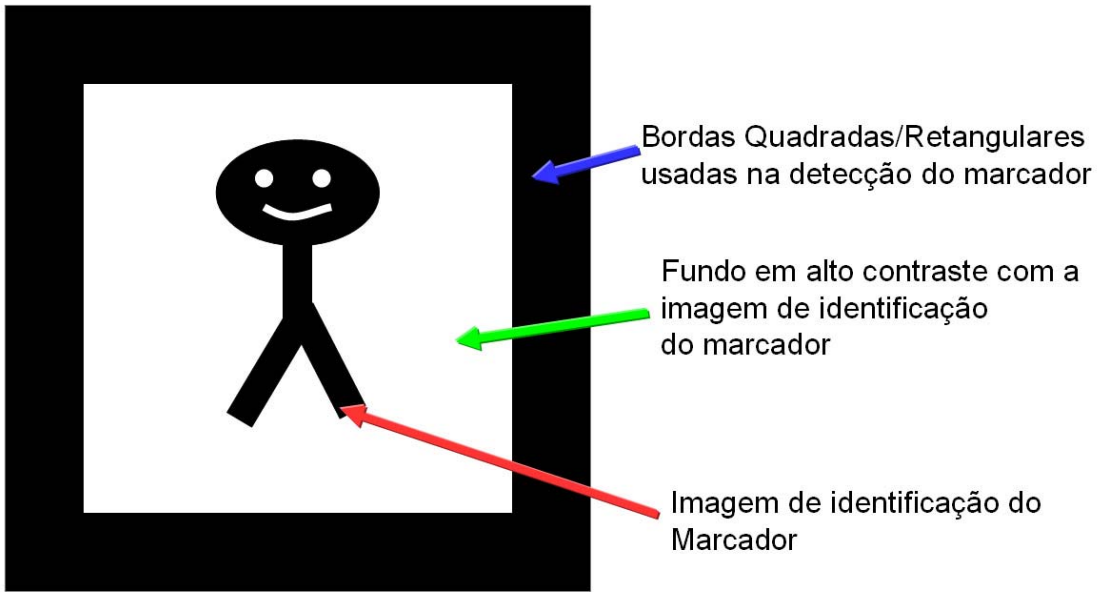

Figura 2.7 – Estrutura de um marcador no ARToolKit.

Para detectar as marcas e mesclar os objetos virtuais ao mundo real de forma satisfatória o software ARToolKit precisa realizar algumas tarefas.

Primeiramente, a imagem de vídeo capturada pela câmera é transformada em uma matriz com valores binários representando as cores Preto e Branco através de um algoritmo conhecido como limiarização ou threshold. Numa das formas mais simples deste algoritmo, são descartados os canais de cor na imagem original, aproveitando-se somente a intensidade de luminância de cada ponto. É definido então um valor de limiar/threshod para a divisão de estados, onde os valores de luminância abaixo do limiar/threshold serão considerados como 0/preto e os valores acima do limiar/threshold serão considerados 1/branco. A Figura 2.8 mostra uma imagem transformada com diferentes valores de limiar/threshold.

Figura 2.8 – Imagem Transformada com Diferentes valores de limiar/threshold.

Como pode ser notado na Figura 2.8, a diferença de valor de limiar/threshold gera resultados bastante diferentes nas áreas coloridas das imagens. Também pode ser notado que em todas as imagens o marcador permanece igual, pois propositalmente ele

já é construído com tonalidades em alto contraste, como se fosse naturalmente limiarizado, dificultando a sua distorção pelo algoritmo.

Em situações reais, os marcadores podem sofrer várias formas de interferência advindas de fatores diversos como luminosidade fraca ou excessiva, reflexão no material do marcador, baixo contraste na impressão da marca, entre outros fatores. É responsabilidade do desenvolvedor evitar e minimizar estas interferências, de forma a facilitar o reconhecimento das marcas.

Após a fase de limiarização, o ArtoolKit faz um exame na imagem, procurando encontrar regiões quadradas ou retangulares. Isso é feito isolando-se as regiões de uma mesma cor e estabelecendo-se relações de geometria nestas regiões. Retângulos e quadrados tem quatro lados com ângulos de 90º entre eles, e este fato é usado para estabelecer onde existem regiões aproximadamente quadradas ou retangulares na imagem. Se forem encontradas áreas na imagem onde se formem 2 ou 3 lados com ângulos de 90º, estes serão descartados, sendo passados para a próxima fase de processamento apenas áreas com os 4 lados em ângulos de 90º.

Depois de encontrar as áreas quadradas ou retangulares na imagem, o ARToolKit examina as imagens no interior dos quadrados (em um processo semelhante a um código de barras), e então os compara com marcadores pré cadastrados em uma base de dados de imagens. Este processo é exemplificado na Figura 2.9

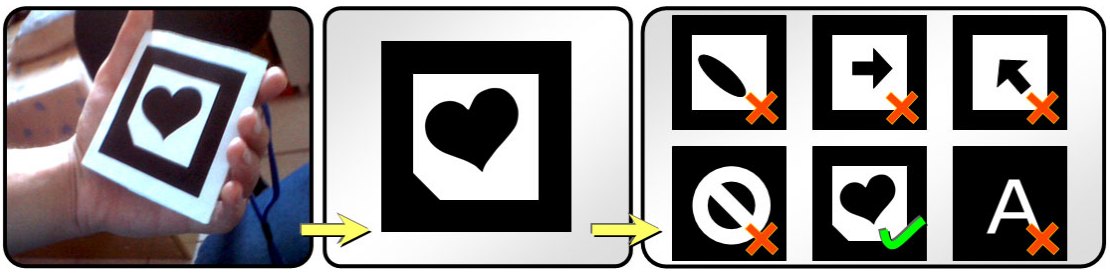

Figura 2.9 – Reconhecimento de Marcadores.

Por conseguinte, se existir similaridade suficiente entre o símbolo capturado e os símbolos pré-cadastrados, o ARToolKit considera que encontrou um dos marcadores de referência. Para que esse processo ocorra de forma rápida o suficiente para acompanhar a taxa de quadros capturada pela câmera, os símbolos pré-cadastrados tem baixas resoluções, tipicamente 16x16pixels. O símbolo a ser reconhecido também pode ser redimensionado no momento da comparação, de forma a facilitar este processo. Assim, as comparações entre os símbolos a serem reconhecidos e a base de dados pode ser feita eficaz. É interessante ressaltar que para cada símbolo a ser reconhecido são necessárias

‘n’ comparações , onde ‘n’ é o número de símbolos armazenados na base de dados. Dessa forma, quanto mais marcas existirem na base de dados maior será o tempo de processamento gasto para as comparações. A quantidade de símbolos na cena capturada pela câmera também influencia o desempenho, pois para cada símbolo são necessárias as ‘n’ comparações mencionadas anteriormente.

Depois de reconhecidas as marcas, o ARToolKit usa o tamanho conhecido do quadrado e a orientação do padrão encontrado para calcular a posição e orientação da câmera virtual em relação a posição e orientação real do marcador. Se as coordenadas da câmera virtual e real estiverem alinhadas, o objeto virtual pode ser sobreposto de forma relativamente precisa sobre o marcador real, acompanhando inclusive as movimentações e mudanças de orientação da câmera real. Assim, é gerada a ilusão visual de presença física do objeto virtual no ambiente real. Dispositivos como HMDs podem ser utilizados para ampliar esta ilusão. A Figura 2.10 exemplifica o alinhamento de um objeto virtual sobre o marcador real.

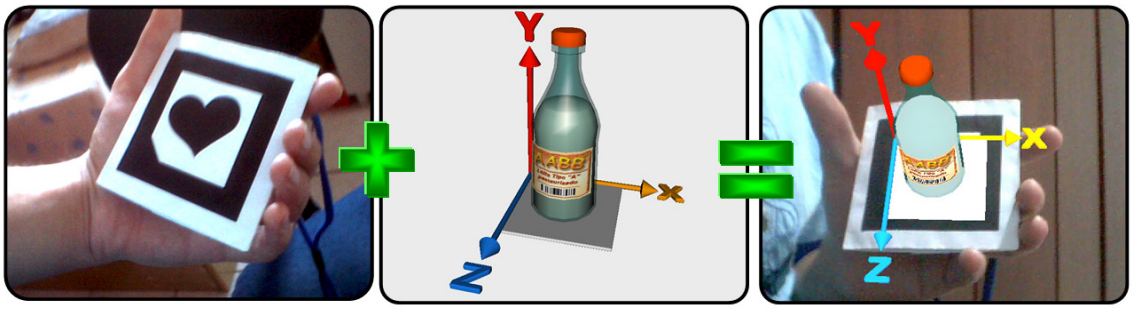

Figura 2.10 – Alinhamento entre marcadores e objetos virtuais.

O software ARToolKit é distribuído de forma aberta, sob a licensa GPU Level2[46], o que permite aos usuários e desenvolvedores terem acesso ao código-fonte do programa, podendo utilizá-lo e modificá-lo para fins não comerciais. Dessa forma, novas características e funcionalidades podem ser introduzidas no software original, dando a este várias maneiras de expansão e flexibilidade na utilização.

O processo de rastreamento através de marcas utilizado pelo ARToolKit ainda está longe de ser o modelo ideal de interatividade proposto pela Realidade Aumentada, no entanto pela sua simplicidade e facilidade na utilização, contribuiu de forma proeminente no processo de popularização da Realidade Aumentada. Com apenas uma webcam e um computador pessoal já é possível utilizar dezenas de aplicações de Realidade Aumentada. Esses e outros motivos foram decisivos na escolha do software ARToolKit como base para a implementação do sistema proposto neste trabalho.

A Figura 2.11 apresenta uma visão geral da estrutura de funcionamento do software ARToolKit.

Figura 2.11 – Estrutura de Funcionamento do software ARToolKit.

2.4. Captura e Tratamento de Sinais Cardiológicos

A captura de sinais cardiológicos, bem como quaisquer outros sinais do corpo humano, é um desafio nada trivial. Para que essa tarefa seja realizada com sucesso, é necessário conhecer com profundidade o funcionamento do órgão ou sistema para o qual os sinais deverão ser capturados.

O coração humano é constituído essencialmente por tecido muscular e nervoso, e sua função primordial é fazer circular o sangue pelo corpo, suprindo assim as demandas do metabolismo celular de todos os órgãos. Essa tarefa é dividida entre os átrios e ventrículos, subdivisões musculares do coração. Os átrios são responsáveis pela recepção do sangue e sua transferência para os ventrículos, e os ventrículos por sua vez, bombeiam efetivamente o sangue para todo o corpo [47]. Após as contrações dos átrios e ventrículos, existe também um período de inatividade muscular, onde acontece a preparação para um novo período de contração. As tarefas se repetem seguidamente, formando um ciclo da atividade cardíaca que perdura durante toda a vida do indivíduo [47,48].

Tanto os átrios quanto os ventrículos, tem seu funcionamento comandado por reações químico-elétricas que acontecem entre as células. E para desencadear estas reações de forma rítmica, existem estruturas de tecido especializadas no coração [48]. Estas estruturas geram um impulso elétrico que se propaga pelo músculo cardíaco, gerando as contrações e conseqüente bombeamento sanguíneo. O impulso também se propaga de forma reduzida para as células nos tecidos próximos ao coração. Assim,

colocando eletrodos na pele do paciente, em regiões opostas do coração, podem ser registrados os potenciais elétricos gerados pela corrente do impulso cardíaco[47]. O registro das variações elétricas do ciclo cardíaco é comumente armazenado em diferentes mídias, tais como papel milimetrado, fitas magnéticas e discos magnéticos e é parte do exame eletrocardiograma (ECG). A Figura 2.12 apresenta uma máquina que realiza exame de ECG e o armazena em papel milimetrado.

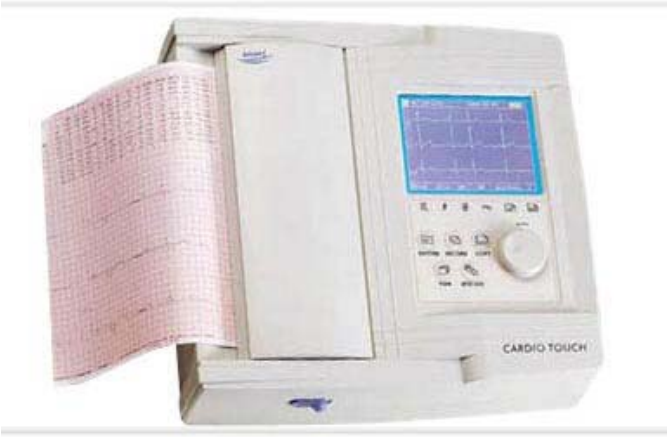

Figura 2.12 – Máquina para Exame de ECG com saída em papel milimetrado [49].

Durante o ciclo cardíaco, acontecem variações de corrente elétrica. Através das curvas percebidas nas variações é possível identificar as fases componentes do ciclo [48]. A Figura 2.13 apresenta um eletrocardiograma com as divisões das fases.

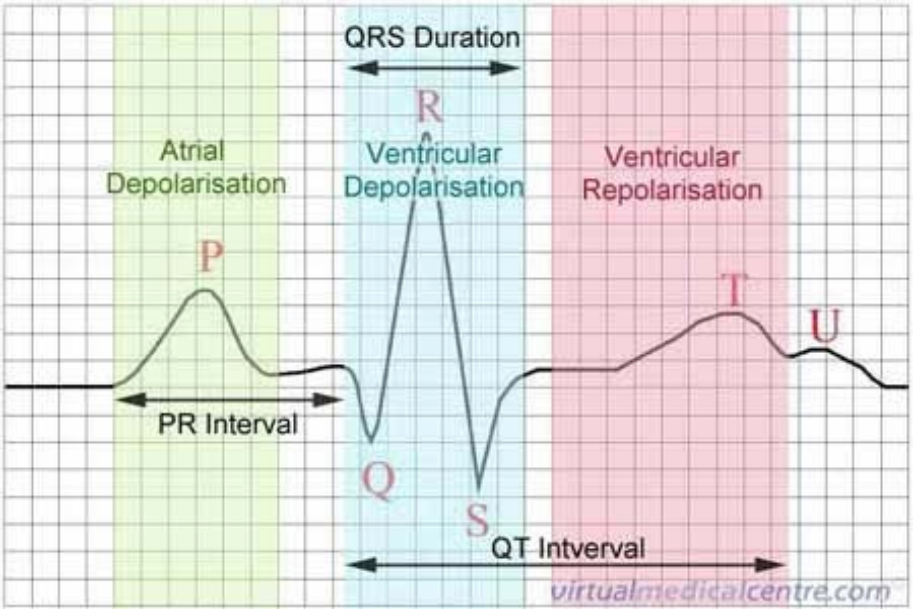

Figura 2.13 – ECG com as divisões das fases do ciclo cardíaco [50].

A despolarização, tanto atrial quanto ventricular (áreas verde e azul), é a fase onde existe a contração muscular no ciclo cardíaco. Após estas fases, acontece a

repolarização, que é um período onde os ventrículos se recuperam da despolarização anterior, e há pouca atividade muscular (área vermelha) [47,48].

Os impulsos elétricos permitem monitorar de forma relativamente precisa as fases do ciclo cardíaco, porém existem maneiras complementares de se monitorar a atividade do coração e outros órgãos, por meios como o som e opacidade sanguínea.

O monitoramento da atividade cardíaca por som é feito tradicionalmente com um estetoscópio, um instrumento acústico capaz de amplificar o som derivado da contração muscular cardíaca [51], e possibilita ao seu utilizador identificar pelos sons ouvidos as atividades do coração e mesmo identificar alguns sintomas de anomalias[52].

Outra forma de monitorar o funcionamento do coração, além de outros dados, é através da opacidade sanguínea. Está forma de monitoramento baseia-se nos coeficientes de absorção da luz vermelha e infravermelha pela hemoglobina rica ou pobre em oxigênio.

Quando a hemoglobina do sangue está rica em oxigênio, ela apresenta como características alta absorção da luz infravermelha e pouca absorção da luz vermelha. O fenômeno inverso ocorre com a falta de oxigênio, fazendo com que haja maior absorção da luz vermelha e pouca absorção da luz infravermelha [53]. Com base nesses fenômenos óticos, é possível construir sensores compostos por LEDs vermelhos e infravermelhos e fotosensor, chamados oxímetros. Os LEDs são posicionados tradicionalmente de forma frontal ao fotosensor, sendo que entre eles fica uma parte translúcida do corpo, como o lóbulo da orelha ou a ponta do dedo polegar. A cada pulsação cardíaca existe um momento onde se altera a oxigenação do sangue na área de medição, devido ao recolhimento do sangue venoso, pobre em oxigênio e sua conseqüente substituição pelo sangue arterial, rico em oxigênio. Essas alterações são monitoradas pelo sensor, possibilitando a medição da freqüência cardíaca e também do nível de oxigenação sanguínea. A Figura 2.14 apresenta o esquema de posicionamento dos LEDs e fotosensor em um oxímetro.

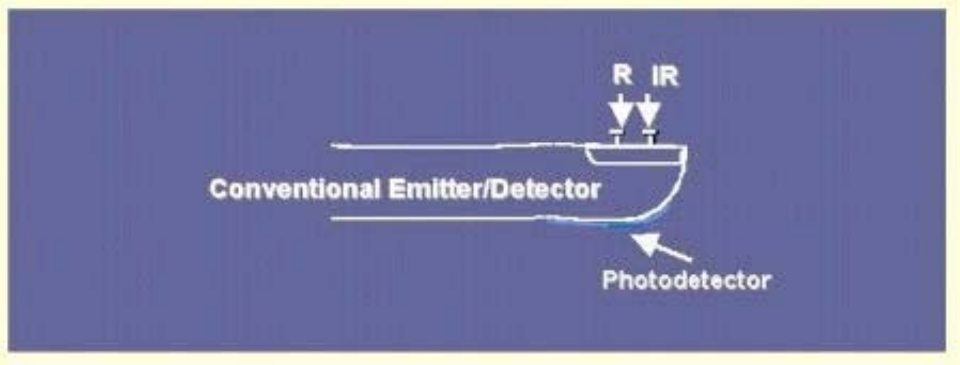

Figura 2.14 – Posicionamento dos LEDs e Fotosensor em um oxímetro[84].

O sensor utilizado no sistema proposto por esta dissertação utiliza princípios de oximetria para seu funcionamento, porém não apresenta o LED vermelho, pois não faz a medição da oxigenação sanguínea, somente mede a freqüência cardíaca.

Além das formas mencionadas, existem diversas outras técnicas para monitoramento e diagnóstico de doenças cardíacas, porém fogem ao escopo desta dissertação, onde foram tratados apenas sinais eletrocardiográficos e provindos de oximetria.

2.5. Conclusões

A pesquisa dos conteúdos teóricos relacionados ao tema da pesquisa foi extremamente válida e absolutamente necessária na definição das metas do sistema a ser desenvolvido, possibilitando uma visão mais ampla das áreas abrangidas e auxiliando na escolha das técnicas necessárias para alcançar os resultados desejados.

Capítulo 3

3. TRABALHOS RELACIONADOS

3.1. Introdução

Esta seção apresenta os trabalhos relacionados com o tema desta dissertação, identificados durante o processo de pesquisa da mesma.

A análise de trabalhos relacionados ao tema de interesse pode ser muito promissora, de várias maneiras:

- ✓ Permite que a visão e conhecimento do pesquisador sobre a área de pesquisa sejam ampliados;
- ✓ Auxilia o pesquisador na definição de metas a serem cumpridas para sua pesquisa;
- ✓ Possibilita ao pesquisador conhecer iniciativas semelhantes pré-existentes e, através destas, concentrar seus esforços em problemas ainda não solucionados ou aperfeiçoar soluções existentes.

Ao final deste capítulo será apresentado um quadro contendo as características e funcionalidades de todos os sistemas previamente apresentados.

3.2. Project on Image Guided Surgery

Este projeto é parte dos resultados conseguidos através de uma parceria que já ocorre há alguns anos entre o MIT (Massachusetts Institute of Technology) [54] e o BWH(Brigham & Women's hospital) [55].

O projeto consiste em utilizar técnicas de Computação Gráfica e Realidade Aumentada como ferramenta auxiliar em intervenções cirúrgicas. Os elementos das simulações são concebidos em uma complexa fase de pré-processamento e sua utilização permite aos médicos visualizar no momento da cirurgia, não somente os órgãos a serem operados mas também os auxílios virtuais previamente criados em perfeito alinhamento com a fisiologia do paciente, facilitando assim a tomada de decisões e diminuindo as chances de fracasso ou seqüelas.

O sistema utiliza as características do próprio paciente na fase de preparação, o que gera modelos e simulações muito mais precisas e personalizadas, pois a cada paciente o processo todo é repetido.

A fase inicial do projeto consiste em reconstruir tridimensionalmente as estruturas anatômicas do paciente onde será realizada a cirurgia. Para conseguir modelos tridimensionais precisos o suficiente para as operações são utilizadas várias imagens de ressonância magnética, sendo que cada imagem corresponde a uma espécie de corte no local de interesse, sendo que a cada corte há um reposicionamento dos sensores em prol de capturar outro corte. A Figura 3.1 mostra um exemplo de ressonância magnética de uma caixa craniana.

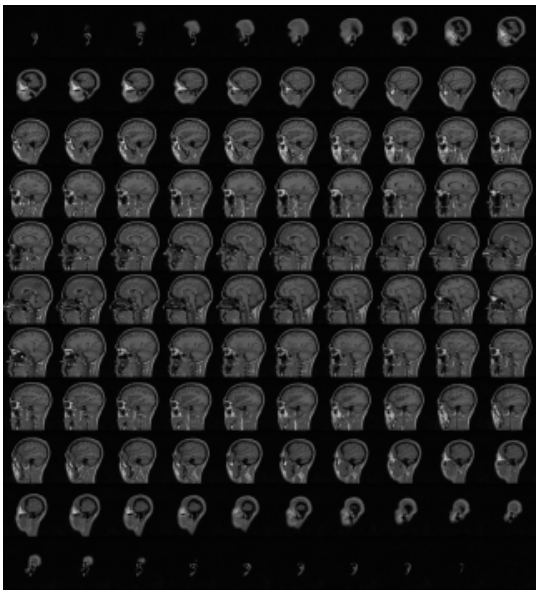

Figura 3.1 – Exemplo de Ressonância Magnética [10].

Essas imagens são processadas e separadas as partes de interesse. Após isso um software utiliza os diversos cortes para criar objetos tridimensionais correspondentes. Esses modelos podem receber atributos como cores e texturas, e alternar sua opacidade de forma a permitir exames de partes mais internas. A Figura 3.2 ilustra exemplos de modelos tridimensionais gerados a partir de imagens de ressonância magnética.

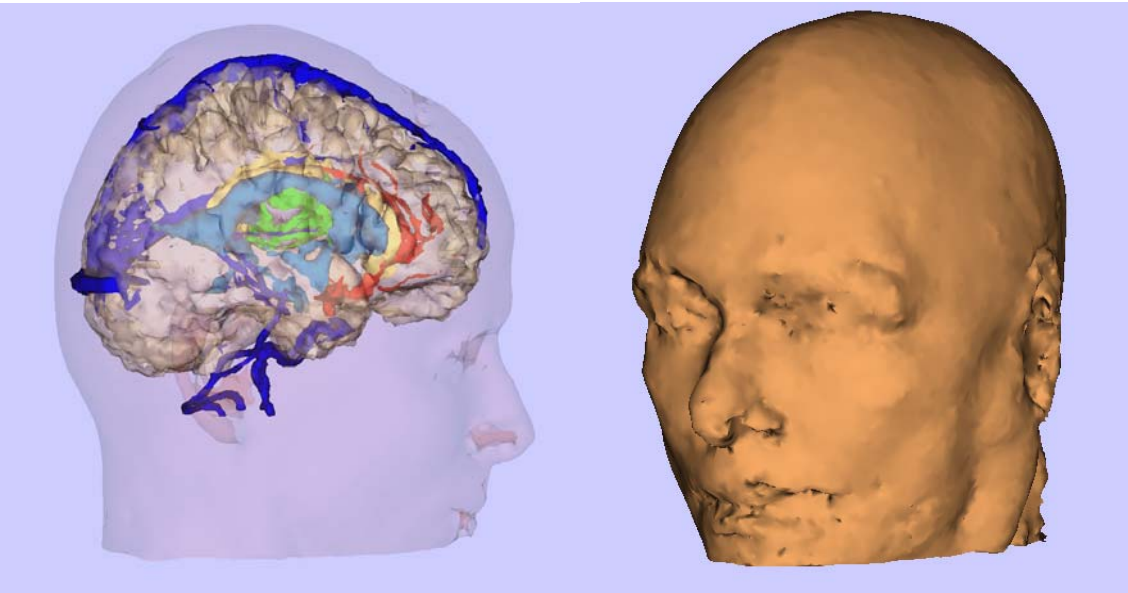

Figura 3.2 – modelos Tridimensionais gerados a partir de imagens de ressonância magnética [10].

A fase de pré-processamento termina logo após os modelos tridimensionais estarem apropriadamente editados. Em seguida esses modelos são utilizados já no momento da cirurgia.

Com os modelos já prontos, faz-se necessária alguma forma de sincronizar o posicionamento, escala e orientação dos objetos virtuais em relação aos modelos reais, de modo que se possa utilizar os modelos virtuais como base segura em ações de risco, como por exemplo a retirada de um tumor no cérebro.

Para tanto, é preciso fazer o rastreamento das coordenadas do paciente correspondentes aos modelos tridimensionais criados. Isso é conseguido utilizando scanners tridimensionais baseados em raios laser e câmeras de vídeo. A precisão conseguida por esta técnica é bastante alta, sendo o erro inferior a 1 mm. Além disso, por utilizar raios laser , é evitado o contato físico direto utilizado em outras técnicas de rastreamento, o que é benéfico pois em condições de cirurgia não seria possível na maioria dos casos utilizar técnicas de contato direto, devido às condições críticas das operações.

A Figura 3.3 apresenta os equipamentos utilizados no sistema e a Figura 3.4 mostra o alinhamento de alguns modelos tridimensionais e pacientes.

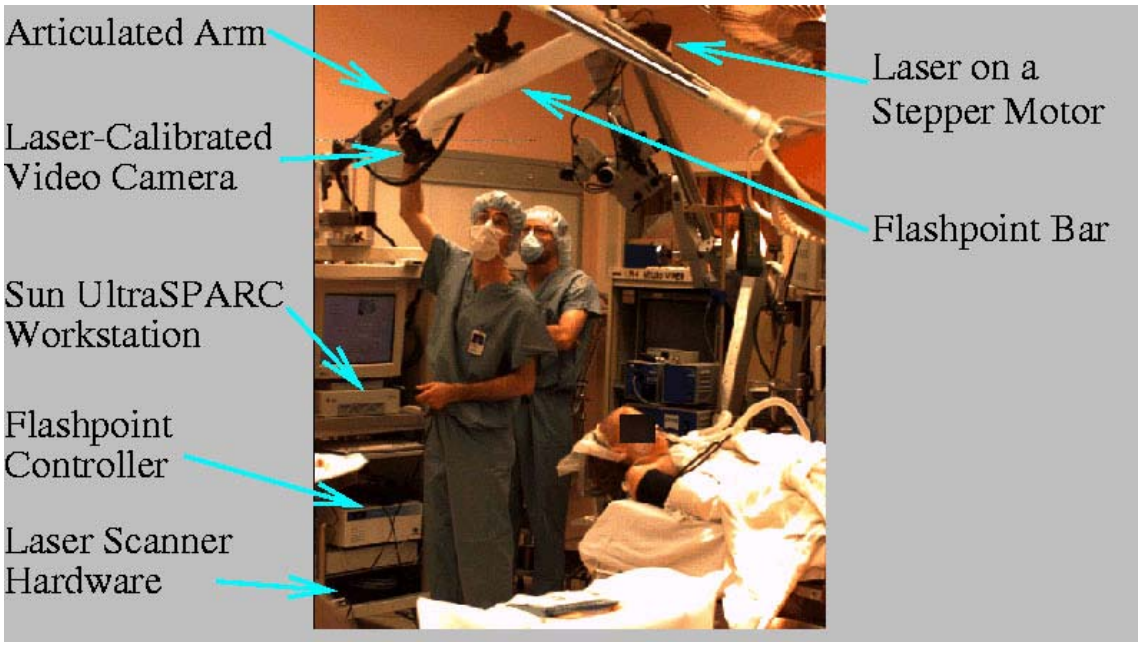

Figura 3.3 – Equipamento para Rastreamento dos órgãos do paciente [10].

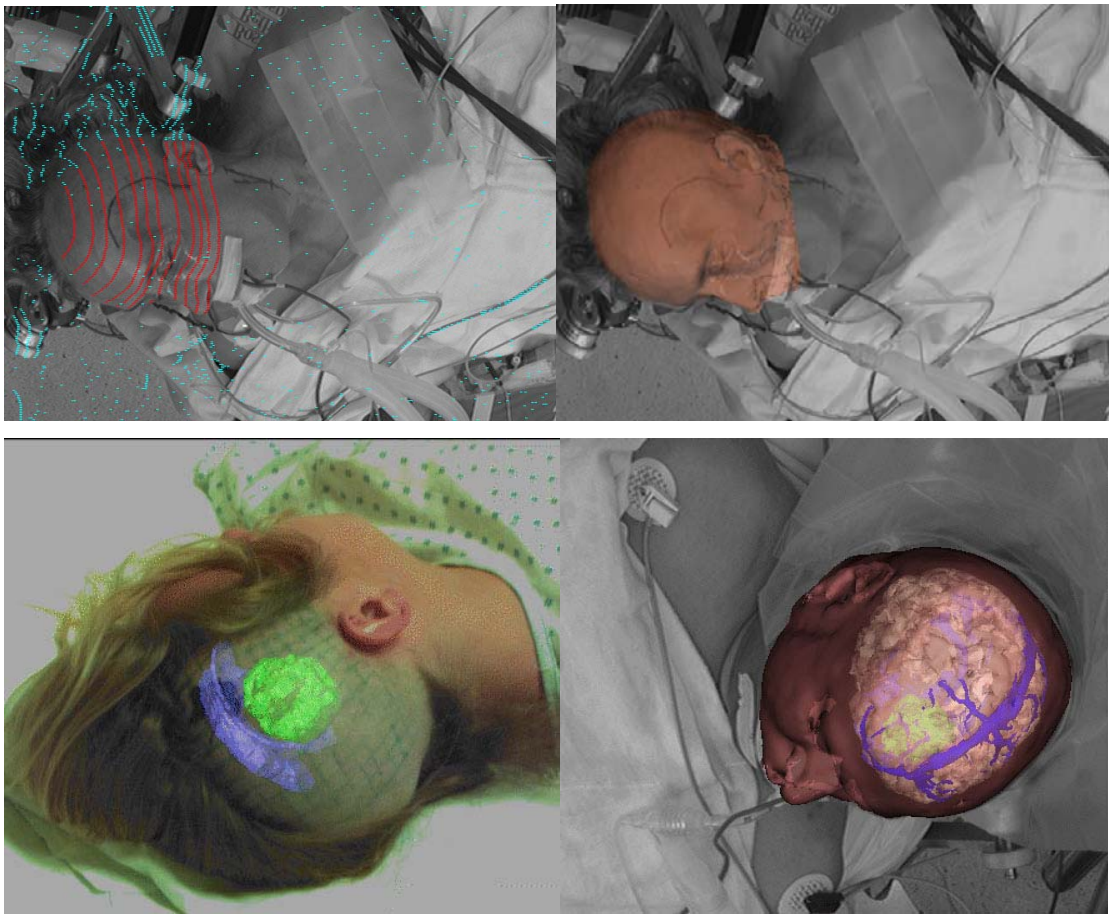

Figura 3.4 – Alinhamento entre pacientes e modelos tridimensionais [10].

Como pode ser visto na Figura 3.4, os modelos virtuais podem ser alinhados com boa precisão aos pacientes, resultando em uma sobreposição bastante confiável.

Para gerar a visualização final, são utilizados monitores comuns de vídeo, sendo que a imagem filmada do paciente fica ao fundo, sobreposta pelos objetos virtuais. É como se o médico tivesse uma visão de Raios-X, podendo observar os órgãos do paciente ainda antes de realizar qualquer corte sobre a pele. Isso permite cirurgias menos invasivas e com menos imprecisão, resultando também em recuperações mais rápidas e bem sucedidas.

O sistema proposto já foi bastante utilizado em intervenções cirúrgicas, com resultados surpreendentes, reduzindo drasticamente os custos de cada intervenção cirúrgica, bem como o tempo de preparo e operação do cirurgião. No entanto, a fase que antecede a simulação é bem extensa e envolve equipamentos de alto custo, inviabilizando a implantação do projeto em larga escala. Além disso, os órgãos reconstruídos com as técnicas apresentadas são a princípio rígidos, não sofrendo quaisquer deformações ao longo do tempo, o que dificulta a utilização do sistema com órgãos que tenham movimentação, como músculos em geral e o coração.

3.3. VIDA: Visualizador Interactivo de Dados biomecânicos Animados

O sistema VIDA foi proposto como uma aplicação para visualizar dados biomecânicos de atletas de uma forma simples e intuitiva, como por exemplo a atividade muscular[18].

O sistema pode fornecer dois modos distintos de funcionalidade. No primeiro deles, o usuário manipula um boneco virtual onde pode visualizar os dados. Já no segundo modo, vídeos com cenas pré-capturadas são utilizados na construção e visualização das simulações.

Os principais objetivos do VIDA são:

- ✓ Construir um modelo tridimensional do atleta que possibilite a interação do biomecânico após a captação de dados. A interação nesse caso é entendida como a aplicação de uma das 3 transformações fundamentais, translação, rotação e escala.
- ✓ Sobrepor aos vídeos capturados com cenas dos atletas o componente muscular virtual gerado.

Para alcançar com sucesso os objetivos propostos, uma série de etapas de processamento foram necessárias. A Figura 3.5 mostra a arquitetura do sistema VIDA.

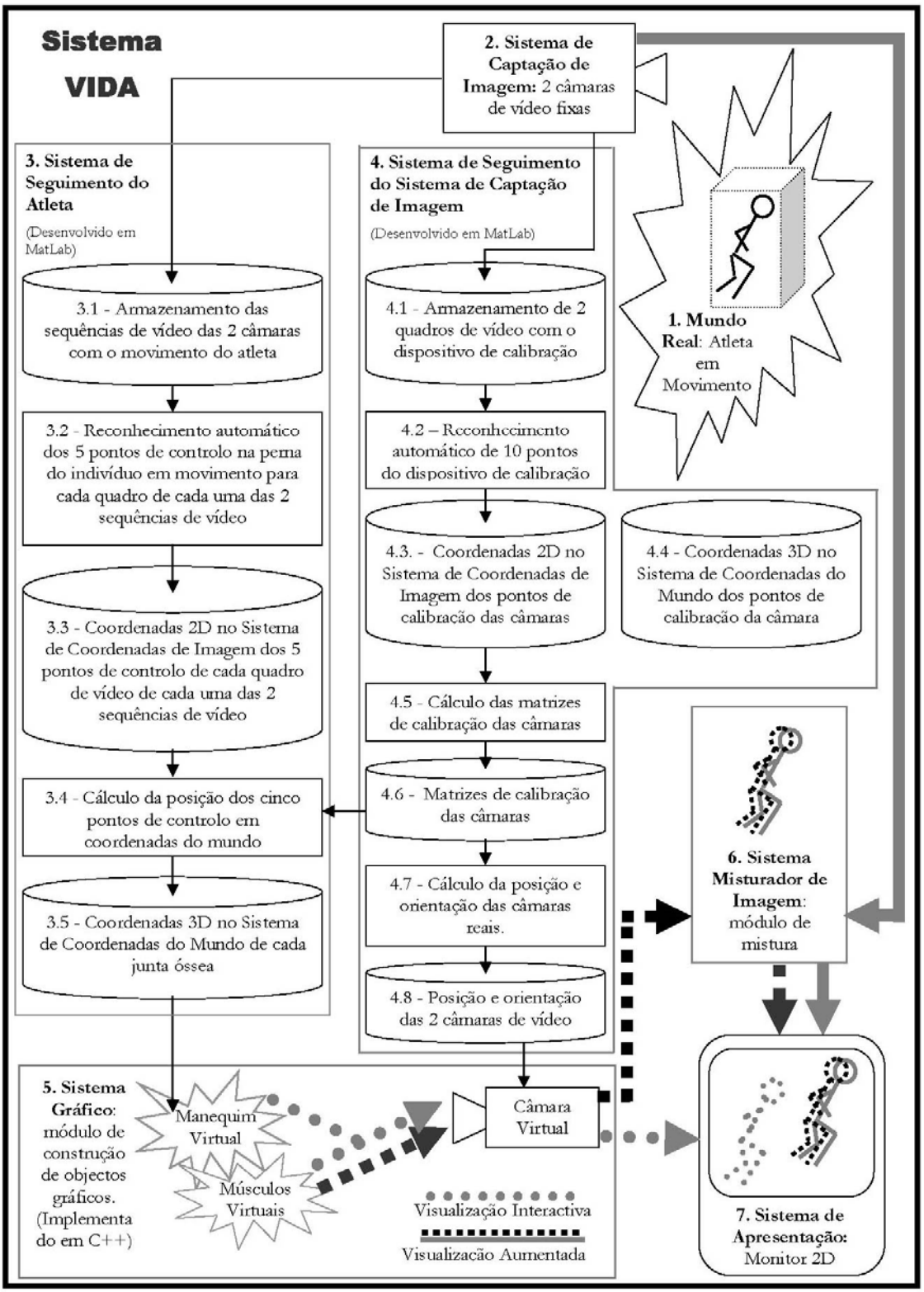

Figura 3.5 – Arquitetura do projeto VIDA [18].

O projeto VIDA não trabalha com dados em tempo real, somente com vídeos pré-capturados com ações de atletas, o que é ressaltado na própria pesquisa como algo a

ser mudado para que a experiência de Realidade Aumentada seja realmente contemplada pelo projeto.

A fase inicial neste caso é a captura de informações visuais do atleta. Como a meta desta fase é obter dados tridimensionais do atleta, algum método de rastreamento se faz necessário. Ao invés de utilizar dispositivos de complexidade elevada como scanners baseados em lasers, o sistema VIDA se vale de um outro recurso bastante interessante, que é a utilização de marcas rastreáveis e câmeras de vídeo estrategicamente posicionadas.

As marcas podem ser construídas para diversos tipos de rastreamento, como cor, formato, luminosidade, entre outras. O VIDA utiliza marcas iridescentes em ambiente de laboratório controlado. Essas marcas tem brilho em baixas condições de iluminação, de forma semelhante ao revestimento de interruptores domésticos, o que facilita o seu isolamento do resto do cenário através da utilização de algoritmos de processamento de imagem. Assim, é possível rastrear a cada quadro filmado a posição das marcas no cenário.

No entanto, o problema das coordenadas tridimensionais ainda persiste, pois as câmeras de vídeo operam de forma essencialmente bidimensional. Para conseguir a terceira coordenada ou profundidade, existem diversos métodos. O método escolhido pelo sistema do VIDA é a combinação de 2 ou mais planos bidimensionais perpendiculares para a aquisição da terceira dimensão. Esses planos são as aquisições de imagem das câmeras, sendo que para cada plano bidimensional uma câmera é necessária. Combinando as imagens de todas as câmeras o sistema é capaz de determinar as coordenadas tridimensionais de cada marca iridescente.

Depois de capturar as coordenadas das marcas nos vídeos, o VIDA associa cada marca a uma marca virtual responsável pelos movimentos de um boneco tridimensional modelado, de forma a sincronizá-lo visualmente com os movimentos realizados pelos atletas no vídeo. Uma câmera virtual com características similares à câmera real que capturou o vídeo de fundo também é necessária para a sincronização ser perfeita. É importante ressaltar que tanto as marcas virtuais quanto as reais são posicionadas nas juntas de movimentação dos atletas e bonecos, como por exemplo, joelho, ombros e cotovelos. Além disso, o sistema conta também com uma forma de visualização do esforço muscular através de linhas com cores estrategicamente estabelecidas, possibilitando verificar pontos de alto esforço em exercícios físicos. A Figura 3.6

demonstra a sincronização dos movimentos do atleta com o boneco virtual e a Figura 3.7 mostra o sistema de visualização de esforço muscular.

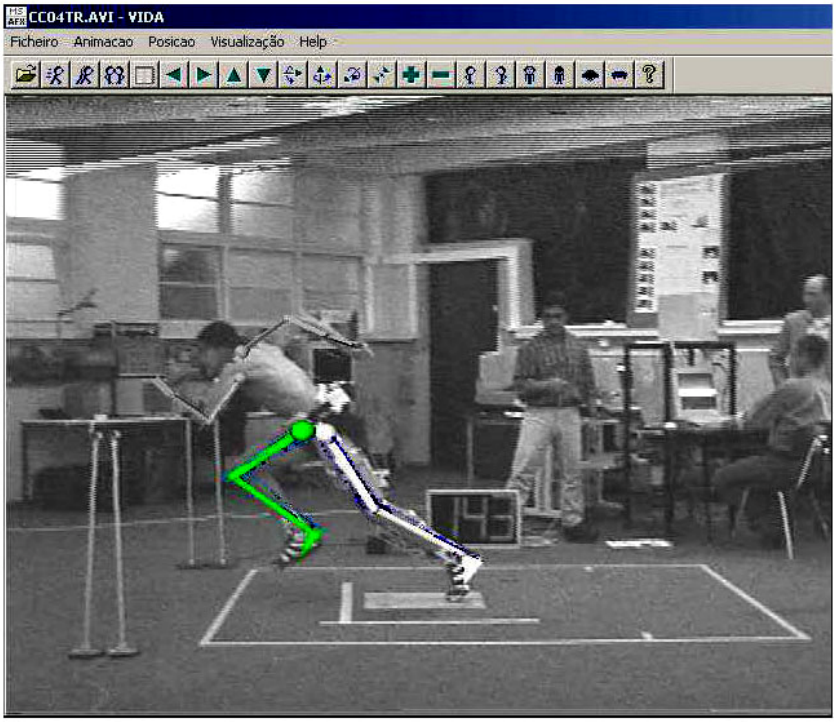

Figura 3.6 – Sincronização do vídeo com boneco virtual [18].

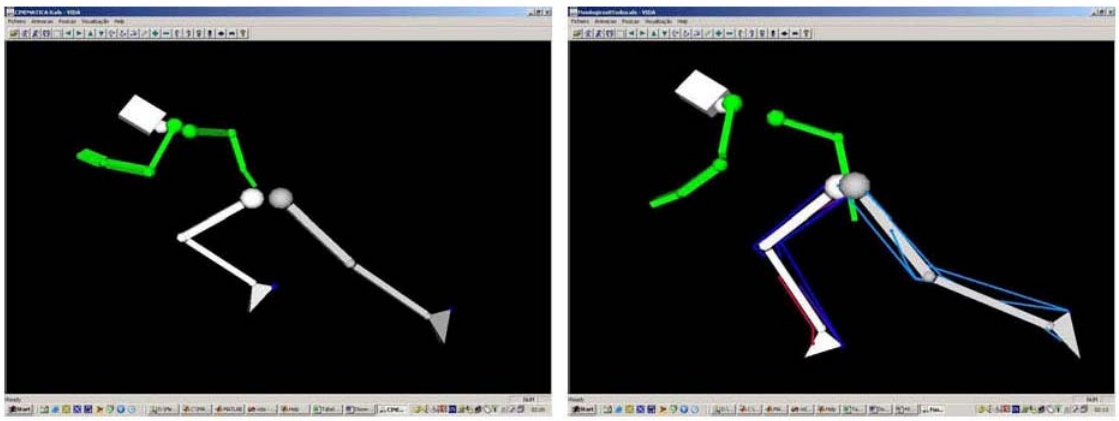

Figura 3.7 – Visualização de esforço muscular [18].

