

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” – UNIVEM
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDERSON KUMAGAI

**ESTUDO E IMPLEMENTAÇÃO DE TÉCNICAS DE REALIDADE
AUMENTADA SEM MARCADORES**

MARÍLIA
2011

ANDERSON KUMAGAI

REALIDADE AUMENTADA UTILIZANDO MARCADORES NÃO
CONVENCIONAIS

Trabalho de Curso apresentado ao Curso de
Ciência da Computação da Fundação de
Ensino “Eurípides Soares da Rocha”,
mantenedora do Centro Universitário
Eurípides de Marília – UNIVEM, como
requisito para obtenção do grau de Bacharel
em Ciência da Computação.

Orientador:
Prof. Ms. LEONARDO CASTRO BOTEGA

MARÍLIA
2011



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Anderson Kumagai

REALIDADE AUMENTADA UTILIZANDO MARCADORES NÃO CONVENCIONAIS

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 9.0 (NOVE)

Orientador: Leonardo Castro Botega

1º. Examinador: Ildeberto de Gênova Bugatti

2º. Examinador: Elvis Fusco

Three horizontal lines with handwritten signatures written over them. The top signature is 'L. Botega', the middle one is 'Ildeberto de Gênova Bugatti', and the bottom one is 'Elvis Fusco'.

Marília, 28 de novembro de 2011.

KUMAGAI, Anderson. **Realidade Aumentada Utilizando Marcadores Não Convencionais**. 2011. 47 f. Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha de Marília, 2011.

RESUMO

A realidade aumentada insere objetos virtuais no ambiente físico através de marcadores impressos em papéis, os quais, ao serem apresentados para uma câmera, são reconhecidos e objetos são projetados no ambiente físico. A realidade aumentada sem marcadores se caracteriza pelo rastreamento da câmera, o que permite a inserção dos objetos virtuais no ambiente físico. Desta forma, o presente trabalho visa o estudo de técnicas para o reconhecimento de objetos sem marcadores e o desenvolvimento de protótipo utilizando a ferramenta OpenCV.

Palavras-chave: realidade aumentada sem marcadores, realidade aumentada com marcadores não convencionais, realidade aumentada.

KUMAGAI, Anderson. **Realidade Aumentada Utilizando Marcadores Não Convencionais**. 2011. 47 f. Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha de Marília, 2011.

ABSTRACT

Augmented Reality insert virtual objects in the physical environment by markers printed on papers which, when presented to the camera, they are recognized and objects are projected on the marker. Markerless Augmented Reality is characterized by the tracking camera which permits the insert of virtual objects in the physical environment. The present work aims the study of techniques to recognize objects without markers and the development of a prototype using the OpenCV.

Keywords: markerless augmented reality, markerless augmented reality with unconventional markers, augmented reality.

LISTA DE ILUSTRAÇÕES

Figura 1: Funcionamento de um sistema de Realidade Aumentada (Kirner <i>et al</i> , 2008).	10
Figura 2: Imagem ilustrando um dispositivo de visão ótica direta (Realidade Aumentada).....	11
Figura 3: Imagem ilustrando um dispositivo de visão direta por vídeo (Realidade Aumentada).....	12
Figura 4: Imagem ilustrando um dispositivo de visão por vídeo baseada em monitor (Realidade Aumentada).	12
Figura 5: Exemplo de quebra-cabeças com palavras (Zorzal <i>et al</i> , 2006).	13
Figura 6: Processo de oclusão do marcador (Zorzal <i>et al</i> , 2005).....	14
Figura 7: Simulação do envasamento em realidade virtual (Zorzal <i>et al</i> , 2006).	15
Figura 8: Simulação do envasamento em realidade aumentada (Zorzal <i>et al</i> , 2006).	15
Figura 9: (a) Arquitetura geral da ferramenta (b) Interface de configuração de visualização em realidade aumentada (Bueno <i>et al</i> , 2005).	16
Figura 10: Visualização das informações em realidade aumentada (Bueno <i>et al</i> , 2005).....	16
Figura 11: Taxonomia das técnicas de MAR monocular (Simões, 2008).....	17
Figura 12: Objeto poligonal rastreado a partir de suas arestas (Simões, 2008).	18
Figura 13: Processo de rastreamento do objeto através da amostragem de pontos (Teichrieb, 2007). .	19
Figura 14: Processo de rastreamento do objeto através da detecção das arestas (Simões, 2008).	19
Figura 15: Técnica baseada no correlacionamento de modelos (Teichrieb, 2007).	21
Figura 16: Técnica baseada em pontos de interesse (Teichrieb, 2007).	21
Figura 17: Aplicação utilizando SfM (Teichrieb, 2007).	22
Figura 18: Reconhecimento de face em uma câmera digital. (Kodak).....	23
Figura 19: Reconhecimento de face em uma foto (Gorodnichy, 2005).	24
Figura 20: Reconhecimento facial através de imagens das faces que se desejam reconhecer (Zhou <i>et al</i> , 2003).	25
Figura 21: Gestos reconhecidos pela aplicação (Manresa <i>et al</i> , 2005).	26
Figura 22: Detecção da mão (Kato <i>et al</i> , 2011).	27
Figura 23: Projeção de um templo grego antigo (Stricker <i>et al</i> , 2001).....	27
Figura 24: Visualização do tumor através do sistema MAR (Mellor, 1995).....	28
Figura 25: Funcionamento do algoritmo de amostragem de pontos (Simões, 2008).	29
Figura 26: (A) e (B) característica 1 (C) característica 2 e (D) característica 3 (Viola <i>et al</i> , 2001).	31
Figura 27: Avaliação das regiões retangulares de uma imagem (Bradski e Kaebler, 2008).	33
Figura 28: Fluxo do algoritmo do protótipo desenvolvido.	34
Figura 29: Função de inicialização.....	35
Figura 30: Função responsável por detectar o objeto de interesse.	37
Figura 31: Reconhecimento da mão e projeção do objeto 3D sobre a mesma.....	39
Figura 32: Reconhecimento de áreas não desejadas.....	39
Figura 33: Reconhecimento da mão em movimento.	40
Figura 34: A alta variação de iluminação prejudica o reconhecimento da mão.	40
Figura 35: Reconhecimento de mais de um objeto de interesse na cena.	41
Figura 36: Não reconhecimento da pose da mão.	41

LISTA DE ABREVIATURAS E SIGLAS

AR: Augmented Reality

AV: Ambientes Virtuais

HMD: Head Mounted Display

MAR: Markerless Augmented Reality

RA: Realidade Aumentada

RV: Realidade Virtual

SfM: Structure from Motion

SLAM: Simultaneous Localization and Mapping

VE: Virtual Environment

VR: Virtual Reality

SUMÁRIO

INTRODUÇÃO.....	8
CAPÍTULO 1 - REALIDADE AUMENTADA.....	9
1.1. Tipos de RA.....	11
1.1.1. Sistema de visão ótica direta.....	11
1.1.2. Sistema de visão direta por vídeo.....	11
1.1.3. Sistema de visão por vídeo baseada em monitor.....	12
1.1.4. Sistema de visão ótica por projeção.....	13
1.2. Aplicações RA.....	13
1.2.1. Educação.....	13
1.2.2. Aprendizado Musical.....	14
1.2.3. Automação Industrial.....	14
1.2.4. Visualização de Informações.....	15
CAPÍTULO 2 - REALIDADE AUMENTADA SEM MARCADORES (MARKERLESS AUGMENTED REALITY).....	17
2.1. Técnicas baseadas em modelos.....	18
2.1.1. Técnica baseada em arestas.....	19
2.1.2. Técnica baseada em fluxo óptico.....	20
2.1.3. Técnica baseada em textura.....	20
2.2. Técnicas baseadas em estruturas e movimentos.....	21
2.2.1. SfM de tempo real.....	21
2.2.2. MonoSLAM.....	22
2.3. Aplicações MAR.....	23
2.3.1. Reconhecimento de Face.....	23
2.3.2. Rastreamento de mão.....	25
2.3.3. MAR aplicado à educação.....	27
2.3.4. MAR aplicado à medicina.....	28
CAPÍTULO 3 - METODOLOGIA.....	29
3.1. Estudo de técnicas.....	29
3.1.1. MAR baseada em arestas.....	29
3.1.2. MAR baseada na detecção da ponta dos dedos.....	30
3.1.3. Detecção de objetos utilizando OpenCV.....	30
3.2. Prototipação.....	33
CAPÍTULO 4 - RESULTADOS OBTIDOS.....	39
CAPÍTULO 5 - CONCLUSÕES.....	42
REFERÊNCIAS.....	43