As melhorias previstas para o VIDA são:

- ✓ Permitir o tratamento e visualização de informações em tempo real, ao invés de somente utilizar sequências de vídeo pré-capturadas;
- ✓ Incorporar visualização através de um HMD do tipo see-through;
- ✓ Substituir os modelos virtuais primitivos de ossos por modelos realistas;
- ✓ Substituir os modelos lineares de esforço muscular por modelos realistas;
- ✓ Calcular as forças de reação nas juntas e mostrá-las visualmente.

3.4. Die-Heart

O projeto Die-Heart tem como objetivo principal, visualizar em ambiente de Realidade Aumentada um modelo deformável do ventrículo esquerdo de um coração humano. Essa visualização permitiria ao médico interagir com o modelo deformável de maneira mais natural, utilizando para tanto um sistema de baixo custo [19].

A motivação para o projeto reside no fato de a Realidade Aumentada ser uma área de pesquisa relativamente recente, cujas aplicações ainda estão sendo definidas, sendo que a Medicina é um dos campos mais promissores. Além disso, pesquisas anteriores desenvolvidas pelo grupo puderam ser aproveitadas nas bases do projeto.

Os dispositivos utilizados pelo sistema do Die-Heart são basicamente um computador comum, com monitor e utilizando um mouse 3D [56], dispositivo que permite maior liberdade de interação em ambientes tridimensionais. A parte do software responsável pela Realidade Aumentada foi desenvolvida com base no software MxToolKit [57]. A Figura 3.8 mostra a Arquitetura do sistema.

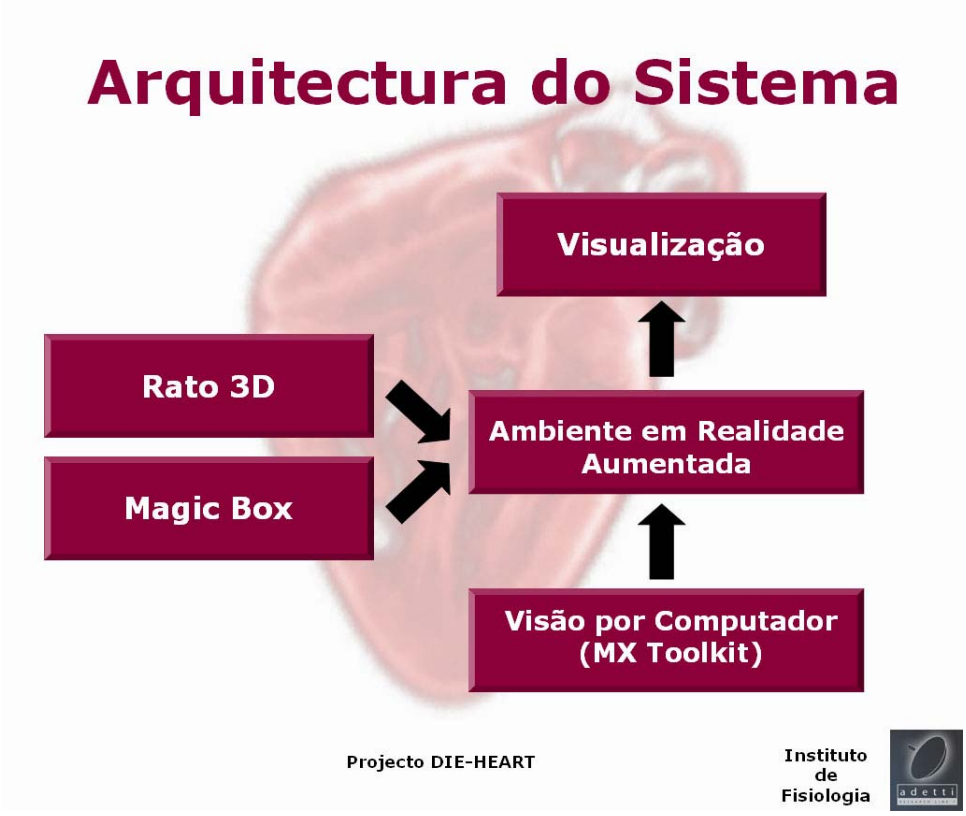

Figura 3.8 – Arquitetura do sistema Die-Heart [19].

O sistema funciona capturando em tempo real uma cena através de uma câmera. Os quadros de vídeo são passados para o software mxToolKit que reconhece marcas monocromáticas em alto contraste e exibe objetos tridimensionais apropriadamente

sobre essas marcas, de acordo com o tamanho, orientação e posição reais, fazendo parecer que os objetos estão realmente no cenário. A manipulação dos objetos virtuais no sistema é feita de duas formas, através das próprias marcas físicas e através do mouse 3D, que permite ao usuário 6 graus de liberdade na utilização.

No caso do sistema Die-Heart, o objeto virtual de interesse é somente o ventrículo esquerdo de um coração humano. Uma das exigências propostas é que o modelo fosse deformável e tridimensional. Para reconstrução com bastante fidelidade do ventrículo, foram realizadas algumas tarefas.

Inicialmente, são capturadas imagens de ecocardiograma, de onde se podem extrair informações visuais referentes à geometria do coração. Após isso essas imagens passam por um processo de segmentação. Como o coração tem diversas particularidades, a técnica de segmentação escolhida é crucial para uma boa separação das partes desejadas do resto da imagem. O primeiro passo é a escolha da área de interesse, no caso, o ventrículo esquerdo do coração, através da delimitação de uma janela de seleção. Isto evita erros induzidos pela segmentação de uma área maior na imagem. Após isso, a área de interesse é passada por um filtro de suavização, o que reduz bastante os ruídos presentes na imagem. A imagem resultante é passada então por um filtro de magnitude de gradiente, que ressalta o contorno da parede interna do ventrículo. Um outro filtro é aplicado depois para intensificar ainda mais os contornos na imagem. Com base nisso a imagem é binarizada e os contornos podem ser redesenhados por algoritmos especializados. O resultado final é um conjunto de linhas descritas geometricamente, independentes da imagem original. A Figura 3.9 ilustra um ecocardiograma que serviu de base para os cálculos e algumas imagens intermediárias geradas no processo de segmentação.

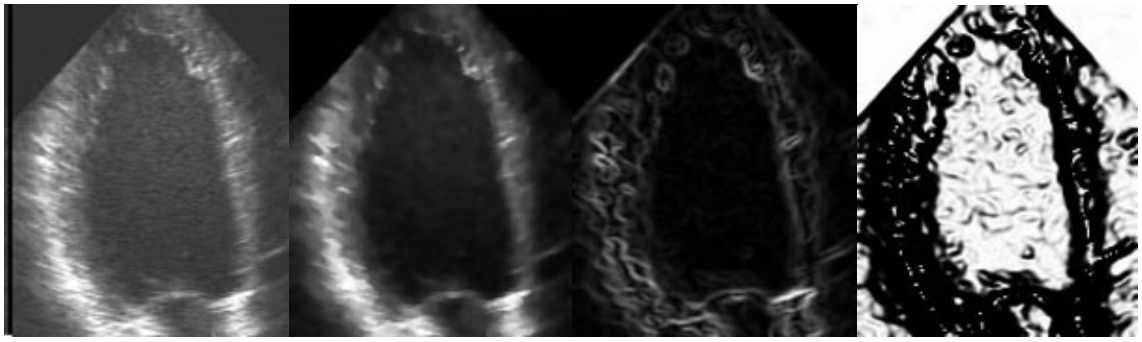

Figura 3.9 – Ecocardiograma e imagens do processo de segmentação [19].

Após a segmentação obtém-se uma série de contornos representando a geometria do ventrículo, porém para que possa ser feita uma reconstrução tridimensional

apropriada é necessário unificar os contornos em uma única linha. Isso é feito no Die-Heart através de algoritmos de média aplicados sobre os contornos de forma inteligente, de forma a não induzir muito erro no resultado. A Figura 3.10 mostra os contornos antes e depois do aperfeiçoamento na fase de sístole e diástole cardíaca.

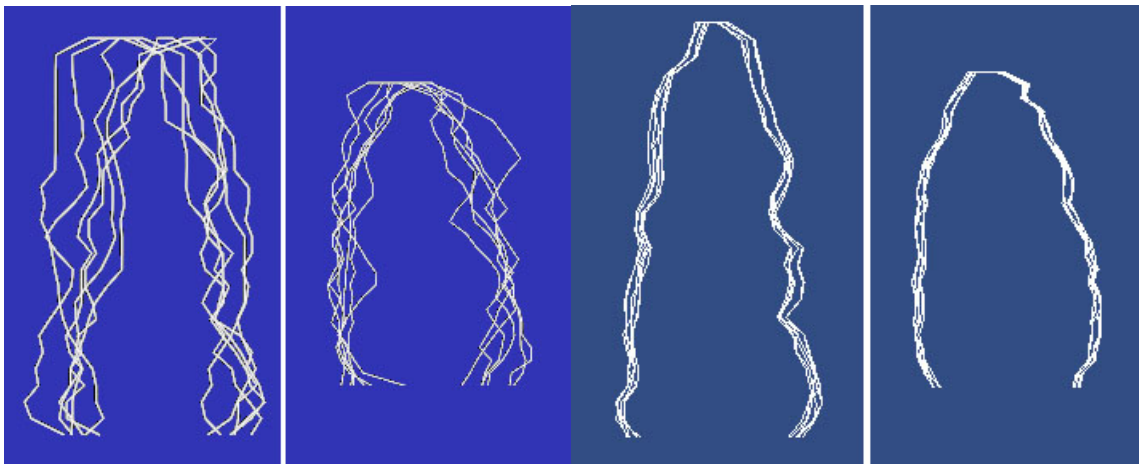

Figura 3.10 – Aperfeiçoamento dos Contornos [19].

Para que os contornos sejam transformados em geometria tridimensional, o Die-Heart utiliza diversas imagens de ecocardiograma capturadas a partir de diferentes ângulos do coração, girando de forma concêntrica a ele. A captura e segmentação dessas imagens rotacionadas gera perfis correspondendo à geometria do coração, que podem posteriormente ser ligados e criada a geometria em si. No sistema atual, pelo menos 6 ângulos diferentes são capturados para recriar a geometria do coração. Um detalhe importante a se ressaltar é que os contornos têm suas quantidades de pontos normalizadas para um número fixo de pontos, com distâncias equivalentes. Isso é algo interessante para a fase de criação da geometria, pois cada ponto num contorno pode ser mais facilmente associado a um ponto do outro contorno. A Figura 3.11 ilustra os contornos gerados a partir dos 6 ângulos capturados e também a conexão de todos os ângulos formando a base da geometria tridimensional a ser reconstruída.

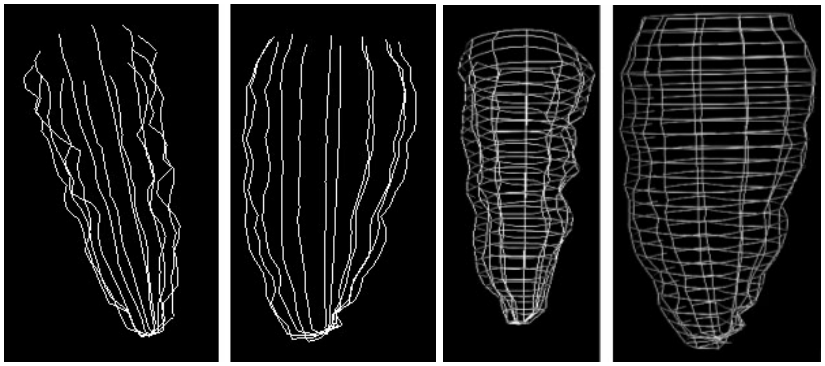

Figura 3.11 – Contornos nos 6 ângulos e criação da geometria tridimensional [19].

Depois da captura e pré-processamento dos dados que servirão para compor a geometria tridimensional do objeto, a próxima fase concentra esforços em reconstruir a geometria tridimensional do objeto propriamente dita. São reconstruídos modelos das diversas fases do ciclo cardíaco presentes entre a sístole e a diástole. Esses modelos servem como pontos-chave para a animação de um ciclo cardíaco. Ao ser dado um valor de batidas por minuto, a animação do ciclo cardíaco começa no primeiro modelo da fase sistólica. De acordo com o tempo de duração de um ciclo cardíaco, o coração virtual vai sendo animado pela interpolação de pontos entre os diversos modelos, da sístole para a diástole e depois é feita uma animação em sentido contrário, de forma a voltar para a sístole o coração virtual, fechando assim o ciclo. Dessa forma, o modelo do ventrículo esquerdo do coração é animado de forma dinâmica, sendo deformado a cada instante para demonstrar o ciclo cardíaco completo. A Figura 3.12 mostra os diversos modelos-chave utilizados na animação do ciclo cardíaco.

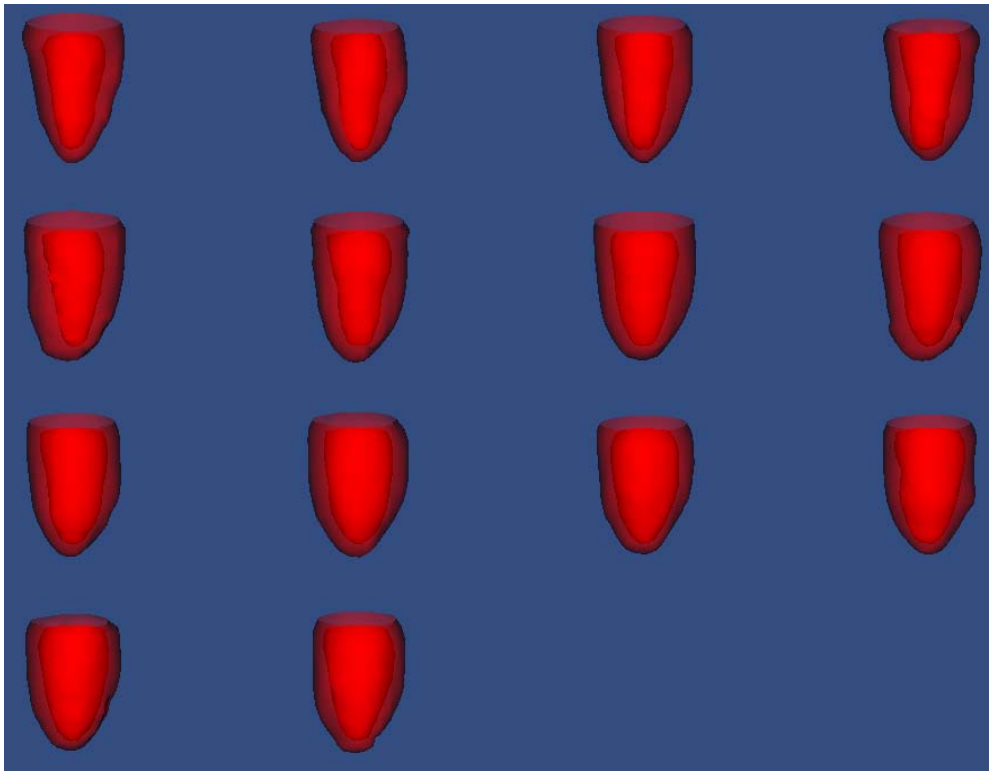

Figura 3.12 – Diversos modelos-chave do ciclo cardíaco [19].

A animação pronta do ventrículo do coração humano pode ser visualizada inicialmente em ambiente de Realidade Virtual, porém o objetivo do Die-Heart é fornecer a visualização da animação em Realidade Aumentada. Por isso foi utilizado o mxToolKit para recriar a visualização em Realidade Aumentada. o MXToolKit utiliza marcas monocromáticas com símbolos em seu interior para realizar o rastreamento e a

sobreposição dos objetos virtuais sobre o ambiente real, de maneira bastante similar ao ARToolKit, detalhado no capítulo 2. A Figura 3.13 mostra algumas visualizações proporcionadas pelo Die-Heart.

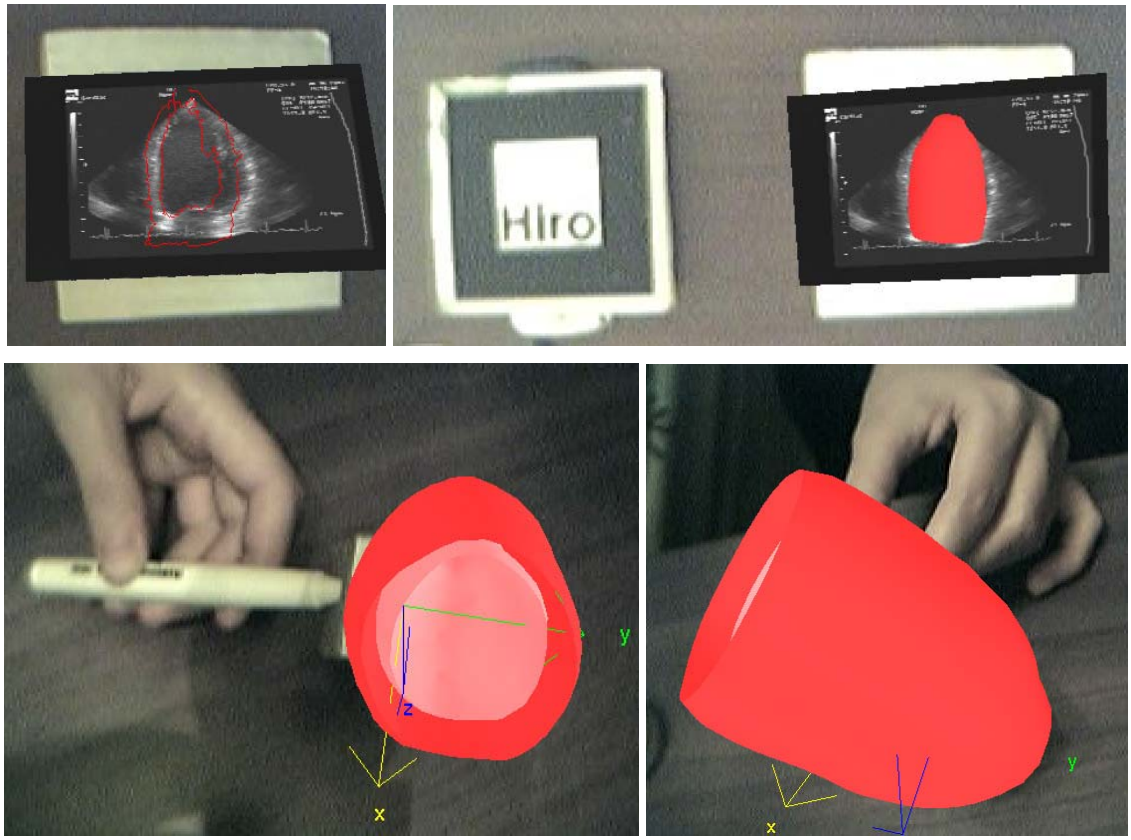

Figura 3.13 – Visualizações do Die-Heart em Realidade Aumentada [19].

O sistema Die-Heart cumpre bem a tarefa a qual foi proposto, mostrando a animação deformável de um ventrículo cardíaco. No entanto a fase de que precede a simulação é extensa, envolvendo um processo bastante complexo de captação dos momentos exatos do ciclo cardíaco desejado, bem como a conversão de todas as imagens captadas para modelos tridimensionais. Imagens com baixa qualidade na captura e processamento, podem distorcer os resultados, deformando o modelo final do ventrículo e causando equívocos na interpretação dos resultados.

3.5. Mitral Valve Surgery Using Pre-Operative Cardiac Models

Este sistema foi desenvolvido com o objetivo de auxiliar o planejamento e a execução de cirurgias cardíacas na válvula mitral, fornecendo um ambiente em Realidade Aumentada com bastante precisão no alinhamento dos objetos reais com os virtuais [20].

As cirurgias cardíacas sempre foram desafiadoras para cirurgiões, desde que foram realizadas as tentativas de intervenção no coração. Uma das maiores causas dos problemas nas cirurgias desse tipo é a dificuldade na visualização das estruturas internas do coração. Essas dificuldades fizeram com que o padrão de cirurgias cardíacas se estabelecesse através do desvio cardiopulmonar, que permite ao cirurgião ter uma visão direta do coração do paciente, melhorando assim a eficácia dos procedimentos. O grande problema com esta técnica é que ela é muito invasiva, acarretando ao paciente longos períodos de recuperação e por vezes seqüelas. Então, para se encontrar um ponto de equilíbrio entre a eficácia dos procedimentos e os períodos de recuperação dos pacientes, técnicas menos invasivas e ao mesmo tempo mais eficazes são desenvolvidas a cada dia.

O procedimento desenvolvido ao longo deste trabalho implementa um sistema que fornece acesso seguro dentro do coração em funcionamento e uma visualização boa o suficiente para assegurar bons resultados nas intervenções.

Para a visualização em tempo real do coração, é utilizada a ecocardiografia transesofágica (ETE) [58,59]. Também é utilizado um sistema de rastreamento magnético para capturar as posições de todos os aparelhos cirúrgicos utilizados no procedimento. Ao final do processo, um ambiente de Realidade Virtual amplia as imagens da ETE com visualizações de imagens pré-operativas e dos instrumentos cirúrgicos rastreados magneticamente. Dessa forma, as informações providas pela ETE podem ser interpretadas como se estivessem em seu contexto tridimensional, o que melhora a navegação durante o procedimento. A Figura 3.14 mostra um diagrama com a estrutura de funcionamento do sistema.

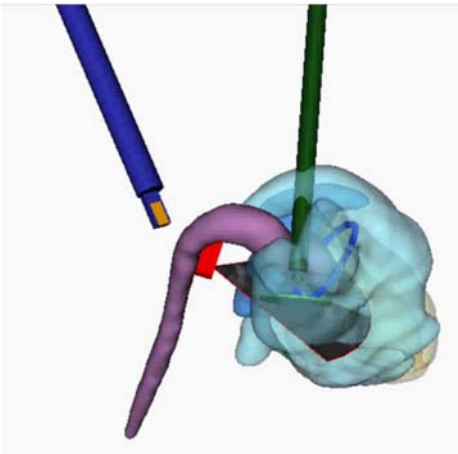

Figura 3.14 – Diagrama com a Estrutura do sistema [20].

Para guiar o reparo na válvula mitral são necessárias informações detalhadas sobre o comportamento dinâmico dos anéis da válvula mitral também da anatomia ao seu redor. Para satisfazer essas restrições foram desenvolvidos modelos tridimensionais pré-operativos com base em imagens capturadas por técnicas de ressonância magnética. Esses modelos representam de forma aproximada a anatomia cardíaca e podem prever com uma precisão de aproximadamente 3 milímetros a localização do alvo da cirurgia, ou seja, a válvula mitral.

O foco deste trabalho é integrar os modelos pré-operativos desenvolvidos ao ambiente de intra-operativo, compensando as discrepâncias de anatomia encontradas e mensurando de alguma forma a precisão de posicionamento alcançada. As tarefas necessárias para alcançar os objetivos propostos foram divididas em diversas fases seqüenciais.

A primeira etapa foi a criação dos modelos tridimensionais pré-operativos. Utilizou-se imagens de ressonância magnética de alta resolução como base da reconstrução tridimensional. As imagens foram adquiridas de um paciente saudável em forma de cortes horizontais, espaçados em 6 milímetros. Os cortes capturados passam então por um processamento de imagens e reconstrução tridimensional, o que resultou nos modelos das estruturas cardíacas. A Figura 3.15 apresenta a imagem de um corte com os modelos tridimensionais já reconstruídos ao lado.

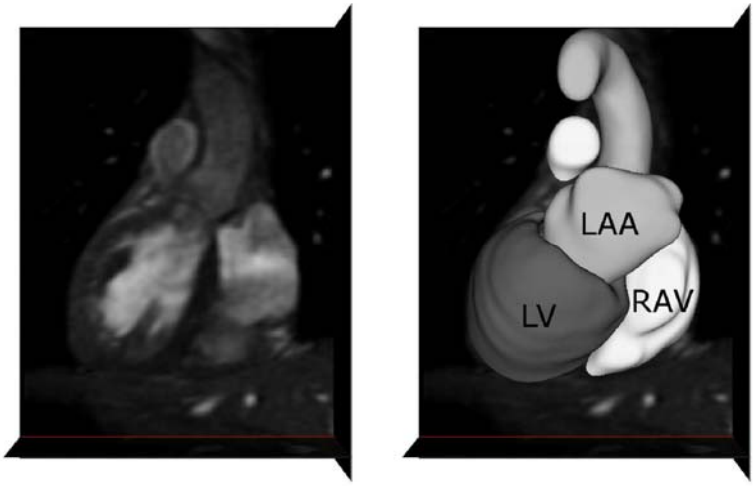

Figura 3.15 – Corte de ressonância Magnética e Modelos tridimensionais [20].

Os modelos tridimensionais reconstruídos através deste processo funcionam como modelos-base que se adaptam a cada paciente, através da média de imagens de ressonância magnética em momentos específicos do ciclo cardíaco. Dessa forma, os

modelos podem ser facilmente adaptados para corresponder mais precisamente às diversas anatomias a serem representadas.

Na próxima fase, os modelos tridimensionais reconstruídos e adaptados são alinhados com o ambiente real de operação. Para isso são capturadas imagens de ultra-som 3D que auxiliam o correto posicionamento dos elementos virtuais. Algumas fases do alinhamento e identificação são feitas manualmente, com a ajuda de um cardiologista experiente. Como resultado é obtido um ambiente onde a visualização intra-operativa é enriquecida pelos modelos virtuais das estruturas anatômicas desejadas. A Figura 3.16 mostra 2 exemplos de visualizações obtidas com o sistema proposto.

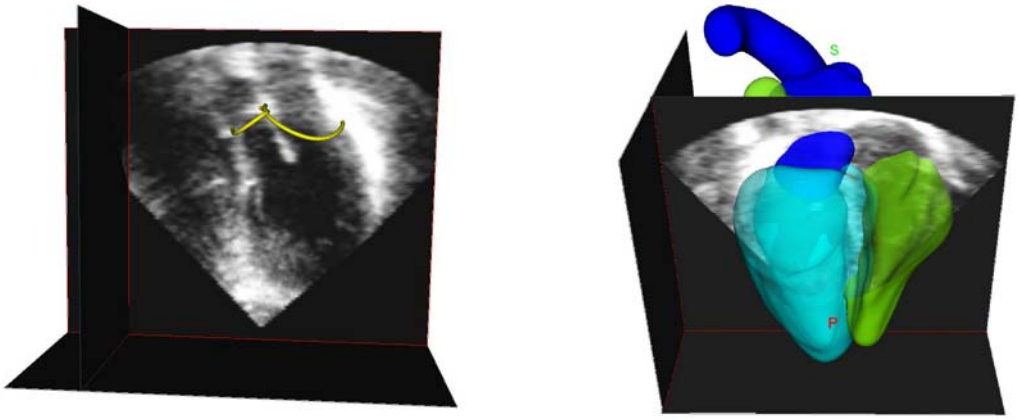

Figura 3.16 – Exemplos de Visualizações obtidas com o sistema proposto [20].

Com o auxílio dos modelos tridimensionais reconstruídos a partir dos cortes de ressonância magnética, o sistema proposto pode auxiliar os cirurgiões no planejamento e mesmo na execução das cirurgias cardíacas. Porém, o sistema no estado em que foi estudado, ainda não foi testado em situações reais de intervenções cirúrgicas, o que é previsto pelos autores como implementação futura. Além disso, não há no sistema a captação e processamento de sinais cardíacos em tempo de execução da simulação, o que seria bastante útil no sentido de sincronizar o comportamento das estruturas anatômicas virtuais ao das reais. Os modelos reconstruídos são estáticos, não correspondendo em termos de comportamento às estruturas anatômicas originais.

3.6. Sumário e Conclusões

Todos os trabalhos relacionados analisados buscam enriquecer de alguma forma a visualização de processos do corpo humano, com o propósito de auxiliar médicos e outros profissionais da área da saúde na análise de comportamentos dos órgãos, diagnóstico de doenças, estudos em fisiologia e mesmo intervenções cirúrgicas. O

projeto Vida é o único dentre os analisados que não utiliza modelos tridimensionais capturados da fisiologia de pacientes reais, embora estejam previstos comportamentos mais realistas em implementações futuras. Quanto aos outros trabalhos, é característica comum entre eles a utilização de imagens tradicionais de diagnóstico como ecocardiogramas e ressonâncias magnéticas na reconstituição da fisiologia dos modelos. Embora as técnicas utilizadas para isso nem sempre sejam similares, os resultados são bastante precisos segundo os autores.

Além da precisão tridimensional dos modelos, em alguns sistemas também procura-se simular o comportamento físico dos órgãos, deformando-os apropriadamente.

Alguma forma de visualização em Realidade Aumentada está presente em todos os sistemas, embora o sistema do VIDA estudado ainda utilize vídeos pré-capturados para realizar a mixagem com os objetos virtuais. Não foi notada uniformidade quanto às plataformas utilizadas para gerar o ambiente de Realidade Aumentada, cada sistema possui uma abordagem específica.

Apesar da precisão alcançada nos modelos gerados por alguns dos sistemas estudados, não é tão visível na maioria deles uma preocupação de proporção similar quanto ao comportamento dos órgãos reconstruídos, no sentido de capturar em tempo de execução dados dos órgãos e utilizá-los para realizar sincronia entre os órgãos reais e os virtuais. Essa sincronia é muito importante em situações reais de aplicação, como cirurgias cardíacas, onde o coração está em constante movimentação e um órgãos virtual estático não corresponde ao real.

Com o objetivo de fornecer uma visão geral de todos os sistemas estudados, a Tabela 3.1 apresenta um quadro contendo os sistemas e suas características.

Características / Sistemas	Modelagem dos órgãos virtuais a partir de dados fisiológicos reais	Processamento de múltiplas entradas de dados	Simulação dos comportamentos dos órgãos reais nos modelos virtuais	Processamento de Sinais Biológicos em Tempo Real	Visualização / Interação em Realidade Virtual	Visualização / Interação em Realidade Aumentada	Utilização em situações reais
Project On Image Guided Surgery	■					■	■
Projecto Vida					■		
Die Heart	■		■			■	
Linte, C. A. et al.	■					■	

Tabela 3.1 – Sistemas Estudados e Características.

As células de cor azul representam as características presentes nos sistemas estudados, enquanto que as células de cor branca representam características não detectadas nos sistemas.

A característica “Modelagem dos órgãos virtuais a partir de dados fisiológicos reais” refere-se à tarefa de modelar os órgãos virtuais baseando-se em informações retiradas de órgãos reais.

A característica “Processamento de múltiplas entradas de dados” refere-se à capacidade do sistema receber como entrada vários tipos de informação e processá-los adequadamente, gerando uma saída apropriada.

A característica “Simulação dos comportamentos dos órgãos reais nos modelos virtuais” representa a simulação de comportamentos dos órgãos reais via deformações geométricas e outras técnicas nos modelos virtuais.

A característica “Processamento de Sinais Biológicos em Tempo Real” refere-se à capacidade dos sistemas processarem sinais biológicos dos órgãos simulados em tempo de execução, adaptando automaticamente a simulação.

A característica “Visualização / Interação em Realidade Virtual” classifica os sistemas quanto à presença de elementos de Realidade Virtual que possibilitem a visualização e a interação com os modelos do sistema, seja por transformações geométricas ou quaisquer outros acionamentos.

A característica “Visualização / Interação em Realidade Aumentada” refere-se à capacidade do sistema de gerar as simulações em ambiente de Realidade Aumentada, permitindo aos usuários inspecionar e interagir com os objetos.

A característica “Utilização em situações reais” classifica os sistemas quanto à sua utilização fora do ambiente da pesquisa, em situações reais como intervenções cirúrgicas ou estudos em fisiologia.

A observação da Tabela 3.1 fornece uma visão geral dos pontos onde há maior ou menor presença nos sistemas estudados, permitindo assim a definição de uma estratégia que explore justamente os pontos de maior carência, visando dessa forma gerar uma contribuição mais efetiva para a área. Os 4 sistemas detalhados neste capítulo foram escolhidos por representarem de forma bastante efetiva, o estado da arte na área de pesquisa desta dissertação, tanto em relação às características quanto em relação às carências de funcionalidades.

O estudo dos sistemas apresentados neste capítulo trouxe como resultado, um melhor discernimento no estabelecimento de metas, definição de características e funcionalidades do sistema a ser implementado.

Capítulo 4

4. ARQUITETURA DO SISTEMA

4.1. Introdução

O desenvolvimento e a especificação de um sistema exige que se faça uma ampla pesquisa em torno dos assuntos envolvidos, bem como das soluções semelhantes já existentes. O objetivo deste capítulo é especificar uma arquitetura para o sistema proposto.

4.2. Arquitetura

A partir dos dados teóricos apresentados no Capítulo 2 e das observações feitas nos trabalhos pesquisados no Capítulo 3, foi possível formular uma estrutura viável para o sistema, buscando satisfazer os objetivos deste trabalho. A Figura 4.1 apresenta os elementos envolvidos no sistema e as principais interações entre eles.

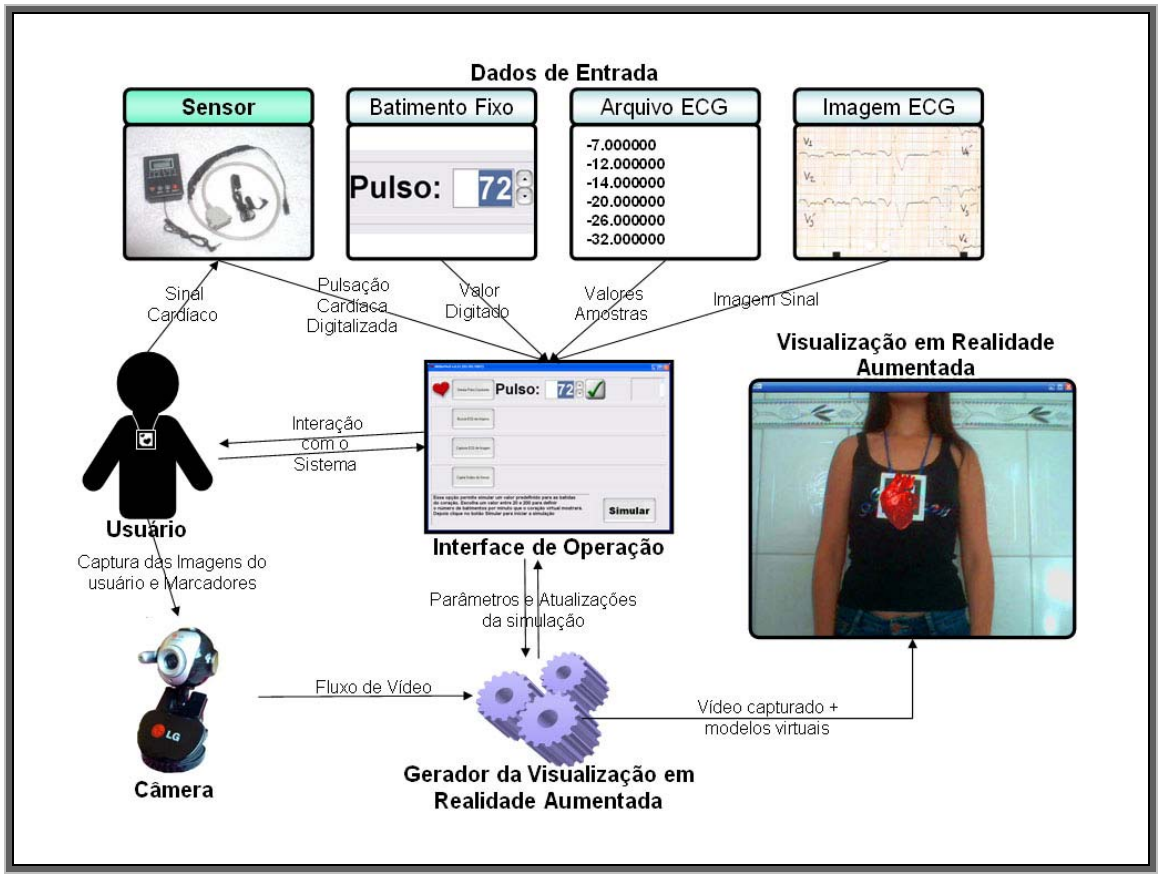

Figura 4.1 – Arquitetura do sistema.

Como pode ser visto, o sistema possui diversas conexões com elementos de hardware e entradas de dados. Também existe uma divisão bem definida entre a interface de operação e o módulo gerador da visualização em Realidade Aumentada, que a partir de agora será referenciado como GVRA. A seguir, são explicados mais detalhadamente os elementos presentes na arquitetura do sistema.

- Usuário – O usuário é uma entidade genérica, podendo representar tanto os sujeitos onde será visualizado o sinal, como também os operadores do sistema. O próprio sujeito poderá ser o operador do sistema. O usuário se comunica com o sistema através da interface de operação, utilizando dispositivos convencionais de interação, como mouse teclado e monitor.

- Sensor – O sensor representa a entrada de dados em tempo real do sistema, capturando informações do sujeito em tempo de execução do sistema, digitalizando essas informações e realimentando a simulação, de forma a atualizar a visualização apropriadamente. O sensor deve ser de um tipo que permita a fácil integração com o sistema, tanto em termos de hardware quanto de software.

- Câmera – A câmera tem o papel de capturar informação visual do mundo real, onde estão o usuário e os marcadores utilizados pelo módulo GVRA. Pode ser representada por uma webcam, um fluxo de vídeo remoto, uma placa de captura de TV ou qualquer outro dispositivo que tenha as características necessárias ao sistema. O vídeo capturado pela câmera será reutilizado posteriormente pelo gerador de Realidade Aumentada, por isso é importante a qualidade deste vídeo ser adequada a alguns parâmetros mínimos de qualidade.

- Batimento Fixo – assim como o sensor, o batimento fixo pode ser considerado um tipo de entrada de dados. Nesse caso, a entrada de dados é manual onde o usuário, através da interface de operação, insere um valor de batidas por minuto do coração para simular. O sistema deve estabelecer valores mínimos e máximos para batidas do coração, de modo a evitar situações que fujam da realidade. Esta entrada de dados

serve para simulações baseadas em parâmetros digitados pelo usuário, em contraste com as entradas de dados, baseadas em dados reais de sujeitos.

- Arquivo ECG – Este tipo de entrada de dados é inserido no sistema como um arquivo contendo os valores de amplitude de um gráfico de eletrocardiograma, em uma taxa de amostragem fixa. Tais arquivos podem ser gerados por dispositivos médicos de diagnóstico cardíaco ou mesmo outros softwares de simulação. O formato padrão de entrada é o arquivo-texto.

- Imagem ECG – Diversos exames cardíacos são gravados em forma de gráficos de linhas, em mídias como papel em folhas ou bobina. De forma a recuperar de forma rápida as informações contidas nesse tipo de entrada de dados, os exames devem ser digitalizados e salvos em arquivos gráficos. O sistema deverá ser capaz de ler os principais formatos gráficos atuais para recuperar arquivos de imagem com exames de eletrocardiograma previamente digitalizados.

- Interface de Operação – A interface de operação é a principal forma de comunicação entre o usuário e o sistema. Para que a interação ocorra de forma suave e bem sucedida, é importante que a interface seja rica em usabilidade, seja auto-explicativa, tenha elementos visuais agradáveis e facilidade na operação, utilizando dispositivos conhecidos como mouse e teclado. Além disso, a interface deve se comunicar de forma eficiente com as outras partes do sistema, desde as entradas de dados até o módulo GVRA.

- Gerador da Visualização em Realidade Aumentada (GVRA) – É o módulo responsável pela união entre o vídeo capturado e recebido da câmera e os elementos virtuais a serem inseridos. Esta união deverá ser feita de forma transparente, com boa sincronização entre o vídeo e os elementos virtuais e também com uma taxa de quadros por segundo alta o suficiente para não prejudicar a sensação de movimento no vídeo final,

sendo considerada a taxa de 15 quadros por segundo o mínimo aceitável. Os parâmetros da animação são recebidos diretamente da interface de operação e verificados sempre que há alterações. Para a sincronização dos elementos virtuais são utilizados marcadores monocromáticos no vídeo capturado. A interface também deverá ser implementada sobre uma linguagem que ofereça acesso ao hardware do computador, funcionalidade utilizada no recebimento de dados do sensor.

- Visualização em Realidade Aumentada – É o resultado final da combinação de todos os elementos do sistema, uma visualização que mescla o vídeo captado pela câmera juntamente com os modelos virtuais. Essa visualização é animada, sendo que os parâmetros da animação são controlados pela interface de operação e efetivamente calculados pelo módulo GVRA. Essa visualização é dinâmica, podendo se alterar parâmetros de entradas e configurações em tempo de execução e visualizar esses resultados refletindo no comportamento da simulação.

4.3. Elementos físicos do sistema

Para possibilitar o desenvolvimento da arquitetura proposta, foram estabelecidos além dos elementos conceituais do sistema em si, alguns requisitos e características dos elementos físicos sobre os quais o sistema deve ser executado. A seguir são descritos estes elementos.

- Computador – Foi decidido que o projeto seria implementado em um computador pessoal, devido a fatores como baixo custo de maquinário, alta disponibilidade no mercado e facilidade na customização e instalação de componentes adicionais. Dentre os principais componentes utilizados em aplicações gráficas para computadores pessoais, os maiores influenciadores de performance são o processador, a memória e a placa de vídeo. O processador utilizado foi um Athlon 64 3000+, a memória foi uma combinação de 2 módulos Kingston DDR400 em modo Dual Channel e a placa de vídeo foi uma GeForce 6800 XT. Utilizando essa configuração foram obtidos bons resultados no sistema, com boa precisão e velocidade, tanto nos cálculos como na exibição da simulação.

- Câmera – A câmera é parte vital no resultado visual final da simulação, pois o fluxo de vídeo por ela produzido é reutilizado e ao final mesclado com os elementos virtuais. Assim, é importante que a qualidade da câmera seja razoável, produzindo uma imagem limpa e nítida, e também uma taxa de quadros por segundo que seja suficiente para uma boa percepção do movimento. Com base nestes fatores foi escolhida a webcam LG WebPRO2, que tem resolução de 640X480 pontos, utiliza interface de conexão USB 1.0 e consegue capturar imagens com taxas superiores a 15 quadros por segundo.

- Monitor – O monitor é o dispositivo onde a simulação final será visualizada. Por isso é importante que ele tenha boa qualidade de cores, resolução e nitidez. Praticamente todos os monitores de computador fabricados atualmente possuem boas características de visualização, sendo a escolha deste componente mais subjetiva que as demais. O monitor utilizado nas simulações foi um CRT LG F700P, utilizando resolução de 1280 X 960 pontos, no formato de tela 4:3.

- Sensor – Na arquitetura proposta, além das entradas de dados baseadas em parâmetros e arquivos existe o sensor, responsável por capturar dados do sujeito em tempo real e transmitir esses dados ao sistema, que atualiza de forma apropriada a visualização da simulação. Dentre os sensores analisados, buscou-se um que tivesse precisão suficiente para demonstrar os resultados, e ao mesmo tempo fosse de fácil integração ao computador utilizado e de baixo custo. Assim, foi adaptado um sensor de pulso de bicicleta ergométrica à porta paralela do computador, possibilitando o monitoramento do sinal em tempo real e com um custo bastante inferior às soluções de aquisição profissionais existentes no mercado. O sensor utilizado é não-invasivo, sendo acoplado ao lóbulo da orelha esquerda do sujeito para fazer as medições.

Os elementos descritos podem ter variações nos componentes e configurações. É muito complexo descrever com precisão o percentual de compatibilidade entre os elementos propostos e as muitas configurações alternativas possíveis, por isso o

recomendado para o bom funcionamento do sistema é utilizar componentes similares aos descritos ou de maior capacidade.

4.4. Elementos de software do sistema

Os elementos físicos do sistema fornecem a base sobre a qual os elementos de software são executados. Na implementação do sistema, os elementos de software também tem um papel fundamental, pois a escolha do software pode influenciar de forma positiva ou não o funcionamento do hardware. Além disso, algumas etapas do sistema necessitaram de ferramentas especiais de software. Sendo assim, os principais elementos de software utilizados no sistema são especificados a seguir.

- Sistema Operacional – O sistema operacional é o software responsável pelo gerenciamento de todos recursos do computador, bem como pelo bom funcionamento dos outros softwares a serem utilizados. A opção escolhida para a implementação do sistema foi o Microsoft Windows XP SP2, pois esse sistema já se encontra em uso há bastante tempo, tendo boa flexibilidade e suporte aos recursos necessários. Todos os outros elementos de software foram escolhidos com base em sua compatibilidade com o sistema operacional, uma vez que é sobre ele que suas soluções serão executadas.

- Linguagens de Programação – É possível encontrar uma grande variedade de linguagens de programação atualmente, cada qual com suas soluções e aplicações. Algumas destas linguagens são menos focadas em soluções específicas, servindo para propósitos mais amplos e gerais. É o caso da linguagem C++ [60], escolhida como a base da implementação do sistema. A linguagem C++ permite ao programador manipular os recursos mais profundos do computador, como portas de comunicação e diversos outros elementos de hardware. Além disso, a linguagem possui capacidades bastante potentes de programação, como estruturas de dados complexas e orientação a objetos, sendo útil para as mais variadas aplicações de software, tais como sistemas operacionais, aplicações matemáticas e computação gráfica. Atualmente, a linguagem C++ existe em alguns pacotes de programação, como o C++ Builder [61], Visual C++ [62], Dev C++ [63] e outros mais. Cabe ao programador analisar os

recursos oferecidos por cada pacote e escolher qual a melhor solução, ou soluções para o projeto. Para o sistema proposto, foram utilizados o C++ Builder na criação da interface de operação e o Visual C++ no módulo gerador da visualização em Realidade Aumentada. O C++ Builder foi escolhido por sua simplicidade na criação de interfaces gráficas de operação, bem como a possibilidade de expansão de funcionalidades através de plugins. Já o Visual C++ foi escolhido por sua facilidade em criar aplicações gráficas, fácil integração com bibliotecas de vínculo dinâmico e porque a biblioteca utilizada para a geração de Realidade Aumentada possui maior suporte e integração com essa ferramenta.

- Biblioteca de Realidade Aumentada – A Realidade Aumentada tem sido implementada de muitas maneiras nos últimos anos. Alguns autores de aplicações optam por desenvolver implementações específicas para as suas soluções, enquanto outros procuram criar bibliotecas de Realidade Aumentada, genéricas o suficiente para um amplo número de aplicações. Uma biblioteca de grande destaque na Realidade Aumentada é o ARToolKit, que possibilita ao desenvolvedor a criação facilitada de aplicações em Realidade Aumentada. Esta biblioteca é de livre utilização para fins não-comerciais, tendo o código aberto e passível de modificações pelos usuários, o que a torna flexível e adequada aos propósitos do sistema. Dentre as versões existentes do ARToolKit optou-se por utilizar a versão 2.52, que têm boa funcionalidade, suporte a modelos tridimensionais em OpenGL [74] ou VRML [73], facilidade de modificação e ótima performance na geração das visualizações.

- Modelagem dos elementos virtuais – Existem hoje muitos softwares para autoria e edição de modelos tridimensionais. Neste segmento, o 3ds Max® da Autodesk® [72] pode ser destacado como o principal representante pela geração dos modelos virtuais modelados para jogos e aplicações de Realidade Virtual. Além disso o 3ds Max® possui integração com as linguagens VRML e OpenGL, ambas muito utilizadas em aplicações de Realidade Aumentada e Virtual. Assim, optou-se pela

utilização do 3ds Max® juntamente com a VRML na criação dos modelos virtuais utilizados pelo sistema.

4.5. Conclusões

A fase de especificação da arquitetura do sistema sistema foi de suma importância no projeto como um todo. Nesta fase foi possível delinear melhor as metas do sistema, as suas limitações e as possibilidades das ferramentas utilizadas. Também é nessa fase que problemas de lógica em potencial surgem e podem ser sanados mais facilmente, antes que o projeto seja de fato implementado e a manutenção e mudanças estruturais se tornem mais complexas, o que de fato foi observado neste projeto. Através da arquitetura proposta, foi possível implementar com sucesso todos os elementos propostos no sistema, satisfazendo assim os objetivos propostos.

Capítulo 5

5. DETALHES DE IMPLEMENTAÇÃO

5.1. Introdução

A fase de especificação da arquitetura do sistema pode ser comparada à criação de uma planta e estudos arquitetônicos de um edifício, onde são definidos os parâmetros do projeto, as restrições a serem seguidas e as melhores ferramentas para a implementação do que foi definido.

Em posse das definições geradas pela arquitetura do sistema, o mesmo pode ser então implementado de acordo com essas definições, usando-as como um guia durante todo o processo. Este capítulo trata justamente disso, os detalhes da implementação do sistema sistema. É importante ressaltar que o sistema foi dividido já na arquitetura em 2 grandes núcleos, a interface de operação e o módulo GVRA. Assim, será seguido este padrão de divisão neste capítulo, começando com o núcleo da interface de operação, imediatamente seguido pelo detalhamento do módulo GVRA.

5.2. Interface de operação

A interface de operação é o meio de comunicação entre o usuário e o sistema. Uma boa interface valoriza o sistema, cativando o usuário, enquanto que uma interface mal projetada pode prejudicar o desempenho e a avaliação dos usuários. Existem diversos tipos de usuários, alguns altamente especializados e proficientes nos processos computacionais e também outros com conhecimentos e familiaridade com o computador, bastante limitados. A interface de operação precisa levar em conta qual o público-alvo de potenciais usuários, de forma a fornecer elementos de interação adequados a esse público.

Atualmente, os dispositivos mais amplamente utilizados em todo o mundo para a interação com sistemas de computador são o teclado e o mouse. Novas interfaces como a Realidade Aumentada e a Realidade Virtual sugerem novos dispositivos, mas a sua popularização entre os usuários de computador ainda não se concretizou de fato. Assim, na implementação do sistema optou-se por seguir o padrão atualmente estabelecido mundialmente, ou seja, a utilização de teclado e mouse aliados a janelas e outros elementos gráficos para a interação com o sistema.

Para a implementação da interface, foi utilizado o ambiente de desenvolvimento do C++ Builder. Este ambiente permite a fácil criação de janelas com elementos de interação tradicionais, como botões, caixas de texto e barras de rolagem. A Figura 5.1 apresenta o ambiente do C++ Builder, com alguns elementos de interação já criados.

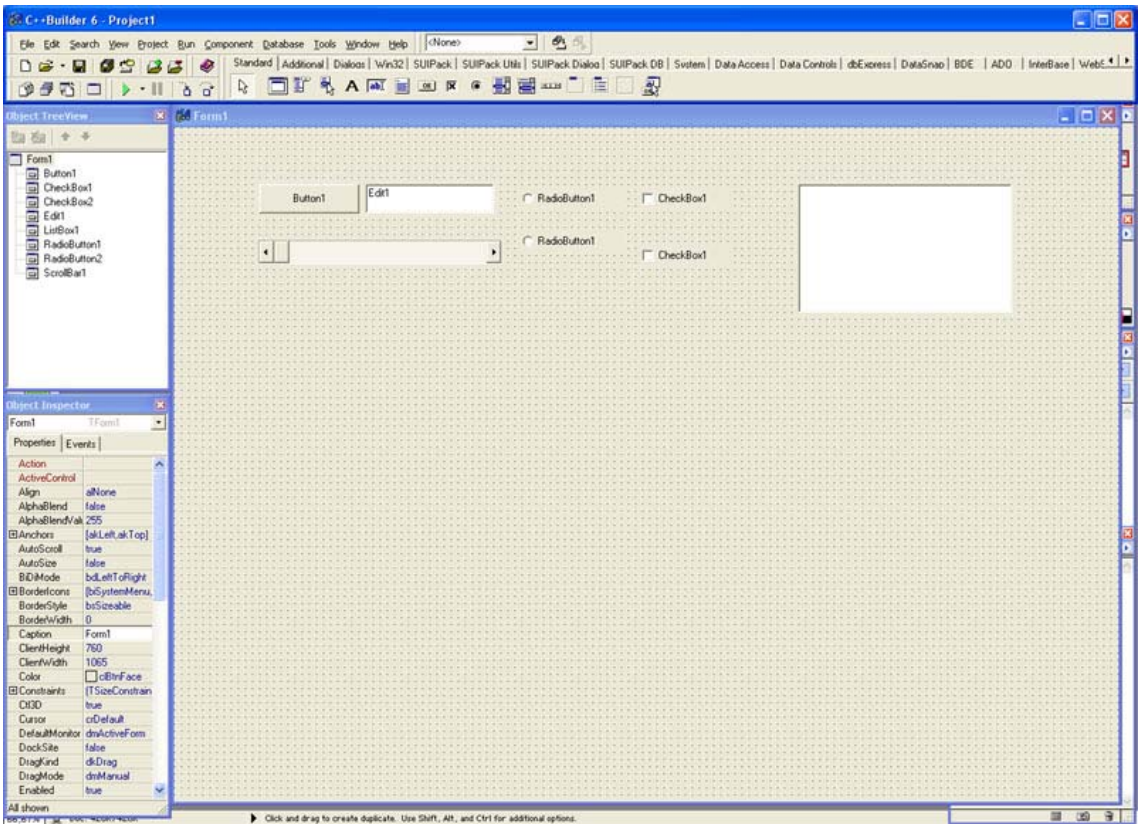

Figura 5.1 – Ambiente do C++ Builder.

Apesar dos componentes do C++ Builder serem bastante familiares para usuários mais experientes, os aspectos visuais desses componentes já estão desatualizados frente aos atuais softwares existentes, o que pode causar uma certa confusão e desconforto para os usuários. Assim foi resolvido que se utilizaria um plugin para converter os elementos antigos para formatos mais atuais ou contar com novos componentes atuais customizáveis. O plugin escolhido foi o Suipack [64], que reúne as estas duas funcionalidades. Com ele é possível tanto converter uma aplicação feita com os componentes originais do C++ Builder quanto criar uma aplicação com componentes totalmente novos do plugin. A Figura 5.2 apresenta os componentes padrão do C++ Builder em contraste com as versões convertidas para os componentes do plugin Suipack.

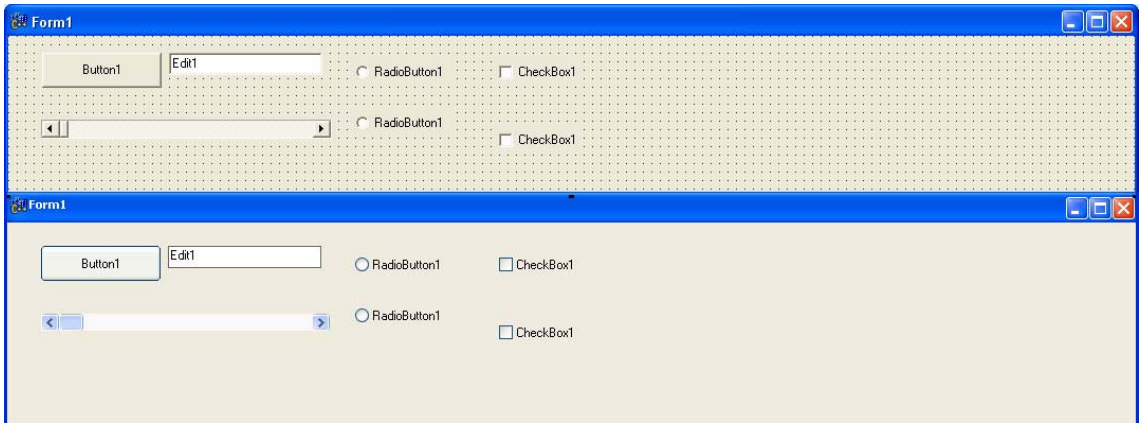

Figura 5.2 – Componentes padrão do C++ Builder x Componentes Suipack.

Além do visual apresentado, o plugin apresenta diversas outras opções, dando flexibilidade às aplicações desenvolvidas.

Com a junção do C++ Builder e o plugin Suipack foi possível a implementação de todos os elementos necessários à interface de operação desejada. A seguir são detalhadas todas as situações presentes na interface de operação.

5.2.1 Tela Inicial.

Um dos objetivos do sistema sistema é a criação de uma interface de fácil utilização e auto-explicativa, com baixos níveis de complexidade, de maneira a ser operada com facilidade por usuários com pouco conhecimento na área de informática e ao mesmo tempo não se tornar incômoda para usuários mais experientes. Assim, quando na criação da tela inicial do sistema, observou-se os seguintes critérios:

- ✓ Textos explicativos do sistema, contendo informações relevantes sobre as opções disponíveis, guiando assim os usuários na utilização de cada elemento do sistema;
- ✓ Visual agradável com cores suaves e padrões estéticos relativamente próximos dos demais softwares existentes atualmente no ambiente computacional;
- ✓ Simplificação das opções iniciais, evitando a sobrecarga de informação para o usuário.

A tela inicial do sistema foi projetada com base nesses critérios e pode ser visualizada na Figura 5.3.

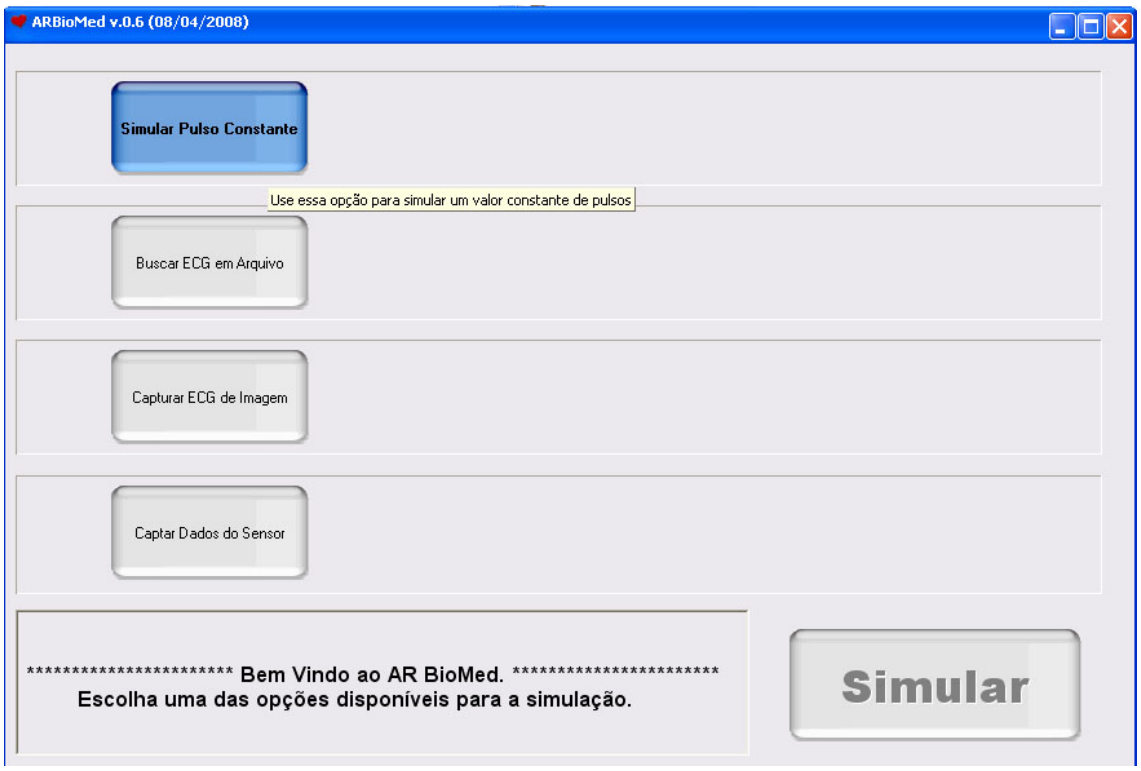

Figura 5.3 – Tela Inicial do sistema.

Como pode ser visto, a interface inicial do sistema é relativamente simples, com informações simples e focadas, guiando o usuário na melhor escolha.

Os 4 botões presentes no lado esquerdo da janela representam as diversas opções de entrada de dados presentes no sistema. Ao clicar em cada uma destas opções, o conteúdo correspondente é exibido ao lado direito do botão. Os botões foram criados em tamanhos ampliados propositadamente, aproveitando o espaço da janela e tornando fácil o alcance via mouse. Também pode se notar que existem elementos gráficos em baixo relevo entre cada opção, enfatizando a divisão entre elas.

Um quinto botão com o título “Simular” está presente ao lado direito inferior da janela, sendo que inicialmente encontra-se desabilitado. Este botão tem a função de consolidar as informações ajustadas nas opções de entradas e enviá-las para o módulo GVRA. O botão é desativado não só inicialmente, mas sempre que houver condições de erro que comprometam a visualização da simulação. Essas condições poderão também ser reportadas ao usuário através de textos explicativos e elementos gráficos intuitivos, como será visto mais adiante.

O quadro à esquerda do botão Simular destina-se a fornecer pequenas quantidades de informação ao usuário, guiando-os nas diversas etapas de utilização do sistema. Nesse quadro são informadas também algumas restrições do sistema, de forma

a prevenir erros na utilização. Em caso de erros, algumas vezes este quadro também é utilizado para informar ao usuário os motivos do erro e como efetuar as devidas correções.

Ao posicionar o mouse por alguns instantes sobre qualquer um dos botões e mesmo o quadro-explicativo, o usuário visualiza uma dica simples de utilização em um pequeno quadro amarelo. Esta é mais uma forma de auxílio ao usuário.

A interface inicial é utilizada pela maioria das opções, exceto a opção "Capturar ECG de Imagem" que sobrepõe sua própria janela à interface inicial, dada a complexidade maior requerida pelos seus componentes. A seguir, são detalhadas todas as opções presentes na interface inicial, na mesma disposição em que se encontram visualmente.

5.2.2 Opção Simular Pulso Constante

Esta opção destina-se a simular um valor constante de pulsação cardíaca, definido em batidas por minuto (bpm). No momento em que o usuário clica no botão correspondente a essa opção, a interface se altera, exibindo os elementos da opção. A Figura 5.4 mostra estes elementos.

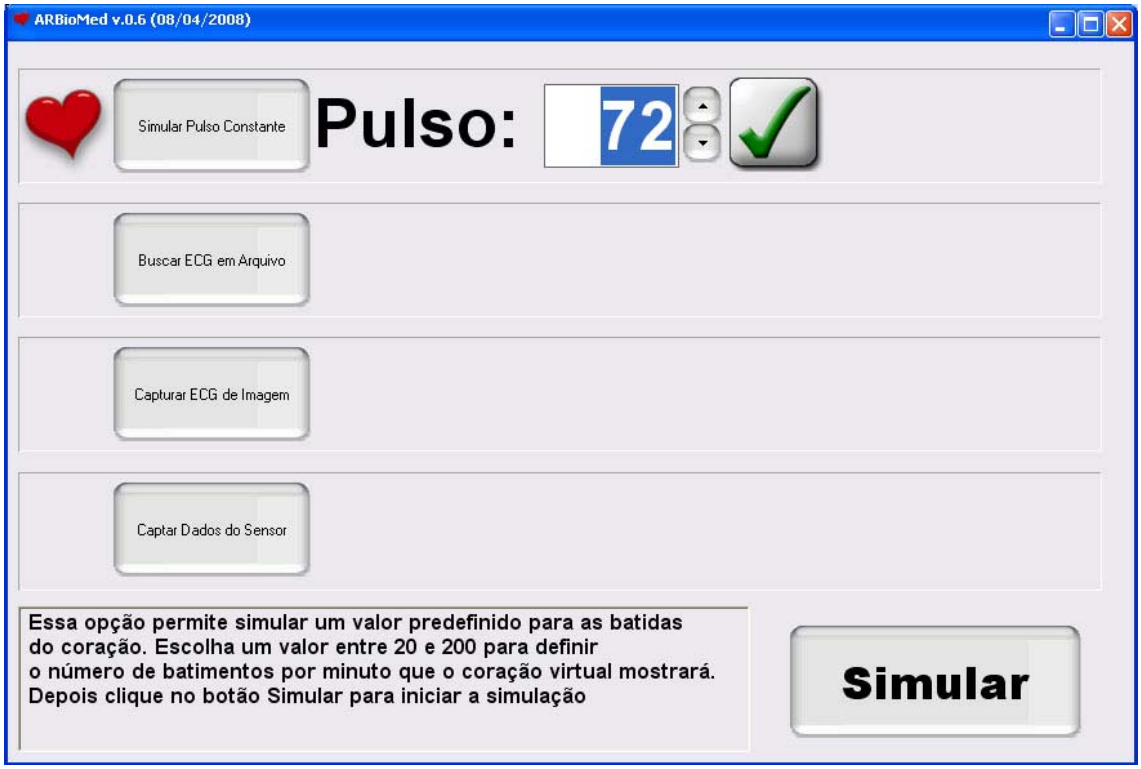

Figura 5.4 – Elementos de Interface da opção Simular Pulso Constante.

Como pode ser notado, aparecem alguns parâmetros editáveis nessa opção. A quantidade de batidas por minuto do coração pode ser definida tanto via teclado,

digitando-se o valor desejado, como através das setas ao seu lado, aumentando ou diminuindo uma unidade no pulso a cada clique do mouse. Também pode ser notado que o botão Simular ficou habilitado, permitindo assim que a simulação seja iniciada ou atualizada. O símbolo de status da opção encontra-se no estado de "certo" em cor verde ajudando a reforçar a sensação de que está tudo certo. Além disso, o quadro de informações muda seu conteúdo, explicando ao usuário os passos necessários para conseguir iniciar a simulação.

Inicialmente, o valor definido para as batidas por minuto do coração é 72, um valor considerado médio na população. Estabeleceu-se que valores menores que 20 ou maiores que 200 bpm não seriam permitidos. Se o usuário tenta digitar um valor maior que 200 ou menor que 20, o sistema desabilita o botão simular, o símbolo de status da opção muda de "certo" em verde para um "errado" em vermelho, mostrando ao usuário que correções são necessárias para tornar a simulação válida. O quadro explicativo também se altera nessa situação, mostrando o que deve ser feito para corrigir o problema. A Figura 5.5 ilustra uma típica situação de erro.

Figura 5.5 – Situação de Erro na Opção Simular Pulso Constante.

Após corrigido o problema, o botão simular é reabilitado e o símbolo de status da opção volta para o estado de "certo". O quadro explicativo também se altera indicando que a simulação pode ser iniciada.

Quando o usuário clica sobre o botão “Simular”, toda a interface se move para o canto inferior direito da tela do computador, pois a janela da simulação em ambiente de Realidade Aumentada é aberta no canto esquerdo superior da tela. Com o deslocamento, a interface não atrapalha a visualização e vice-versa. O botão simular também muda seu título para “Atualizar”, indicando que a simulação está sendo executada. A Figura 5.6 apresenta os cenários antes (esquerda) e depois (direita) do clique sobre o botão Simular.

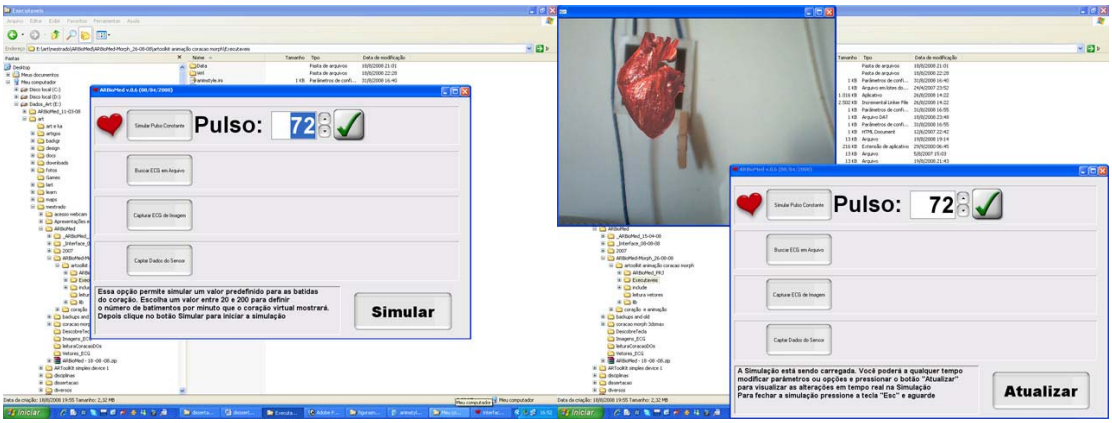

Figura 5.6 – Cenário resultante do clique sobre o botão Simular.

Além desse comportamento, uma importante operação é executada, a consolidação das informações escolhidas na opção e o envio dessas informações para a simulação. Isso é feito mediante um arquivo-texto, denominado config.ini que é lido periodicamente pelo módulo GVRA. Este arquivo contém todos os dados necessários da opção simulada, além de algumas informações adicionais utilizadas somente pelo módulo GVRA. A Figura 5.7 apresenta o arquivo config.ini gerado pela configuração da opção Simular Pulso Constante nos seus valores padrão.

Figura 5.7 – Arquivo config.ini gerado pela opção Simular Pulso Constante.

O arquivo gerado pela opção Simular Pulso Constante é muito simples, uma vez que o único parâmetro de especial interesse para o módulo GVRA é o pulso.

O formato de dados do arquivo é bastante padronizado. O primeiro valor existente é sempre a opção que se deseja simular. Este valor é utilizado posteriormente para a filtragem e devido processamento no módulo GVRA. O segundo valor neste caso é a pulsação escolhida. É importante ressaltar que utilizando o primeiro valor, o sistema é capaz filtrar e prever a estrutura dos tipos de dados subseqüentes.

Os valores são sempre separados por caracteres de espaços. A opção por arquivos-texto é justificada pela pouca complexidade dos arquivos gerados e também pelo fato de permitir monitoramento facilitado por usuários avançados.

A qualquer momento nessa opção, o usuário pode clicar em qualquer um dos outros botões escolhendo uma opção diferente, porém o pulso permanece no valor editado.

5.2.3 Opção Buscar ECG em Arquivo

Esta opção destina-se a recuperar e simular gráficos de ECG provenientes de arquivos-texto contendo os valores de amplitudes em formato numérico de ponto flutuante. A taxa de amostragem determina o espaçamento entre os valores de amplitude, por isso é necessário seu conhecimento prévio, uma vez que esta informação não vem junto ao arquivo.

Os arquivos podem ter sido gerados por equipamentos médicos, outros softwares de simulação ou até pela opção Capturar ECG de Imagem do sistema, conforme será visto adiante. A localização desses arquivos pode ser variada, podendo estar armazenados em disquete, disco rígido, CD, DVD, PenDrive, entre outras mídias. O papel do sistema é possibilitar a recuperação dos arquivos no máximo de mídias compatíveis, pré-visualizar o gráfico gerado e então realizar a simulação.

Assim, ao clicar na opção Buscar ECG em Arquivo, o usuário obtém os elementos de interface ilustrados na Figura 5.8.

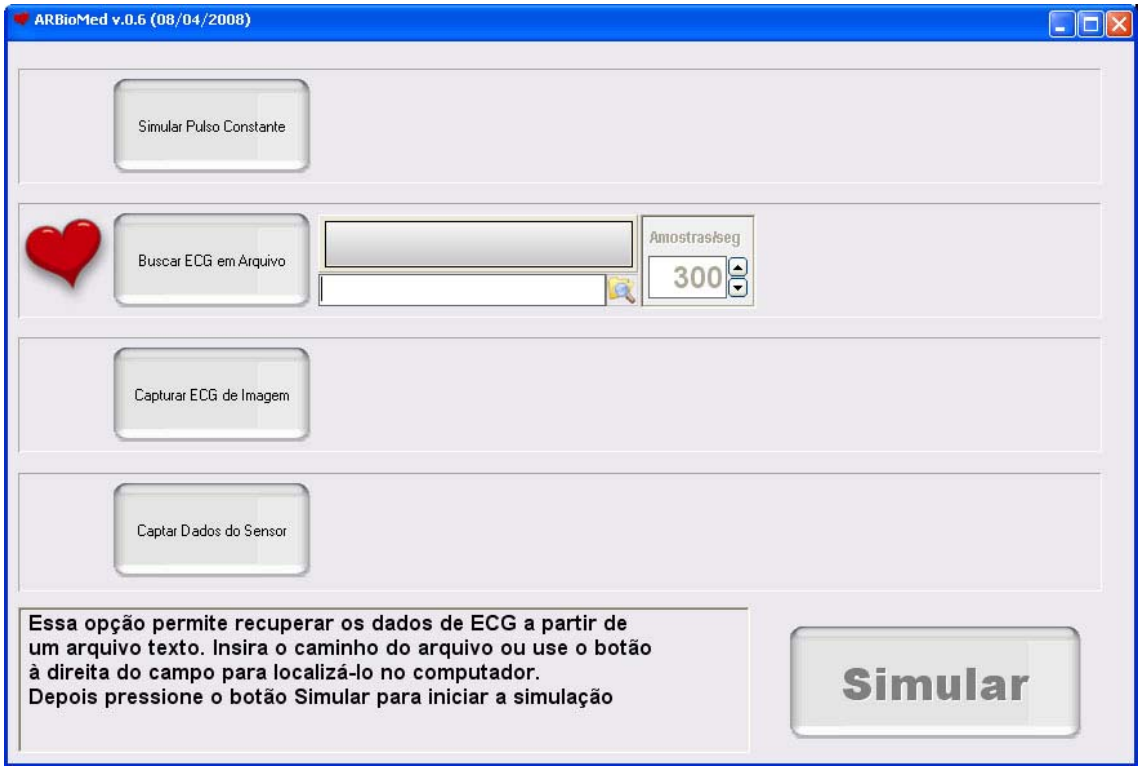

Figura 5.8 – Elementos de Interface da opção Buscar ECG de Arquivo.

Como pode ser visto, o quadro explicativo altera seu texto, indicando ao usuário os passos necessários para utilizar a opção. O primeiro passo é localizar o arquivo-fonte das informações. A interface da opção Buscar ECG de Arquivo permite ao usuário digitar o caminho completo do arquivo no campo de texto à direita do botão da opção, ou localizá-lo através de um componente especial do C++ Builder, chamado OpenDialog, representado pelo ícone de uma pasta e lupa. Esse componente exibe uma interface gráfica que permite ao usuário navegar entre as unidades e pastas disponíveis até encontrar o arquivo desejado.

Após o usuário digitar o caminho do arquivo e pressionar a tecla Enter ou encontrar o arquivo através do OpenDialog, é gerado um gráfico de pré-visualização do sinal, mostrando as curvas geradas pelas amostras, localizado logo acima do campo de texto com o caminho do arquivo. Se o arquivo escolhido não contiver dados válidos de eletrocardiograma, o gráfico não é gerado corretamente, ou senão apresenta anomalias. Por isso essa etapa exige bastante atenção do usuário, caso contrário ocasionará falhas na simulação.

O campo de texto à direita do gráfico intitulado Amostras/seg é uma configuração de fundamental importância no momento da simulação. Esse valor representa a quantidade de amplitudes do sinal capturadas por segundo pelo

equipamento ou software autor do arquivo a simular. O valor padrão do sistema é de 300 amostras por segundo, podendo ser facilmente alterado através de digitação no campo de texto ou através das setas ao lado, que aumentam ou diminuem em 1 unidade a quantidade de amostras a cada clique. A Figura 5.9 apresenta a pré-visualização de um arquivo de eletrocardiograma e o ajuste da taxa de amostragem por digitação no campo de texto.

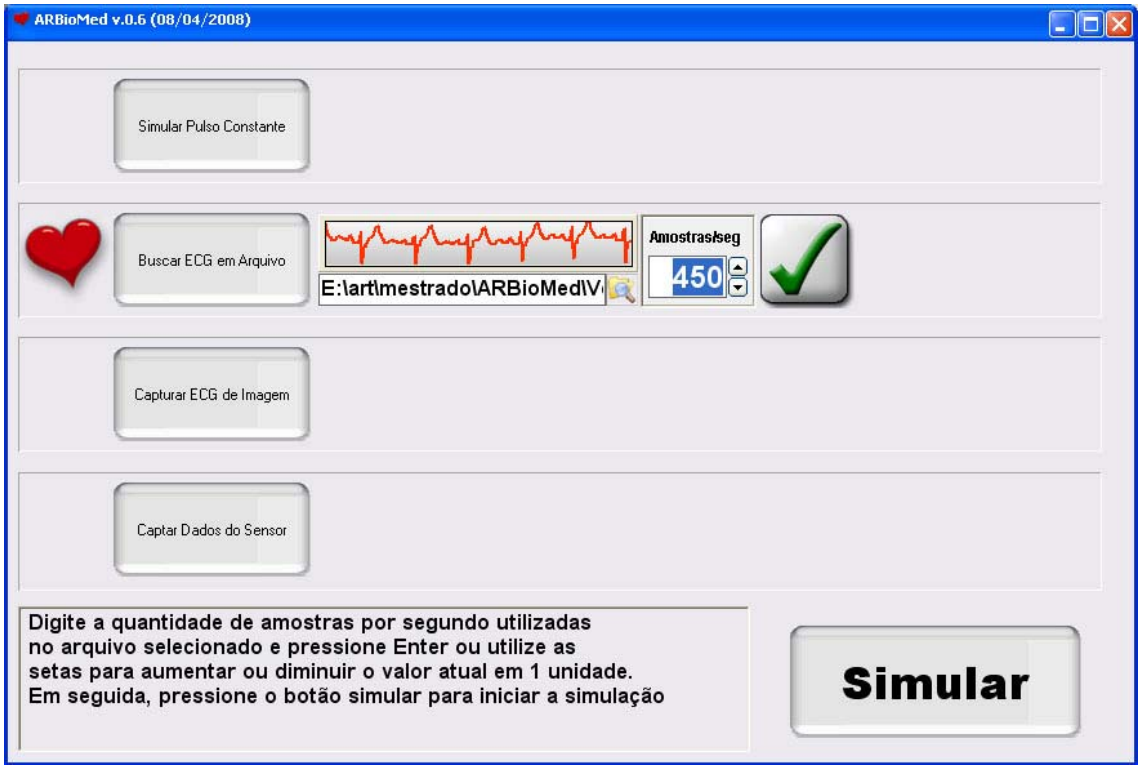

Figura 5.9 – Pré-visualização de um arquivo de ECG e ajuste de Amostras/seg.

Pode se notar que o sinal de "certo" aparece mostrando a simulação pode ser feita com sucesso. Vale lembrar que nesse caso não é analisado o estado do gráfico do arquivo, somente se ele existe. O botão Simular também é habilitado permitindo o início ou atualização da simulação. Ao clicar sobre ele, novamente é utilizado o arquivo config.ini para passar os parâmetros configurados para o módulo GVRA. O conteúdo do arquivo neste caso é o número da opção, a quantidade de amostras por segundo e o caminho completo do arquivo com um caractere especial simbolizando o fim do caminho. O arquivo passa por algumas filtragens antes de alimentar a simulação propriamente dita, por isso optou-se por passar somente o caminho do arquivo, evitando assim duplicidade desnecessária de arquivos no disco rígido. O caractere especial utilizado para indicar o fim do caminho é o sinal de menor (<) que não aparece em caminhos de arquivo e é utilizado pelo módulo GVRA quando na leitura do arquivo

config.ini. O comportamento da interface do sistema é o mesmo que na opção anterior, deslocando-se para o canto direito e inferior da tela para permitir uma boa visualização da simulação. Vale ressaltar que se a simulação estiver sendo executada e ajustada para outra opção, ela se atualizará quase que imediatamente após a solicitação através do botão Simular/Atualizar, refletindo os novos parâmetros configurados.

5.2.4 Opção Capturar ECG de Imagem

Apesar de a maioria dos modernos equipamentos médicos de eletrocardiograma apresentarem facilidade de integração com os computadores, seja através de acesso direto ou, como no caso anterior, pela geração de arquivos de relatório e diagnóstico, ainda há bastante utilização de exames cardiológicos com saída em papel, como é o caso dos gráficos de eletrocardiograma. Para recuperar as informações contidas nesse tipo de exame de forma automatizada, a melhor solução é o processamento de imagem sobre as informações visuais e a conseqüente extração dos valores dos gráficos.

A primeira fase em um sistema que use processamento de imagem é a aquisição [59]. É nesta fase que as imagens são capturadas do mundo real e transformadas para dados digitais, permitindo assim a sua manipulação. A aquisição pode ser feita por diversos tipos de dispositivos, como câmeras digitais, aparelhos de fax, scanners de mesa e scanners de mão. O foco do sistema não é a aquisição das imagens de eletrocardiograma, mas sim o processamento das mesmas. Porém, para que essa etapa ocorra de forma eficaz, é necessário conhecer algumas das características utilizadas na fase de aquisição da imagem, como a resolução utilizada e sua relação com as medidas originais dos exames de eletrocardiograma. Essas características possibilitam ao sistema mensurar o intervalo de tempo entre as amostras do gráfico, de forma parecida com o parâmetro Amostras/seg da opção Buscar ECG em arquivo. O resultado das configurações é notado somente no momento da simulação, onde efetivamente são utilizados os parâmetros.

Dada a maior complexidade da opção Capturar ECG de Imagem, foi decidido que as configurações desta opção seriam melhor dimensionadas em uma janela exclusiva, sobrepondo a interface inicial. A janela criada pode ser visualizada na Figura 5.10.

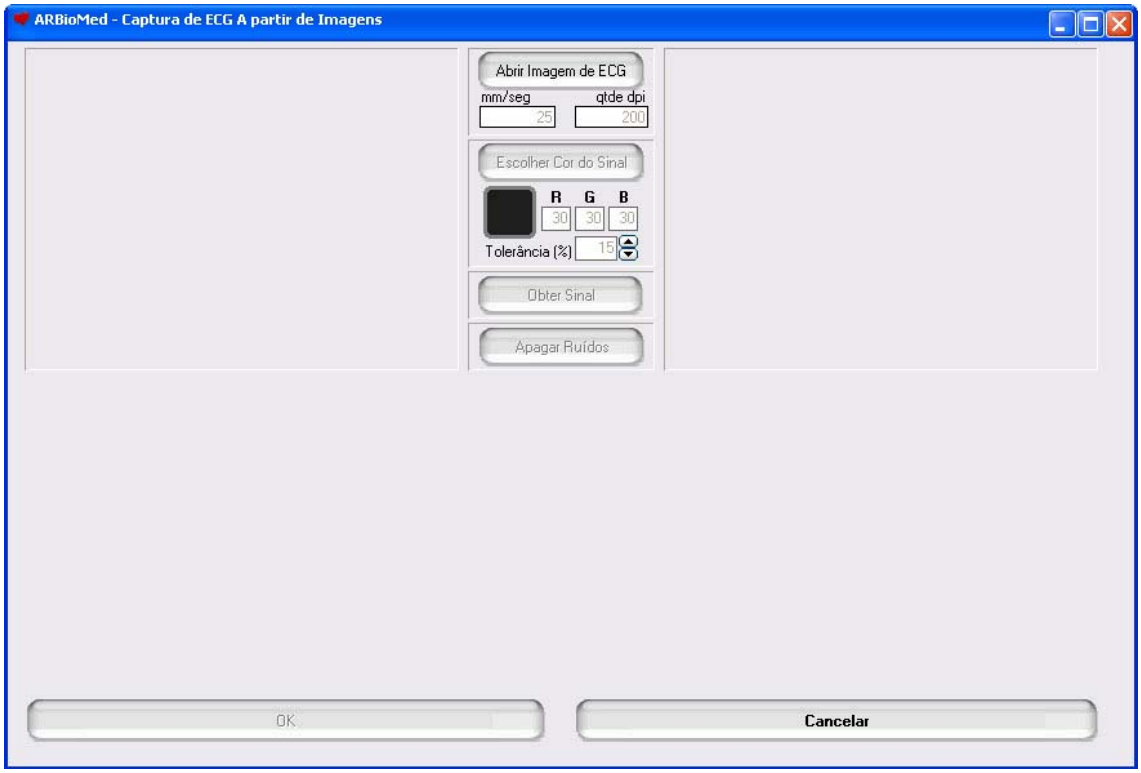

Figura 5.10 – Interface da Opção Capturar ECG de Imagem.

Os elementos da interface visualizada foram projetados de modo a guiarem o usuário nas etapas necessárias à simulação. Porém, nessa opção é necessário que o usuário conheça alguns dos parâmetros para obter de forma mais fácil os resultados esperados. Isso porque o sistema tem configurações padrão que não são adequadas a todas as situações e devem ser ajustadas apropriadamente em cada caso. Um pequeno treinamento e esclarecimento é válido nesse caso. De qualquer forma, os elementos da interface são programados para irem se ativando somente quando as etapas anteriores forem concluídas. Essas restrições são a forma encontrada de guiar o usuário pela opção, ainda que parcialmente.

Inicialmente a interface apresenta somente 2 botões ativos, o “Cancelar” e o “Abrir Imagem de ECG”. O botão “Cancelar” tem múltiplas funções ao longo da interface como será visto posteriormente, mas nesse momento, ao pressioná-lo, a interface da opção Capturar ECG de Imagem é descartada e pode ser visualizada a interface inicial do sistema novamente. Já o botão “Abrir Imagem de ECG” é a primeira etapa no processo de extração do eletrocardiograma a partir de uma imagem. Ao clicar nesse botão, o usuário obtém uma janela de diálogo onde pode localizar o arquivo gráfico que contém os gráficos de eletrocardiograma. Além dos elementos padrão de

navegação, este diálogo apresenta uma pequena pré-visualização da imagem, facilitando a localização do arquivo para o usuário.

Quando o usuário encontra o arquivo desejado e clica no botão abrir, ocorre um processo inicial de identificação do formato da imagem. O C++ Builder só permite originalmente a abertura direta de arquivos do tipo bitmap, com extensão ".bmp". Essa é uma séria restrição, visto que atualmente existem formatos bem mais populares que o bitmap, alguns deles bastante utilizados por scanners e outros dispositivos de aquisição. Assim, foi necessária a utilização de um software auxiliar com o objetivo de ampliar as possibilidades de formatos gráficos. O software escolhido foi o IrfanView [65], que é de utilização gratuita para fins não comerciais. Este software permite a abertura de uma ampla variedade de formatos gráficos, bem como a conversão deles para o formato bitmap, aceito pelo C++ Builder. Além disso, o IrfanView possui a funcionalidade "linha de comando" que permite ao utilizador automatizar tarefas e realizar conversões sem ter que recorrer à interface gráfica do mesmo. Essa funcionalidade foi utilizada no sistema, tornando o processo de conversão transparente ao usuário final.

Para realizar a conversão apropriada, assim que o usuário clica no botão "Abrir" da janela de diálogo, o sistema analisa a extensão do arquivo. Se a extensão for ".bmp", o sistema faz a abertura da imagem de forma direta. Se a extensão for outra, o sistema faz uma chamada ao software IrfanView pedindo a conversão para o formato bitmap. É informada a localização do arquivo para o IrfanView e a localização do arquivo bitmap a ser gerado é definida como sendo a pasta de trabalho do sistema. O arquivo gerado é temporário, sendo que logo após a sua leitura pelo sistema o mesmo é apagado do disco.