INTRODUÇÃO

No surgimento do computador, as interfaces utilizavam chaves e lâmpadas para se comunicarem com o computador em linguagem de máquina. Ao passar dos anos, as interfaces foram evoluindo, com o surgimento de consoles, interface baseada em comandos e a interface multimídia, a qual permanece até hoje. Porém, é necessária a utilização de dispositivos para a interação com a mesma, como mouse, teclado e monitor.

Com o surgimento da Realidade Virtual foi possível aumentar a interação do usuário com o computador, tornando-a mais natural e transportando-o para um ambiente virtual. Porém, para realizar essa interação mais natural faz-se necessário o uso de equipamentos especiais, os quais, em primeira instância, causavam desconforto e dificuldade de interação, além da dificuldade de acesso aos mesmos.

Com a evolução tecnológica, surgiu a Realidade Aumentada, a qual possibilita a sobreposição de objetos virtuais em uma cena real. Enquanto a Realidade Virtual necessita de equipamentos especiais para a interação com os objetos do ambiente virtual, a Realidade Aumentada não apresenta essas restrições, tornando-se mais natural e abrangente.

Existem dois tipos de Realidade Aumentada: baseada em marcadores e sem marcadores. A primeira consiste no uso de marcadores impressos em papéis para a projeção dos objetos virtuais, enquanto a segunda baseia-se nas características do ambiente real. Apesar de ambas sobrepor objetos virtuais na cena real, a Realidade Aumentada baseada em marcadores necessita dos mesmos para poder interagir com os objetos.

O presente projeto consiste na implementação de um protótipo que possibilite a detecção e projeção de objetos sem a utilização de marcadores, aumentando assim a interação do usuário com os objetos virtuais, além de entender o funcionamento da técnica baseada em arestas utilizando a ferramenta OpenCV para a implementação do protótipo com a linguagem C++ e a validação do protótipo criado, reconhecendo objetos sem marcadores recebidos por um dispositivo de entrada, que no caso será uma webcam. No capítulo 1 serão apresentados conceitos sobre Realidade Aumentada, tipos e algumas aplicações relacionadas. No capítulo 2 serão apresentados conceitos sobre Realidade Aumentada sem Marcadores, os tipos e aplicações relacionadas. No capítulo 3 será apresentada a metodologia utilizada neste projeto, bem como a descrição de técnicas e o desenvolvimento de um protótipo. No capítulo 4 serão apresentados os resultados obtidos através da implementação e validação do protótipo. No capítulo 5 serão apresentados as conclusões.

CAPÍTULO 1 - REALIDADE AUMENTADA

Na década de 40 e 50, as interfaces utilizavam chaves e lâmpadas para se comunicarem com o computador em linguagem de máquina. Os consoles de vídeo surgiram somente na década de 60, porém com gráficos bem rudimentares. Com o surgimento de microprocessadores, nas décadas de 70 e 80 surgiu a interface baseada em comandos. A partir da interface baseada em comandos, foi possível explorar técnicas de multimídia, porém os usuários ficam restritos a tela do monitor e aos menus e ícones (Kirner *et al*, 2008).

A Realidade Virtual pode ser definida como interface a qual utiliza representações tridimensionais mais próximas da realidade do usuário e, através de dispositivos multissensoriais, interagir com os mesmos em tempo real, criando interações mais reais e rompendo a barreira da tela (Kirner *et al*, 2008)(Botega *et al*, 2009).

Na década de 60, a realidade virtual surgiu com o desenvolvimento do primeiro sistema gráfico interativo, porém só ganhou força na década de 90, pois o avanço tecnológico possibilitou a execução de sistemas de computação gráfica em tempo real. A utilização de equipamentos sofisticados, como luva, capacetes, óculos, etc., é necessária para transportar o usuário para o ambiente da aplicação, e em alguns casos, este transporte, em primeira instância, causava desconforto e dificuldades de interação (Kirner *et al*, 2008)(Botega *et al*, 2009).

Realidade Aumentada pode ser definida como a mistura de mundos reais e virtuais em algum ponto da realidade/virtualidade contínua, que conecta ambientes completamente reais a ambientes completamente virtuais. A Realidade Aumentada é uma tecnologia recente capaz de enriquecer a visão do mundo real com a sobreposição de objetos 3D virtuais. Sua primeira aparição ocorreu na década de 90, porém só ficou mais acessível a partir do ano 2000, com a tendência de técnicas de visão computacional, software e dispositivos com melhor custo-benefício. Realidade Aumentada (RA ou AR, Augmented Reality) é uma variação de Ambientes Virtuais (VE, Virtual Environments), ou Realidade Virtual (RV ou VR, Virtual Reality) como é mais comumente chamada (Platonov e Langer, 2007)(Kirner, 2007)(Azuma, 1997).

Seu principal objetivo é aprimorar a percepção sensorial. Os sistemas de Realidade Aumentada possuem as seguintes características: processamento em tempo real, combinação de elementos virtuais com o ambiente real e a utilização de elementos virtuais 3D. A figura 1 como funciona um sistema de Realidade Aumentada (Kirner *et al*, 2008).

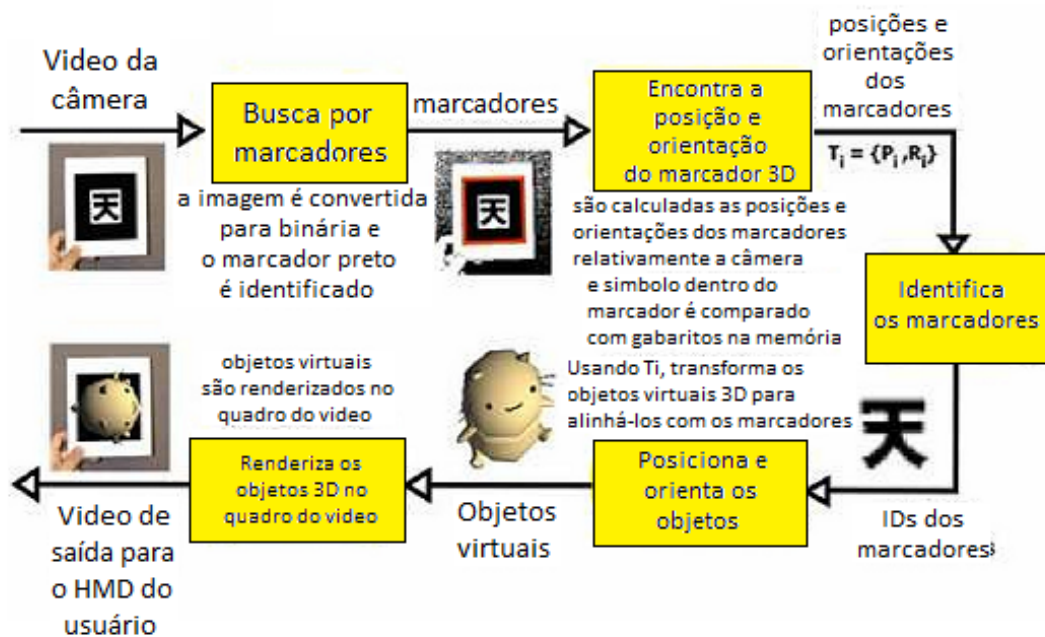


Figura 1: Funcionamento de um sistema de Realidade Aumentada (Kirner *et al*, 2008).