Assim que o arquivo é aberto, o sistema mostra uma miniatura sua no baixo relevo à esquerda da janela, com informações básicas sobre as dimensões da imagem em pixels. Além disso, é criada uma estrutura de dados do tipo matriz com três dimensões, a primeira e segunda de acordo com as dimensões do arquivo de imagem e a terceira com três elementos, cada um representando um canal de cor da imagem. Essa estrutura é utilizada posteriormente para acelerar o processamento da imagem, pois o acesso aos componentes da imagem através dos componentes do C++ Builder é mais lento. Também é criada outra matriz, esta de duas dimensões, utilizando como quantidades de elementos as dimensões da imagem, para ser utilizada posteriormente na obtenção e edição do sinal. Ambas as matrizes são criadas utilizando uma funcionalidade da linguagem C++ conhecida como alocação dinâmica de memória [66]. Essa funcionalidade permite a criação de matrizes e outras estruturas de dados em

tempo de execução, com quantidades variáveis de elementos e dimensões. A Figura 5.11 ilustra o processo de alocação dinâmica em C++ para uma matriz tridimensional.

```
imgOrig = new int**[larg];
for(int i=0;i<larg;i++){
   imgOrig[i]=new int*[alt];
}
for(int i=0;i<larg;i++){
   for(int j=0;j<alt;j++){
      imgOrig[i][j]=new int[3];
   }
}
```

Figura 5.11 – Alocação Dinâmica de uma matriz tridimensional em C++.

É importante ressaltar que após a alocação das matrizes, a matriz tridimensional é preenchida com todos os pixels da imagem, o que dependendo da resolução da imagem pode ser um processo lento, pois o acesso à imagem é feito pelos componentes padrão do C++ Builder.

Com a imagem devidamente aberta no sistema, vários dos elementos da interface se habilitam, tornando possível a sua manipulação e configuração de parâmetros. A Figura 5.12 mostra a interface já com os novos elementos habilitados.

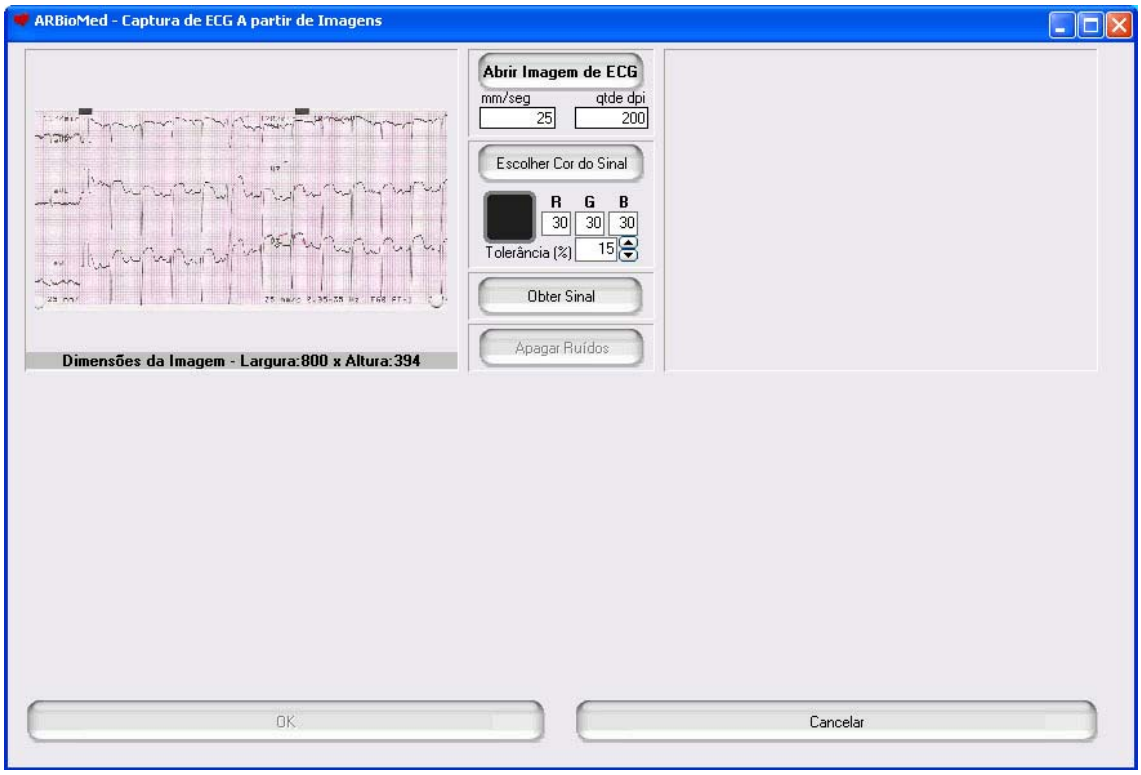

Figura 5.12– Interface após abertura da imagem.

Os componentes estão organizados de cima para baixo, segundo a ordem de seqüência das etapas.

O parâmetro “mm/seg” é uma configuração que estabelece quantos milímetros de gráfico foram capturados por segundo. Em algumas imagens de eletrocardiograma consta esta informação, podendo o usuário utilizá-la. Este parâmetro é muito importante na obtenção da taxa de amostragem do sinal, necessária no momento da simulação. O valor padrão de 25 foi obtido em uma imagem de eletrocardiograma real capturada.

O parâmetro “qtde dpi” indica a resolução utilizada no momento da aquisição da imagem. Essa resolução indica quantos pontos são capturados por polegada real de gráfico e é utilizado em conjunto com o parâmeto “mm/seg” no estabelecimento da taxa de amostragem. Como essa resolução é bastante variável de acordo com o dispositivo de aquisição, ficou estabelecido o valor de 200 dpi como um valor médio, não servindo no entanto como uma base segura de utilização em todos os casos, apenas como um valor padrão a ser configurado apropriadamente.

A taxa de amostragem do sinal é calculada da seguinte maneira:

1. Divide-se o valor do campo “qtde dpi” por 25,4(quantidade de milímetros em 1 polegada) para obter-se a quantidade de pontos efetivamente digitalizados por milímetro da imagem original.
2. Multiplica-se o valor obtido anteriormente pelo valor do campo “mm/seg”. Assim, obtém-se a quantidade existente de pontos ou “amostras” em um segundo de gráfico digitalizado.

A taxa de amostragem calculada nesse caso é altamente dependente da resolução na qual a imagem que contem o gráfico de ECG foi digitalizada, podendo haver perdas de amostras em caso de resoluções muito baixas ou duplicação de amostras com resoluções muito altas. Entre as duas situações, é mais interessante a que gera duplicação de amostras, pois na simulação é feita uma filtragem do sinal que leva em conta todas as amplitudes presentes no gráfico para gerar a visualização, e no caso da perda de amostras, os valores perdidos seriam ignorados, afetando a visualização gerada.

Após a edição dos parâmetros responsáveis pelo cálculo da taxa de amostragem, o próximo passo é identificar de alguma forma o sinal do gráfico em meio ao restante da imagem. Ao observar alguns eletrocardiogramas em papel, é possível notar que a cor do gráfico geralmente é diferente da cor de fundo, ainda que às vezes seja uma diferença sutil. Essa diferença pode ser aproveitada com sucesso na separação entre o gráfico e o restante da imagem. Para tanto, é necessária uma técnica de processamento de imagens que se baseie em detalhes cromáticos para separar detalhes. Essa técnica existe e é

conhecida como Chroma Key [67,68]. A técnica consiste em separar objetos de interesse de um fundo utilizando como elemento de diferenciação a cor do fundo ou do objeto, sendo que para uma boa separação a cor de separação deve ser bastante homogênea. A técnica de Chroma Key tem sido utilizada no cinema há muito anos, pois permite que os personagens sejam filmados em estúdios e posteriormente inseridos em outros cenários, facilitando o trabalho dos criadores de efeitos especiais.

Para separar com sucesso objetos de interesse de um fundo através de técnicas como o Chroma Key, é necessário analisar alguns pontos:

- ✓ Uniformidade da Cor – As cores no mundo real raramente são perfeitamente uniformes. Efeitos indesejados de luz, sombras ou mesmo defeitos injetados pelas lentes das câmeras podem influenciar o resultado final das cores nas imagens capturadas. Até os próprios materiais dos objetos podem apresentar variações indesejadas de cor. Um sistema de processamento de imagem robusto deve ser capaz de lidar com esse tipo de imperfeições, de forma a tratá-las apropriadamente e ainda assim chegar ao resultado desejado;
- ✓ Cor de Recorte – a cor utilizada como recorte não pode estar presente no objeto, caso contrário haverão recortes no mesmo, o que geralmente é indesejado;
- ✓ Tipo de Recorte – o recorte em uma imagem pode ser baseado tanto na cor de fundo predominante quanto na cor do elemento a destacar, de acordo com a situação.

O sistema utiliza uma técnica derivada do Chroma Key, valendo-se da cor do gráfico para o separar do restante da cena. Para lidar com o problema da falta de uniformidade nas cores do gráfico, foi implementado um método de tolerância, onde a cor escolhida como a do gráfico pode variar em um certo percentual antes de ser invalidada. Este percentual pode ser facilmente ajustado, e também observadas as modificações causadas por esse procedimento.

As outras restrições fogem do controle do sistema, visto que este trabalha com exames pré-capturados, onde não houve qualquer controle ou cuidado referente aos requisitos deste sistema. Mesmo assim, algumas ferramentas foram desenvolvidas com o objetivo de prover meios de correção de erros encontrados, o que será visto adiante.

Para escolher a cor do sinal, o usuário pode tanto clicar no botão “Escolher Cor do Sinal” quanto digitar os valores para cada canal de cor nos campos R, G e B. Um quadrado mostra a cor escolhida à medida que o usuário digita os valores e pressiona

Enter. Ao clicar no botão “Escolher Cor do Sinal”, a janela se altera, mostrando a imagem em seu tamanho natural e mostrando barras de rolagem para navegação caso a imagem seja muito grande. Nesse modo, o cursor torna-se uma cruz sendo acompanhado pelos campos com os valores dos canais de cor bem como pelo quadrado de pré-visualização da cor. Quando o usuário movimenta o cursor, os valores dos canais de cor e o quadrado se alteram de acordo com a cor do pixel na imagem sobre a qual o centro do cursor estiver. Isso permite ao usuário escolher qualquer tonalidade presente na imagem simplesmente posicionando o cursor sobre ela. Quando a cor desejada é encontrada, o usuário pode clicar com o botão esquerdo do mouse finalizando a escolha e retornando a janela a seu estado anterior. A Figura 5.13 exemplifica a seleção de uma cor através do botão “Escolher Cor do Sinal”.

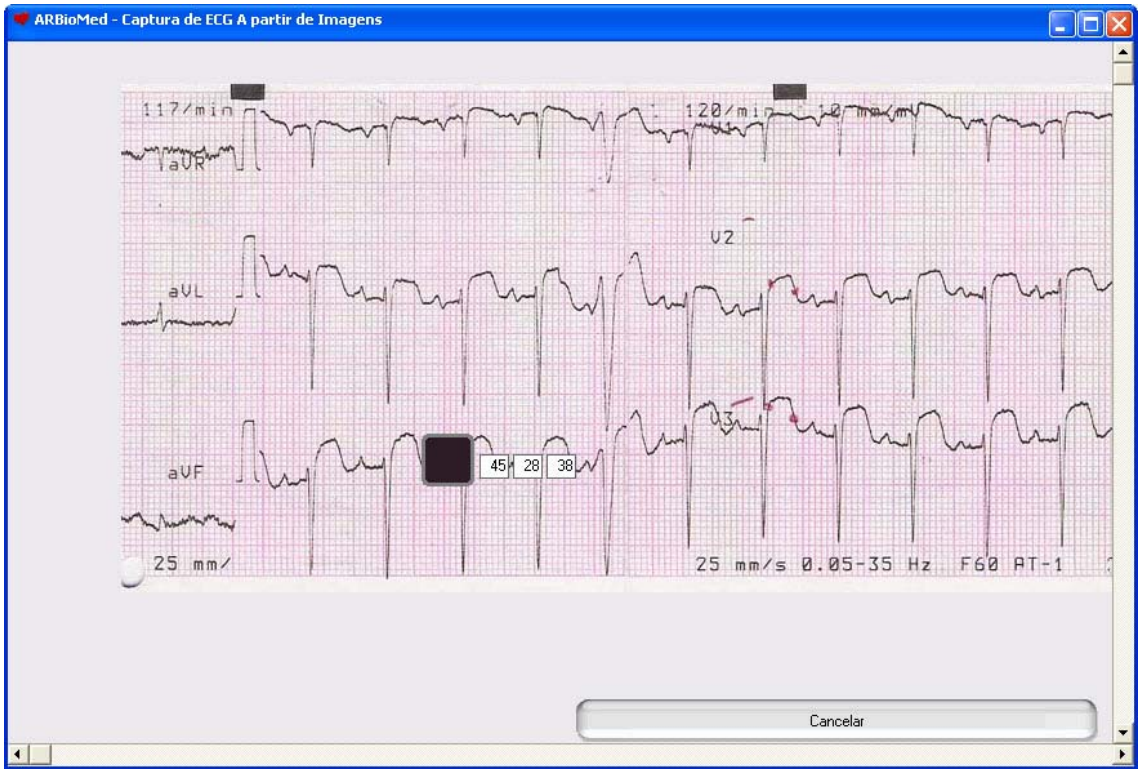

Figura 5.13 – Seleção de cor pelo botão “Escolher Cor do Sinal”.

Caso o usuário desista da seleção, o botão Cancelar também pode ser utilizado para retornar à janela anterior sem alterar os parâmetros de cor do sinal.

Após a seleção da cor do sinal, ainda é possível alterá-la através dos campos de cada canal de cor. Nesse momento, todos os passos necessários para processar a imagem já foram efetuados, assim o usuário pode clicar no botão “Obter Sinal”. Ao fazer isso, após alguns instantes de processamento, o quadro em baixo relevo à direita superior da janela se altera, apresentando uma prévia do sinal detectado, baseada nas

configurações escolhidas. Nesta prévia, os pontos pretos representam os pixels descartados e os pontos brancos representam os pixels reconhecidos como sinal. Também é exibida uma prévia do gráfico ECG obtido, na parte inferior da janela.

Nesse momento já é possível pressionar o botão “OK” e retornar à interface inicial do sistema, porém na maioria das situações é necessária ainda alguma manipulação do gráfico para obtenção de bons resultados. A Figura 5.14 mostra o estado da janela quando o botão “Obter Sinal” foi pressionado.

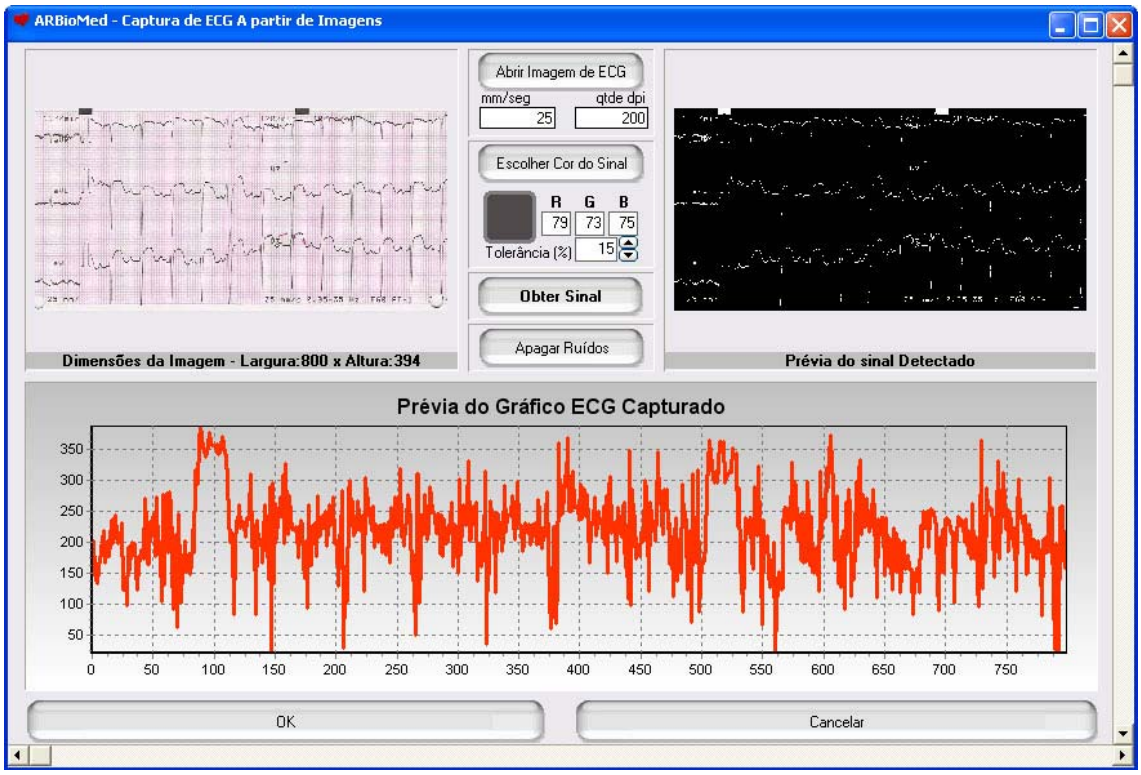

Figura 5.14 – Estado da Janela após pressionamento do botão “Obter Sinal”.

Quando o sinal capturado não apresenta um resultado satisfatório, como mostrado na Figura 5.16, uma das soluções possíveis é ajustar o percentual de tolerância quanto à cor escolhida. Esse percentual por padrão se encontra em 15%, valor definido como razoável para a maioria das imagens utilizadas nos testes. O ajuste da tolerância para menos pode ajudar na retirada de ruídos que talvez tenham sido reconhecidos como sinal e para mais pode ajudar a reduzir as falhas no reconhecimento de sinais com mais variação de cor. Em ambas as situações, é preciso um novo clique sobre o botão “Obter Sinal” para se visualizar a diferença nos resultados.

Mesmo com o ajuste de tolerância sobre a cor do sinal, pode ocorrer o reconhecimento de ruídos, por motivos diversos. Assim, seria interessante o sistema permitir a edição do sinal detectado, apagando os pixels erroneamente detectados. Esta

funcionalidade foi implementada através do botão “Apagar ruídos”. No momento em que o usuário clica sobre este botão, a janela se altera de forma parecida com o que acontece no botão “Escolher Cor do Sinal”, mostrando em tamanho natural a prévia do sinal detectado juntamente com barras de rolagem para navegação em imagens maiores. Além disso, o cursor se altera para um círculo com borda verde e preenchimento preto. Este círculo é, na verdade, um apagador que permite ao usuário retirar as áreas indesejadas do sinal, clicando sobre elas. Quando é detectado o clique e o arrastar do mouse, os pontos brancos na área onde se encontra o cursor são apagados, ou seja, convertidos para preto. Adicionalmente, a cor de borda do cursor muda de verde para vermelho, indicando que a função apagar está em execução. O tamanho do círculo apagador pode ser controlado através do campo “Raio do Pincel” por digitação ou pelas setas que aumentam ou diminuem o raio em 1 unidade. Também estão presentes os botões “OK” e “Cancelar”, sendo que o “OK” valida as transformações efetuadas sobre a imagem e retorna a janela a seu estado anterior, ao passo que o botão “Cancelar” ignora as transformações na imagem simplesmente retornando a janela a seu estado anterior.

A captura de ECG a partir de imagens implementada no sistema possui uma restrição de processamento de apenas uma linha de gráfico por vez. Assim, imagens que contenham 2 ou mais linhas de gráfico precisam ser previamente separadas em algum software de edição de imagem, ou então apagadas as linhas extra utilizando a ferramenta apagador do sistema. Para possibilitar esse tipo de edição, o sistema utiliza o pixel mais inferior e o mais superior de sinal detectado para estabelecer os valores máximos e mínimos no gráfico reconhecido.

Outro fator importante a ser considerado são as falhas de continuidade nos gráficos reconhecidos. Quando ocorre uma falha, o sistema faz um preenchimento da falha criando uma interpolação linear entre o último valor reconhecido antes da falha até o próximo pixel do sinal que marca o fim da falha. Apesar de simples, esta técnica combinada à tolerância de cor resulta em gráficos contínuos e com poucas discrepâncias, mesmo porque as falhas geralmente encontradas são de poucos pixels, não influenciando de forma tão proeminente o gráfico final obtido. A Figura 5.15 exemplifica o processo de apagar ruídos do gráfico ECG.

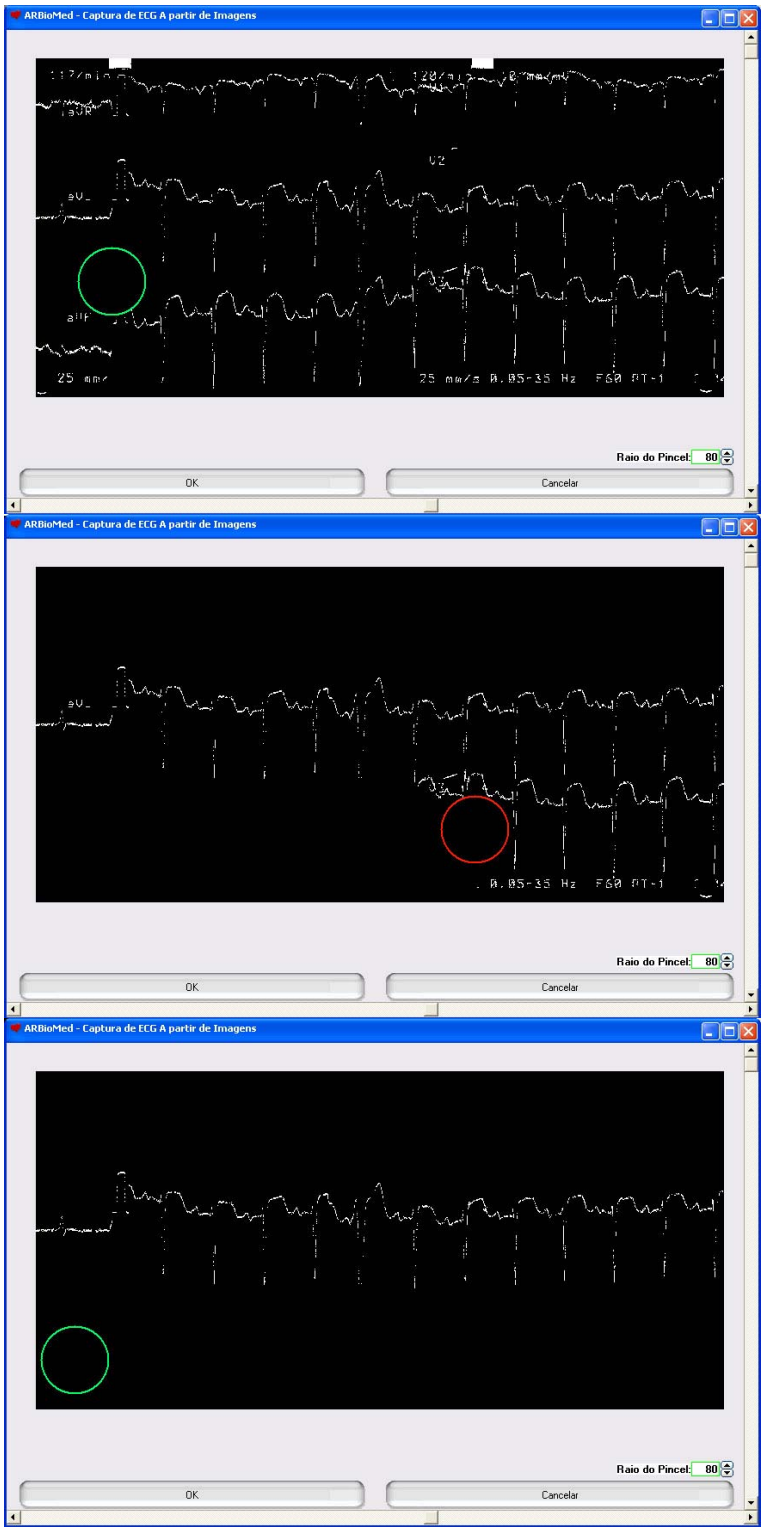

Figura 5.15 – Processo de Apagar Ruídos do gráfico ECG.

Ao clicar no botão “OK”, a janela volta a seu estado anterior, com a exibição da prévia já atualizada do sinal detectado e mostra também a visualização do gráfico ECG reconhecido. A Figura 5.16 apresenta a janela com as informações atualizadas.

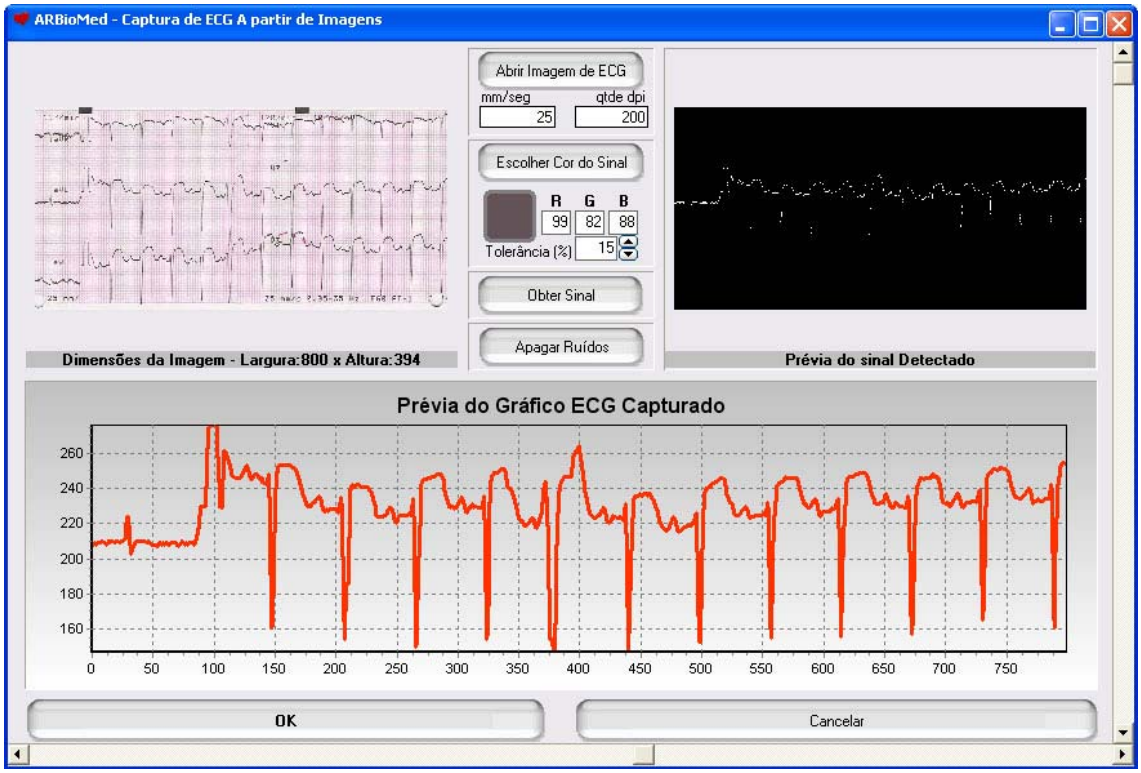

Figura 5.16 – Janela com as informações atualizadas.

Como pode ser visto, o gráfico foi reconhecido de forma satisfatória pelo sistema, não apresentando discrepâncias ou falhas. Assim, pode-se clicar no botão “OK” para retornar à interface inicial do sistema.

Neste momento o gráfico ECG já se encontra completamente processado e armazenado em uma estrutura de dados do tipo vetor. A taxa de amostragem também foi calculada e armazenada, portanto é possível realizar a simulação. Adicionalmente, foi criado um botão “Salvar Gráfico” que permite ao usuário salvar o gráfico reconhecido em formato texto, passível inclusive de simulação posterior com a opção Buscar ECG de Arquivo. Quando o usuário clica sobre o botão “Simular”, o sistema envia ao módulo GVRA os dados de opção, taxa de amostragem e um caminho para o gráfico ECG reconhecido. O gráfico é salvo em um arquivo texto temporário, contendo as amplitudes do sinal reconhecidas na imagem, sendo bastante similar aos arquivos de ECG da opção Buscar ECG em Arquivo. A Figura 5.17 mostra a interface final da opção Capturar ECG de Imagem.

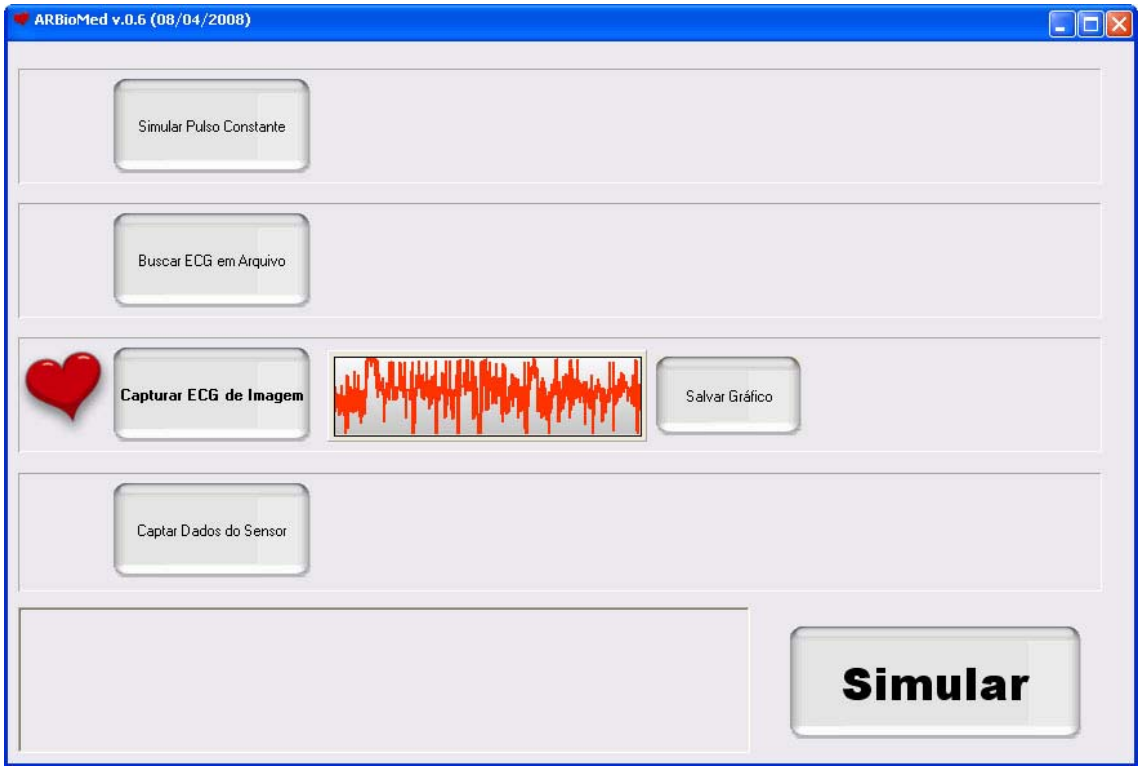

Figura 5.17 - Interface final da opção Capturar ECG de Imagem.

A tarefa de apagar os ruídos é bastante dependente da ação do usuário, pois não é trivial a elaboração de um algoritmo capaz de ignorar todo o tipo de ruídos existentes na imagem, visto que nem todos eles se apresentam de maneira padronizada em todas as situações. Assim, ruídos aleatórios como marcas à caneta escritas pelo médico na mesma cor do sinal e mesmo algumas marcações dos papéis utilizados nos exames exigem a ação do usuário para o correto reconhecimento do sinal.

5.2.5 Opção Captar Dados do Sensor

Um dos principais objetivos do sistema é a simulação em tempo real do batimento cardíaco, através de sensores apropriados. A opção Captar Dados do Sensor tem poucas configurações, servindo basicamente para definir o tipo da simulação, monitorar o funcionamento do sensor e fazer medições aproximadas do pulso cardíaco.

A Figura 5.18 mostra a interface inicial da opção Captar Dados do Sensor.

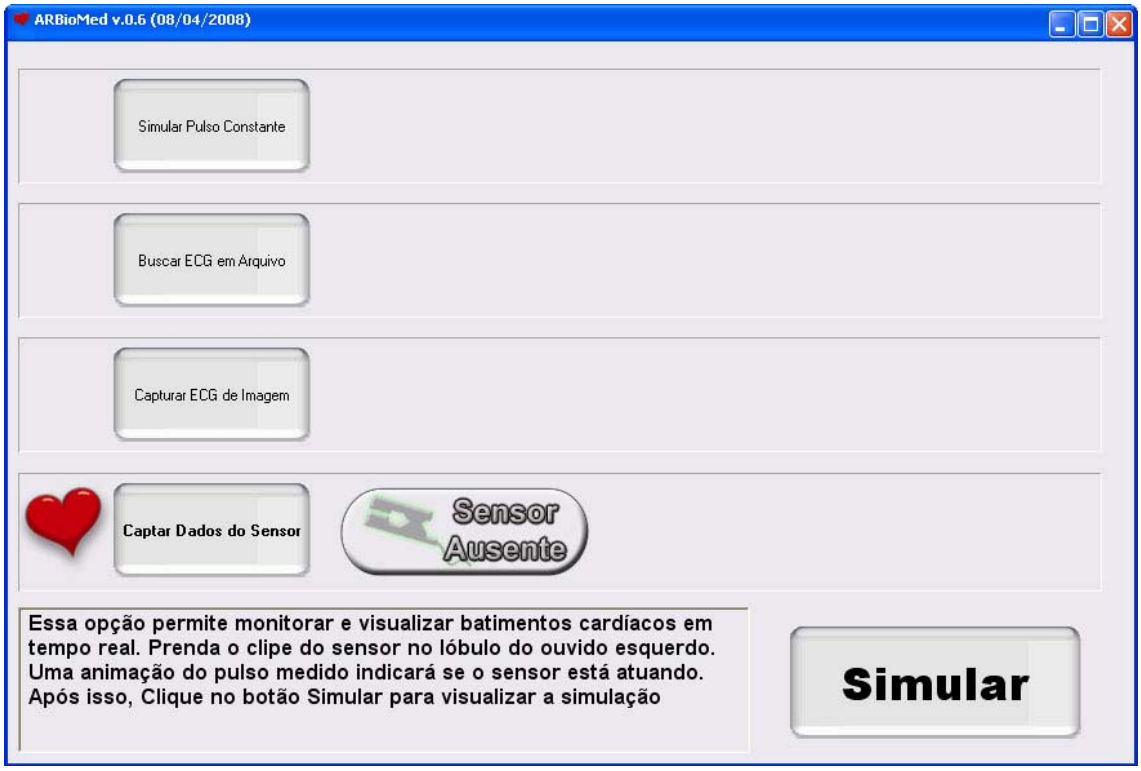

Figura 5.18 – Interface Inicial da opção Captar Dados do Sensor.

Inicialmente, a interface da opção mostra apenas um ícone com a mensagem de "Sensor Ausente". Esse ícone possui outro estado, chamado "Sensor Presente" que aparece somente quando é detectada atividade do sensor. É possível notar que o quadro explicativo fornece algumas informações de como proceder quando à utilização do sensor. Essas informações referem-se ao sensor adaptado utilizado no sistema.

O sensor é composto de 2 partes, o sensor de luminosidade que é acoplado ao lóbulo da orelha do usuário, e o dispositivo que filtra o sinal recebido do sensor e fornece a visualização do mesmo em formato digital. Como se trata de um sensor de bicicleta ergométrica, foi necessária uma adaptação eletrônica no mesmo para que pudesse fornecer dados à porta paralela. Inicialmente, foi utilizado um acoplador óptico [69] diretamente sobre o circuito amplificador do sinal para ao mesmo tempo proteger a porta paralela e digitalizar o sinal. Porém, isso gerou uma queda de corrente muito grande que prejudicou o funcionamento do sensor. Assim, foi feito um pequeno amplificador em separado para o sinal do acoplador óptico, baseado num transistor bc548c. Dessa forma, foi possível manter o funcionamento normal do sensor e ao mesmo tempo digitalizar o sinal e enviá-lo para uma das entradas da porta paralela. Com o objetivo de facilitar a montagem e desmontagem do sensor, foi utilizado um

conector do tipo P2 macho no sensor e um conector P2 fêmea na entrada da porta paralela.

A Figura 5.19 apresenta o sensor utilizado e a figura 5.20 exemplifica o esquema de funcionamento do circuito de amplificação desenvolvido.

Figura 5.19 – Sensor utilizado no sistema.

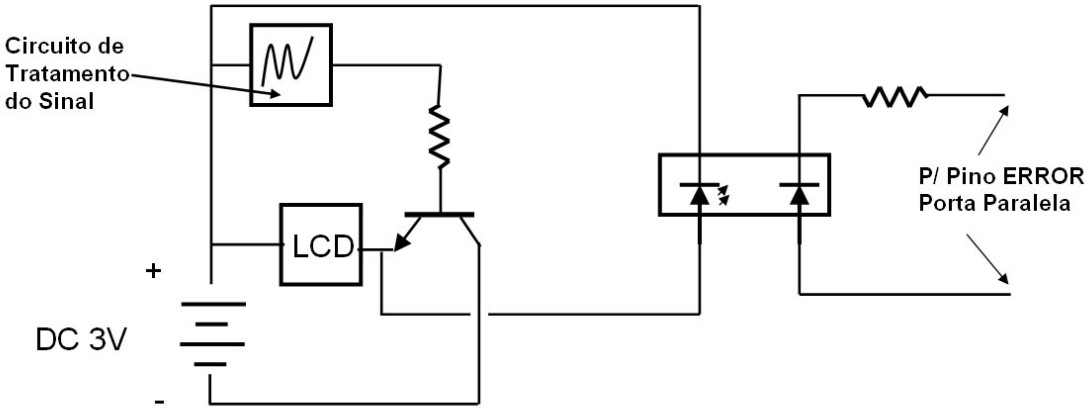

Figura 5.20 – Esquema de funcionamento do circuito de amplificação desenvolvido.

A interface da opção Captar Dados do Sensor fica constantemente monitorando a porta paralela em busca de alterações no seu estado. Para realizar esse monitoramento foi utilizada a biblioteca de vínculo dinâmico InpOut32 [70] em conjunto com os

recursos da linguagem C++. Essa biblioteca possui funções para entrada e saída na porta paralela. A função de entrada utilizada retorna o estado lógico de cada entrada da porta.

Quando o estado lógico da entrada utilizada se altera, o sistema reconhece a mudança e imediatamente muda o estado do ícone “Sensor ausente” para “Sensor Presente”. Também é exibido um ícone sinalizador do pulso cardíaco cada vez que um novo pulso é recebido. Ao lado deste ícone surge uma medição aproximada do pulso, em batidas por minuto. Esta medição é atualizada periodicamente, refletindo as mudanças ocorridas. Para se evitar mudanças muito bruscas no pulso foi utilizado um algoritmo de média sobre os valores recebidos. A Figura 5.21 mostra a interface com o sensor em funcionamento.

Figura 5.21 – Interface com sensor em funcionamento.

Após monitorar o funcionamento do sensor e acoplá-lo ao lóbulo esquerdo na orelha do sujeito, pode-se iniciar a simulação normalmente, clicando no botão Simular. No momento em que isso é feito, a medição de pulso pela interface é interrompida na interface de operação, com o objetivo de não sobrecarregar o acesso à porta paralela, deixando-o livre para a simulação. Assim como nas demais opções, se o módulo GVRA estiver gerando simulações para outras opções, ele se atualizará quase que instantaneamente refletindo a mudança de opção na interface.

5.3. Gerador da visualização em Realidade Aumentada (GVRA)

Após todas as configurações e opções serem realizadas na interface de operação, entra em cena o módulo gerador da visualização em Realidade Aumentada. Este módulo utiliza os parâmetros recebidos da interface de operação em conjunto com os dados visuais captados pela câmera e, dependendo da opção escolhida, os dados recebidos do sensor de pulso para produzir a visualização final da simulação, contendo o vídeo da câmera adicionado de elementos virtuais, caracterizando assim o ambiente de Realidade Aumentada.

Para gerar esta visualização, são necessários alguns requisitos adicionais não tratados no módulo da interface de operação, como os modelos virtuais utilizados, as animações realizadas pelos modelos e todos os comportamentos e ações utilizados no ambiente de Realidade Aumentada. A seguir são explicados os principais conceitos relacionados a esses aspectos.

5.3.1 Objetos Virtuais

Em um sistema de Realidade Aumentada, a escolha dos objetos virtuais é extremamente importante, pois através destes que a simulação se realiza. Assim, características como modelagem, texturas e iluminação devem ser consideradas. Algumas dessas características são detalhadas no Anexo 1.

O sistema foi projetado para utilizar poucos objetos virtuais, pois trata da simulação do batimento cardíaco, envolvendo assim basicamente um modelo virtual para o coração. Além do modelo virtual cardíaco, também estão presentes mais 2 objetos virtuais, utilizados para delimitar o tamanho da caixa torácica do sujeito e, consequentemente, redimensionar o coração virtual de acordo com as medidas obtidas.

O modelo virtual do coração foi adaptado de uma versão em VRML existente. A Figura 5.22 apresenta o modelo original com um visualizador de arquivos VRML.

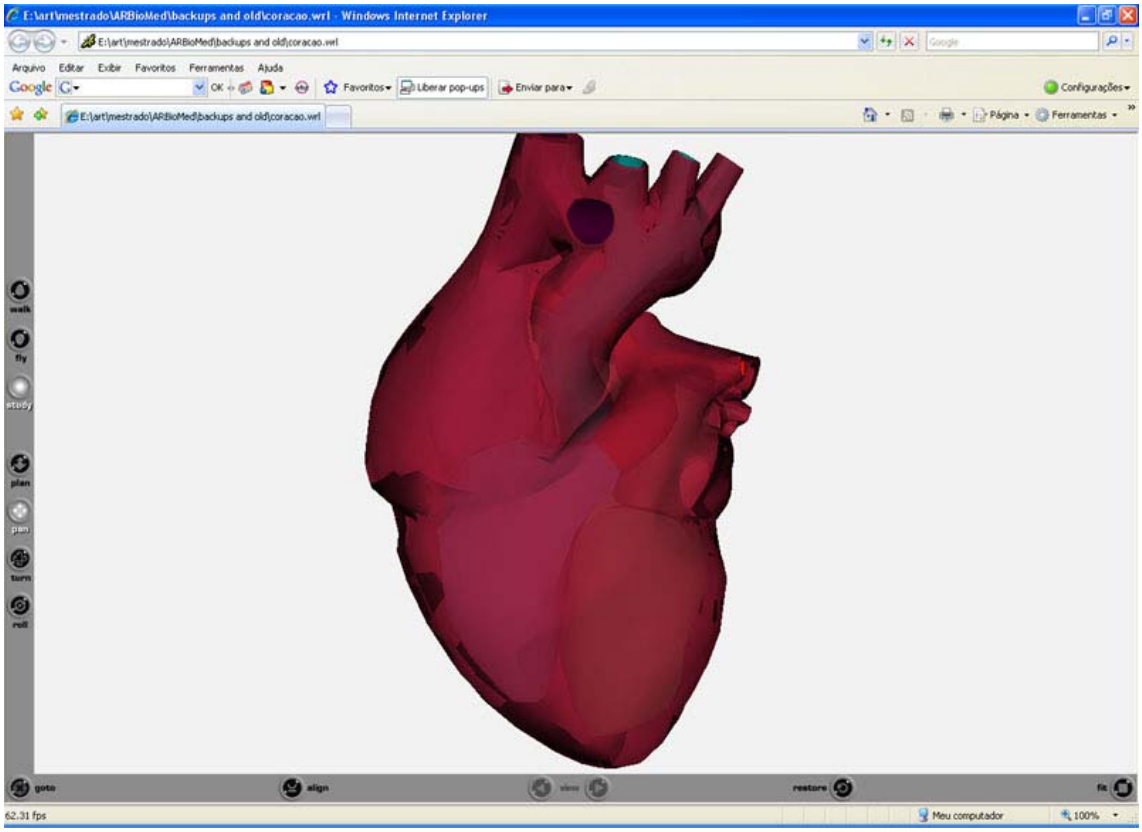

Figura 5.22 – Modelo Original do Coração Virtual.

O modelo original continha praticamente todas as estruturas anatômicas do coração modeladas, o que aumentava consideravelmente a sua complexidade geométrica. Para editá-lo foi feita a sua importação no software 3ds Max® e retirados os submodelos desnecessários à simulação feita pelo sistema. Mesmo assim, a contagem de faces no modelo continuou bastante alta, o que poderia prejudicar o desempenho da simulação, principalmente em computadores de menor capacidade. O 3ds Max® tem um conjunto de ferramentas que permite ao usuário manipular os elementos da geometria dos modelos, reduzindo ou aumentando sua complexidade, de acordo com a necessidade. Com a utilização dessas ferramentas foi possível reduzir a complexidade no modelo de 7270 para 980 faces poligonais, sem, contudo, acarretar muita distorção no formato geométrico do mesmo. A Figura 5.23 mostra uma comparação entre os dois modelos.

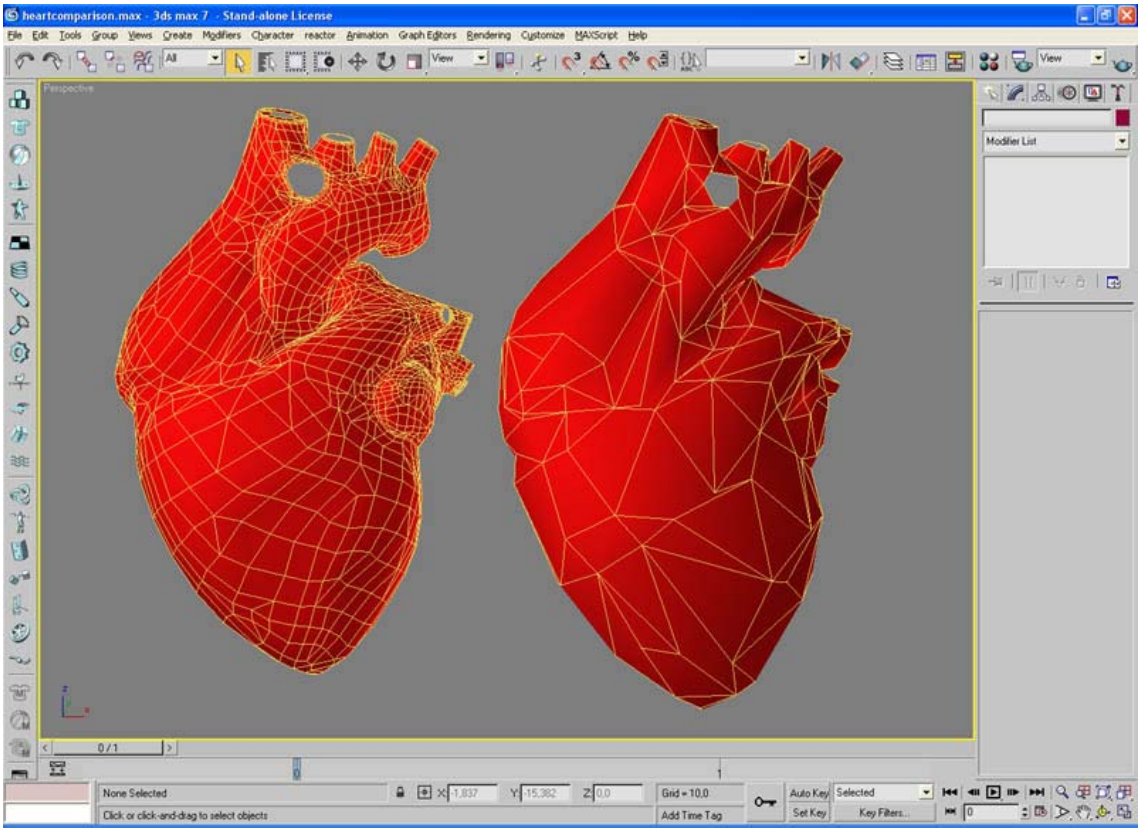

Figura 5.23 – Comparação entre o modelo original do coração virtual com o editado.

Além da geometria excessivamente carregada, a aparência final do coração virtual utilizado estava muito simples, utilizando apenas uma cor sólida como material. Com o objetivo de tornar o modelo mais realista, foi utilizada uma textura de tecido muscular para envolver todo o modelo. Para que a textura fosse aplicada de forma homogênea sobre o modelo, técnicas de mapeamento foram necessárias. Um outro aspecto bastante explorado em jogos e aplicações de Realidade Virtual é a iluminação e sombreamento. Como os recursos neste quesito são limitados, é comum a utilização de uma técnica onde as sombras e efeitos de iluminação como alto brilho e realce especular são misturados à textura original e reaplicados ao objeto. O 3ds Max® possui essa funcionalidade, por isso ela foi utilizada em conjunto com a textura original. A Figura 5.24 apresenta a textura já editada, juntamente com o coração virtual com e sem a textura para uma melhor comparação.

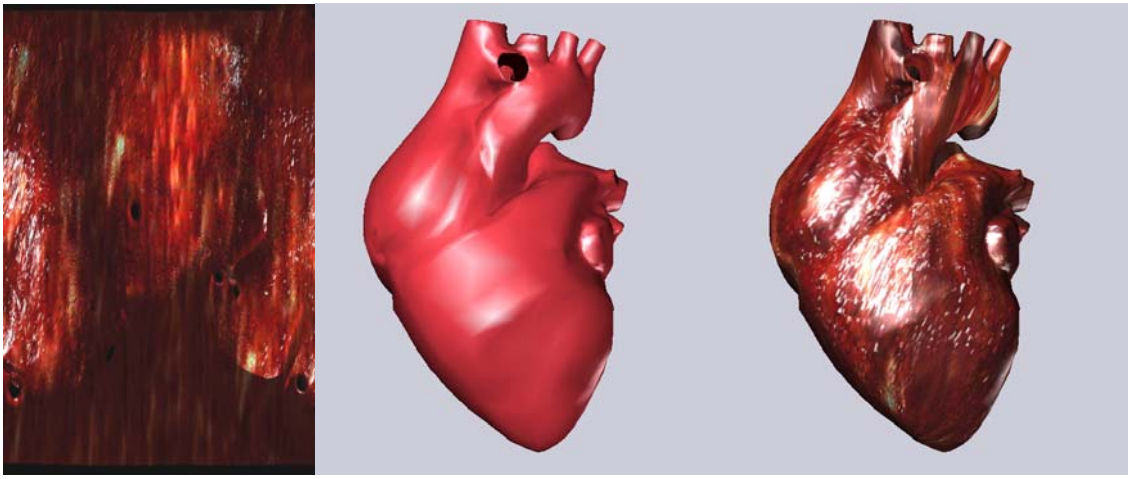

Figura 5.24 – Textura editada e corações virtuais com e sem textura.

Além dos aspectos visuais, o coração virtual também precisa apresentar o comportamento de batimento cardíaco, através de animações apropriadas. A princípio, a forma de animação utilizada foi a escala do órgão como um todo. A animação nesse modo é calculada já no ambiente de Realidade Aumentada, não sendo necessárias portanto, quaisquer modificações no modelo do coração.

A vantagem da animação de escala é que o modelo final gerado é pequeno, ideal para computadores com pouca memória e capacidades gráficas. Porém, essa forma de animação foge bastante do comportamento real do órgão, que expande e contrai a sua musculatura a cada ciclo cardíaco. Assim, foi criada uma forma alternativa de animação, onde a geometria do coração é realmente alterada, caracterizando o ciclo cardíaco de forma mais realista.

O grande problema com a animação dos sub-elementos dos modelos (vértices, arestas e faces) é a sobrecarga nos recursos do computador, visto que há a necessidade de interpolação das coordenadas de cada vértice do modelo a cada quadro da animação. Quanto mais vértices tiver o modelo, mais força computacional é necessária. Apesar do modelo utilizado não apresentar uma quantidade elevada de faces para os padrões atuais, em um ambiente de Realidade Aumentada perde-se muito do poder de processamento em tarefas como o processamento de cada quadro de vídeo e o reconhecimento das marcas.

Outra complicação é que o ARToolKit, a biblioteca de Realidade Aumentada utilizada neste projeto não permite acesso aos sub-elementos dos modelos virtuais utilizados quando estes estiverem descritos como modelos VRML, o que é o caso da simulação do sistema. Essa restrição torna complexo o processo de controle da

animação, e este controle é absolutamente necessário na simulação, pois em cada uma das opções, o ciclo cardíaco precisa ter sua velocidade e ritmo controlados a todo momento.

Uma forma de resolver os problemas supracitados, seria o desenvolvimento de uma rotina para carregamento e animação de modelos via OpenGL dentro do ARToolKit, o que é possível, uma vez que o OpenGl é suportado pela biblioteca.

No entanto, uma outra alternativa de implementação mais simples foi utilizada, utilizando os recursos do próprio ARToolKit.

Ao examinar a arquitetura do ARToolKit, pode-se ver que cada objeto virtual é, em primeira instância, associado a uma marca virtual específica. Porém, é possível facilmente alterar esta característica e associar múltiplos modelos a um único marcador. Também é possível controlar a visibilidade dos modelos, a cada quadro da animação. Assim, alternando a visibilidade de modelos num marcador, pode-se criar a ilusão de animação. Pode-se citar como exemplo a animação de uma bola murchando, criada a partir de algumas poses intermediárias entre a bola cheia e a bola completamente murcha. A Figura 5.25 ilustra os modelos intermediários do exemplo dado.

Figura 5.25 – Poses de uma animação.

Cada quadro da animação é salvo num modelo virtual. Em seguida, todos os modelos da animação são associados a um marcador no ARToolKit. Por fim, para se criar a ilusão de animação, a cada quadro de vídeo gerado, apenas um dos modelos fica visível enquanto todos os outros da sequência são desativados. No próximo quadro, o modelo subseqüente fica visível e todos os outros são desativados.

O processo de ativação dos modelos é de controle bastante simples no ARToolKit, pois para cada modelo carregado é associada uma identificação única através da função “read_VRMLdata”, e esta identificação é criada de maneira a permitir seu acesso com índices numéricos, numa espécie de vetor de objetos virtuais. A identificação permite não só o controle de visibilidade, mas também outras características como a escala e translação sobre o marcador.

A técnica de animação adotada possui as vantagens de não sobrecarregar a execução da aplicação com a interpolação de sub-elementos do modelo virtual e

fornecer amplo controle sobre a ordem e tempo de execução de cada pose do modelo. As desvantagens são de que os modelos precisam ser todos carregados ao início da aplicação, o que dependendo da quantidade de modelos pode ser um processo lento. Animações com muitos quadros podem ser impossibilitadas de serem utilizadas com esta técnica.

Para o modelo virtual do coração, foram criadas 20 poses representando o ciclo ativo do coração. Nestas poses é feita a deformação da malha correspondente aos músculos cardíacos dos átrios e ventrículos. O período de descanso onde o coração se prepara para um novo ciclo é criado simplesmente parando a animação no último quadro existente durante todo o período do descanso. No próximo ciclo, a animação recomeça normalmente, a partir do primeiro quadro.

Além do modelo do coração, foram criados mais dois modelos virtuais, usados nos marcadores de captura das dimensões da caixa torácica. Estes modelos são apenas setas indicativas de direção coloridas. A Figura 5.26 apresenta esses modelos.

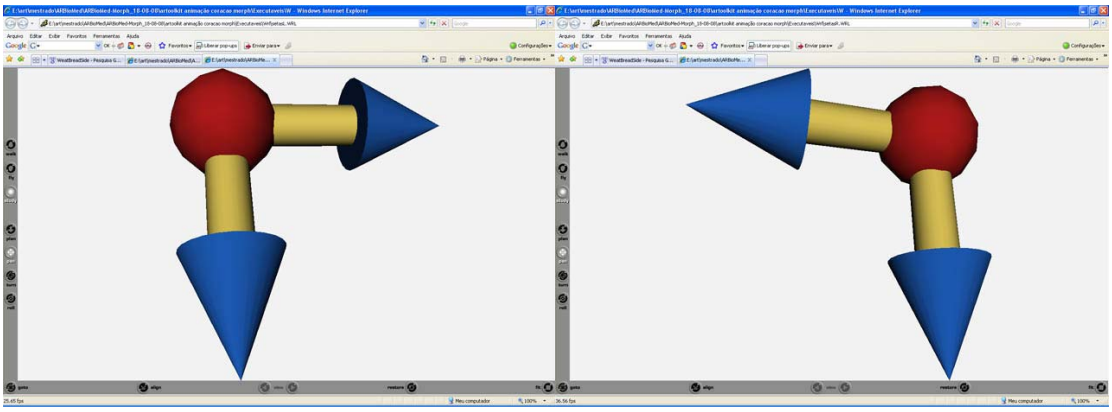

Figura 5.26 – Modelos virtuais adicionais utilizados no sistema.

5.3.2 Ações e Comportamentos no ambiente de Realidade Aumentada.

A interação no ambiente de Realidade Aumentada implementado com marcadores pode ser feita de formas variadas. Algumas aplicações evitam ao máximo a utilização de dispositivos tradicionais de interação como mouse e teclado, fazendo da manipulação de marcadores a única forma de interação. Mas também existem aplicações que utilizam bastante esses dispositivos, em conjunto com os marcadores. O sistema está caracterizado por utilizar tanto marcadores como dispositivos tradicionais na sua operação.

No sistema, os marcadores agregam as funções de posicionamento do coração virtual e delimitação do tamanho da caixa torácica. Ao definir as figuras identificadoras dos marcadores, foi dispensada alguma atenção no sentido de facilitar os desenhos das

marcas às suas funções no sistema. O marcador do coração utiliza uma forma de coração como símbolo base e um pequeno chanfro no canto do marcador. Os marcadores delimitadores das dimensões torácicas utilizam duas letras de abreviatura e também uma seta indicando o canto a que pertencem. As letras foram escolhidas com base nas siglas em inglês, sendo UL (Up Left) para o delimitador superior esquerdo e DR (Down Right) para o delimitador inferior direito. Os delimitadores complementares são criados pelo módulo GVRA por espelhamento. Além disso, para facilitar a utilização, o marcador do coração virtual foi confeccionado em forma de crachá, com tamanho de corda regulável. Assim, o usuário não precisa ficar o tempo todo segurando o marcador junto ao peito. Os marcadores delimitadores das dimensões torácicas foram feitos da forma comum, em papel branco com revestimento ao fundo de papel cartão, para conferir rigidez ao marcador. Esses marcadores são utilizados somente no início da simulação, assim não é necessária nenhuma forma de fixação ao corpo do sujeito. A Figura 5.27 apresenta os marcadores utilizados já confeccionados.

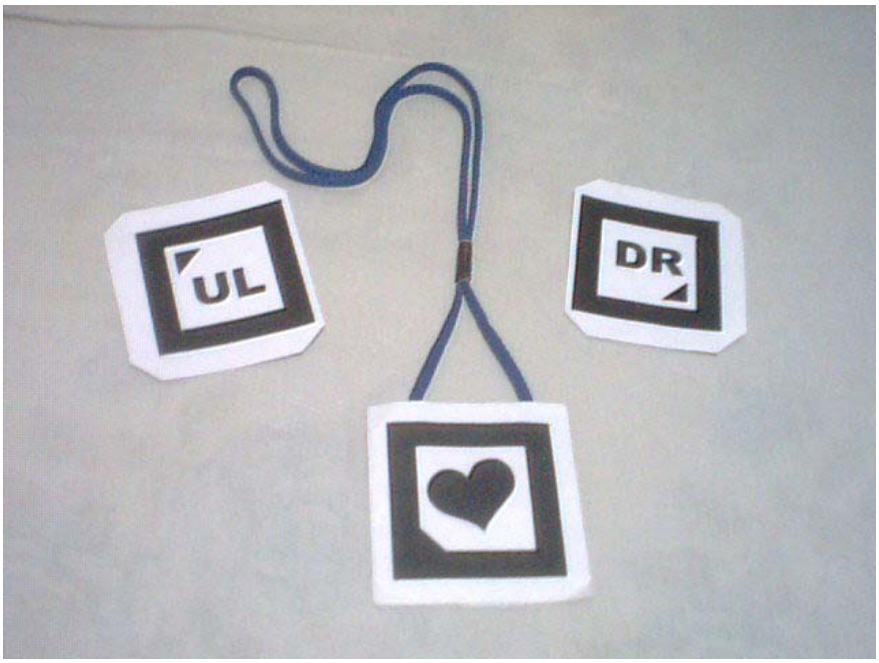

Figura 5.27 – Marcadores utilizados no sistema.

Como foi dito, o marcador do coração está presente durante toda a simulação, sendo para tanto utilizado como uma espécie de crachá. O coração virtual aparece no centro do marcador.

Os marcadores UL e DR são utilizados apenas uma vez, no início da simulação para capturar as medidas da caixa torácica do sujeito. Uma vez estabelecidas estas medidas, os marcadores podem ser guardados. O marcador UL é colocado sobre o

ombro esquerdo do sujeito e o marcador DR é colocado no encontro do abdome com a bacia, no lado direito do corpo. O sistema mostra os objetos virtuais sobre os marcadores quando os está reconhecendo. O calculo das dimensões torácicas é feito levando-se em conta as distâncias entre os marcadores, nos eixos x e y. Foi atribuído maior peso no cálculo para o eixo x, que determina a largura do tórax. Quando há o correto dimensionamento do coração virtual em relação ao tamanho da caixa torácica, pode-se fazer a oclusão de um dos dois marcadores para finalizar a medição, pois o sistema só faz as medições quando os dois marcadores estão sendo reconhecidos. A Figura 5.28 ilustra duas situações de redimensionamento do coração.

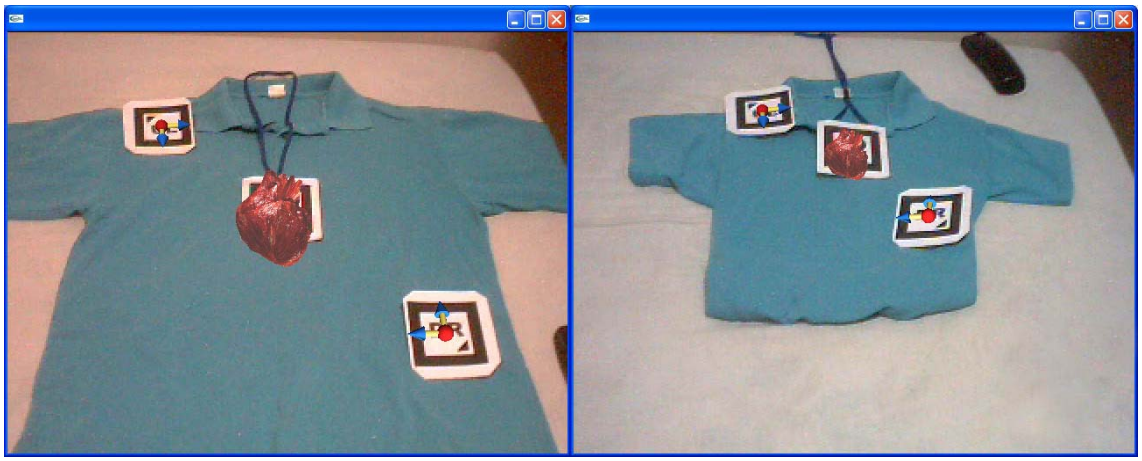

Figura 5.28 – Redimensionamento do coração virtual.

Como pode ser visto, o coração virtual é redimensionado automaticamente de acordo com as medidas torácicas do usuário. No entanto, o redimensionamento efetuado até o momento não é baseado em estudos fisiológicos, somente funcionando como melhoria estética das proporções no órgão virtual. A animação não é afetada por esta função, pois após o cálculo da animação é refeita a escala do órgão. Assim, a qualquer momento podem ser feitos ajustes no dimensionamento do coração virtual, caso haja necessidade.

Além destes comportamentos, é possível manusear o marcador do coração, girando-o ou colocando-o mais perto da câmera para inspecionar detalhes em sua geometria ou da animação.

5.3.3 Comunicação com a interface de operação

O sistema permite ao usuário realizar diversos tipos de simulação, sem que para isso precise reinicializar o sistema. Assim, o módulo GVRA pode inicialmente processar dados da opção pulso constante e logo após receber informações para

atualizar a simulação com dados de uma imagem processada na interface de operação. Para acompanhar esse processo, foi definido que a comunicação com a interface de operação se daria através de arquivos-texto. A primeira utilização destes arquivos é no momento da inicialização da simulação, quando o arquivo config.ini é lido para se saber os parâmetros iniciais da simulação. Neste arquivo é lida a opção escolhida pelo usuário, bem como seus parâmetros. A leitura da opção serve para redirecionar o módulo GVRA para uma de suas diversas funções, cada qual adaptada a uma opção da interface de operação.

Quando a simulação já está sendo executada, o módulo GVRA continua monitorando periodicamente um outro arquivo chamado atualiza.ini. Este arquivo fica normalmente com o valor ajustado para 0. Quando o usuário faz alguma alteração na interface de operação e pressiona o botão “Atualizar”, o conteúdo do arquivo é alterado para 1. Na próxima leitura do arquivo pelo módulo GVRA, ele detecta uma alteração, altera o arquivo para 0 novamente e lê o arquivo config.ini em busca das alterações. Essa política de leituras periódicas foi implantada visando reduzir o acesso desnecessário a disco, pois ler todo o arquivo config.ini e compará-lo ao seu estado anterior seria bastante trabalhoso e geraria mais acesso ao disco. O intervalo de monitoramento do arquivo atualiza.ini foi definido com base na quantidade de quadros por segundo que o computador consegue gerar, não chegando a 2 segundos de animação. Assim qualquer atualização terá um retardo máximo de 2 segundos para poder ser visualizada na simulação.

A comunicação por arquivos foi escolhida pelo motivo da simplicidade na sua implementação face a soluções mais arrojadas como threads e sockets, e por permitir fácil visualização das mensagens pela simples abertura dos arquivos. É possível inclusive corrigir manualmente alguns parâmetros caso haja necessidade e o usuário tenha conhecimento avançado do funcionamento do sistema.

5.3.4 Tratamento da opção Simular Pulso Constante

A opção Simular Pulso Constante gera apenas um parâmetro para o módulo GVRA, a quantidade de batidas por minuto. Como um minuto é um tempo bastante grande em termos de Realidade Virtual, foi escolhido o segundo como unidade de tempo para a geração das animações.

Cada computador tem suas características físicas como processador, memória e placa de vídeo, e de software, como sistema operacional, versões diversas de aplicativos, softwares antivírus e outros. Assim, é impossível fazer previsões de

performance com base nas características do computador. No entanto, para gerar uma animação fiel às características escolhidas, neste caso a quantidade de batidas por minuto, é necessário monitorar a performance gráfica do computador, de alguma forma. Felizmente, o ARToolKit tem uma função bastante útil para contagem de tempo que permite ao desenvolvedor saber a quantidade de quadros por segundo que o computador está gerando, criando-se um contador de quadros no laço principal e periodicamente dividindo o valor desta variável pela quantidade de tempo. Sabendo-se a quantidade de quadros gerada por segundo no computador, podem ser feitos ajustes na animação de forma a corrigir problemas de performance e dar como saída resultados fiéis aos parâmetros recebidos.

A animação simples do coração foi definida como a escala do mesmo partindo de 90% do seu tamanho total, indo até 100% e voltando a 90%. Esse movimento representa o período ativo do ciclo cardíaco, onde o coração contrai seus músculos para bombear o sangue. O período de relaxamento do ciclo é criado simplesmente mantendo a escala do coração virtual em 90% até o próximo ciclo.

Supondo que o usuário deixou o valor de batidas por minuto na sua configuração padrão, que é 72 batidas por minuto, pode-se dividir esse valor pela quantidade de segundos em um minuto para saber quantas batidas por segundo o coração realizará. Neste caso o valor obtido seria 1,2 batidas por minuto (72/60). Isso significa que em um segundo de animação precisa ser simulado um ciclo completo e mais 20% de um outro ciclo, compreendendo aí a parte ativa e a parte inativa do ciclo. Como a escala é feita em 10% para ir até a escala máxima e -10% para voltar ao valor de repouso, pode-se somar estas porcentagens e conseguir 20% de movimento de escala do coração no período ativo do ciclo. Sendo que o período de relaxamento do ciclo tem quase a mesma duração do período ativo em condições normais [65], foi decidido que a animação teria tempos iguais para ambos os períodos.

Assumindo a taxa hipotética de 30 quadros por segundo de animação, isso significa que nesses 30 quadros deverá acontecer a animação 1.2 ciclos cardíacos. Para saber a quantidade de quadros para cada ciclo, basta dividir a taxa de quadros por segundo (30) pela quantidade de ciclos a realizar em um segundo (1,2), obtendo-se o valor de 25 quadros para um ciclo completo. Os períodos do ciclo cardíaco foram definidos como tendo a mesma duração, logo cada período pode utilizar metade dos quadros de um ciclo completo. Assim, tanto o período ativo quanto o de relaxamento utilizariam, nesse caso, 12,5 quadros de animação. Um quadro não pode ser dividido

por 2, logo um dos dois períodos nesse caso teria um quadro a mais, o que é bem comum em aplicações de Realidade Virtual e Aumentada. Usando esse valor como base, pode-se calcular a movimentação em cada quadro do ciclo ativo. A escala total do ciclo é de 20% como foi comentado anteriormente. Tendo o ciclo 12/13 quadros por segundo para sua execução, os 20% totais podem ser divididos por essa duração para se saber a escala do coração a cada ciclo. Assim, a escala em cada quadro sob essas configurações ficaria em 1,66% (20/12) ou 1,53% (20/13). Assim, os cálculos apresentados podem ser resumidos através das seguintes fórmulas:

- ✓ QuadrosCiclo = fps/bpm/60
- ✓ EscalaQuadro = QuadrosCiclo/2/20

Para exemplificar algumas situações reais, foi desenvolvida a Tabela 5.1, contendo valores aleatórios nos parâmetros, ilustrando os possíveis resultados.

bpm	quadros/s	quadros ativo	quadros inativo	escala total %	escala / quadro %
72	30	15	15	20	1,33
72	45	22	23	20	0,91
60	30	15	15	20	1,33
60	30	15	15	20	1,33
100	80	40	40	20	0,50
100	80	40	40	20	0,50

Tabela 5.1 – Exemplos de situações possíveis para a animação simples

Utilizando a taxa de quadros por segundo gerada pelo computador, foi possível criar as animações de forma que correspondessem com bastante precisão aos parâmetros escolhidos pelo usuário. Assim, uma simulação de pulso constante onde o valor escolhido de batimentos por minuto foi 72, apresentará em um minuto de simulação 72 ciclos cardíacos, ou algo muito próximo disso.

Como foi dito anteriormente, além da animação simples baseada na escala do órgão, foi desenvolvida uma animação mais complexa e realista, onde é simulada a atividade dos músculos cardíacos através da troca de modelos com as diferentes deformações presentes em um ciclo do coração. Esta animação foi desenvolvida propositalmente com 20 poses cardíacas para o período ativo do ciclo. As poses executadas em sequência crescente mostram a deformação do coração, alterando sua geometria a cada quadro. Na opção Simular Pulso Constante, é utilizado praticamente o

mesmo cálculo demonstrado anteriormente para saber a deformação do coração. Como a alteração é prevista nos cálculos como um valor de 0 a 20%, o valor em cada quadro é arredondado para o valor inteiro mais próximo, desprezando a parte fracionária. Este valor é utilizado para definir a pose que estará visível em determinado quadro. De acordo com a taxa de quadros por segundo gerada no computador onde está sendo executada a simulação, pode haver repetição de quadros em computadores rápidos, ou pulos de quadros em computadores mais lentos. Com isso, também é mantida a sincronia de ciclos cardíacos simulados com a quantidade requisitada nas opções da interface de operação.

Para alterar o tipo da animação, não foi provida nenhuma interface especial de operação, somente um arquivo chamado “animstyle.ini” que contém um valor numérico que pode estar em 0 ou 1. O valor 0 indica que o sistema utilizará animação simples, baseada em escala, enquanto que o valor 1 indica a escolha pela animação mais complexa e realista. Esse arquivo é lido uma única vez, quando a simulação é iniciada, não sendo possível alterar o tipo de animação em tempo de execução.

5.3.5 Tratamento da opção Buscar ECG em Arquivo

A opção Buscar ECG em Arquivo gera como parâmetros o caminho do arquivo e a quantidade de amostras por segundo do sinal. Com base nesses parâmetros o módulo GVRA deve gerar a animação simples ou complexa do ciclo, de acordo com a configuração do arquivo animstyle.ini.

A primeira etapa consiste em ler o arquivo que contém os valores do gráfico, utilizando o caminho contido no arquivo config.ini. todos os valores são armazenados em uma estrutura de dados do tipo vetor.

Com os dados capturados, o sistema encontra o menor e o maior valor presentes no sinal. Depois disso, cria um novo vetor com o mesmo tamanho do vetor das amostras. Nesse vetor são armazenados os valores do vetor original reescalados para valores entre 0 e 1. Para fazer essa transformação usa-se o maior valor como o máximo (1) e o menor valor como o mínimo (0). Todos os outros valores são reescalados subtraindo o mínimo e dividindo o restante pelo máximo subtraído do mínimo, pois essa subtração indica a extensão total do intervalo. Por exemplo, o vetor hipotético V={5, 6, 1, 3, 12, 15, 9} tem como valor mínimo 1 e como valor máximo 15. Cada valor será reescalado subtraindo-se dele o mínimo 1 e dividindo o resultado pelo máximo 15 menos o mínimo 1. Assim, o primeiro valor que é 5 será reescalado como 0,28 ((5-1)/(15-1)) e os próximos valores, incluindo o mínimo e o máximo seguirão a mesma

regra. O novo vetor reescalado ficará como V2={ 0.28, 0.35, 0, 0.14 , 0.78, 1, 0.57}. Esse processo é útil para filtrar quaisquer grandezas vindas do vetor original e padronizar a saída. Os novos valores armazenados podem ser utilizados para recuperar o vetor original se for necessário, simplesmente multiplicando-os pelo máximo subtraído do mínimo.

Na opção buscar ECG em Arquivo, o novo vetor com valores entre 0 e 1 é utilizado para guiar a animação simples por escala ou complexa por deformação. No entanto um problema frequentemente surgirá nesse tipo de simulação, a perda de dados. Como a taxa de amostragem utilizada em gráficos de eletrocardiograma é geralmente igual ou superior a algumas centenas de amostras por segundo, a maioria dos computadores pessoais não possuem recursos para gerar essa quantidade de quadros por segundo. Além disso, a própria visão humana não é capaz de identificar mudanças significativas a taxas tão altas de animação [71]. Por esse motivo, para recriar a animação de forma satisfatória, o gráfico é percorrido de acordo com a quantidade de quadros por segundo que está sendo gerada pelo computador.

Inicialmente, é calculado um valor base de movimentação pelo gráfico, dividindo a taxa de amostragem pela quantidade de quadros por segundo gerada pelo computador. Por exemplo, se a taxa de amostragem foi mantida no valor padrão da interface de operação que é 300, e a quantidade de quadros por segundo gerada está em 30, o valor de movimentação será 10 (300/30). Isso significa que em cada quadro de animação, o vetor será percorrido 10 unidades adiante. O problema causado por esta técnica é a perda de detalhe nesse intervalo de 10 amostras. Para minimizar esta perda, foi criado um mecanismo onde ao invés de se utilizar simplesmente o valor base do quadro para gerar a animação, utiliza-se uma média entre um pequeno intervalo que começa antes do valor base e se estende até um pouco depois do valor base. O tamanho do intervalo foi definido como 5% da taxa de amostragem para a animação simples e 15% para a animação com deformação.

Com o valor do quadro devidamente calculado, é possível criar as duas formas de animação do sistema. A animação simples usa o valor multiplicado diretamente por 10 que é a máxima escala positiva no órgão virtual. Assim, um valor de 0.5 é interpretado como escala positiva de 5% no órgão e um valor de 1 é interpretado como a escala máxima, ou seja, 10% do órgão virtual. O órgão já começa com 90% do seu tamanho original, sendo assim quando há a máxima escala ele alcança o seu tamanho

original. Ao final do vetor do sinal, é dado um comando que move o sinal para o seu início, reiniciando o ciclo.

O funcionamento da animação por deformação é praticamente o mesmo da animação simples, exceto que a multiplicação do valor do quadro é por 20, que é a quantidade de quadros da animação por deformação.

Tanto na animação simples quanto na animação por deformação, o funcionamento da escala automática é mantido por esta opção.

5.3.6 Tratamento da opção Capturar ECG de Imagem

A interface de operação é responsável por todo o processamento de imagem envolvido na opção Capturar ECG de Imagem. O resultado desse processamento é um vetor com as amplitudes do sinal, similar ao processado nos arquivos de ECG utilizados na opção Buscar ECG em Arquivo. Os parâmetros recebidos da interface de operação foram inclusive condicionados para o mesmo formato da opção citada. Assim, quando o módulo GVRA recebe a opção Capturar ECG de Imagem, ele faz o processamento dos dados de maneira exatamente igual ao que é feito na opção Buscar ECG em Arquivo. As duas formas de animação são mantidas nesta opção, bem como a escala automática do coração virtual.

5.3.7 Tratamento da opção Captar Dados do Sensor

O objetivo da opção Captar Dados do Sensor é a simulação em tempo real do batimento cardíaco. Esta simulação depende inteiramente dos dados capturados pelo sensor utilizado. Assim, nesta opção o único parâmetro recebido da interface de operação é justamente o número da opção.

Para determinar a freqüência do batimento cardíaco através do sensor, é necessário monitorá-lo constantemente em busca do sinal que determina o pulso. Quando o sinal de pulso é detectado, pode-se medir o intervalo de tempo em milissegundos entre o pulso atual e o anterior. A partir desse intervalo é possível determinar a quantidade de batidas por minuto, dividindo um minuto pelo intervalo. Por exemplo, se o intervalo entre um pulso e outro foi de 750 milissegundos, ou 0,75 segundos, dividindo 1 minuto (60 segundos) por esse valor, a freqüência cardíaca obtida é de 80 batidas por minuto. Porém, como o sensor utilizado não é de alta precisão, podem acontecer solavancos no pulso normal, causados por mau contato nos terminais ou erros de medição. De forma a atenuar essas imperfeições, foi definido que a freqüência cardíaca seria baseada na média dos 5 últimos valores lidos. Quando não foram lidos ainda os 5 valores, a média é feita entre os valores já lidos.

Como o módulo GVRA funciona basicamente sobre os quadros de animação gerados, foi embutida a função de monitoramento do sensor nos quadros de animação, assim, a cada quadro é feita uma leitura do sensor. Quando é detectado o pulso, é calculado o intervalo com o pulso anterior e logo após o pulso atual substitui o valor do pulso anterior, estabelecendo assim um ciclo de atualização.

As duas formas de animação do sistema utilizam a freqüência cardíaca obtida do sensor para serem geradas. O parâmetro da freqüência cardíaca é similar ao existente na opção Simular Pulso Constante, assim, foi possível adaptar as mesmas técnicas utilizadas nesta opção para a opção Captar Dados do Sensor. Na prática a simulação ocorre como se fosse o pulso constante sendo atualizado constantemente. Se houver alteração na freqüência cardíaca do usuário, quase que imediatamente poderá ser notada a mudança no sistema. A opção de escala automática também está presente nesta opção.

5.4. Conclusões

A implementação do sistema foi uma tarefa desafiadora, por envolver muitos conceitos e áreas distintas de conhecimento. A divisão do sistema em dois grandes módulos facilitou bastante a implementação, pois cada módulo tem seu espaço de funcionamento e escopo bem definidos, embora haja comunicação entre eles.

Dentre as maiores dificuldades encontradas, pode-se destacar desenvolvimento das ferramentas para o processamento de imagens de eletrocardiograma e os mecanismos das animações geradas no módulo GVRA.

Capítulo 6

6. RESULTADOS E LIMITAÇÕES

6.1. Introdução

Com a finalização da fase na qual o sistema foi implementado, é bastante adequada a realização de um conjunto de testes. Tal prática pode indicar erros não detectados na modelagem e implementação do sistema, bem como destacar as limitações decorrentes da arquitetura e recursos adotados.

Para o teste do sistema, foram selecionados 2 sujeitos, com biótipos bem contrastantes.

O primeiro sujeito é um indivíduo do sexo masculino, pesando aproximadamente 110kg, com uma altura aproximadamente de 1,79m e idade de 20 anos. O segundo sujeito é um indivíduo do sexo feminino, com peso aproximado de 52kg, altura aproximada de 1,63m e 17 anos de idade.

O contraste entre as características dos sujeitos serviu como base para o teste da funcionalidade de escala automática do coração virtual. Com esses usuários foram testadas todas as opções do sistema. Também foi avaliado o ambiente final de Realidade Aumentada, quanto ao reconhecimento de marcadores e a sua estabilidade de exibição do coração virtual.

Nas opções Buscar ECG em Arquivo e Capturar ECG de Imagem, foram utilizadas amostras reais de gráficos e imagens de eletrocardiograma, vindas de um banco de dados de sujeitos. Além das amostras do banco de dados, foram propositalmente inseridas amostras falsas, criadas para testes controlados das funcionalidades das opções.

A seguir, as funcionalidades do ambiente de Realidade Aumentada são testadas, bem como as opções do sistema sobre os sujeitos selecionados.

6.2. Preparação do ambiente de Realidade Aumentada

O ARToolKit, assim como outras aplicações de Realidade Aumentada baseadas em marcadores, utiliza a visualização da câmera para localizar os marcadores e fazer o reconhecimento dos mesmos. Para que o reconhecimento seja feito mais facilmente, algumas recomendações devem ser seguidas:

- ✓ Cadastrar os marcadores em situações de iluminação semelhantes às que serão utilizadas no ambiente da simulação
- ✓ Evitar situações de excesso ou falta de luminosidade sobre os marcadores, que prejudicam seu reconhecimento.
- ✓ Calibrar o foco da câmera de captura no momento do cadastramento dos marcadores e também da simulação.
- ✓ Manter os marcadores à distâncias não muito grandes da câmera, pois isso dificulta a diferenciação e reconhecimento dos mesmos.
- ✓ Criar marcas com diferenças marcantes nos seus símbolos de identificação, evitando assim o reconhecimento equivocado devido à semelhança nos padrões.

As recomendações citadas são válidas para qualquer sistema que se baseie no ARToolKit e outras plataformas semelhantes. Além disso, em ambos os sujeitos foram utilizadas essas recomendações, buscando manter o ambiente de experimentação bastante semelhante para os 2 sujeitos selecionados.

6.3. Escala automática do coração virtual

O teste da escala automática do coração virtual serve para verificar se essa funcionalidade traz bons resultados em sujeitos reais, pois no capítulo anterior já foram mostradas duas situações de redimensionamento do coração virtual em uma camiseta, podendo ser visualizadas na Figura 5.30.

Tal como foi implementada, a escala automática é por padrão desabilitada, permitindo a utilização do sistema mesmo sem a presença dos marcadores específicos dessa funcionalidade. A habilitação da escala automática acontece quando os 2 marcadores estão visíveis na visualização da câmera. Para o teste dessa funcionalidade foi utilizada a opção “Simular Pulso Constante”, com o valor mínimo permitido de bpm, ou seja, 20 batidas por minuto. Como meio de comparação, foram capturadas imagens no período inativo do ciclo cardíaco. A Figura 6.1 apresenta os 2 sujeitos selecionados, à esquerda com a escala automática desativada e à direita com a funcionalidade ativada.

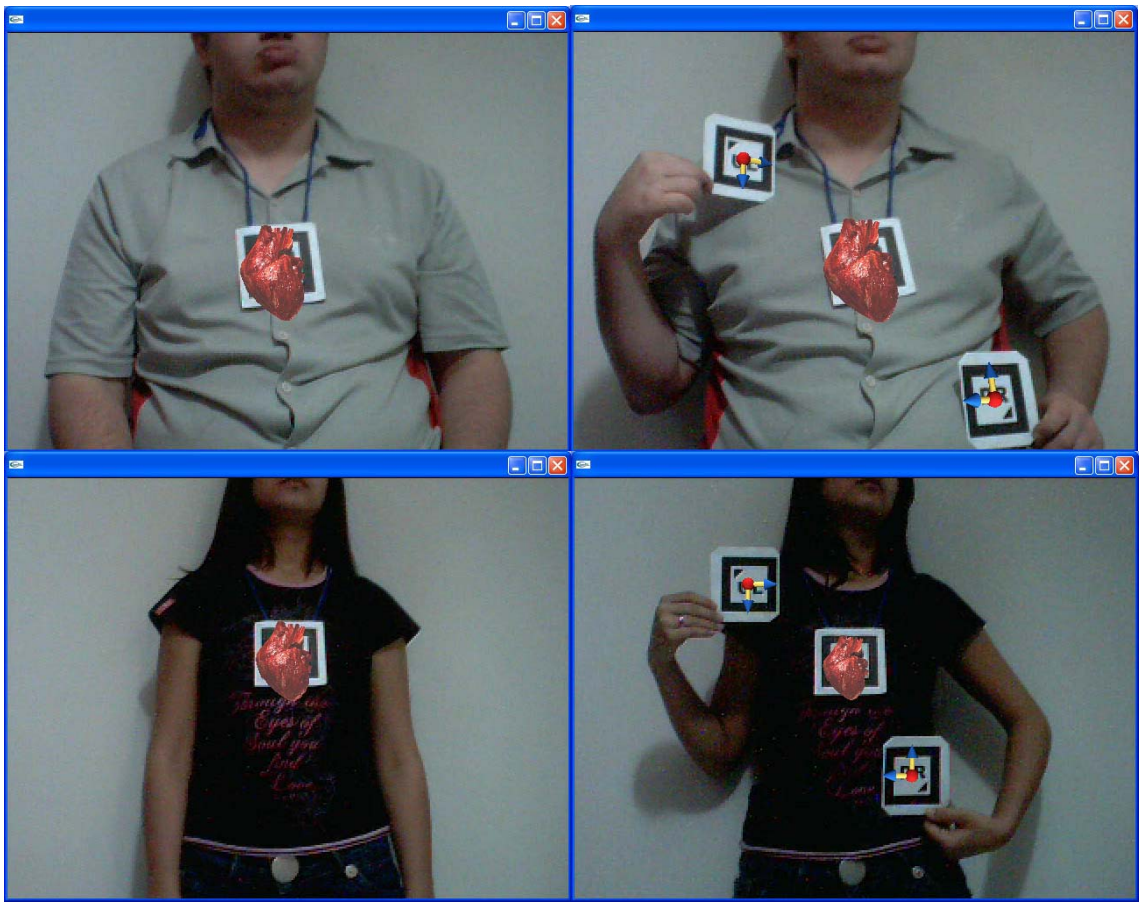

Figura 6.1 – Teste da Funcionalidade de escala automática do coração virtual.

Como pode ser visto, a ativação da escala automática gera mudanças na escala do coração virtual, redimensionando-o de acordo com o tamanho da caixa torácica do sujeito. Os resultados conseguidos são esteticamente agradáveis, melhorando o realismo visual da simulação. Porém, como já foi comentado no capítulo anterior, esta função, até o momento, não está baseada em dados fisiológicos reais, utilizando para o redimensionamento somente as distâncias horizontais e verticais dos marcadores, sendo que a distância horizontal, que determina a largura da caixa torácica, tem maior peso no cálculo do redimensionamento. Para atingir maior grau de realismo, além de utilizar dados fisiológicos relativos à caixa torácica no redimensionamento, seriam necessárias algumas informações adicionais do sujeito, como estilo de vida (sedentário, mediano, atleta) e histórico de doenças cardíacas, que podem exercer influência sobre o tamanho final do coração.

6.4. Opção Simular Pulso Constante

Como esta opção só tem como parâmetro de configuração a quantidade de batidas por minuto, a maior preocupação é com o ambiente de Realidade Aumentada.