A mesma necessidade do rastreamento da câmera, cuja sua principal função é detectar o usuário como um todo, permitindo que o mesmo interaja com o ambiente através de ações como agarrar objetos, empurrar, tocar, etc. Vários dispositivos foram desenvolvidos para realizar o rastreamento, porém com a popularização da webcam e o desenvolvimento dos processadores, o rastreamento óptico passou a ser uma realidade, tendo a vantagem da disponibilidade e baixo custo (Kirner, 2007).

Com a utilização de um software, como ARToolKit, é possível desenvolver aplicações de Realidade Aumentada e através de técnicas de visão computacional é possível rastrear o marcador e calcular a posição do objeto virtual. O rastreamento do marcador é feito através da transformação da imagem recebida pelo dispositivo de entrada em preto e branco (binário) para então buscar regiões quadradas. Caso encontre alguma região, o símbolo que contém dentro da mesma é verificado em seu diretório por outro semelhante, e através do mesmo é calculado o tamanho e a posição do objeto virtual em relação ao marcador (Zorzal *et al*, 2005).

Diferente da Realidade Virtual, que necessita a utilização de equipamentos misturadores e ambientes fechados, a Realidade Aumentada não possui restrições com equipamentos ou ambientes, podendo ser mais abrangente e universal (Kirner *et al*, 2008).

A Realidade Aumentada permite o usuário visualizar o mundo real, com objetos virtuais sobrepostos ou compostos com o mundo real. A visualização do mundo real com objetos virtuais ocorre através de um monitor ou de algum dispositivo especial, como por

exemplo, um capacete de Realidade Virtual (HMD – Head Mounted Display). A Realidade Aumentada com marcadores possui limitações, fazendo com que o usuário não consiga interagir de forma direta com o objeto, gerando a necessidade de implementar marcadores impressos em papéis para que consiga reconhecer e representar algum objeto. Já a mesma sem marcadores (do inglês Markerless Augmented Reality, MAR) se caracteriza por executar o rastreamento do objeto por uma câmera, processo que permite a correta inserção dos elementos virtuais na cena real (Simões, 2008)(Kirner, 2007).

1.1. Tipos de RA

Existem 4 tipos de realidade aumentada baseadas no tipo de equipamento utilizado para a exibição da cena captada:

1.1.1. Sistema de visão ótica direta

Utiliza óculos ou capacetes que permitem visualizar a cena real com a projeção de objetos virtuais ajustadas com a mesma. A vantagem deste tipo de RA é que o usuário pode mover-se livremente, porém é necessário o uso de equipamentos especiais e o equipamento responsável por adquirir a cena é fixo. A figura 2 mostra um sistema de visão ótica direta.

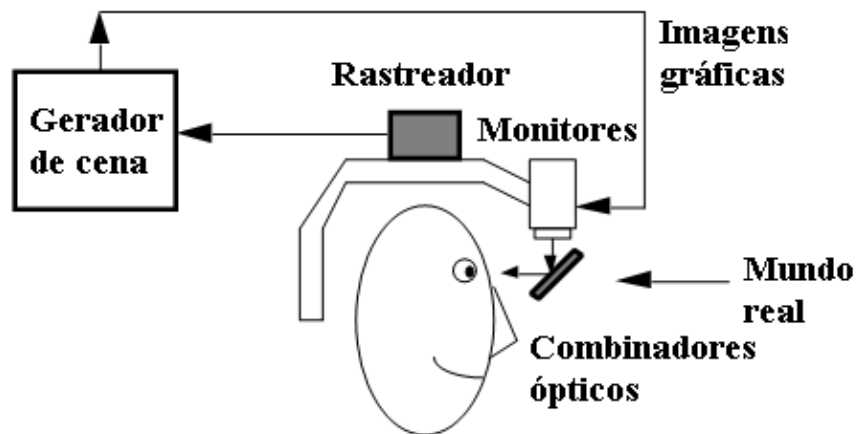


Figura 2: Imagem ilustrando um dispositivo de visão ótica direta (Azuma, 1997).

1.1.2. Sistema de visão direta por vídeo

Caracterizado pela utilização de capacetes com micro câmeras acopladas, a qual captura a cena real e os objetos virtuais são inseridos, e a mesma é mostrada para o usuário através de monitores acoplados no capacete. A vantagem deste tipo de RA é que o usuário

pode mover-se livremente e a cena adquirida é gerada de acordo com a posição do usuário, porém neste tipo também se faz necessário o uso de equipamentos especiais. A figura 3 mostra um sistema de visão direta por vídeo.

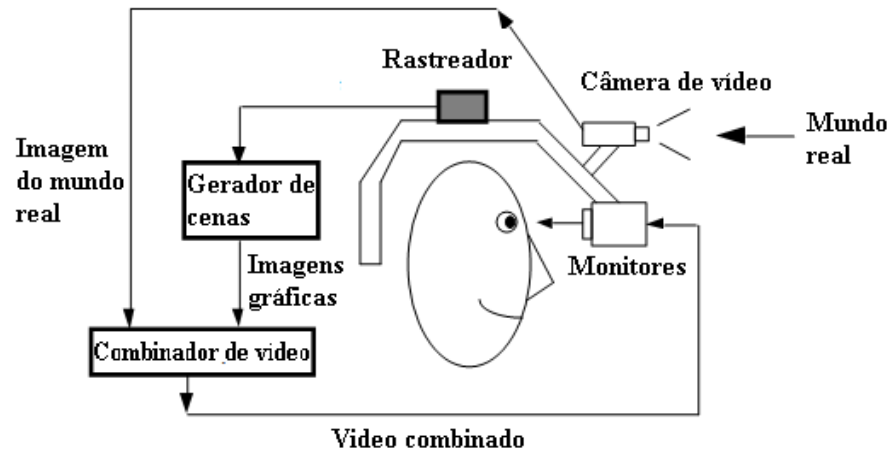


Figura 3: Imagem ilustrando um dispositivo de visão direta por vídeo (Azuma, 1997).

1.1.3. Sistema de visão por vídeo baseada em monitor

Caracterizado pela utilização da webcam, a qual captura a cena real, insere objetos virtuais a mesma e apresenta para o usuário através do monitor. A vantagem deste tipo de RA é o uso de equipamentos baratos e fáceis de adquirirem, porém o mesmo não permite o usuário se mover livremente. A figura 4 mostra um sistema de visão por vídeo baseada em monitor.

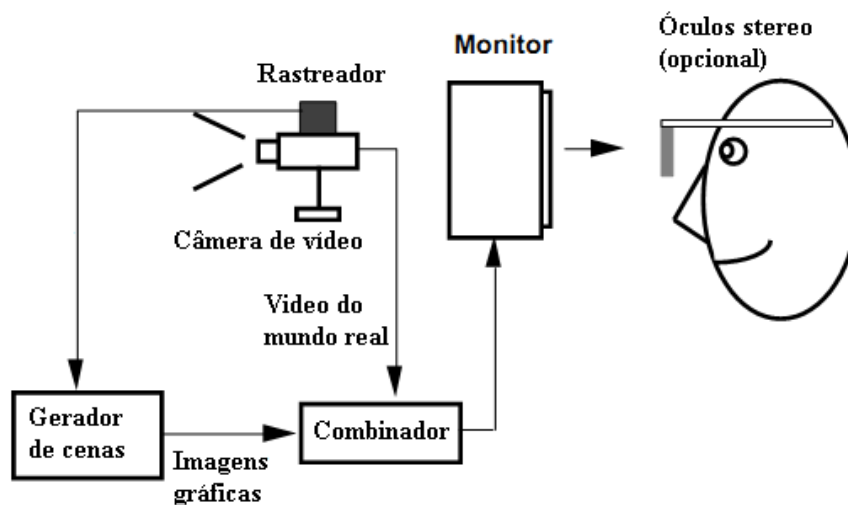


Figura 4: Imagem ilustrando um dispositivo de visão por vídeo baseada em monitor (Azuma, 1997).