Sendo o marcador do coração confeccionado em forma de crachá, o sujeito pode realizar a simulação sentado, em pé ou mesmo deitado. Com o objetivo de manter o conforto dos sujeitos, todos os testes foram realizados com os sujeitos sentados. A escala automática foi mantida desativada nesses testes, pois já tendo sido feita sua validação, foi mais interessante manter o coração virtual com tamanho padrão nos 2 usuários, para fins de comparação de resultados.

Pelo fato de a simulação ser realizada por animação, é difícil ilustrar o funcionamento da mesma. Assim, foi estabelecido que para ilustrar a simulação, seriam capturados 5 quadros da animação gerada, tanto no modo simples por escala, quanto no modo complexo por deformação do coração virtual. As formas de animação só serão ilustradas uma vez, pois o seu comportamento é o mesmo para todas as opções.

Para o teste da opção "Simular Pulso Constante", foram utilizadas as configurações na interface de operação apresentadas na Figura 6.2.

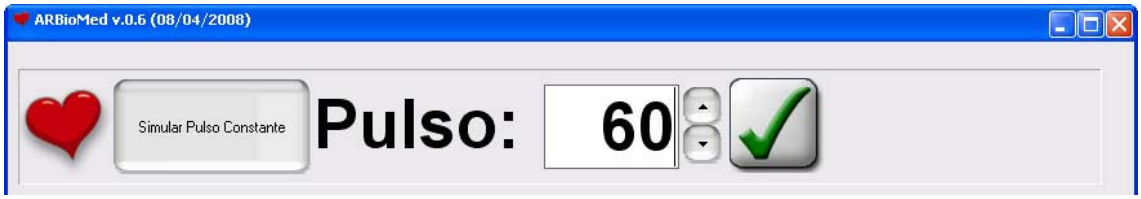

Figura 6.2 – Configurações para teste da opção Simular Pulso Constante.

Ao pressionar o botão "Simular", a simulação levou aproximadamente 6 segundos para iniciar, estando ativa por padrão a forma simples de animação. A Figura 6.3 apresenta 5 quadros de animação simples para os 2 sujeitos selecionados.

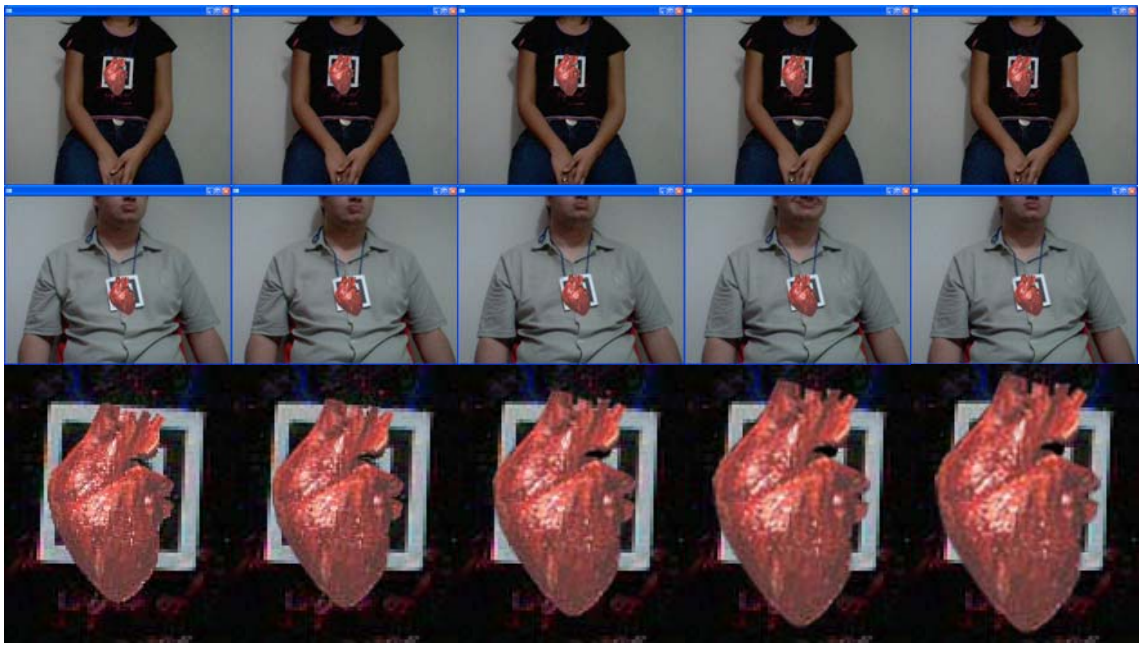

Figura 6.3 – Quadros da animação simples com os 2 sujeitos.

Neste teste, foi observado que a resolução mais adequada para a permanência da interface de operação juntamente com a janela de simulação é de 1280 x 960 pontos ou superior, sendo que em resoluções inferiores a essa, o espaço se torna pequeno para acomodar os 2 módulos.

Em seguida, foi fechada a simulação, através do pressionamento da tecla “Esc” do teclado. Ao tentar fechar a simulação através do botão fechar da janela, ocorre um erro decorrente da implementação do ARToolKit utilizado. Assim, seria interessante sanar esse problema ou pelo menos instruir o usuário para utilizar a tecla “Esc” e alertá-lo a respeito dessa falha.

Outro teste feito ainda na opção 1 foi a forma mais complexa da simulação, através da mudança do valor 0 para 1 no arquivo animstyle.ini. A Figura 6.4 mostra o arquivo config.ini padrão.

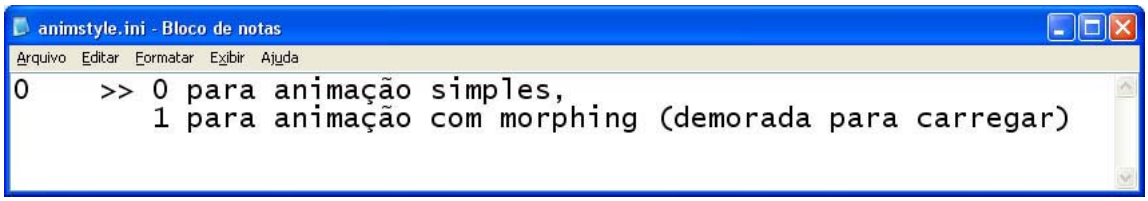

Figura 6.4 – Arquivo animstyle.ini padrão.

Quando foi pressionado o botão “Simular”, a simulação com animação complexa demorou aproximadamente 45 segundos para ser apresentada, o que é conseqüência da demora no carregamento dos 20 modelos que compõem a animação complexa. Essa demora pode desorientar o usuário, fazendo-o pensar que a simulação não está sendo carregada. Assim, seria válido no caso da animação complexa, alertar o usuário através da interface de operação que haverá um período longo de espera até o carregamento da simulação. O resultado da animação após o carregamento é bem mais realista que na animação simples, e como os modelos já estão todos carregados na memória, a animação flui suavemente, sem sobrecarga aparente no sistema. A Figura 6.5 apresenta 5 quadros da animação complexa para os 2 sujeitos.

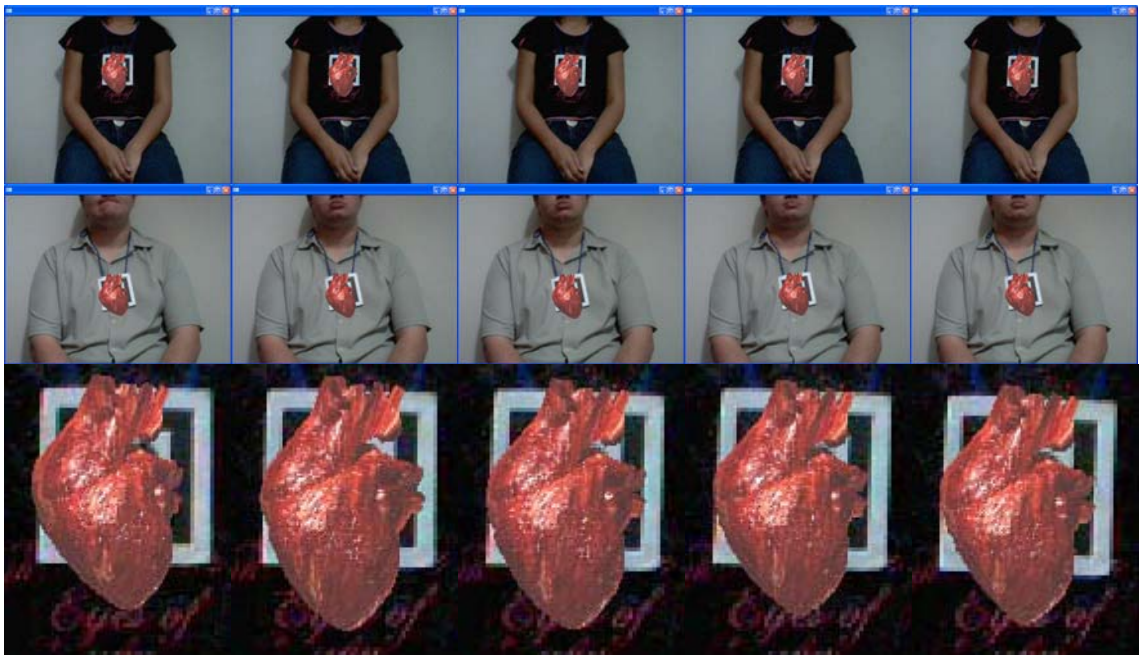

Figura 6.5 – Quadros da animação complexa para os 2 sujeitos.

Foi testada ainda a funcionalidade de atualizar os parâmetros na interface de operação em tempo de execução da simulação. Para tanto, foram inseridos novos parâmetros na opção e pressionado o botão "Atualizar". A Figura 6.6 apresenta as novas configurações inseridas.

Figura 6.6 – Novas Configurações inseridas na interface de operação.

Após o pressionamento do botão "Atualizar", em aproximadamente 1 segundo foi possível notar as mudanças da interface refletindo no comportamento da simulação, tanto no modo simples de animação como no complexo.

Um último teste feito na opção "Simular Pulso Constante", foi a fidelidade da animação gerada, visto que nesta opção é simulado um valor fixo de batidas cardíacas por minuto. Com o auxílio de um relógio simples, foram contadas as batidas do coração virtual durante um minuto para diferentes quantidades de batidas por minuto. A diferença entre o valor solicitado e a simulação foi, em todos os casos, inferior a 1 batida para mais ou para menos. Assim, considera-se que a simulação apresentou boa fidelidade na visualização.

6.5. Opção Buscar ECG em Arquivo

A opção “Buscar ECG em Arquivo” exige como entrada um arquivo-texto contendo os valores das amplitudes para o gráfico de ECG. Além disso, o parâmetro “Amostras/seg” influencia grandemente o resultado final da simulação, à medida que define quantas amostras do arquivo serão utilizadas por segundo de animação. Para o teste dessa opção, foram utilizados arquivos de ECG de um banco de dados real de sujeitos. Adicionalmente, foram criados arquivos com valores controlados para melhor testar o comportamento da simulação.

O primeiro teste feito foi a abertura de arquivos, tanto pelo caminho digitado completamente, quanto através da utilização do diálogo de abertura. Em ambos os casos, ao tentar abrir um arquivo inválido, como era previsto, o sistema indicou o erro ocorrido e desabilitou o botão simular, forçando o sujeito a abrir um novo arquivo. A Figura 6.7 apresenta a inserção de um erro e a reação do sistema.

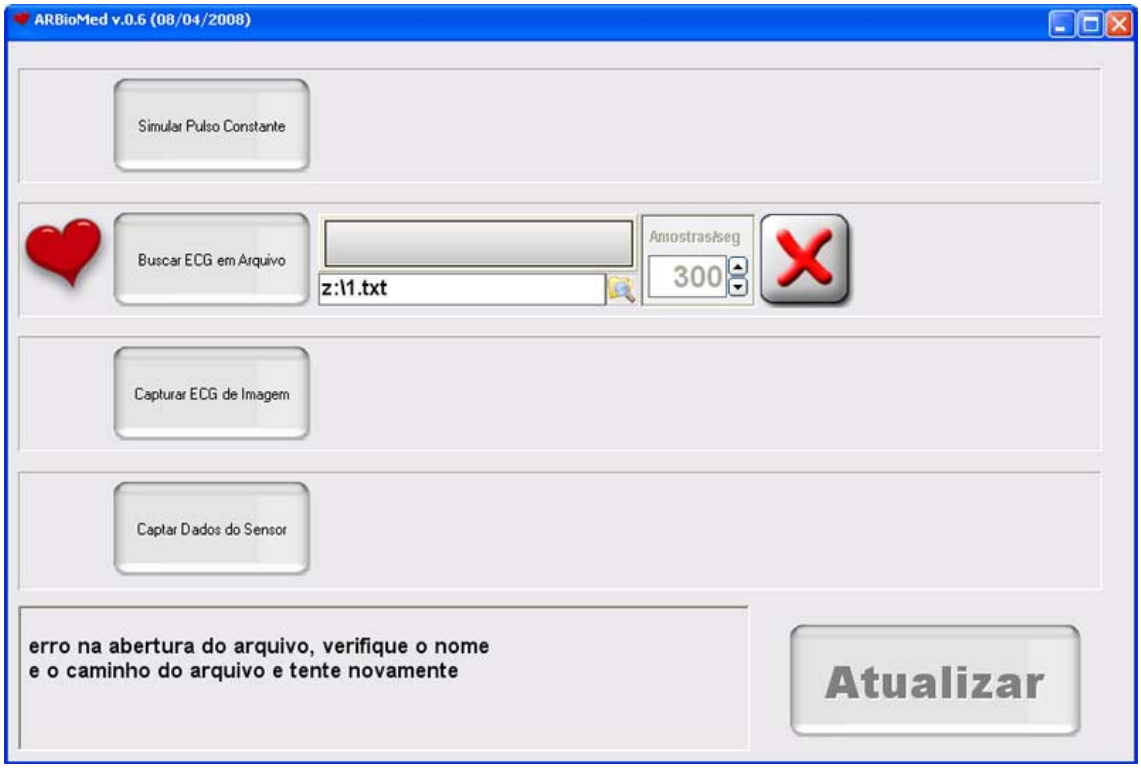

Figura 6.7 – Tratamento de erro na abertura de arquivo ECG.

Ao abrir um arquivo válido, o sistema apresentou um gráfico de pré-visualização para o sujeito e habilitou a configuração do parâmetro “Amostras/seg”, bem como o botão “Simular”. O valor de 300 amostras por segundo que é o padrão do sistema foi mantido nos testes iniciais. A Figura 6.8 apresenta um gráfico aberto com sucesso na opção “Buscar ECG em Arquivo”.

Figura 6.8 – Gráfico aberto com sucesso na opção Buscar ECG em Arquivo.

Ao pressionar o botão Simular, tanto na animação simples quanto na complexa, os tempos de carregamento foram os mesmos da opção anterior. Quanto ao resultado da simulação, na animação simples, a escala do coração obedeceu principalmente às curvas mais acentuadas do gráfico, refletindo o período de maior atividade muscular do coração, nos ventrículos [47,48]. Já na animação complexa, as poses cardíacas foram corretamente associadas às amplitudes detectadas, e a simulação ocorreu de forma bem fluida. Para exemplificar melhor esta situação, a figura 6.9 apresenta um gráfico gerado artificialmente, em formato de onda, e sua respectiva visualização na forma de 5 quadros da animação simples.

Figura 6.9 – Visualização da animação simples a partir de gráfico de controle.

Outro teste efetuado foi a alteração do parâmetro “Amostras/seg” e a atualização da simulação em tempo de execução. Com taxas de amostras por segundo muito distantes das originais de captura, aconteceram distorções visíveis na simulação, tanto na forma simples quanto na complexa de animação. Assim, da correta configuração deste parâmetro depende o bom funcionamento desta opção.

O último teste envolveu a troca do arquivo utilizado em tempo de execução da simulação, bem como o ajuste do parâmetro “Amostras/seg”. Nesse teste foi utilizado o mesmo arquivo de onda mostrado na figura 6.9. Este arquivo contém 72 amostras com amplitudes de 0 a 1000. Foi determinada a taxa fictícia de amostras por segundo de 72, fazendo com que a onda seja simulada completamente em um segundo de animação. A Figura 6.10 apresenta as configurações utilizadas.

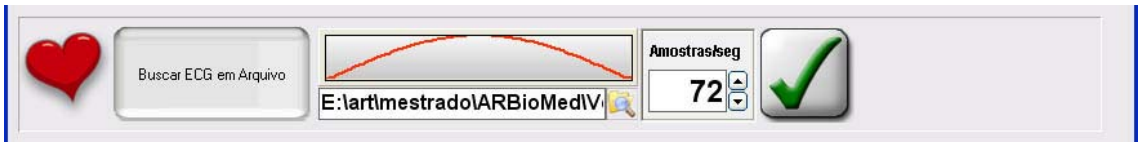

Figura 6.10 – Configurações utilizadas no teste final da opção Buscar ECG em Arquivo.

A atualização da simulação ocorreu quase que simultaneamente ao pressionar do botão "Atualizar". O comportamento das formas de animação simples e complexa foi conforme o previsto, realizando o ciclo de animação completo em um segundo e de uma forma que correspondeu ao padrão da onda. Foi feita ainda uma alteração no parâmetro "Amostras/seg" para 144. Teoricamente, essa alteração deveria dobrar a velocidade da animação, uma vez que foi dobrada a taxa de amostragem. Ao verificar o comportamento da simulação, houve de fato o aumento na velocidade de animação, para o dobro da anterior.

Assim foram finalizados os testes da opção Buscar ECG em Arquivo.

6.6. Opção Capturar ECG de Imagem

De todas as opções, a que mais possui parâmetros e configurações na interface de operação, é a opção Capturar ECG de Imagem. A maior quantidade de parâmetros é diretamente relacionada ao maior processamento e edição necessários, uma vez que o sinal é capturado de meios analógicos como o papel, onde há frequentemente ruídos a serem retirados para uma boa extração do sinal ali contido.

Para testar as funcionalidades desta opção, foram utilizadas algumas imagens vindas de um banco de dados real de exames de ECG, bem como algumas imagens manipuladas de forma controlada, para melhor controle nos testes. A Figura 6.11 apresenta algumas das imagens utilizadas nos testes sendo que as imagens à esquerda da figura são exemplos de arquivos reais de ECG, e as imagens da direita são arquivos de testes gerados de forma controlada via computador.

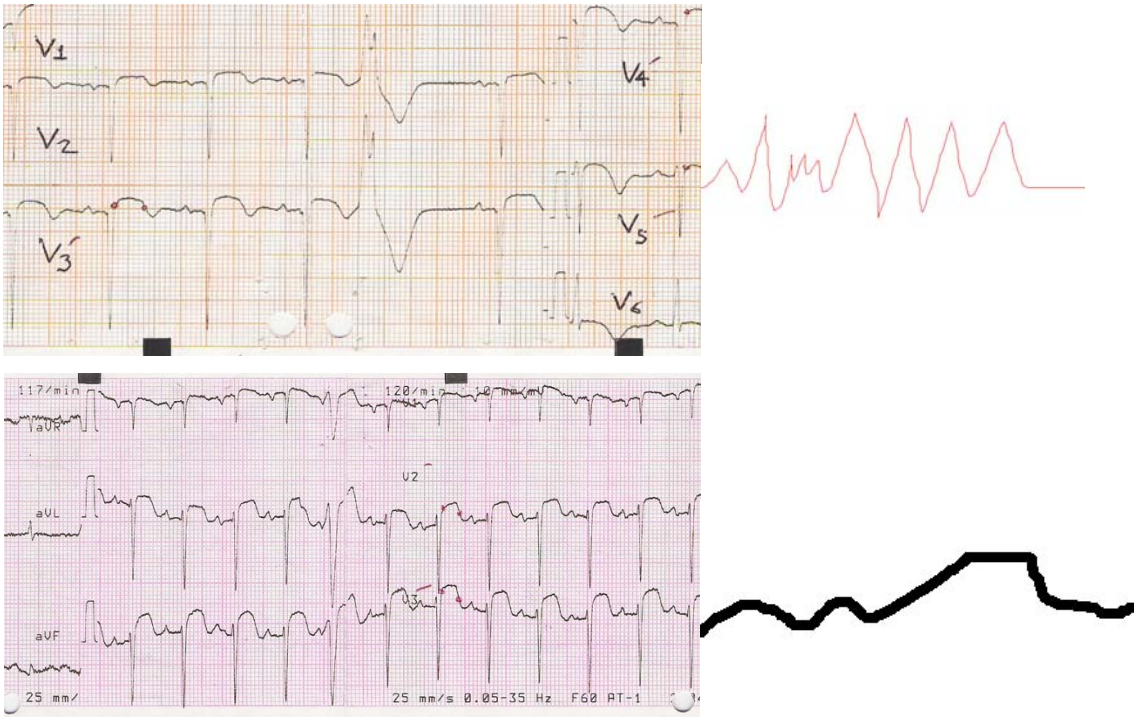

Figura 6.11 – Imagens utilizadas nos testes.

Como pode ser visto, nas imagens reais há inscrições e diversos símbolos gráficos embaralhados junto ao sinal. A presença desses elementos serve para testar a capacidade do sistema de extrair o sinal mesmo em meio a uma quantidade considerável de ruídos.

O primeiro teste feito tem relação com a abertura de arquivos gráficos. Ao pressionar o botão “Abrir Imagem de ECG” a caixa de diálogo exibida já filtra os tipos de arquivo mais comuns aceitos pelo sistema. Como todos os formatos além do bitmap (*.bmp) são abertos de forma terceirizada pelo Irfan View, se houver a tentativa de abrir um arquivo não suportado, consequentemente não haverá a conversão pelo Irfan View e o sistema apresenta uma mensagem de erro padrão. Quando o formato é suportado pelo Irfan View, aparece brevemente uma janela do sistema, enquanto ocorre a conversão. Essa situação é ilustrada pela Figura 6.12.

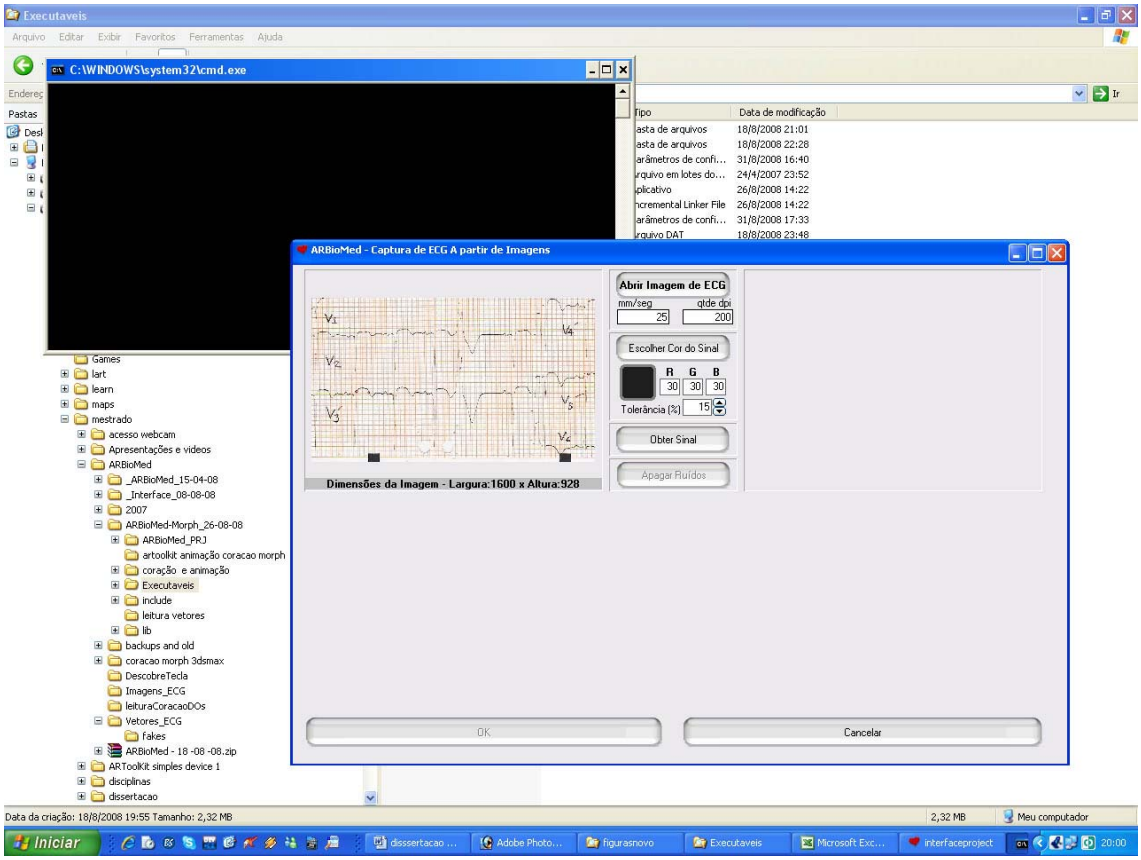

Figura 6.12 – Conversão de Imagem pelo Irfan View.

Nos testes, foram abertas com sucesso imagens nos formatos bmp, jpg, png, gif e wmf. As imagens abertas tinham resoluções bem diversificadas e o sistema foi capaz de tratá-las adequadamente. Todas as imagens foram abertas com tempos inferiores a 5 segundos, embora é possível que imagens muito grandes alcancem ou superem esse intervalo de tempo.

O próximo passo é a escolha da cor do sinal. Essa tarefa foi testada primeiramente no modo semi automatizado. Este modo é assim chamado, porque o usuário não precisa saber os valores a inserir nos canais R, G e B, para selecionar a cor do sinal, como acontece no modo manual. Ele pode simplesmente posicionar o cursor sobre um ponto da imagem a filtrar, para que aquela cor seja automaticamente capturada pelo sistema como sendo o sinal. A figura 6.13 apresenta a utilização desta forma de escolha.

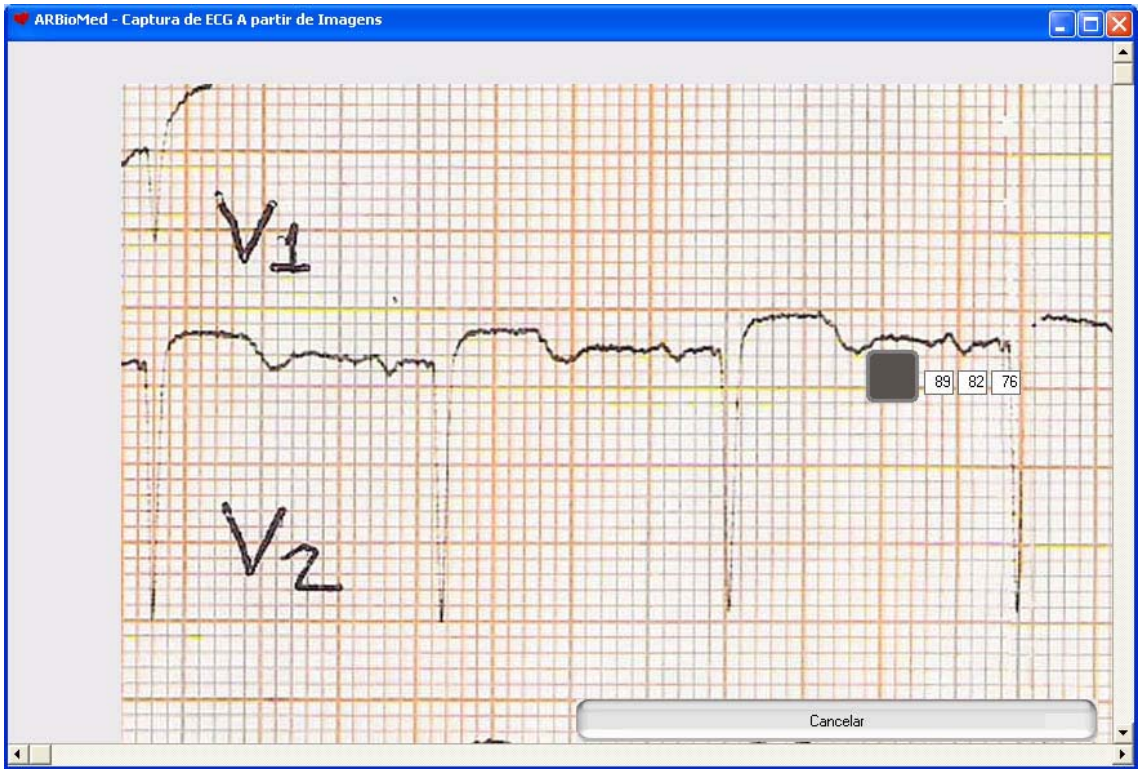

Figura 6.13 – Escolha semi automatizada da cor do sinal.

Nesse teste, o sistema se comportou de forma adequada, permitindo a escolha do sinal facilmente. Para imagens maiores podem ser utilizadas as barras de rolagem para navegar pelas áreas da imagem. O botão "Cancelar" também obteve funcionamento correto, não afetando a cor previamente escolhida do sinal.

O próximo teste foi a alteração manual da cor do sinal, através da edição dos campos de texto com as intensidades dos canais de cor RGB. Esse teste também foi bem sucedido, sendo que a cor do sinal correspondeu fielmente aos valores digitados nos campos. A Figura 6.14 ilustra uma escolha manual de cor.

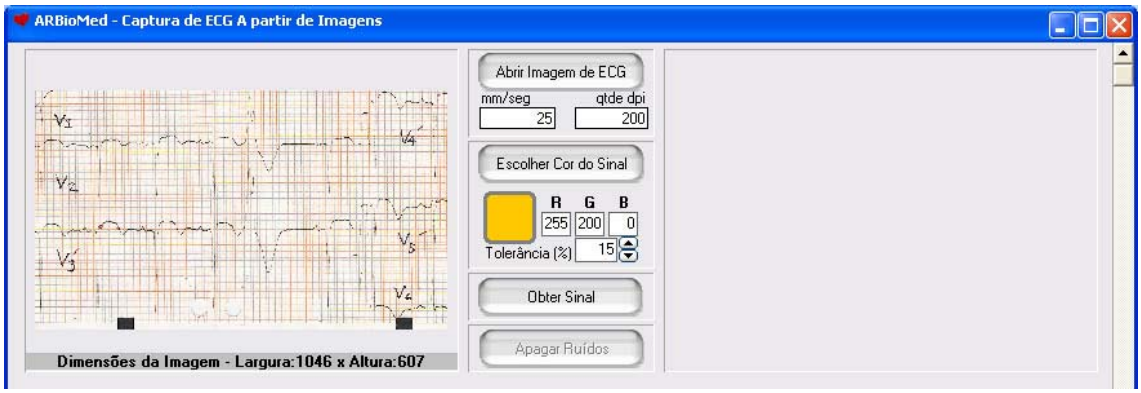

Figura 6.14 – Escolha manual da cor do sinal.

O próximo teste envolveu a obtenção primária do sinal a partir da cor escolhida. Inicialmente, foi deixado o valor de tolerância padrão de 15 %. Este valor foi estabelecido como padrão através de uma série de testes realizados com diferentes imagens de ECG, enquanto acontecia implementação do sistema. Ao clicar sobre o botão “Obter sinal”, foi obtido um resultado bastante satisfatório com a tolerância padrão, sendo que aumentando muito este valor, ocorreu o reconhecimento de ruídos como sinal, o que é indesejado.A diminuição do valor em demasia acarretou erosão no sinal detectado, fazendo com que ele perdesse detalhes. Tanto a erosão do sinal, quanto o reconhecimento indevido de ruídos, são explicados pela escolha incorreta do percentual de tolerância, pois esse percentual serve para estabelecer um limite máximo de variação na cor do sinal a ser reconhecido. Ao se diminuir esse limite, pixels do sinal com mais variação do que o permitido são ignorados ocorrendo assim a erosão. Ao se aumentar demasiadamente esse limite, não ocorre mais a erosão, mas além dos pixels do sinal, pixels com cores próximas da cor do sinal o suficiente para passarem pelo limiar de tolerância, acabam sendo reconhecidas também como sinal, ocorrendo injeção de ruído no sinal. Assim, é de suma importância a correta configuração do parâmetro da tolerância, fazendo-se testes até chegar ao ponto ideal. A Figura 6.15 apresenta o reconhecimento de um sinal com 15% de tolerância sobre a cor do sinal (esquerda), e o mesmo sinal sendo reconhecido com tolerância de 35% (direita).

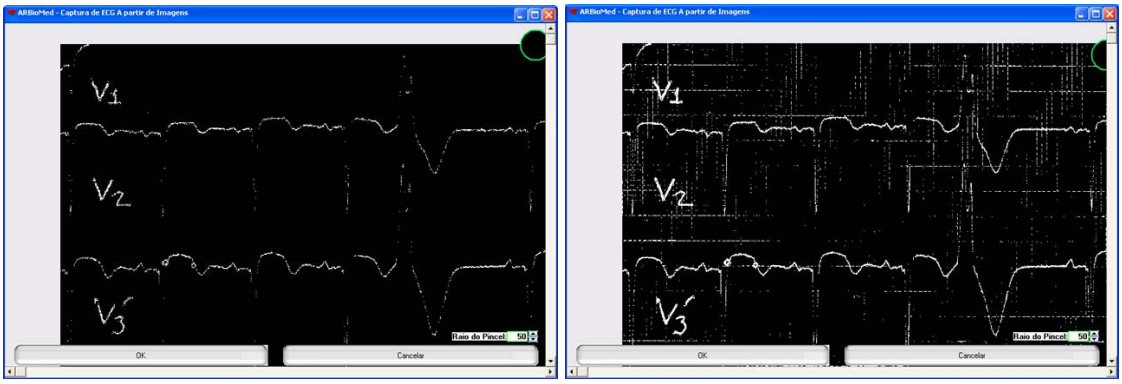

Figura 6.15 – Reconhecimento do Sinal com tolerâncias de 15% e 35%.

Mesmo com o melhor ajuste de tolerância, o reconhecimento do sinal quase nunca é perfeito, pois alguns dos elementos gráficos presentes na imagem podem possuir exatamente a mesma cor do sinal, ou algo muito próximo disso, dificultando a filtragem. Assim, a utilização da ferramenta “Apagar Ruídos” é imprescindível.

Os próximos testes foram relativos à ferramenta “Apagar Ruídos”. Inicialmente, foi testada a funcionalidade básica da ferramenta, sendo apagados todos os ruídos

encontrados. Ao clicar com o mouse sobre a imagem e arrastar, todo o conteúdo do local foi corretamente apagado, ficando preto. Como a imagem real usada continha 3 linhas de gráfico, foi necessário apagar 2 linhas pois o sistema só processa uma linha por vez. Para facilitar a tarefa foram utilizados vários tamanhos de pincéis, definidos através do campo "Raio do Pincel" presente na ferramenta. Ao terminar a tarefa, clicando-se sobre o botão "Ok" a janela retornou, após alguns momentos de processamento, para seu estado anterior, mostrando a prévia do sinal reconhecido atualizada e um gráfico ilustrando como será armazenado o sinal após seu reconhecimento. A Figura 6.16 apresenta esse cenário.

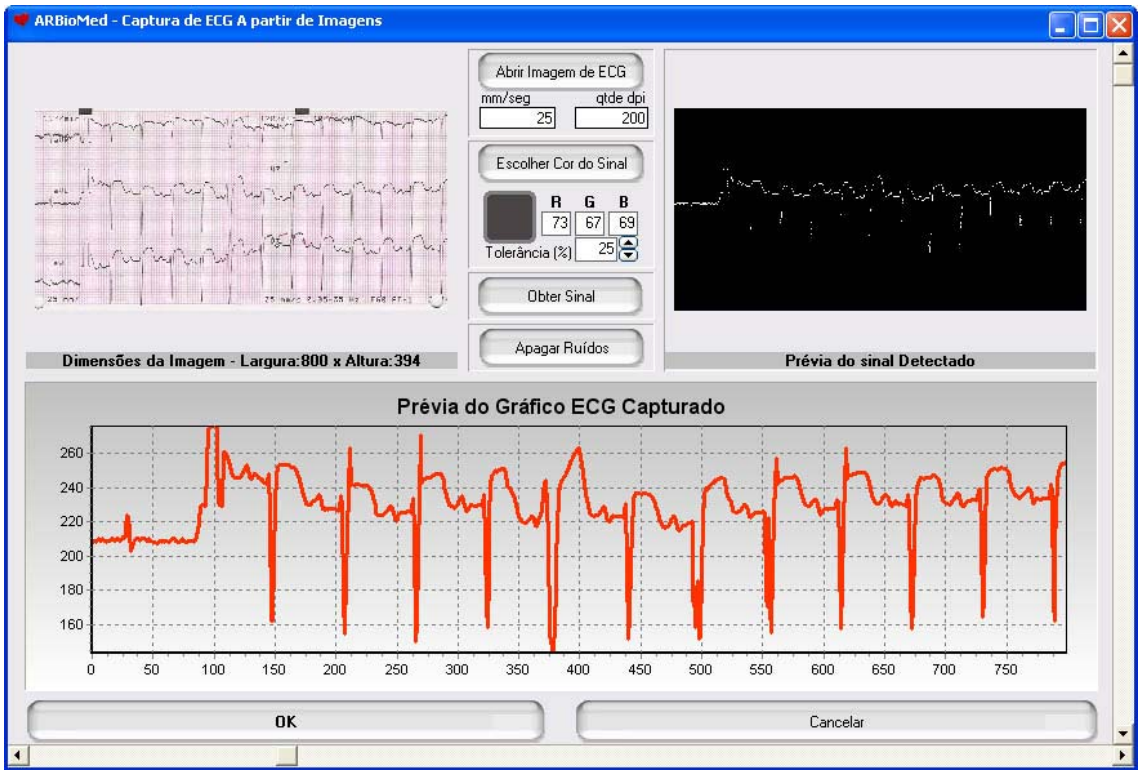

Figura 6.16 – Prévia atual do sinal reconhecido e gráfico mostrando seu armazenamento.

O próximo teste foi feito no sentido de verificar a capacidade da ferramenta corrigir falhas de continuidade no sinal. Para testar isso, foi utilizada uma das imagens de controle. Primeiramente, foi feita a abertura desta imagem de forma normal. Após isso, selecionou-se a cor do sinal e foi pressionado o botão "Obter Sinal". O sinal obtido foi perfeito, sem falhas, pois a imagem foi criada de forma digital, com cores sólidas e sem ruídos. Foram induzidas falhas no sinal, apagando alguns trechos do mesmo e pressionado o botão "Ok". A Figura 6.17 apresenta a imagem de controle, o gráfico com falhas induzidas e o resultado da correção de falhas pelo sistema.

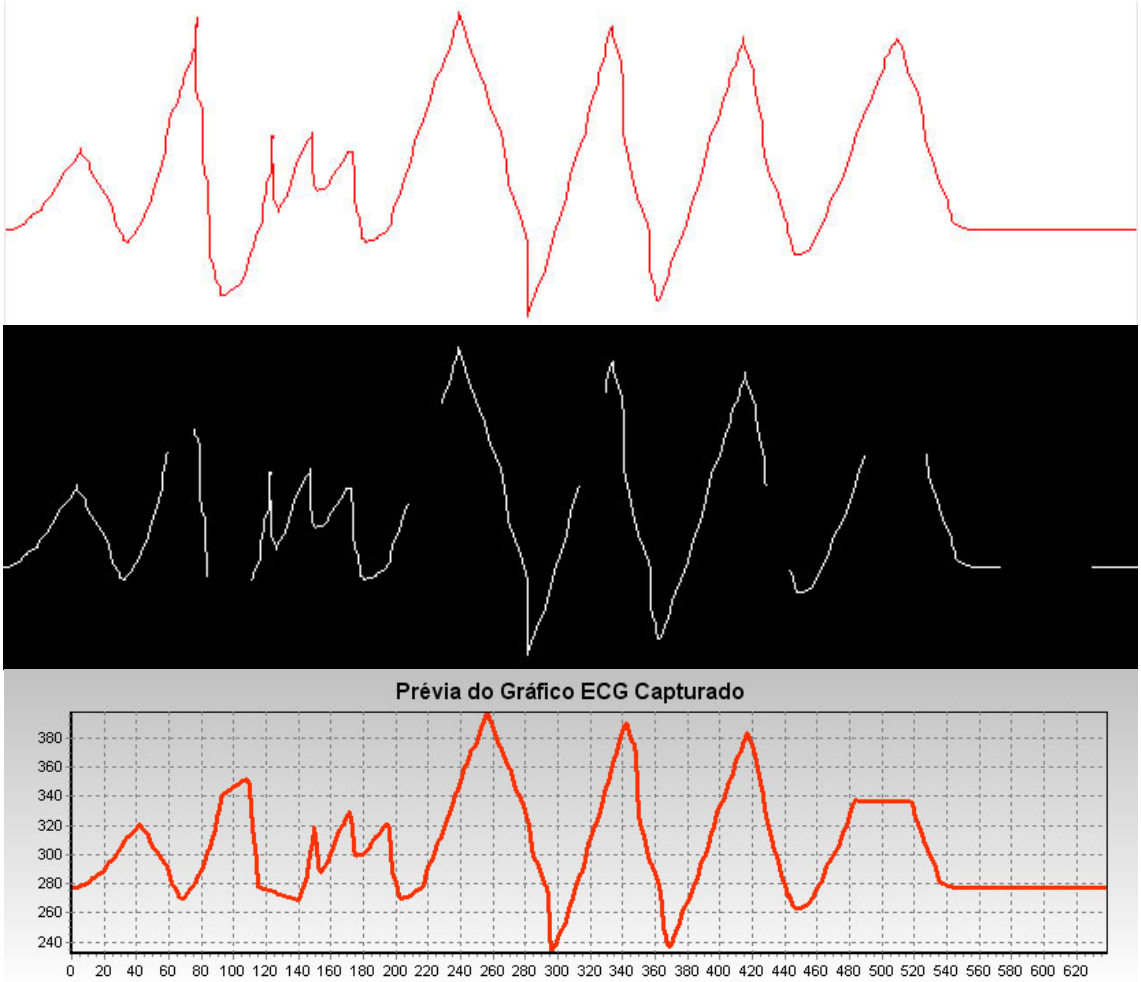

Figura 6.17 – Imagem de controle, gráfico com falhas e correção de falhas resultante.

Como pode ser visto, nos casos onde haviam muitos detalhes na parte apagada, houve perdas consideráveis de sinal, porém nos locais onde havia uma certa linearidade na transição do gráfico, a reconstrução foi praticamente perfeita. Ao realizar este teste, foi sentida a falta de uma ferramenta para recriar certos trechos de sinal apagado.

Um último teste ainda na ferramenta “Apagar Ruídos” foi o funcionamento do botão Cancelar presente na ferramenta. A função deste botão é desfazer todas as operações de apagamento realizadas no sinal reconhecido, retornando-o para seu estado anterior. Esta opção demonstrou funcionar perfeitamente, sendo que quando há o clique sobre o botão Cancelar, alguns momentos são gastos para restaurar o gráfico para seu estado anterior, não excedendo, no pior dos casos testados, 3 segundos de processamento.

Com o sinal já devidamente reconhecido, o próximo passo é pressionar o botão “Ok” para retornar o sistema à sua interface principal. Antes disso, porém, é necessário ter configurado corretamente os parâmetros “mm/seg” e “qtde dpi”. O parâmetro

“mm/seg” muitas vezes está informado no próprio arquivo gráfico. O parâmetro “qtde dpi”, no entanto, pode ser difícil de se saber, principalmente se a imagem for obtida sem informações adicionais. Uma forma simples de se determinar esse parâmetro, é usando as divisões milimetradas frequentemente encontradas nos exames de ECG. Pode-se nesse caso escolher como cor do sinal a cor das divisões milimetradas e após isso selecionar a opção “Apagar Ruídos”. O raio do pincel é medido em pixels, assim é possível mensurar, aproximadamente, a quantidade de pixels presente em um bloco da imagem. Alterando o tamanho do cursor para corresponder a 25/26 divisões milimetradas na imagem, pode-se medir através do raio do pincel, a metade da quantidade de dpi utilizada na digitalização da imagem. Assim, esse valor multiplicado por 2, pode ser usado com boa precisão como a configuração para o campo “qtde dpi”.