1.1.4. Sistema de visão ótica por projeção

Caracterizado por utilizar superfícies do ambiente real para a projeção dos objetos virtuais e a visualização não necessita de nenhum equipamento auxiliar. Este sistema é mais apropriado em casos onde a perda da imagem é arriscada, como andar pela rua ou dirigir um carro, podendo causar acidentes graves.

1.2. Aplicações RA

A RA pode ser aplicada em diversas áreas como entretenimento, educação, treinamentos, etc. Neste tópico serão apresentadas algumas aplicações de RA.

1.2.1. Educação

A Realidade Aumentada pode ser aplicada no desenvolvimento de jogos, como um quebra-cabeça educacional. Algumas versões eletrônicas do quebra-cabeça foram desenvolvidas, porém a interação do usuário com o quebra-cabeça era limitada, pois era necessário o uso de dispositivos como mouse e teclado para poderem montar o mesmo. Com o surgimento da mesma, tornou-se possível unir o ambiente virtual com o real, aumentando a interação do usuário com o aplicativo. Com a introdução de objetos virtuais ao ambiente real, o usuário pode interagir diretamente, de forma mais natural com o jogo. Vários quebra-cabeças foram desenvolvidos, como os quebra-cabeças com palavras, onde ao formar uma palavra, a mesma projeta o objeto relacionado. A figura 5 mostra alguns exemplos (Zorzal *et al*, 2006).



Figura 5: Exemplo de quebra-cabeças com palavras (Zorzal *et al*, 2006).

1.2.2. Aprendizado Musical

A Realidade Aumentada pode ser aplicada também no aprendizado musical, focando áreas distintas. Uma das técnicas utilizadas foi a oclusão de marcadores, os quais são associados a notas musicais, e ao colocar a mão sobre um determinado marcador previamente posicionado sobre a câmera, o aplicativo executa o som da nota associado ao mesmo. Para realizar esta ação, foi necessário acionar uma thread em paralelo com o aplicativo de Realidade Aumentada. Com a oclusão de vários marcadores, é possível criar melodias personalizadas. A área explorada é a área rítmica, onde o usuário precisa acionar os marcadores no tempo correto para conseguir reproduzir a melodia corretamente. A figura 6 mostra o processo de oclusão do marcador (Zorzal *et al*, 2005).

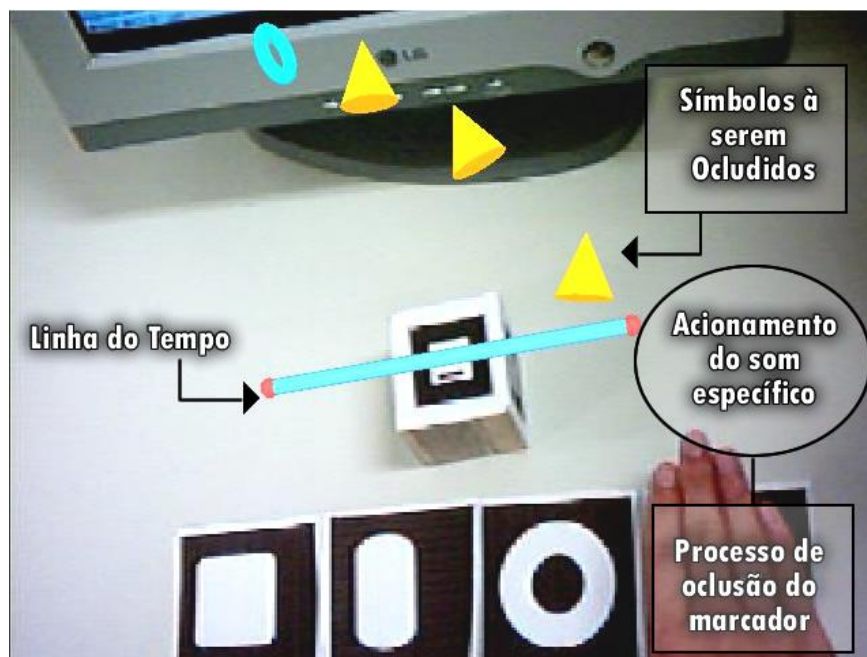


Figura 6: Processo de oclusão do marcador (Zorzal *et al*, 2005).

1.2.3. Automação Industrial

Sistemas Industriais se caracterizam pelo alto sincronismo e precisão entre as máquinas. Uma das formas de se simular um sistema é através da realidade virtual, onde é possível implementar um cenário bastante similar ao real. Porém, a mesma necessita de equipamentos específicos para interagir no ambiente criado. Em (Kirner *et al*, 2006) é utilizado além da realidade virtual a realidade aumentada, que não necessita de equipamentos especiais para ser utilizado. Para realizar a simulação, foi utilizado a ferramenta ARToolKit e VRML (Kirner *et al*, 2006).

A simulação utiliza dois marcadores, onde um recebe o cenário implementado em VRML, que será fixo, enquanto o segundo receberá o objeto que deverá receber as ações do maquinário de forma a simular o sistema automático. Conforme o usuário percorre o cenário com o objeto, o cenário realiza as ações que serão feitas no cenário real. As figuras 7 e 8 mostram a simulação em VRML e os marcadores sendo utilizados. O sistema a ser simulado é o envasamento (Zorzal *et al*, 2006).



Figura 7: Simulação do envasamento em realidade virtual (Zorzal *et al*, 2006).



Figura 8: Simulação do envasamento em realidade aumentada (Zorzal *et al*, 2006).

1.2.4. Visualização de Informações

Visualizar informação consiste em transformá-las em formas visuais, para que seja possível observar e entendê-las. Em (Bueno *et al*, 2005) são utilizadas diversas formas de visualizar uma informação, como a codificação de cores, que consiste em associar cores a objetos, como linhas, planos ou faces. Outra forma seria a técnica de faces de Chernoff que relaciona as informações com características faciais (Bueno *et al*, 2005).

Nesta aplicação foram utilizadas duas ferramentas: Meta3D++ e ARToolKit para a visualização da informação em realidade aumentada. A figura 9 mostra a estrutura da ferramenta Meta3D++ e a figura 10 a visualização da informação utilizando a realidade aumentada (Bueno *et al*, 2005).

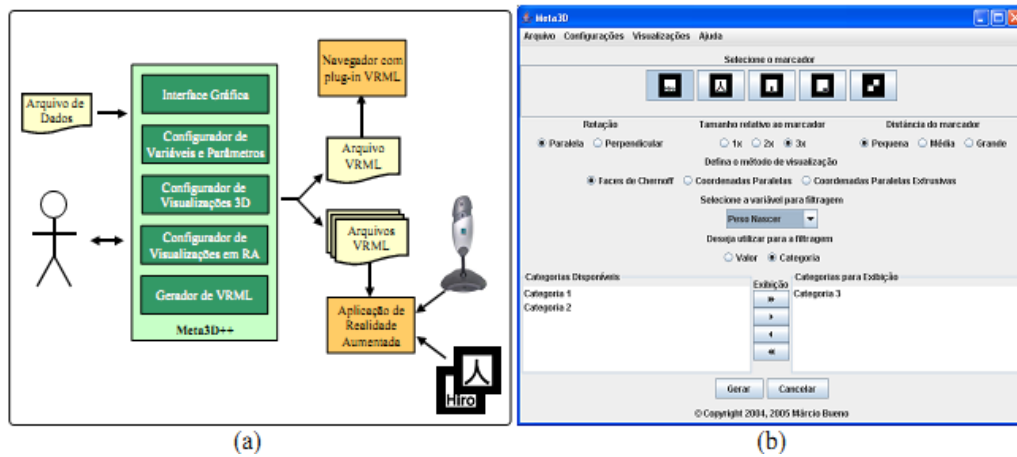


Figura 9: (a) Arquitetura geral da ferramenta (b) Interface de configuração de visualização em realidade aumentada (Bueno *et al*, 2005).

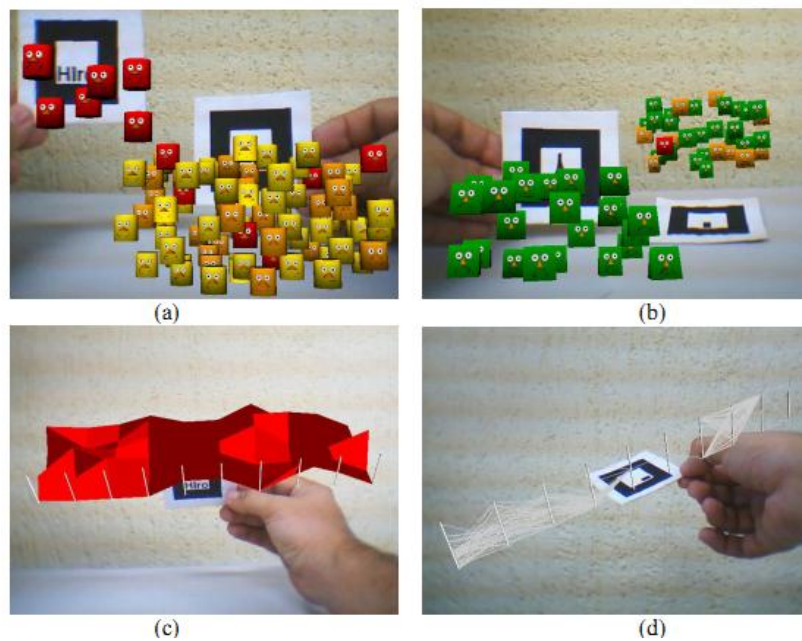


Figura 10: Visualização das informações em realidade aumentada (Bueno *et al*, 2005).

No capítulo 2 será apresentado a teoria sobre Realidade Aumentada sem Marcadores, bem como os tipos e suas vantagens.

CAPÍTULO 2 - REALIDADE AUMENTADA SEM MARCADORES (MARKERLESS AUGMENTED REALITY)

Sistemas baseados em MAR integram objetos 3D na cena real sem o uso de marcadores. Porém, qualquer parte da cena real pode ser utilizada como marcador, podendo ser rastreado e estimada a posição do objeto. Algumas vantagens dos sistemas MAR são que os mesmos contêm rastreadores especializados e robustos, além de possibilitar extrair características da cena real. Porém, realizar o mesmo é complexo e apresentam restrições (Teichrieb, 2007).

Para o reconhecimento de objetos sem marcadores, é necessário o conhecimento prévio do mundo real, que são modelos 3D utilizados para auxiliar na estimativa da posição da câmera em relação ao objeto (Simões, 2008).

As técnicas sem marcadores objetivam realizar o rastreamento da câmera através de informações inerentes às cenas, como a relação entre características da cena que se perpetuam ao longo das imagens capturadas pela câmera ou modelos 3D gerados a partir dos objetos presentes na cena, sem a inserção de nenhum objeto sintético. A Figura 11 ilustra a Taxonomia das técnicas de MAR monocular (Simões, 2008).

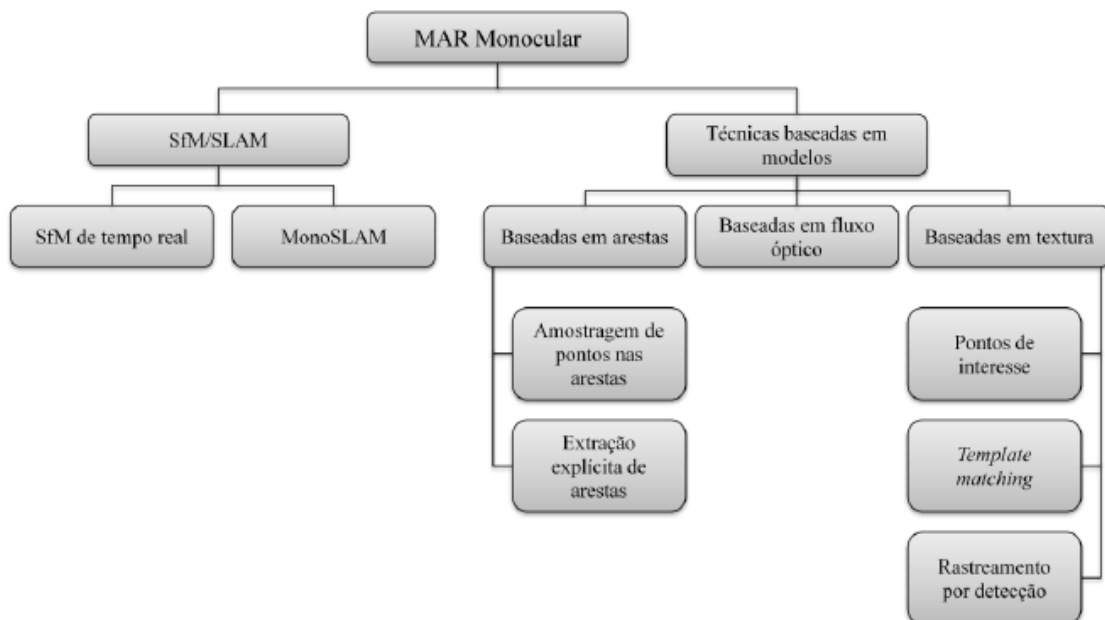


Figura 11: Taxonomia das técnicas de MAR monocular (Simões, 2008).

As técnicas desenvolvidas em MAR podem ser classificadas em dois tipos fundamentais: técnicas baseadas em modelos e técnicas baseadas em estrutura e movimentos (Structure from motion - SfM). Nas técnicas baseadas em modelos é necessário um

conhecimento prévio sobre o mundo real, que é guardado na forma de um modelo 3D utilizado para auxiliar na estimativa da pose do objeto pela câmera. Nas técnicas SfM, o movimento da câmera ao longo da sequência de quadros é estimado sem nenhum conhecimento prévio da cena, sendo adquirido durante o rastreamento. As técnicas baseadas em modelo são, em geral, mais simples do que as baseadas em SfM, porém depende da visibilidade do objeto para estimar a posição do objeto a ser inserido na cena real, enquanto a segunda não tem essa restrição. A Figura 12 ilustra o rastreamento das arestas de um objeto sem a utilização de marcadores impressos em papéis (Simões, 2008) (Teichrieb, 2007).

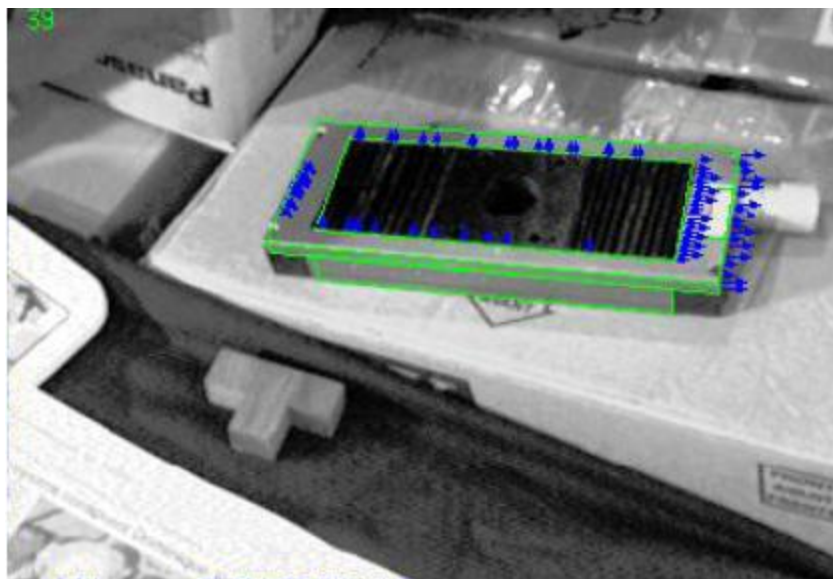


Figura 12: Objeto poligonal rastreado a partir de suas arestas (Simões, 2008).