De volta à interface principal, um novo botão “Salvar gráfico” aparece. A função para este botão é gerar arquivos similares aos abertos pela opção “Buscar ECG em arquivo”. Isso foi testado e os arquivos gerados realmente são similares aos do banco de dados, possuindo a quantidade de amostras igual à quantidade de pixels na largura da imagem de origem do sinal. A Figura 6.18 apresenta a interface atualizada e o trecho de um arquivo salvo pela funcionalidade do botão “Salvar Gráfico”.

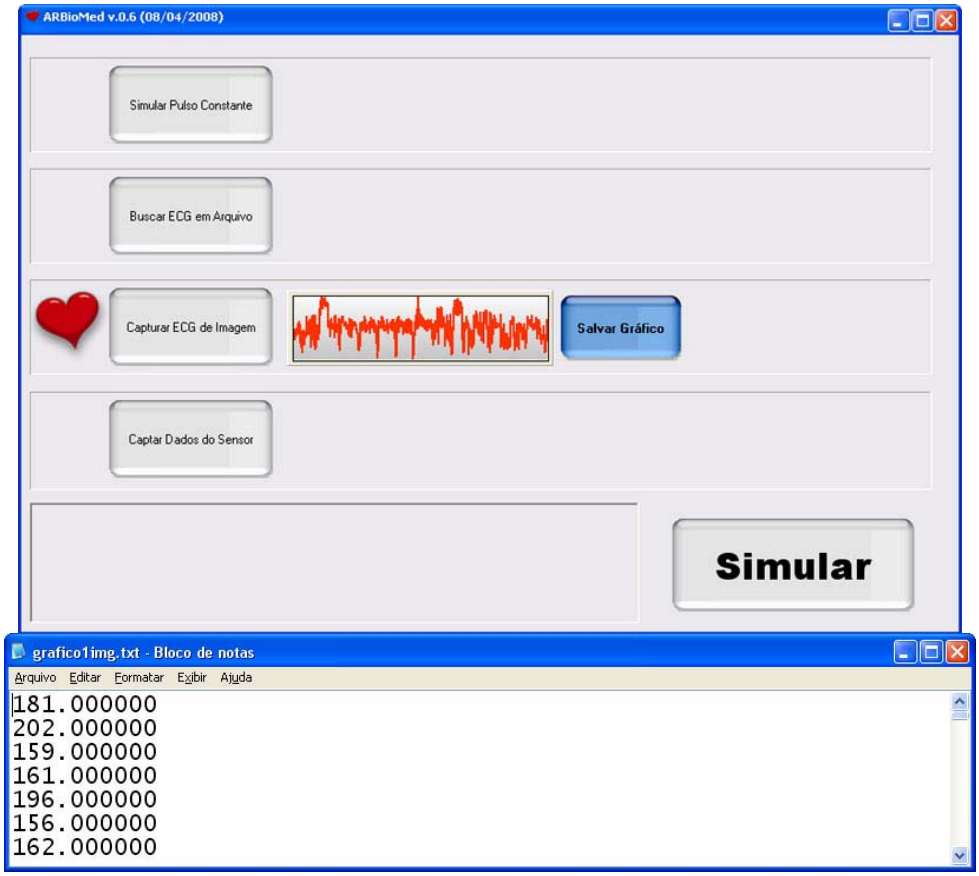

Figura 6.18 – Interface Atualizada e arquivo ECG gerado a partir de Imagem.

Ao pressionar o botão Simular, as animações tanto no modo simples quando no modo complexo, se comportaram exatamente como na opção “Buscar ECG em arquivo”, visto que a saída para a simulação é um arquivo similar ao utilizado pela opção citada. Ao se alterar os valores dos campos “mm/seg” e “qtde dpi”, os resultados na simulação também se alteram drasticamente, por isso devem ser ajustados com atenção.

Um último teste foi realizado para averiguar o suporte à modificação do arquivo de imagem em tempo de execução da simulação. Nesse caso foi notada uma leve sobrecarga no sistema, devido ao processamento de imagem concorrendo com a execução da simulação. Mas o resultado foi como esperado, a simulação se adaptou de forma praticamente instantânea aos novos parâmetros, após o clique do botão Atualizar.

6.7. Opção Captar Dados do Sensor.

Esta opção não tem nenhum parâmetro a ser ajustado na interface de operação, tendo ela como função somente o monitoramento de funcionamento do sensor. Assim, foi acoplado o sensor nos 2 sujeitos para teste de funcionamento. Nesse ponto pôde ser averiguado que o sensor utilizado não é muito robusto para suportar as diversas anatomias dos sujeitos, pois em um dos sujeitos houve limitações do sensor em reconhecer as batidas do coração, o que demandou maior tempo de ajuste do sensor no lóbulo da orelha até se conseguir uma medição correta.

Na interface foi testada a detecção da presença do sensor e a fidelidade da medição dos batimentos cardíacos. A presença do sensor foi detectada com sucesso nos 2 sujeitos, porém, a fidelidade da medição das batidas, apresentou inicialmente valores incorretos no sujeito que teve limitações com o sensor, só sendo corrigidos quando foi conseguido o correto ajuste do sensor no sujeito. A Figura 6.19 apresenta o comportamento da interface para os 2 sujeitos, mostrando à esquerda a medição correta e à direita uma das medições incorretas capturadas enquanto se estava fazendo o ajuste do sensor ao sujeito.

Figura 6.19 – Interface da Opção Captar Dados do Sensor para os 2 sujeitos.

O próximo teste foi a simulação da opção, onde foi testado se o sistema conseguiria recriar de forma satisfatória as batidas cardíacas com freqüência similar à medida pelo sensor. Ao pressionar o botão “Simular”, a medição e atividades da interface de operação foram interrompidas. O carregamento da simulação no modo simples de animação levou pouco mais de 9 segundos, sendo que essa demora adicional é provavelmente devida à utilização das funções de aquisição do sensor. No modo complexo de animação, a demora no carregamento chegou a pouco mais de um minuto. Em relação à fidelidade da freqüência cardíaca virtual, tanto no modo simples quanto no complexo de animação, houve atraso mínimo, similar ao ocorrido na opção “Simular Pulso Constante”, mesmo porque os sistemas de animação nas 2 opções são bastante similares. Para saber a medida da pulsação foi utilizado o visor LCD do sensor, que mostra a quantidade de batidas por minuto do coração. A figura 6.20 apresenta a simulação com o sensor para os 2 sujeitos.

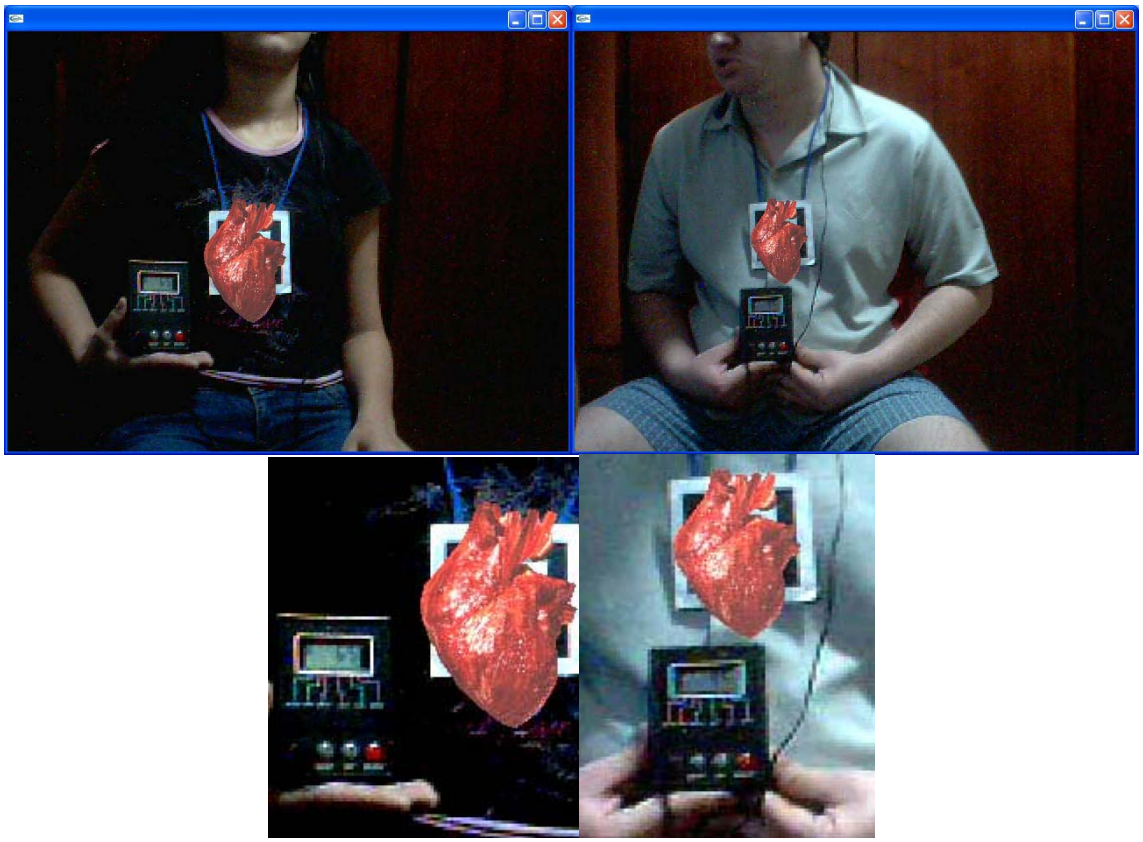

Figura 6.20 – Simulação com o sensor para os 2 sujeitos, sendo 65 o valor de bpm do sujeito à esquerda e 75 o valor de bpm do sujeito à direita.

A diferença de tamanho nos corações virtuais dos sujeitos é devida ao fato de a captura ter sido realizada em diferentes momentos de animação para os 2 sujeitos.

Não foi notado durante os testes nenhum problema de acesso à porta paralela ou de comunicação entre o computador e o sensor, sendo que a única limitação encontrada foi em relação à versatilidade de utilização do sensor por diferentes sujeitos, o que dificultou a aquisição da pulsação cardíaca em um dos sujeitos.

6.8. Testes adicionais

Além do teste individual em cada opção do sistema, também foram realizados testes no sistema como um todo, visando mensurar a real capacidade de integração do mesmo. Para tanto, foram ativadas cada uma das opções e iniciada a simulação. Depois da simulação estar funcionando, foram sendo feitas trocas para as opções todas as opções restantes, e atualizada a simulação. Isso foi feito de todas as opções para todas as opções, sem exceção. Esse teste foi bem sucedido, tendo a simulação aceitado normalmente a mudança entre todas as opções e se atualizado conforme os novos parâmetros de forma transparente ao sujeito.

6.9. Limitações Detectadas

Durante os testes, puderam ser detectadas algumas limitações no sistema. São elas:

- Reconhecimento das marcas muito sensível à condições de luminosidade e reflexos – Essa limitação prejudica o bom funcionamento do sistema à medida que impede a detecção dos marcadores ocasionando a perda de sincronia entre os objetos reais e virtuais;
- Animação Complexa muito lenta para Carregar – A animação complexa foi desenvolvida otimizando a execução, mas da forma como está implementada, ocasiona grande demora no carregamento da simulação, prejudicando o bom funcionamento da mesma;
- Processamento de apenas uma linha de ECG por vez – Essa limitação da opção "Capturar ECG de Imagem", torna um pouco cansativa a extração de sinais partindo de imagens, uma vez que é preciso apagar todas as demais linhas existentes para extrair o gráfico ECG;
- Sensor com pouca capacidade de adaptação – O sensor utilizado é bem simples, projetado originalmente para utilização em esteiras e bicicletas ergométricas de academias. Nos testes, foi notado que o seu funcionamento não é uniforme em todos os sujeitos, exigindo bastante paciência para se ajustar o sensor e alcançar resultados;
- O coração virtual modelado é opaco, não mostrando as estruturas internas do coração, como os átrios e ventrículos que se revezam durante o ciclo cardíaco;
- A escala automática realizada no coração virtual não é baseada em dados fisiológicos da caixa torácica, nem utiliza informações adicionais dos sujeitos, como estilo de vida e histórico de doenças cardíacas, podendo haver distorções no seu funcionamento;
- Validação do sistema – Para o teste do sistema foram escolhidos apenas 2 sujeitos, sendo que nenhum deles é portador de deficiências cardíacas. Também não foram testadas condições variadas de exercícios físicos, que seriam particularmente úteis nas simulações envolvendo o sensor e possibilitariam visualizar a alteração do ritmo cardíaco frente às diferentes situações de esforço físico. Uma outra forma interessante de se avaliar o sistema e sua real funcionalidade seria a interação com profissionais de

Cardiologia, estudantes de áreas relacionadas e mesmo pessoas leigas, através da utilização do sistema e alguma forma de pesquisa de satisfação dos usuários, contendo parâmetros de avaliação do sistema

6.10. Sumário e Conclusões

O teste do sistema foi de suma importância para avaliar as funcionalidades desenvolvidas, detectar limitações na arquitetura e na implementação do sistema, e ter uma visão melhor de todo o processo. Cada teste realizado permitiu de alguma maneira a descoberta de pontos até então ignorados parcial ou totalmente, o que certamente foi válido para a formulação de melhorias e futuras implementações no sistema existente.

Capítulo 7

7. CONCLUSÕES E TRABALHOS FUTUROS

7.1. Introdução

A proposta desta dissertação foi o estudo de técnicas de Realidade Aumentada e diversas outras áreas de conhecimento relacionadas, com o objetivo de prover uma ferramenta capaz de simular e visualizar com facilidade, o batimento cardíaco humano. Para alcançar estes objetivos, foram realizadas diversas tarefas, indo desde o estudo de sistemas existentes com temas convergentes ao proposto e até a examinação de assuntos mais específicos como a fisiologia do coração e a teoria da Cardiografia.

Todo esse processo possibilitou, ao fim desta dissertação, a disponibilização de um sistema em Realidade Aumentada para a visualização de sinais cardíacos, bem como a percepção de algumas conclusões e melhorias passíveis de serem acrescidas ao sistema atual.

Assim, a seguir são apresentadas as conclusões obtidas com este trabalho, seguidas das futuras melhorias para este sistema.

7.2. Conclusões

Ao início desta pesquisa, a proposta para o sistema a ser desenvolvido envolveu a investigação das diversas fontes de dados de sinais cardíacos. Esta investigação conduziu às atuais opções existentes no sistema, contemplando os meios mais comuns de armazenamento e troca de conhecimento sobre o funcionamento do ciclo cardíaco.

Longe de ser perfeito, o atual sistema é apenas um protótipo, um estudo de viabilidade da integração das tecnologias de Realidade Aumentada e Medicina. É necessário um maior aperfeiçoamento em alguns aspectos do mesmo antes de atingir níveis plenamente satisfatórios de produtividade. Apesar dessas limitações, os testes realizados mostraram que existem sim grandes possibilidades de aplicação para o sistema proposto, e que com a sua utilização é possível auxiliar diversos grupos de pessoas. Sujeitos com problemas cardíacos podem visualizar arritmias, aceleração dos batimentos, e simular seus próprios exames de eletrocardiogramas, bem como armazená-los de forma digital, caso estejam impressos em papel. Estudantes das Ciências da Saúde podem entender alguns princípios de funcionamento do coração,

inserir valores fixos de batimento e estabelecer paralelos entre os gráficos de exames e o funcionamento do coração. Além destes, pessoas comuns que busquem expandir seus conhecimentos podem estudar diversos parâmetros do coração, bem como visualizar em tempo real seus batimentos cardíacos, em ambiente de Realidade Aumentada.

Agora, com o objetivo de situar o sistema entre as aplicações existentes na sua área de concentração, a Tabela 7.1 traz novamente os sistemas discutidos no Capítulo 3, juntamente com o sistema, mostrando suas características e funcionalidades.

Características / Sistemas	Modelagem dos órgãos virtuais a partir de dados fisiológicos reais	Processamento de múltiplas entradas de dados	Simulação dos comportamentos dos órgãos reais nos modelos virtuais	Processamento de Sinais Biológicos em Tempo Real	Visualização / Interação em Realidade Virtual	Visualização / Interação em Realidade Aumentada	Utilização em situações reais
Project On Image Guided Surgery	■					■	■
Projecto Vida					■		
Die Heart	■		■			■	
Linte, C. A. et al.	■					■	
Sistema Desenvolvido		■	■	■		■	

Tabela 7.1 – Características dos Sistemas Estudados e do sistema desenvolvido.

Como pode ser visto, as características para as quais o sistema foi proposto foram alcançadas, podendo ser caracterizado como um sistema que aceita múltiplas entradas e formas de dados cardíacos, simula comportamentos fisiológicos dos órgãos reais através de órgãos virtuais, inclusive processando esses dados em tempo real e devolvendo a resposta visual em ambiente de Realidade Aumentada para o sujeito.

Dentre as características da Tabela não contempladas pelo sistema, pode-se destacar como maior ponto de possibilidade de melhorias a “Modelagem dos órgãos virtuais a partir de dados fisiológicos reais”, pois através desse tipo de técnica será possível ao sistema aproximar-se ainda mais dos comportamentos fisiológicos reais do coração.

A utilização do sistema em situações reais não foi ainda contemplada, pois o sistema esteve até o momento em fase de projeto e implementação, mas não há impedimentos de qualquer ordem para que isso seja alcançado.

Dentre as maiores contribuições trazidas pelo sistema, é possível destacar:

- O processamento das formas mais utilizadas de armazenamento de sinais cardíacos, com destaque para a possibilidade de resgatar dados armazenados em mídias analógicas como os eletrocardiogramas em papel.
- O processamento dos sinais cardíacos em tempo real, devolvendo de forma imediata e realista, respostas visuais para os seus sujeitos.
- A visualização dos órgãos em ambiente de Realidade Aumentada, que propiciam uma interação mais natural entre o sujeito e a simulação, habilitando-o a ver seu próprio coração bater sobre o peito, em uma visualização enriquecida do mundo real.

7.3. Trabalhos Futuros

No capítulo anterior, foram detectados alguns pontos de maior deficiência do sistema. Alguns deles podem ser mais facilmente resolvidos, enquanto outros podem fugir ao controle e escopo deste projeto. Assim, são propostas a seguir as soluções mais viáveis para os problemas encontrados.

- Reconhecimento das marcas muito sensível à condições de luminosidade e reflexos – Apesar de serem as mais populares formas de Realidade Aumentada, aplicações baseadas em algoritmos de limiarização para a detecção de padrões como o ARToolKit, constantemente apresentam este tipo de problema, uma vez que as próprias condições físicas dos materiais dos quais são feitos os marcadores os limitam e criam condições de deficiências, como sombras e reflexos de luz. O processamento de imagem, juntamente com as técnicas de visão computacional ainda têm um longo caminho pela frente, até que sejam desenvolvidas técnicas que possibilitem, com perfeição, reconhecer objetos e toda sorte de situações visuais existentes no mundo real. Enquanto essas técnicas não existirem ou forem inviáveis para os computadores pessoais, podem ser adotadas medidas paliativas de modo a melhorar os problemas de reconhecimento das marcas, como a confecção das mesmas em materiais mais foscos, com baixos níveis de refletividade. O controle das condições de iluminação e da qualidade da câmera, se usados com sabedoria podem reduzir grandemente os problemas de reconhecimento.
- Animação Complexa muito lenta para Carregar – O carregamento de muitos modelos é uma tarefa complexa e que demanda recursos

computacionais intensos. A forma de animação complexa criada, utiliza 20 modelos para realizar o ciclo cardíaco completo. Cada modelo possui sua complexidade geométrica e mapeamento de texturas que precisam ser carregados para a simulação ser executada. Possíveis soluções em software para a continuidade de utilização dessa forma de animação, são a redução mais drástica da complexidade geométrica do modelo de coração simulado, a utilização de texturas em resoluções mais moderadas e mesmo estudos quanto à redução na quantidade de modelos da animação. Porém soluções mais eficientes existem, como o desenvolvimento de uma função para carregamento e animação de objetos em OpenGl ou Directx.

- Processamento de apenas uma linha de ECG por vez – Este problema pode ser sanado se for trabalhada uma forma de o sujeito definir os limites de cada linha e processá-las de forma separada, armazenando-as em diferentes canais de dados. Uma forma bastante intuitiva de se fazer isso seria através de uma ferramenta similar ao apagador de ruídos já implementado, onde ao invés de apagar, seriam pintadas as áreas de cada linha de gráfico.
- Sensor com pouca capacidade de adaptação - O propósito do sensor utilizado era de mostrar e validar a possibilidade de integração entre a Realidade Aumentada e o processamento em tempo real de sinais captados por sensores. Como esse objetivo foi alcançado, para futuras implementações podem ser utilizados sensores de melhor fidelidade, como oxímetros de pulso hospitalares ou eletrocardiógrafos com saídas digitais. A integração com a porta paralela também pode ser facilmente portada para a porta serial, tornando as ligações com o computador mais compactas e discretas. Além disso, nos computadores mais modernos nota-se a tendência geral de não dar mais suporte à porta paralela. Assim, uma forma mais definitiva de atualizar a interface do sistema com os sensores, seria a implementação da conexão via USB, pois estes conectores estão presentes em todos os computadores atuais.
- Validação do sistema – O sistema foi testado com apenas 2 sujeitos, pretende-se futuramente, avaliá-lo junto a um conjunto maior de indivíduos, sendo incluídos neste grupo portadores de doenças cardíacas. Também pretende-se avaliar diferentes situações de esforço físico, como caminhadas,

período de repouso e esforço físico intenso, avaliando o comportamento do sistema perante essas variações. Outro ponto importante é a interação com profissionais de Cardiologia, estudantes e todo o restante do público-alvo do sistema, através da utilização do sistema tanto como usuários quanto como sujeitos e de uma avaliação em forma de pesquisa dos diversos aspectos relevantes do sistema.

7.4. Considerações Finais

A Realidade Aumentada ainda não é a interface perfeita de interação, mas já demonstra suas potencialidades em muitas e muitas áreas, surgindo como mais uma forma de auxiliar o Homem na sua compreensão do mundo e de si mesmo. E este é o propósito que permeou a pesquisa e desenvolvimento do sistema, o servir à Humanidade através da tecnologia.

Espera-se que esta pesquisa possa servir como fonte de auxílio e informação para outros estudantes e pesquisadores, auxiliando-os na tomada de decisões quanto aos temas aqui apresentados e servindo como um guia para o desenvolvimento de aplicações e pesquisas relacionadas a este tema.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Moriwaki, K. **Between the skin and the garden: new modes of interaction in the wearable data environment**. In Horizon Zero, vol.7. Disponível em http://www.horizonzero.ca/textsite/wear.php?is=16&file=7&tlang=0. Acesso em 27 de Agosto de 2008.

[2] Preece, J. **Human-computer interaction**. Addison-Wesley Pub Co,1994.

[3] Terzian, F. **Com as mãos ou com a mente.** In Revista Exame, Edição 905, Novembro de 2007. Disponível em: http://portalexame.abril.com.br/revista/exame/edicoes/0905/tecnologia/m0142205.html. Acesso em 27 de Agosto de 2008.

[4] Tori, Romero; Kirner, C.; Siscouto, R.. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. 1. ed. Porto Alegre: Sociedade Brasileira de Computação - SBC, 2006. v. 1. 369 p.

[5] PlayStation. (2008) **PlayStation Global**. Disponível em: http://www.playstation.com/. Acesso em 28 de março de 2008.

[6] Wii. (2008) **Wii at Nintendo.** Disponível em : http://wii.nintendo.com/. Acesso em 28 de março de 2008.

[7] Xbox. (2008) **Microsoft Xbox.** Disponível em: http://www.microsoft.com/xbox/. Acesso em 28 de março de 2008.

[8] Azevedo, E. (2007) **Cinema X Videogames.** Disponível em: http://br.geocities.com/elinaldoazevedo3/analises/CinemaVideogames.pdf. Acesso em 28 de março de 2008.

[9] MATRIS. **Markerless real-time Tracking for Augmented Reality Image Synthesis.** Disponível em: http://www.ist-matris.org/. Acesso em 27 de agosto de 2008.

[10] MIT (1999). **Project on Image Guided Surgery**. Disponível em: http://groups.csail.mit.edu/vision/medical-vision/surgery/surgical_navigation.html. Acesso em 28 de março de 2008.

[11] Shuhaiber, J.H. **Augmented Reality in Surgery.** *Arch Surg:2004;139:170-174*. Disponível em: http://archsurg.ama-assn.org/cgi/content/full/139/2/170#ACK. Acesso em 11 de Novembro de 2007.

[12] Lopes, K.; Junior, E. A. L.; Cardoso, A. **Uma aplicação de realidade aumentada no auxílio da adaptação de próteses em membros superiores**. *In: I WARV, 2005, Workshop de Aplicações de Realidade Virtual*. Uberlândia- MG 2005

[13] Pereira, J. M. L. B. **A Realidade Aumentada na Engenharia Biomédica: Estado da Arte.** in Actas da 5ª Workshop em Engenharia Biomédica, Secção de Sistemas e Controlo, Instituto Superior Técnico/UTL, Lisboa, Portugal, 2000**.** Disponível em: http://ltodi.est.ips.pt/jbraz/ficheiros/pubs/2000_5web_jbraz.pdf. Acesso em 11 de Novembro de 2007.

[14] Arruda, A. **Interpretação de Exames ainda Determina Diagnósticos.** Folha Online – Equilíbrio, Novembro de 2004. Disponível em: http://www1.folha.uol.com.br/folha/equilibrio/noticias/ult263u3738.shtml. Acesso em 27 de agosto de 2008.

[15] Monteiro, R. H. **Videografias do Coração.** História, Ciência e Saúde - Manguinhos, Rio de Janeiro, v. 9, n. nº 2, p. 452-453, 2002. Disponível em: http://www.fac.org.ar/tcvc/llave/c378/monteira.PDF. Acesso em 27 de agosto de 2008.

[16] Penna, C. C. **O Processo de Diagnóstico.** in Artigos Gratuitos Online – Artigonal. Disponível em: http://www.artigonal.com/medicina-artigos/o-processo-de-diagnostico-395104.html. Acesso em 27 de agosto de 2008.

[17] Penna, C. C. **Sistemas de Apoio Diagnóstico.** in Artigos Gratuitos Online – Artigonal. Disponível em: http://www.artigonal.com/medicina-artigos/sistemas-de-apoio-diagnostico-395187.html. Acesso em 27 de agosto de 2008.

[18] Pereira, J. B. **VIDA: Visualizador Interactivo de Dados biomecânicos Aumentados**. in Actas do 12º Encontro Português de Computação Gráfica, Instituto Superior de Engenharia do Porto, Porto, Portugal, Outubro de 2003. Disponível em: http://ltodi.est.ips.pt/jbraz/ficheiros/pubs/2003_12epcg_jbraz.pdf. Acesso em 28 de março de 2008.

[19] Rosário, L. B. et al. **Die Heart - Augmented Reality Visualization of the Left Ventricle of the Human Heart, using a Tangible Interface.** Disponível em: http://www.adetti.pt/DieHeart/AugmentedReality.htm. Acesso em 28 de março de 2008.

[20] Linte, C. A. et al. **On Enhancing Planning and Navigation of Beating-Heart Mitral Valve Surgery Using Pre-operative Cardiac Models.** in Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. Disponível em: http://www.imaging.robarts.ca/~tpeters/content/grouppubs/2007/C_Linte_EMBS_2007_Manuscript.pdf. Acesso em 28 de março de 2008.

[21] Kirner, C. ; Tori, R. **Realidade Virtual: Conceitos, Tecnologia e Tendências**. 1ed. São Paulo, 2004.

[22] Xupa Xipa. **Vídeo Game por Mouse e Teclado.** Disponível em: http://img511.imageshack.us/img511/7633/fullsetrv4.jpg . Acesso em 09 de julho de 2008.

[23] Dreyer, V. **A Máquina do Tempo.** Disponível em: http://www.artcoustic.com/Latest%20News/Goethe%20cinema%20front.jpg. Acesso em 09 de Julho de 2008.

[24] Kopp, C. **Helmet Mounted Sights and Displays.** Disponível em: http://www.ausairpower.net/hmd-technology.html. Acesso em 17 de Junho de 2008.

[25] Raposo, Alberto; Szenberg, F; Gattass, M; Celes, W. **Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração**. In: A. M. S. Andrade, A. T. Martins, R. J. A. Macêdo (eds.), Anais do XXIV Congresso da Sociedade Brasileira de Computação, Vol. 2, XXIII JAI - Livro Texto, Cap. 7, p.289 - 331. SBC, Brazil, 2004. Disponível em: http://www.tecgraf.puc-rio.br/~abraposo/pubs/JAI2004/JAI2004-cap7.zip. Acesso em 17 de Junho de 2008.

[26] Lipton, L. (1982) Foundations of the Stereoscopic Cinema – A Study in Depth. Van Nostrand Reinhold Company.

[27] Hansma, Paul K. 1996. View Camera Focusing in Practice. Photo Techniques, March/April 1996, 54–57.

[28] Ray, Sidney F. 2002. Applied Photographic Optics. 3rd ed. Oxford.

[29] Wikipédia. **Luva Virtual.** Disponível em: http://pt.wikipedia.org/wiki/Luva_digital. Acesso em 24 de junho de 2008.

[30] Vrealities. **Virtual Head Mounted Display.** Disponível em: http://www.vrealities.com/5dt.html. Acesso em 09 de julho de 2008.

[31] Blay, F. **Virtual Reality Technology.** Disponível em: http://www.fbley.de/lev2_vrtech.htm. Acesso em 09 de julho de 2008.

[32] SaukValley. **Graduate student develops virtual reality game glove.** Disponível em: http://www.saukvalley.com/articles/2006/12/06/news/state/374701713391302.txt. Acesso em 10 de julho de 2008.

[33] Beardsley, Paul; Raskar, Ramesh; Forlines, Clifton; Baar, Jeroen V. **Interactive Projection.** in International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2004 Emerging technologies. Disponível em: http://www.merl.com/papers/docs/TR2004-107.pdf. Acesso em 24 de junho de 2008.

[34] Milgram, P. et al. **Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum, Telemanipulador and Telepresence Technologies.** SPIE, V. 2351, p. 282-292, 1994.

[35] Kirner, C. ; TORI, R. **"Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade".** In: Claudio Kirner; Romero Tori. (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. 1ed. São Paulo, v. 1, p. 3-20. 2004.

[36] Azuma, R. T. **Tracking Requirements for Augmented Reality.** Communications of the ACM, 36(7):50-51, July 1993.

[37] Bajura, M.; Neumann, U. **"Dynamic Registration Correction in Video-Based Augmented Reality Systems".** IEEE Computer Graphics & Applications, v.15, n.5. p.52-60. 1995.

[38] Boman, D. K. **International Survey: Virtual Environment research.** IEEE Computer, 28(6):57-65. Junho. 1995.

[39] Feiner, S. et al. **Knowledge-Based Augmented Reality.** Communications of the ACM, 36(7):52- 62. Julho. 1993.

[40] Billinghurst, M. et al. **"The MagicBook - Moving Seamlessly between Reality and Virtuality".** Computer Graphics and Applications, v. 21, n.3, p.2-4. 2001.

[41] Santin, R. et al. **"Ações interativas em Ambientes de Realidade Aumentada com ARToolKit".** VII Symposium on Virtual Reality, SBC, p. 161-168. 2004.

[42] Zhou, Z. et al. **"Interactive Entertainment Systems Using Tangible Cubes".** Australian Workshop on Interactive Entertainment, p. 19-22. 2004.

[43] Azuma, R. T. et al. **Recent Advances in Augmented Reality.** IEEE Computer Graphics and Applications, v. 21, n. 6, p. 34-37, 2001.

[44] Azuma, R. T. **A Survey of Augmented Reality.** Presence: Teleoperators and Virtual Environments, v.6, n.4, p. 355-385.1997.

[45] ARToolKit. **ARToolKit HomePage.** Disponível em: http://www.hitl.washington.edu/artoolkit/. Acesso me 24 de junho de 2008.

[46] Free Software Foundation . **GNU General Public License v2.0 - GNU Project - Free Software Foundation (FSF).** Disponível em: http://www.gnu.org/licenses/gpl-2.0.html. Acesso em 01 de Julho de 2008.

[47] Guyton, A. C. ; Hall, J. E. **Tratado de Fisiologia Médica.** 9ª Ed. Guanabara Koogan.1996.

[48] Carvalho, A. P. ; Costa, A. F. **Circulação e Respiração, Fundamentos de Biofísica e Fisiologia.** 2ª Ed. MEC. 1976.

[49] InterCromo. **Catálogo de Eletrocardiógrafos.** Disponível em: http://intercromo.com/montra/images/Produtos/bionet/ECG%20DE%2012%20CANAIS%20C%20MONITOR.jpg. Acesso em 27 de agosto de 2008.

[50] VirtualMedicalCentre. **Electrocardiogram.** Disponível em: http://www.virtualcancercentre.com/Investigations.asp?sid=28. Acesso em 27 de agosto de 2008.

[51] Lyons, A. S. ; Petrucelli, R.J. **The Nineteenth Century: The Beginnings of Modern Medicine**. In: Medicine - an illustraded history. New York, Abradale Press-Harry N. Abrams Inc, Publishers, 1987. p. 510-11.

[52] Epstein, O. et al. **Clinical Examination.** Londres, Editora Mosby, 1997, pags .

[53] Wright J. **Introduction to pulse oximetry**. Br J Perioper Nurs. 2003; 13:456–460.

[54] MIT. **MIT.** Disponível em: http://web.mit.edu/. Acesso em 09 de julho de 2008.

[55] BWH. **Brigham and Women's Hospital.** Disponível em: http://www.brighamandwomens.org/. Acesso em 09 de julho de 2008.

[56] Wikipedia. **Mouse (computing).** Disponível em: http://en.wikipedia.org/wiki/Mouse_%28computing%29. Acesso em 10 de julho de 2008.

[57] mxToolKit. **mxToolKit manual.** Disponível em: http://chumbalum.swissquake.ch/mxtk/index.html. Acesso em 10 de julho de 2008.

[58] Aptec. **Ecocardiografia Transesofágica.** Disponível em: http://www.aptec.pt/index.php?option=com_content&task=view&id=55&Itemid=0. Acesso em 17 de julho de 2008.

[59] Otto, C. M. **Fundamentos de Ecocardiograma Clínica.** Editora Elsevier, 3ª Ed, 2005.

[60] Maia, R. **A Linguagem C++.** Disponível em: http://www.inf.puc-rio.br/~maia/apres/cpplang.pdf. Acesso em 06 de agosto de 2008.

[61] Borland. **Borland: C++ Builder 2006.** Disponível em: http://www.borland.com/br/products/cbuilder/index.html. Acesso em 06 de agosto de 2008.

[62] Microsoft. **Visual C++ Develop Center.** Disponível em: http://msdn.microsoft.com/en-us/visualc/default.aspx. Acesso em 06 de agosto de 2008.

[63] Bloodshed. **Bloodshed Software – Dev-C++.** Disponível em: http://www.bloodshed.net/devcpp.html. Acesso em 06 de agosto de 2008.

[64] Sunisoft. **Delphi VCL Skin & Web Update Builder & Art/Icon Design – Sunisoft**

[65] IrfanView. **IrfanView – Official HomePage.** Disponível em: http://www.irfanview.com/. Acesso em 11 de agosto de 2008.

[66] Deitel, M.D. ; Deitel, P. J. **C++ Como Programar.** 5ª Ed. Pearson / Prentice Hall, 2006.

[67] Gonzalez, R. C.; Woods, R. E. **Processamento de Imagens Digitais.** 1ª Ed. 2000. Edgar Blücher, 2000.

[68] Ashihkmin, M. **High Quality Chroma Key.** Disponível em: http://bmrc.berkeley.edu/courseware/cs294-3/fall97/projects/ashikhmin/. Acesso em 11 de agosto de 2008.

[69] Quines, R. **Protegendo a Porta Paralela com Acoplador Óptico.** Disponível em: http://www.rafaelquines.com/blog/2007/04/protegendo-a-porta-paralela-com-acoplador-optico/. Acesso em 11 de agosto de 2008.

[70] Logix4u. **How Inpout32.dll works.** Disponível em: http://logix4u.net/Legacy_Ports/Parallel_Port/How_Inpout32.dll_works_.html. Acesso em 12 de agosto de 2008.

[71] 100FPS.COM. **How many frames per second can the human eye see?.** Disponível em: http://www.100fps.com/how_many_frames_can_humans_see.htm. Acesso em 29 de Maio de 2008.

[72] Autodesk. **Autodesk® 3ds Max®.** Disponível em: http://www.autodesk.com/3dsmax. Acesso em 05 de maio de 2008.

[73] W3C. **VRML Virtual Reality Modeling Language.** Disponível em: http://www.w3.org/MarkUp/VRML. Acesso em 05 de maio de 2008.

[74] OpenGl. **OpenGl – "The Industry Standard for High Performance Graphics".** Disponível em: http://www.opengl.org/. Acesso em 05 de maio de 2008.

[75] Ferreira, Levi. **Computação Gráfica – Radiosidade.** Disponível em: http://www.inf.ufsc.br/~awangenh/CG/radiosidade/index.html. Acesso em 05 de maio de 2008.

[76] Bertulani, C. **Luz e Cor.** Disponível em: http://www.if.ufrj.br/teaching/luz/cor.html. Acesso em 29 de Maio de 2008.

[77] Wikipedia. **Frame Rates.** Disponível em: http://en.wikipedia.org/wiki/Frame_rate. Acesso em 29 de maio de 2008.

[78] Walrath, J. **30 Frames per Second vs. 60 Frames per Second.** Disponível em: http://www.daniele.ch/school/30vs60/30vs60_1.html. Acesso em 29 de maio de 2008.

[79] Wikipedia. **Key Frame.** Disponível em: http://en.wikipedia.org/wiki/Key_frame. Acesso em 01 de junho de 2008.

[80] Wikipedia. **Tweening.** Disponível em: http://en.wikipedia.org/wiki/Tweening. Acesso em 01 de junho de 2008.

[81] Ramos, Z. M. **Actualidade.** Disponível em: http://zmramos.110mb.com/Blog/actualidade.php. Acesso em 09 de julho de 2008.

[82] Shopping Online. **Shopping Online.** Disponível em: http://www.safarishop.com.br/pl_site/images/powerlight_mais01.jpg. Acesso em 09 de julho de 2008.

[83] Brunoproducoes. **FAQ.** Disponível em: http://www.brunobrissosproducoes.com/faq.html. Acesso em 09 de julho de 2008.

[84] Poets, C. ; Barker, S. J. ; Hay, B. ; Miyasaka, K. Pulse Oximetry – Principles. Disponível em: http://www.oximetry.org/IMAGES/cpo2.gif . Acesso em 27 de agosto de 2008.

APÊNDICE 1 - PUBLICAÇÕES

A seguir são apresentadas as publicações resultantes das pesquisas realizadas no decorrer deste trabalho:

[1] BUCCIOLI, Arthur Augusto Bastos; ZORZAL, Ezequiel Roberto; CARDOSO, Alexandre; KIRNER, Claudio; LAMOUNIER JÚNIOR, Edgard. Usando Realidade Aumentada para visualização de Informações de Tráfego em Redes de Computadores. In: WRA2006 - III WORKSHOP SOBRE REALIDADE AUMENTADA, 2006, Rio de Janeiro, RJ, Brasil. 2006.

[2] BUCCIOLI, Arthur Augusto Bastos; LAMOUNIER JÚNIOR, Edgard; CARDOSO, Alexandre; KIRNER, Cláudio. A Utilização De Realidade Aumentada Com Objetos Virtuais Dinâmicos No Treinamento De Medicina. In: II WORKSHOP DE APLICAÇÕES DE REALIDADE VIRTUAL - WARV2006, Pernambuco. 2006.

[3] BUCCIOLI, Arthur Augusto Bastos; LAMOUNIER JÚNIOR, Edgard; CARDOSO, Alexandre; KIRNER, Cláudio. ARBioMed – Sistema Em Realidade Aumentada Para Visualização E Simulação De Sinais Eletrocardiográficos. In: Workshop de Realidade Virtual e Aumentada 2007 - WRVA'07, 2007,Itumbiara,MG,Brasil. 2007.

ANEXO A – TAREFAS RELACIONADAS À REALIDADE VIRTUAL

Para que seja possível uma implementação bem sucedida de um sistema de Realidade Virtual, é necessário primeiro realizar diversas tarefas relacionadas com a criação do ambiente virtual, bem como a definição das formas de interação que estarão presentes nesse mesmo ambiente. Dentre as principais destas tarefas, a modelagem, texturização e a animação podem ser destacadas e serão detalhadas a seguir.

O mundo real é composto por elementos tridimensionais com características geométricas bem definidas. Para reproduzir essas características em um ambiente virtual é necessário utilizar aproximações, visto que mesmo os computadores mais avançados apresentam limitações de memória e processamento. Foram desenvolvidas diversas técnicas computacionais com o objetivo de contornar as limitações do hardware, porém a mais proeminente e amplamente utilizada na Realidade Virtual é a representação poligonal de objetos tridimensionais, presente na maioria das linguagens e pacotes gráficos de modelagem tridimensional, bem como nas placas gráficas com capacidades tridimensionais.

Nesta técnica, os objetos são modelados e representados a partir de polígonos interligados em um ambiente tridimensional. A Figura A.1 mostra a representação de uma esfera utilizando o sistema poligonal, bem como os elementos presentes nesse sistema.

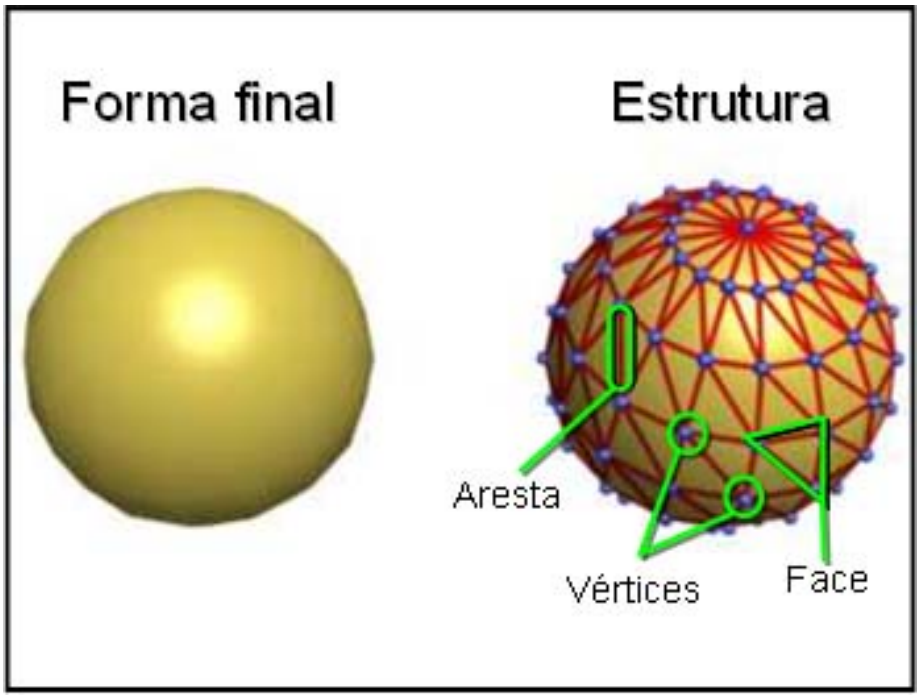

Figura A.1 – Sistema de Modelagem Poligonal.

A utilização do sistema poligonal torna possível a representação de grande parte das entidades do mundo real, porém a construção de modelos baseada unicamente na definição de polígonos pode mostrar-se demasiadamente lenta, o que tornaria inviável a produção de modelos mais detalhados em curto prazo. Para resolver essas questões existem técnicas visando facilitar o processo de modelagem. Por exemplo, uma esfera pode ser definida por uma fórmula matemática e posteriormente traduzida para uma representação poligonal. Também é possível utilizar outras formas primitivas geradas a partir de fórmulas e parâmetros, como cubos e cilindros e combina-las, produzindo assim formas mais complexas. Essa forma de modelagem é amplamente utilizada em softwares de modelagem como o 3ds Max® [72] da Autodesk® e também em linguagens de Realidade Virtual como a VRML [73] e OpenGl [74]. As Figuras A.2 e A.3 exemplificam as primitivas do 3ds Max® e da VRML, respectivamente e a Figura A.4 demonstra a utilização de primitivas combinadas para produzir objetos mais complexos.

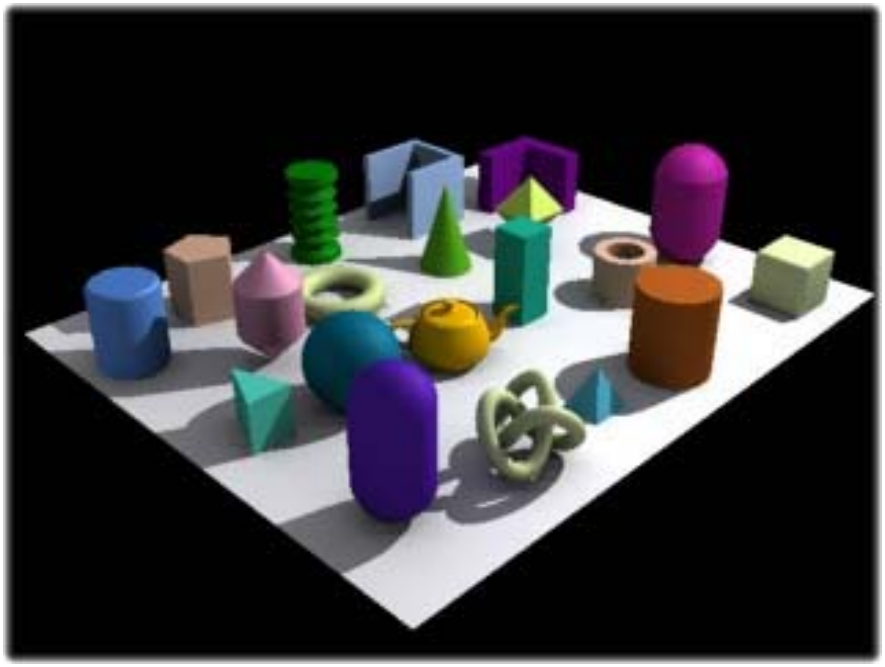

Figura A.2 – Primitivas do 3ds Max®.

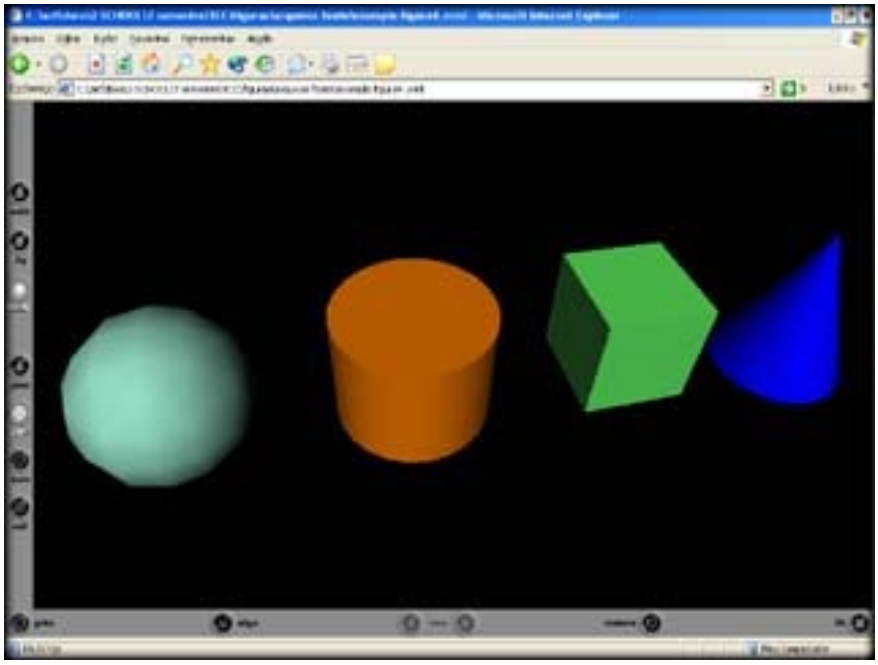

Figura A.3 – Primitivas do VRML.

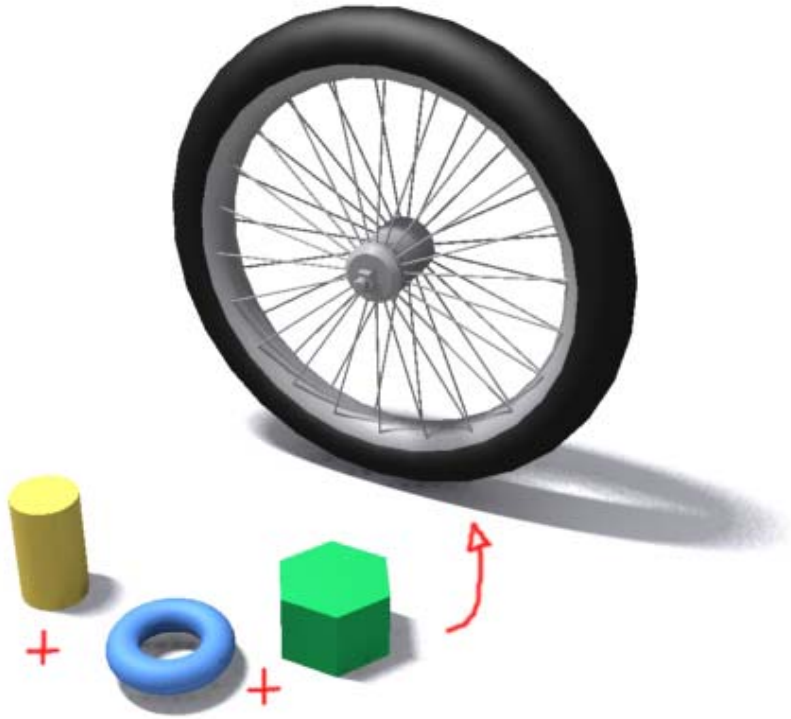

Figura A.4 – Criação de Objetos Partindo de Primitivas.

Existem situações onde a simples combinação de primitivas não satisfaz as necessidades do modelo a ser criado. Em geral, modelos orgânicos ou com muitas

curvas reduzem drasticamente a utilidade das primitivas, fazendo com que seja necessária a manipulação dos elementos poligonais para a criação dos modelos. Num caso como esses é muito mais produtivo criar os modelos em um pacote de modelagem como o 3ds Max® do que fazê-lo diretamente em uma linguagem de Realidade Virtual como a VRML definindo polígono por polígono, pois nestes pacotes são oferecidas diversas ferramentas que tornam menos árdua a tarefa de modelar através de polígonos. Algumas dessas ferramentas lidam diretamente com os elementos básicos da modelagem poligonal como vértices e arestas enquanto outras, definidas por fórmulas matemáticas, podem alterar inteiramente a geometria dos objetos ou suas subpartes de diversas maneiras úteis em casos específicos. A Figura A.5 demonstra algumas etapas da criação de uma nave espacial utilizando o ambiente do software 3ds Max®, enquanto a Figura A.6 apresenta a utilização da ferramenta MeshSmooth, projetada para ser utilizada na suavização e aumento de detalhes em modelos poligonais.

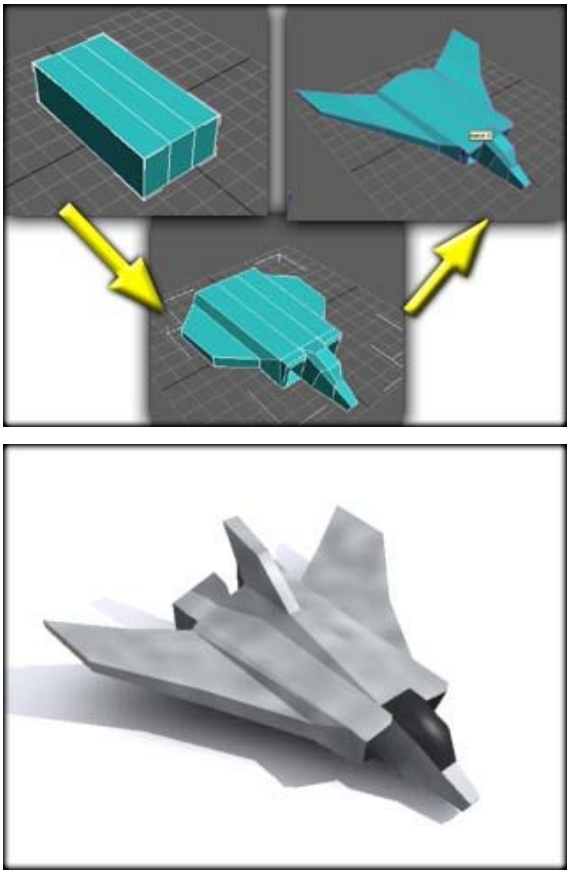

Figura A.5 – Criação de um Modelo Utilizando Modelagem Poligonal.

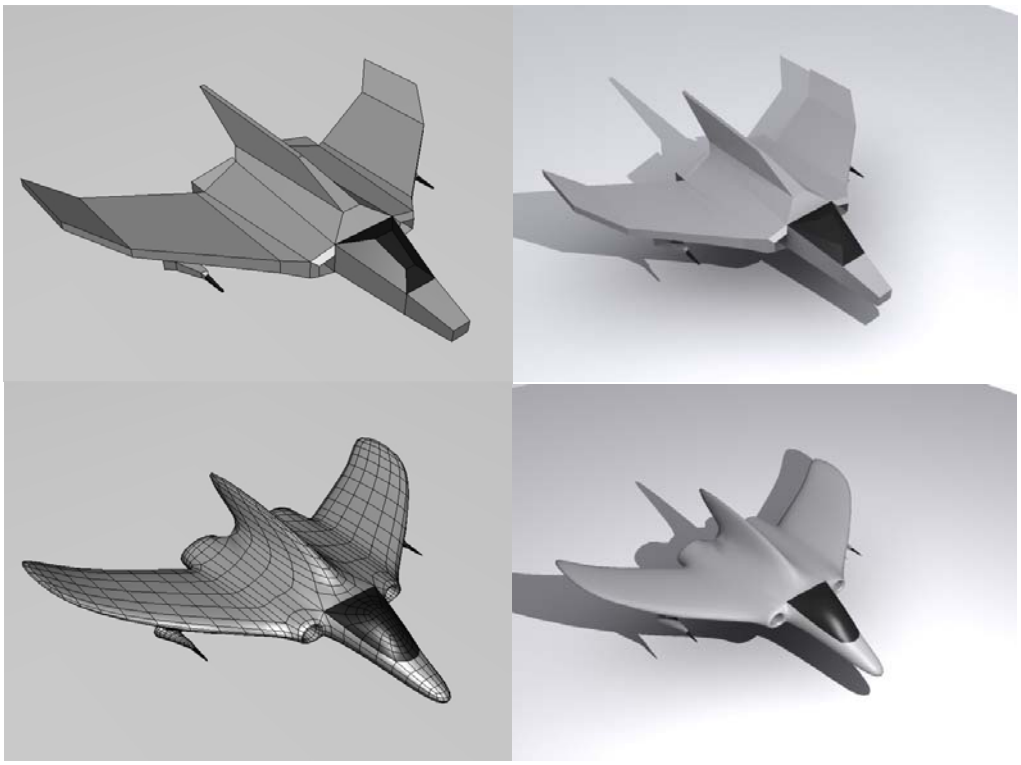

Figura A.6 – Utilização da Ferramenta MeshSmooth para suavização de modelos.

A boa utilização das ferramentas disponíveis nos pacotes de modelagem é fundamental para possibilitar a criação de modelos altamente detalhados em uma quantidade de tempo aceitável. No entanto, quando se trata de aplicações em tempo real, nem sempre é possível detalhar demais um determinado modelo, pois as placas aceleradoras possuem limites bem definidos para o processamento de polígonos, devendo assim os objetos serem feitos com a menor quantidade possível de polígonos, mantendo assim a performance da aplicação.

A tarefa de modelagem é muito importante para definir corretamente os detalhes geométricos dos objetos, porém por si só não permite que os objetos apresentem um nível de realismo satisfatório. Para complementar a modelagem existe a tarefa de texturização, cuja função primordial é definir a aparência dos objetos, através de características como cores, nível de brilho, polimento, transparência, entre outras. A Figura A.7 mostra um mesmo modelo geométrico com diversas configurações de texturas aplicadas.

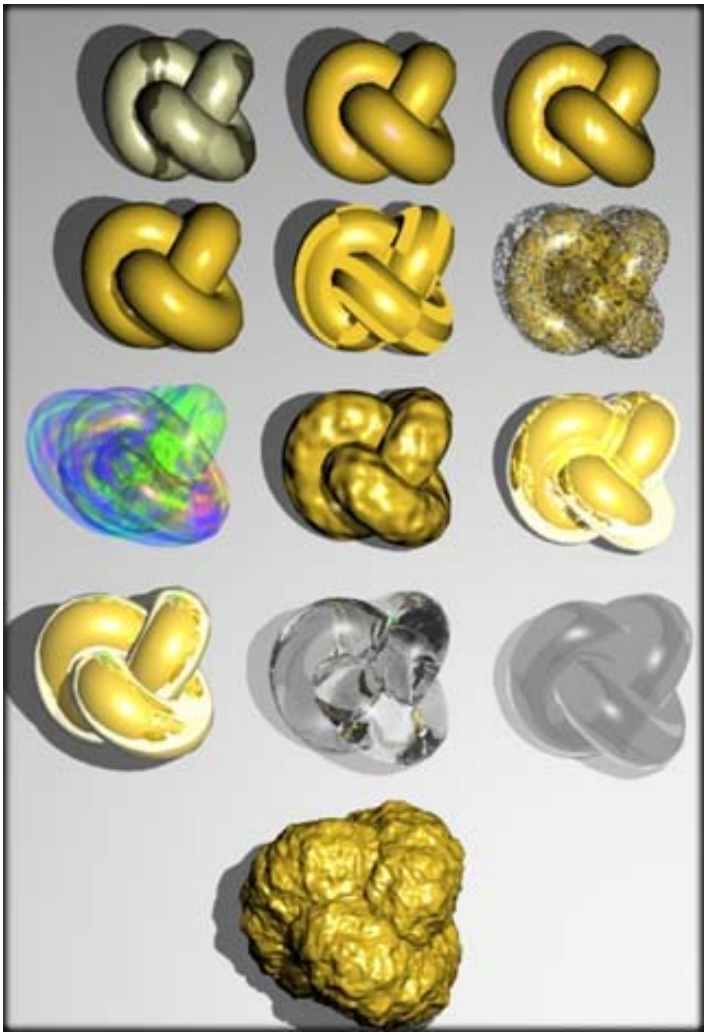

Figura A.7 – Diferentes configurações de texturização sobre um modelo.

Assim como na modelagem, é possível definir as propriedades de texturização dos objetos tanto de forma direta, utilizando-se de linguagens como OpenGl e VRML, ou através de softwares especializados como o 3dsMax. Freqüentemente é mais simples e produtivo utilizar as ferramentas providas pelos softwares para estas tarefas, pois a atualização das aparências nos objetos pode ser observada em tempo real, o que nem sempre acontece quando se está lidando com as linguagens de Realidade Virtual. Outro fato que encoraja o uso de tais softwares é a considerável complexidade representada por algumas das tarefas da texturização, como o mapeamento que será detalhado posteriormente.

A etapa de texturização envolve tanto os aspectos de aparência do objeto como cor, textura, rugosidade, como também o comportamento dos objetos em relação à luz, definindo, por exemplo, a refletância da luz, o formato desta refletância, a intensidade, cor, nível de polimento e transparência. A forma como essa separação é implementada

depende fortemente da linguagem de Realidade Virtual ou software gráfico utilizado. Como exemplo dessa separação pode-se citar a linguagem VRML, onde são configuradas separadamente as propriedades de comportamento perante a luz e as propriedades de texturização, tais como utilização de imagens e vídeos para definir a aparência dos objetos e opções de mapeamento. A junção destas duas configurações molda o comportamento e aparência final do material a ser aplicado sobre os objetos. A Tabela A.1 apresenta um trecho de código em VRML onde é definido um material com parâmetros de luminosidade e uma imagem como textura, bem como um objeto para exibir essas definições de texturas. A Figura A.8 apresenta o resultado visual para o código da Tabela A.1.

```
#VRML V2.0 utf8
  Transform {
     children Shape {
        appearance Appearance {
          material Material {
              shininess 1
              specularColor 0.2 0.2 0.2
          }
          texture ImageTexture { url "textura.jpg" }
        }
        geometry Box { size 1 1 1}
     }
  }
```

Tabela A.1 – Código em VRML para aplicação de Texturas em Objetos.

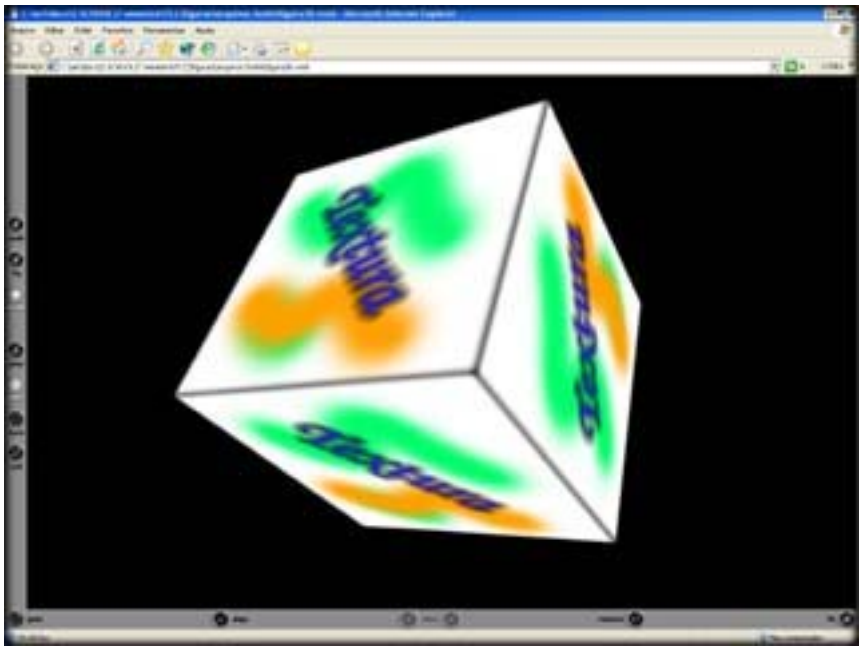

Figura A.8 – Resultado Visual para o Código da Tabela 1.

Como pôde ser observado, o código da Tabela A.1 produziu a imagem de um cubo com uma textura do tipo imagem aplicada a ele, bem como algumas reações à luz, embora menos visíveis devido à natureza geométrica do cubo. O parâmetro “shininess”

do código especifica o nível de polimento de um objeto, influenciando o tamanho dos reflexos perante a luz. Já o parâmetro “specularColor” serve para definir a cor dos reflexos apresentados pelo material na presença de luz. Existem em VRML outros parâmetros úteis para se definir o comportamento do material perante a luz, como o “transparency” que controla o nível de opacidade do material e o “diffuseColor” que define a cor de exibição do material em caso de não se utilizar texturas.

Para o exemplo dado foi utilizada uma textura do tipo imagem, no caso uma imagem no padrão jpeg. É comum tanto os softwares de modelagem como as linguagens de Realidade Virtual suportarem diversos formatos de imagens para utilização como texturas, bem como alguns formatos de vídeo. Texturas também podem ser criadas através de parâmetros e fórmulas matemáticas como os fractais. Uma das vantagens destas texturas é que elas geralmente podem ser criadas em vários tamanhos, não estando sujeitas a problemas de baixa resolução, bastante comuns se tratando de imagens e vídeos pré-capturados. Outra grande vantagem é a flexibilidade de modificação, pois a simples alteração dos parâmetros pode produzir texturas completamente diferentes. A Figura A.9 demonstra uma textura do tipo fractal criada no 3dsmax, bem como a interface de modificação dos seus parâmetros.

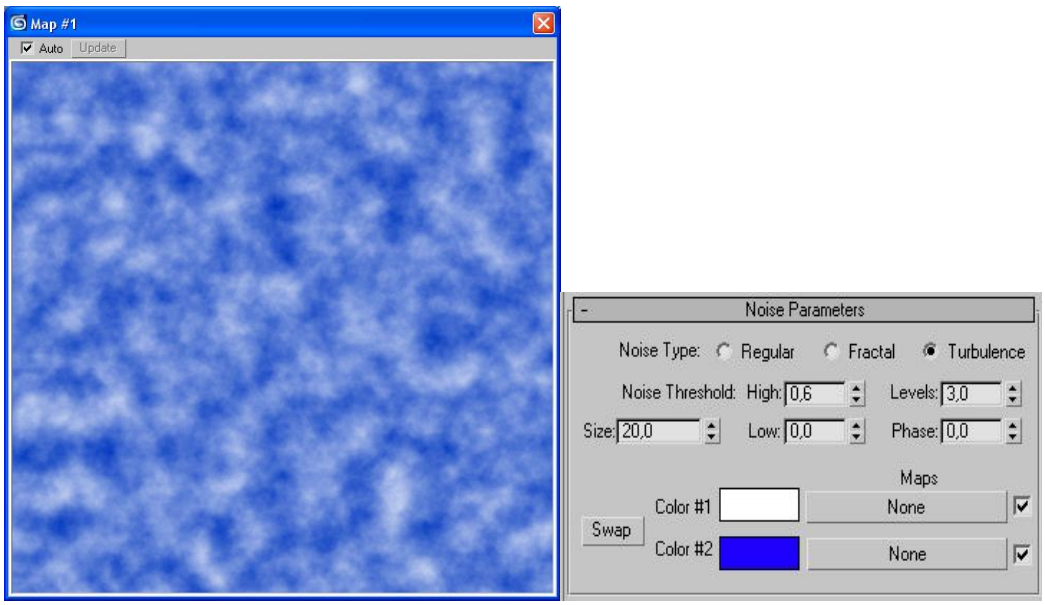

Figura A.9 - Textura Fractal criada em 3dsMax.

Tanto as texturas pré-capturadas quanto as criadas por parâmetros destinam-se à aplicação sobre objetos tridimensionais previamente modelados. Porém, a natureza dessas texturas é em geral plana, ou seja, bidimensional com largura e altura num

formato quadrado ou retangular. Para que essas texturas possam ser utilizadas é necessário informar à linguagem ou software de modelagem como será a forma de aplicação dessas texturas, como será a distribuição de cada pedaço da textura sobre a superfície do objeto. Essa distribuição é chamada mapeamento, e pode ser analogamente comparada ao ato de embalar um presente, onde o presente representa o objeto tridimensional e o papel de presente representa a textura a ser aplicada.

Existem diversas formas de se mapear uma textura sobre um objeto e cada linguagem e software as implementa de formas particulares. As formas mais comuns de mapeamento são baseadas em formulas matemáticas e algoritmos de distribuição sobre superfície baseados em sólidos primitivos como esferas, cubos e cilindros. A forma de utilização dessas formas de mapeamento é semelhante à modelagem de objetos através de primitivas. Em primeiro lugar é necessário analisar a forma do objeto a mapear e buscar uma forma de mapeamento que mais se adeque a esse formato. Com base nisso a textura é mapeada sobre o objeto de destino. A Figura A.10 exemplifica alguns tipos de mapeamento presentes no 3dsMax.

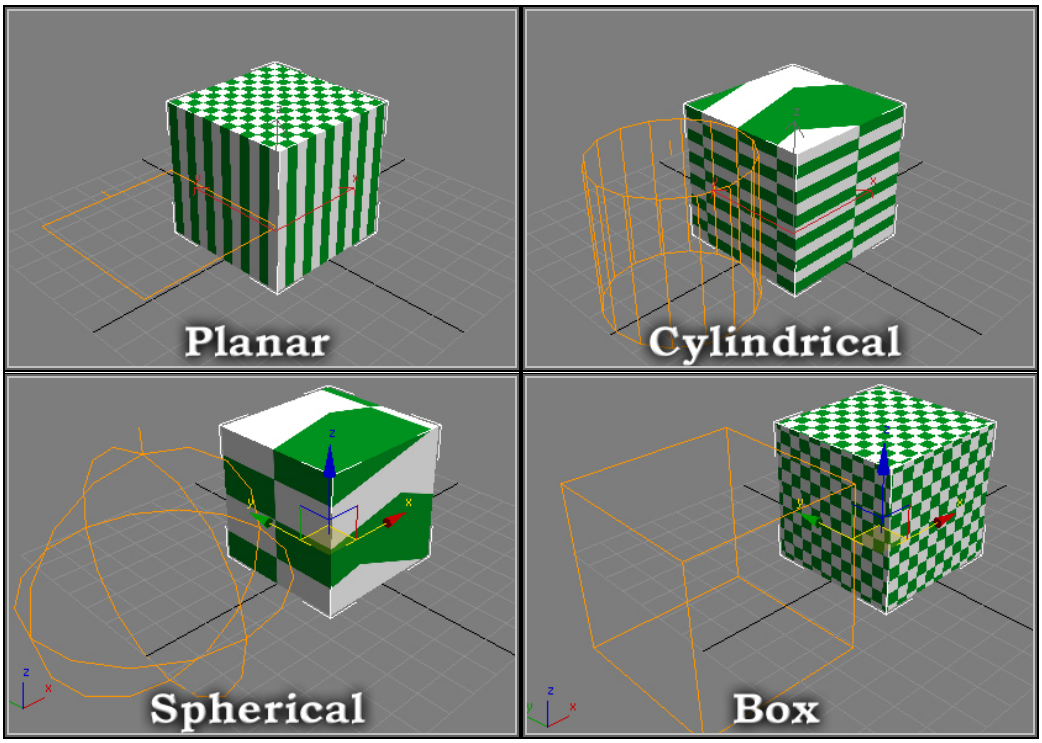

Figura A.10 – Algumas Formas de Mapeamento do 3dsMax.

No exemplo da Figura A.10 fica claro que mapeamento que mais se aproximou do ideal foi o Box, pois como o seu formato geométrico é bastante similar ao do objeto de destino da textura, a distorção resultante foi insignificante. Cabe ao artista saber

escolher entre cada forma de mapeamento com eficácia, sempre buscando a forma que mais se aproxime do objeto de destino.

Além dos mapeamentos simples, é possível também atribuir ao objeto mais de um mapeamento, o que possibilita a aplicação de texturas com sucesso em objetos de maior diversidade geométrica, onde um mapeamento único acarretaria sérias distorções sobre a textura aplicada.

Como foi dito anteriormente, a texturização define para os objetos, além das características de cor e texturas, as características de resposta à iluminação. Essas características são notadas quando existem luzes virtuais no ambiente de Realidade Virtual, através das mudanças de pontos de alto brilho e sombras nos objetos.

No mundo real, a luz é composta por milhões de fótons emitidos por uma ou mais fontes de luz que ao se depararem com algum objeto são por ele parcialmente absorvidos, sendo que essa absorção é diretamente relacionada com a cor do objeto. Por exemplo, se um objeto é vermelho, ele tenderá a absorver os espectros de todas as cores, exceto o vermelho, que será refletido para outros objetos, onde serão absorvidos mais e mais fótons, até que haja o fim das reflexões, quando todos os fótons foram absorvidos. Esse fenômeno é conhecido como radiosidade [75]. Representar computacionalmente o funcionamento da luz, tal como ela é no mundo real, não algo trivial, mesmo com os melhores computadores e placas de vídeo.

A Realidade Virtual presume aplicações capazes de gerar um certo número de quadros por segundo, de forma que a interatividade aconteça praticamente em tempo real. Para que os cálculos de iluminação possam ser feitos em velocidade aceitável em aplicações de Realidade Virtual, foram criadas diversas técnicas de iluminação mais simples, cujos resultados não são os melhores possíveis visualmente, mas possibilitam o cálculo suficientemente rápido para satisfazer os requisitos de interatividade da Realidade Virtual.

Para representar as fontes de luz mais comuns no mundo real são implementadas luzes virtuais correspondentes, presentes comumente nas linguagens de Realidade Virtual e nos softwares de modelagem. O foco destas implementações está na forma de emissão da luz, ou seja, se ela é unidirecional ou multidirecional, de raios paralelos ou cônicos. Assim, as principais fontes de luz virtuais podem ser classificadas em:

- Omnidirecionais – Emitem luz para todas as direções, partindo de um ponto central e espalhando luz pela cena. Dependendo da linguagem ou aplicativo podem haver controles sobre a perda de luz ao longo do espaço, controles de

suavização, emissão de sombras, entre outros. É uma luz pontual e a forma do feixe de luz se assemelha a uma esfera. Os raios ou fótons nesse tipo de luz são cônicos entre si.

- Spots/Cônicas – São luzes que emitem luz direcionada, de forma cônica. Assim a luz parte de um ponto central e se espalha na direção para a qual está apontada a luz, assumindo o feixe de luz uma forma semelhante a um cone. Os raios ou fótons nesse tipo de luz são cônicos entre si. De acordo com a linguagem ou aplicativo podem haver controles iguais aos das Omnidirecionais e controles adicionais sobre a direção e abertura do feixe de luz.
- Direct/Direcionais – Este é o tipo de luz que mais diverge das anteriores, pois a emissão de luz nesse caso não parte de um ponto central, e sim de uma área central. O feixe de luz assemelha-se a um cilindro e os raios ou fótons são paralelos entre si. Assim como nas luzes anteriores, dependendo da linguagem ou aplicativo, podem haver controles gerais sobre as propriedades da luz, bem como controles adicionais de diâmetro do feixe e mesmo a forma deste.

A Figura A.11 ilustra a forma de iluminação para cada tipo de luz virtual, bem como exemplos a elas correspondentes que existem no mundo real.

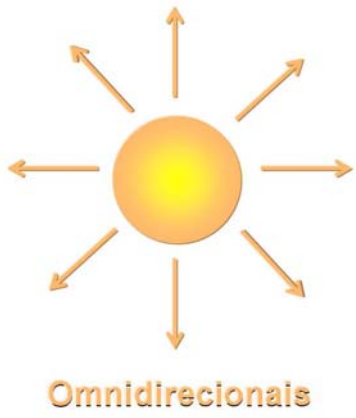

Figura A.11 – Luzes Virtuais e suas Correspondentes no Mundo Real [44,45,46].

As tarefas anteriormente explanadas concentram-se especificamente na criação da cena tridimensional e os objetos nela contidos, além de definir propriedades visuais de iluminação e texturas. Porém, a implementação de um sistema de Realidade Virtual completo prevê não somente objetos estáticos com boa texturização e iluminação, mas também animações automatizadas ou controladas pelo sujeito, constituindo-se assim a interatividade total, ficando não apenas restrita à inspeção dos objetos, mas sim estendida à interação através de controles virtuais como alavancas e botões.

Quanto à animação, diversas são as técnicas presentes na Realidade Virtual, permitindo animar não somente os objetos criados, mas também luzes virtuais, pontos de vista, texturas, transparência, entre outros parâmetros. Entretanto, estas técnicas quase que na sua totalidade compartilham o mesmo conceito de funcionamento utilizado nos desenhos animados e no cinema.

O movimento visualizado em sistemas de Realidade Virtual, desenhos animados e produções cinematográficas é somente uma ilusão, conseguida graças a uma pequena demora na visão humana em interpretar dados visuais captados pela retina. Assim,

através da exibição de quadros estáticos em seqüência e com razoável velocidade, é possível criar a ilusão de movimento, sendo que cada quadro representa uma fração de tempo na animação original. Esse fato é especialmente válido em produções cinematográficas, onde há o registro pelas câmeras de movimentação real de personagens e cenários, com uma taxa de quadros por segundo normalmente fixada entre 24 e 30 quadros por segundo. Nas aplicações de Realidade Virtual no entanto, a taxa de quadros por segundo pode variar consideravelmente, pois de acordo com a complexidade da cena, a placa de vídeo e o processador podem sofrer variações de performance, o que resulta em diferentes taxas de quadros por segundo. Essas variações são mais facilmente percebidas quando as taxas de quadros caem para quantidades menores que 30 quadros por segundo, pois abaixo destes valores a ilusão de movimento percebida pelo olho humano começa a se perder [76,77,78].

Para gerar uma animação em sistemas de Realidade Virtual, existem basicamente duas técnicas distintas. Na primeira delas, conhecida como keyFraming [79] ou animação por quadros-chave, cada quadro da animação é criado informando-se todos os parâmetros a serem animados, como por exemplo as coordenadas de localização de um objeto. Dependendo da plataforma da aplicação, é possível animar os mais diversos parâmetros, tais como posicionamento, rotação, escala, iluminação e materiais. A vantagem desta técnica de animação é a sua precisão, pois cada fração da animação é plenamente especificada em seus parâmetros. Em contrapartida, a criação de animações a partir desta técnica pode se mostrar muito improdutiva, principalmente se não se utilizar alguma ferramenta auxiliar, pois a quantidade de quadros a se criar mesmo em pequenas animações é frequentemente alta, aumentando bastante a quantidade de dados a parametrizar. A Figura A.12 demonstra a animação da movimentação de uma esfera feita em 6 quadros e, ao final, ilustra a composição da animação final.

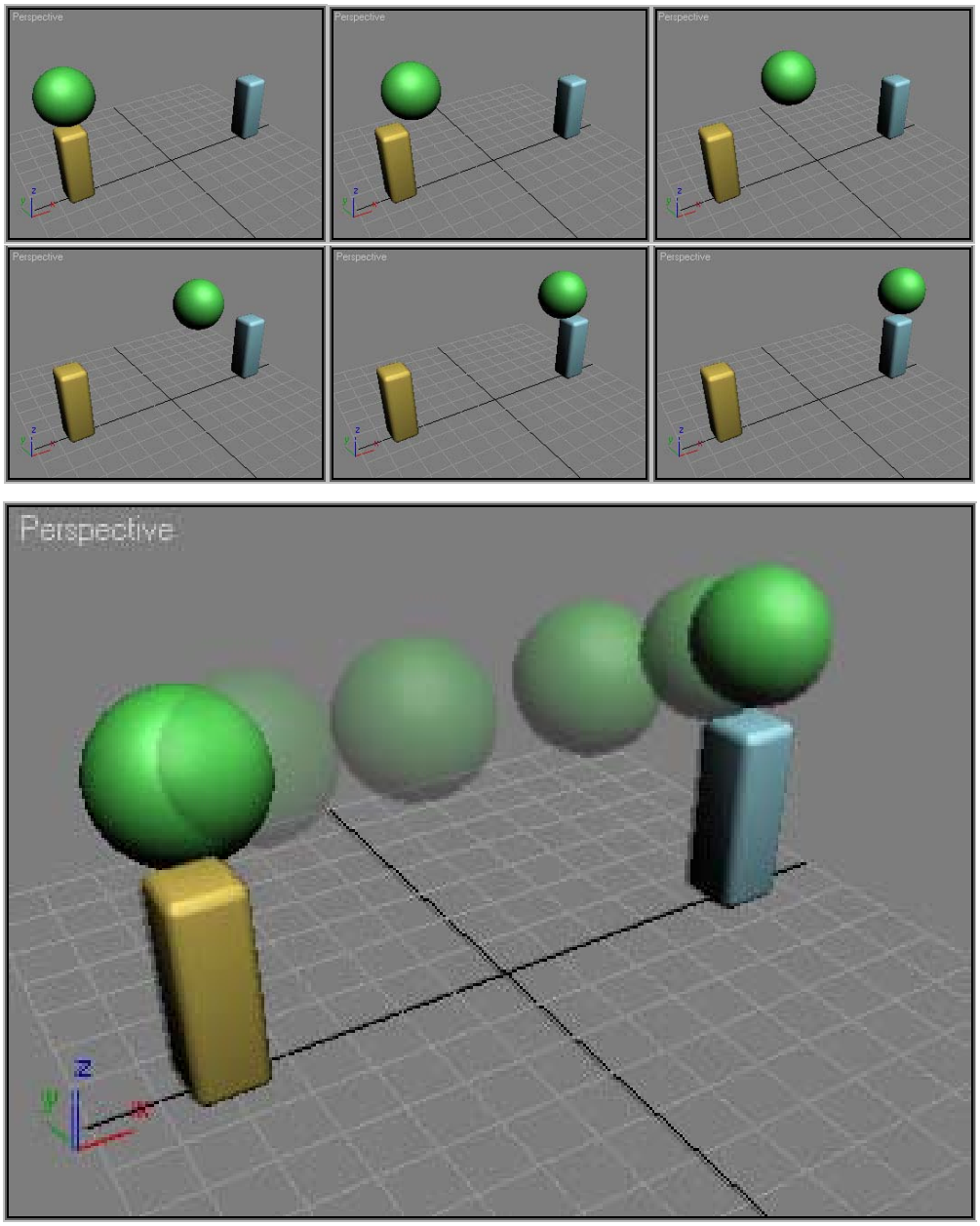

Figura A.12 – Animação de uma esfera por keyFraming.

Além da técnica de keyFraming, existe outra abordagem de animação muito utilizada em ambientes de Realidade Virtual conhecida como Tweening [80] ou animação por interpolação. Nesta técnica, ainda são definidos os parâmetros a serem animados nos objetos e/ou características, porém ao contrário da técnica de keyFramming simples, nas animações feitas por interpolação não são definidos os estados de animação em cada quadro da animação final. Ao invés disso, são definidos quadros-chave com mudanças de parâmetros a serem animados em intervalos desejados da animação. A aplicação então, baseada nos quadros chaves previamente definidos, deduz os parâmetros a animar nos quadros intermediários mesclando proporcionalmente

as informações entre os quadros-chave, criando assim uma transição linear de um estado para outro. A criação de um determinado quadro intermediário leva em conta a posição do mesmo em relação aos 2 quadros-chave que delimitam o intervalo de animação, quanto mais próximo ele estiver de um quadro-chave, mais semelhante será seu estado ao deste quadro, de forma análoga a um gradiente de cores.

Para se calcular precisamente o estado de um quadro intermediário, primeiro é necessário conhecer o valor total de deslocamento de estado entre um quadro-chave e outro, bem como a quantidade de quadros entre eles. Em posse dessas informações, calcula-se o deslocamento de um único quadro-intermediário, dividindo o deslocamento total pela quantidade de quadros do intervalo+1, pois o quadro-chave final é contado na como fazendo parte da mudança de estado. Por exemplo, a animação da esfera demonstrada na Figura A.12 possui um total de 6 quadros, sendo 2 deles os quadros-chave que definem o estado inicial e final da animação e os outros 4 quadros intermediários a serem gerados. Assim, a quantidade de quadros usada como base para os cálculos é 4+1= 5. A animação da esfera ocorreu no eixo X. No estado do quadro-chave inicial o valor em X para a esfera foi definido como -50 e no estado final como 50. O total de deslocamento nesse caso foi de 100 unidades |(-50 -(50))|. Para definir o sinal dessas unidades verifica-se se o estado final é maior ou menor que o inicial. Sendo maior o sinal é positivo, sendo menor o sinal é negativo. Neste caso o sinal ficou positivo, pois 50 é maior que -50. A seguir, calcula-se o deslocamento de 1 único quadro intermediário, dividindo o deslocamento total já com sinal pela quantidade de quadros, no caso 100/5 = 20. Com base neste valor pode-se calcular todos os quadros intermediários. O estado do primeiro quadro intermediário é calculado somando-se o estado do quadro-chave inicial com o valor de deslocamento de cada quadro previamente calculado, neste caso -50 + 20 = 30. Os próximos quadros usam sempre os estados dos anteriores como base e somam o deslocamento, gerando assim uma animação suave. A Figura A.13 exemplifica o processo de animação por interpolação.

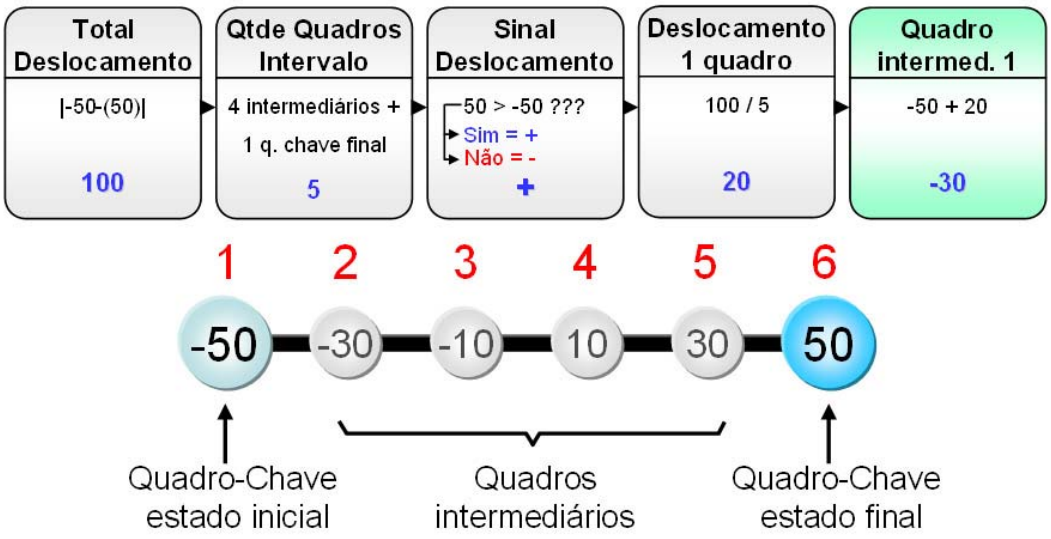

Figura A.13 – Animação por Interpolação.

O método de animação por interpolação tem a vantagem de gerar poucas quantidades de dados, deixando a tarefa de gerar quadros intermediários para a aplicação. É válido lembrar que, por padrão, as animações são sempre lineares de um estado para outro, não havendo mudanças bruscas de um estado para outro. Alguns softwares e linguagens implementam variações da interpolação linear, de forma a suprir necessidades de animação específica, através de ferramentas que modificam os parâmetros da interpolação. Nesses casos a forma de calcular os quadros intermediários pode ser bem diferente da interpolação linear previamente exemplificada.

Como desvantagens do método de animação por interpolação, pode-se ressaltar que, apesar de gerar pouco volume de dados na animação, como a tarefa de gerar os quadros é transferida para a aplicação e isso muitas vezes é feito em tempo de execução, pode haver sobrecarga no sistema e, consequentemente, queda de performance. Na animação por quadros-chave a queda de performance é menor, pois geralmente o custo computacional de leitura de parâmetros é bem menor que o de cálculo de quadros intermediários. Isso é válido principalmente em aplicações que animem grandes quantidades de parâmetros simultaneamente. Por isso, ao criar uma animação, o artista deve ter em mente o escopo e propósito de sua aplicação e optar pelo método mais adequado de animação, balanceando com eficiência a performance e o tamanho da aplicação.

Além das já comentadas, existem outras técnicas de animação utilizadas em sistemas de Realidade Virtual, porém de uma forma ou de outra essas técnicas baseiam-

se nos mesmos princípios básicos da animação por quadros-chave e da animação por interpolação. Além disso, o presente trabalho foi desenvolvido utilizando apenas as duas técnicas apresentadas, por isso não foram detalhadas aqui as outras formas de animação.