2.1. Técnicas baseadas em modelos

Técnicas baseadas em modelos utilizam modelos 3D do objeto presente na cena. Os mesmos são comparados com a imagem 2D e o resultado são os objetos presentes na cena, sua posição e pose (Teichrieb, 2007).

A vantagem das técnicas baseadas em modelos é a possível interação entre o mundo real e virtual. Porém, precisa do conhecimento prévio da cena e o objeto rastreado precisar estar presente na cena, caso contrário não será possível projetar o objeto 3D virtual na cena real (Teichrieb, 2007).

As técnicas baseadas em modelos podem ser divididas em 3 categorias: baseada em arestas, baseadas em fluxo óptico e baseada em textura. Nos subtópicos 2.1.1, 2.1.2 e 2.1.3 serão explicados o funcionamento e vantagens de cada categoria (Teichrieb, 2007).

2.1.1. Técnica baseada em arestas

A técnica baseada em arestas, como o próprio nome já diz, baseia-se nas arestas do objeto para rastreá-lo. Para extrair as arestas do objeto de interesse o modelo do mesmo é projetado sobre a imagem da cena real e é minimizado o deslocamento entre a projeção do modelo e o objeto da imagem (Teichrieb, 2007).

Por sua baixa complexidade, a técnica baseada em arestas são fáceis de implementar e ter uma boa performance, porém não suportam movimentos rápidos da câmera pois a mesma não consegue obter uma quantidade alta de frames em 1 segundo (Teichrieb, 2007).

Técnica baseada em aresta pode ser dividida em duas subcategorias: as técnicas baseadas em amostragem de pontos e as técnicas baseadas na detecção explícita das arestas. Na primeira abordagem, a recuperação dos parâmetros de câmera se dá através da associação entre pontos extraídos das arestas com pontos pertencentes à imagem. Já na segunda a correlação criada se dá entre as arestas completas. A figura 13 ilustra o processo de rastreamento do objeto através da amostragem de pontos e a figura 14 mostra a detecção das arestas (Simões, 2008) (Teichrieb, 2007).

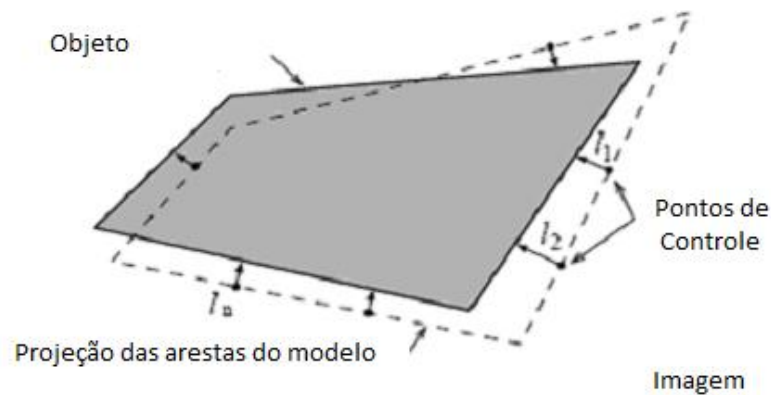


Figura 13: Processo de rastreamento do objeto através da amostragem de pontos (Teichrieb, 2007).

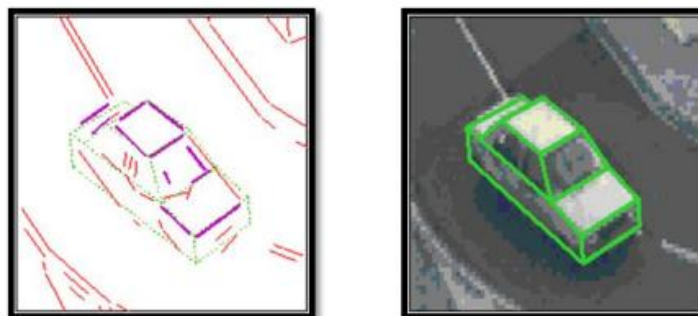


Figura 14: Processo de rastreamento do objeto através da detecção das arestas (Simões, 2008).

A vantagem da técnica baseada em amostragem de pontos é sua generalização, podendo detectar curvas, enquanto a detecção explícita de arestas é mais robusta, ou seja, consegue detectar de uma forma melhor, tendo pouca sensibilidade a luminosidade por exemplo, o objeto de interesse. Já a desvantagem da primeira técnica citada é a falta de robustez na fase de correlacionamento, a qual pode ocorrer erro ao estimar a posição, enquanto a segunda não possui generalização como a amostragem de pontos, pois a detecção das arestas acaba sendo restrita para polígonos (Teichrieb, 2007).

2.1.2. Técnica baseada em fluxo óptico

A técnica baseada em fluxo óptico consiste no rastreamento através de informações temporais, ou seja, do movimento da projeção na imagem. As vantagens do fluxo óptico são: mudanças mais suaves entre as poses além do processamento necessário para a inicialização. Já as desvantagens são: tendência a acumular erros durante a sequência de uma cena, variação de luminosidade e grandes deslocamentos provocam erros de rastreamento (Teichrieb, 2007).

2.1.3. Técnica baseada em textura

A técnica baseada em textura consiste na utilização da textura do objeto para o rastreamento. A mesma pode ser dividida em duas subcategorias: baseada no correlacionamento de modelos e baseada em pontos de interesse.

A técnica baseada em correlacionamento de modelos utiliza a informação global da imagem ao invés de características. Já a técnica baseada em pontos de interesse utiliza informações de características localizadas, ao contrário da técnica citada anteriormente. A vantagem da segunda técnica citada é a menor intensidade de processamento, além de ser pouco sensível a variação de luminosidade. As figuras 15 e 16 mostram as técnicas baseadas em correlacionamento de modelos e pontos de interesse, respectivamente (Teichrieb, 2007).

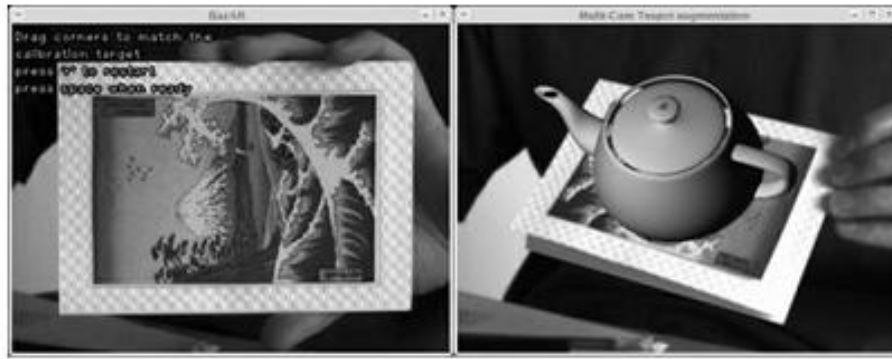


Figura 15: Técnica baseada no correlacionamento de modelos (Teichrieb, 2007).

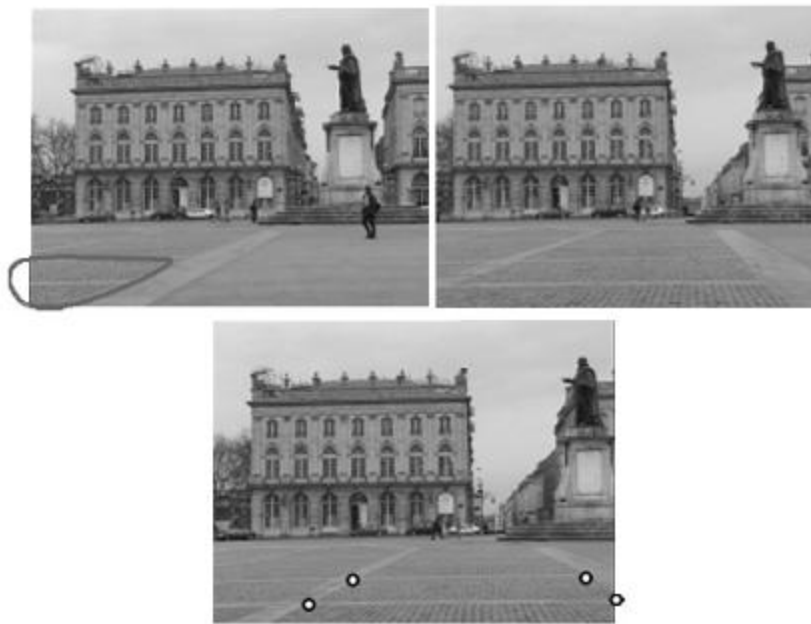


Figura 16: Técnica baseada em pontos de interesse (Teichrieb, 2007).

2.2. Técnicas baseadas em estruturas e movimentos

Algumas técnicas, ao invés de dependerem do conhecimento prévio da cena para realizar o rastreamento, estimam o deslocamento da câmera sem o uso da mesma. Porém, as mesmas são muito complexas, contendo algumas restrições relacionadas à estrutura de tempo real (Teichrieb, 2007).

2.2.1. SfM de tempo real

SfM é uma técnica clássica em visão computacional a qual sua pipeline não se preocupa com as restrições de tempo real, porém a mesma é suportada. Alguns algoritmos

presentes neste fluxo demandam alto processamento para realizar sua tarefa. Geralmente o pipeline é composto pelas seguintes fases: rastreamento de características, extração e refinamento da matriz fundamental, estimativa da pose da câmera e auto calibração. A figura 17 mostra uma aplicação baseada na técnica SfM (Teichrieb, 2007).



Figura 17: Aplicação utilizando SfM (Teichrieb, 2007).

2.2.2. MonoSLAM

SLAM (Simultaneous Localization and Mapping) é bastante utilizado na comunidade robótica. O mesmo é efetuado através de métodos de inferência Bayesiana sequencial (Teichrieb, 2007).

MonoSLAM foi baseada na metodologia de probabilidade SLAM utilizando uma câmera que pode se mover livremente com amplo ângulo de visão. O mesmo estima a pose da câmera sobre o ambiente e cria um mapa disperso do mesmo, porém é restrito para ambientes fechados, movimentos suaves da câmera e imagem monocromática, pois pode ocorrer erros em ambientes abertos e a quantidade de dados seria muito grande para ser processada (Teichrieb, 2007).

2.3. Aplicações MAR

Neste tópico serão apresentadas aplicações MAR, as quais podem ser aplicadas em diversas áreas, como educação, entretenimento, segurança, etc.

2.3.1. Reconhecimento de Face

O reconhecimento de face pode ser comumente observado na área de entretenimento. Por exemplo, nas câmeras digitais, onde as mesmas formam quadrados ao redor dos rostos das pessoas que serão fotografadas. A figura 18 ilustra o reconhecimento de face em uma câmera digital.



Figura 18: Reconhecimento de face em uma câmera digital. (Kodak)

O reconhecimento de face pode ser utilizado mais na área de segurança, para a verificação de passaportes e identificar criminosos, porém é necessário seguir algumas regras para o reconhecimento do mesmo como: a pessoa a ser fotografada deve estar olhando diretamente para a câmera, não mostrar expressões não usuais e não utilizar objetos que possam obstruir alguma parte do rosto. A figura 19 mostra o reconhecimento de face em uma foto (Gorodnichy, 2005).

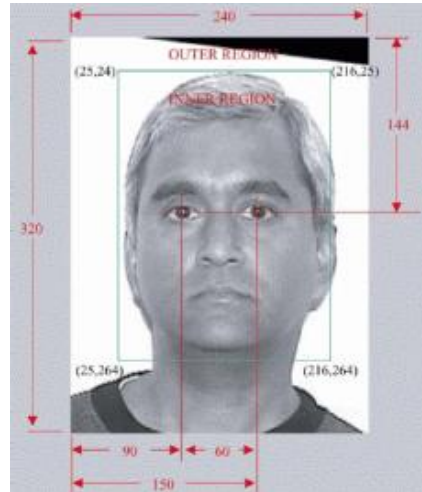


Figura 19: Reconhecimento de face em uma foto (Gorodnichy, 2005).

O reconhecimento facial pode ser utilizado também em vídeos. Para realizar o reconhecimento, dentre os vários existentes, citarei dois métodos: baseadas em imagens ou em outro vídeo. O primeiro método consiste em comparar os frames do vídeo com a imagem da face que se deseja detectar, porém este método contém alguns desafios, como a qualidade da imagem do vídeo e variação de posição e iluminação. A figura 8 demonstra o reconhecimento através do primeiro método (Zhou *et al.*, 2003).

O método baseado em vídeo consiste na utilização de vídeos, de uma forma bem semelhante ao método anterior, ao invés de imagens, para o reconhecimento facial. O mesmo é dividido em duas partes: aprendizado e reconhecimento. Em sua primeira parte é feito a aprendizagem do exemplar que se deseja reconhecer através dos vídeos. Os exemplares reconhecidos são, então, armazenados para uso no reconhecimento. Já na segunda parte é utilizado métodos de probabilidade, pois os mesmos permitem manipular de forma sistemática a incerteza e juntar informações temporais. A figura 20 mostra o reconhecimento de face através de um vídeo e a faces que a aplicação possui como modelo para o reconhecimento (Zhou *et al.*, 2003).



Figura 20: Reconhecimento facial através de imagens das faces que se desejam reconhecer (Zhou *et al*, 2003).

2.3.2. Rastreamento de mão

O rastreamento de mão pode ocorrer tanto com a ajuda de equipamentos quanto sem. Um exemplo seria luvas coloridas ou marcadores nas mãos ou dedos. A utilização de fundos controlados podem facilitar ainda mais o rastreamento, porém todas essas condições impõe restrições ao usuário, fazendo com que nem todos possam utilizá-lo devido a necessidade das mesmas serem atendidas (Manresa *et al*, 2005).

Porém em (Manresa *et al*, 2005) propõe o rastreamento de mão sem a utilização de equipamentos ou fundos controlados. A aplicação utiliza uma webcam que realiza a captura os gestos que servirão de comandos de entrada para o jogo 3D implementado em (Manresa *et al*, 2005). Os gestos reconhecidos são limitados, e cada um é relacionado a uma ação. A figura 21 demonstra os gestos reconhecidos pela aplicação.

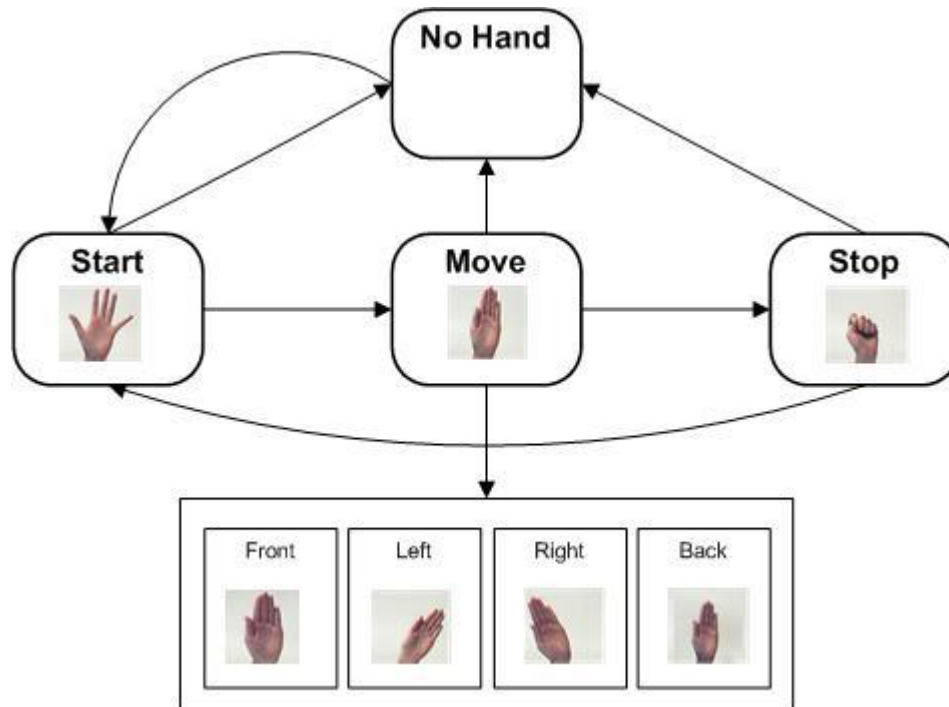


Figura 21: Gestos reconhecidos pela aplicação (Manresa *et al.*,2005).

Como a webcam produz imagens de baixa qualidade, podendo afetar o desempenho do reconhecimento da mão, foi utilizado um algoritmo para o rastreamento da mesma (Manresa *et al.*, 2005).

O reconhecimento da mão é feito, primeiramente, por uma aplicação que aprende sobre a cor da pele da mão. Para que isto ocorra, a mão ocupa a maior parte da cena, e através de um modelo Gaussiano é representada a função de probabilidade da densidade da cor da pele (Manresa *et al.*, 2005).

Em (Kato *et al.*, 2011) é utilizado uma técnica semelhante a (Manresa *et al.*, 2005), porém utilizado em aplicações para smartphones, onde o aplicativo detecta a mão através do tom da pele, as pontas dos dedos e a estimativa de sua posição. Através do modelo estatístico Gaussiano, o aplicativo detecta as cores que são próximas as da pele e as que não são para cada canal de cor através do calculo da probabilidade dos mesmos pelo produto da probabilidade pelos canais.

Para detectar a ponta dos dedos, primeiramente, é escolhida a região que contém a maior área com o tom de pele, e através desta região é feito contorno de toda a região com o mesmo tom e os vértices são considerados como a ponta dos dedos.

Para estimar a pose da mão é utilizado o resultado da detecção da ponta dos dedos, supondo que a palma seja um plano, resolvendo uma homografia pela ponta dos cinco dedos. A figura 22 mostra a detecção da mão.

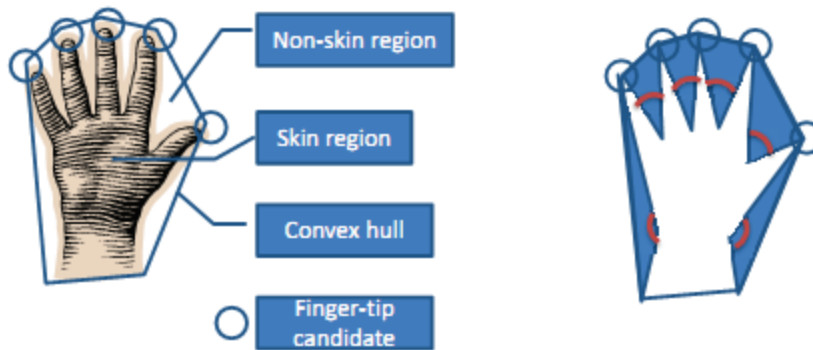


Figura 22: Detecção da mão (Kato *et al*, 2011).

2.3.3. MAR aplicado à educação

Sistemas baseados em MAR podem ser aplicados também na arqueologia, onde os usuários podem ver monumentos históricos através da projeção dos mesmos em uma cena real. O objetivo do sistema é o aprendizado da cultura de um determinado local. A visualização do monumento virtual pode ser feita através de capacetes que possuem monitores e câmeras acoplados. A figura 23 mostra a projeção de um templo antigo utilizando o sistema MAR (Stricker *et al*, 2001).



Figura 23: Projeção de um templo grego antigo (Stricker *et al*, 2001).

2.3.4. MAR aplicado à medicina

O uso de computadores na medicina tem aumentado drasticamente nos últimos anos. Várias ferramentas foram criadas, como a visualização 3D da anatomia do corpo humano, diagnósticos auxiliados por computadores, rastreamento de alergia, etc. Uma das áreas que vem crescendo é a utilização da computação para dar assistência ao planejamento e execução de uma cirurgia. A neurocirurgia é uma área que pode ser bastante beneficiada, pois é necessária extrema precisão para minimizar os danos colaterais. A figura 24 mostra a utilização de sistemas MAR aplicados a neurocirurgia (Mellor, 1995).

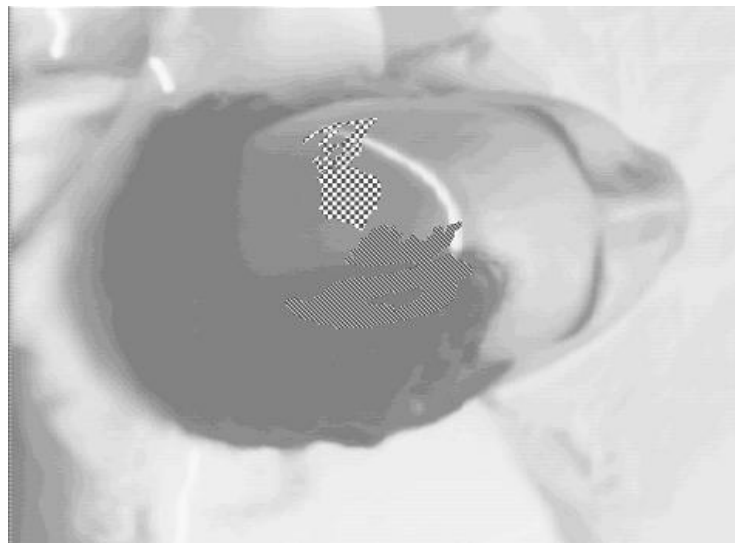


Figura 24: Visualização do tumor através do sistema MAR (Mellor, 1995).

No capítulo 3 será apresentada a metodologia utilizada no presente projeto.

CAPÍTULO 3 - METODOLOGIA

A RA baseada em marcadores, apesar de projetar o objeto na cena real, limita-se na utilização dos mesmos, sendo necessário que o marcador esteja presente na cena para projetar o objeto 3D e o usuário interagir com o mesmo. Com a MAR, a utilização dos mesmos não se faz necessária, podendo aumentar a interação entre o usuário e o objeto 3D, já que o mesmo interagirá diretamente com o objeto ao invés do marcador impresso em papel. O objetivo do presente trabalho é aumentar a interação do usuário através da implementação de técnicas MAR em um protótipo e sua validação.

3.1. Estudo de técnicas

Neste tópico serão apresentadas as técnicas estudadas para o desenvolvimento do protótipo do presente projeto.

3.1.1. MAR baseada em arestas

Uma das técnicas estudadas foi a MAR baseada em arestas. Para esta técnica, é necessário guardar um modelo 3D do objeto que se deseja reconhecer. A mesma é bastante aplicada quando se quer reconhecer objetos poligonais, os quais são mais fáceis de reconhecer através de suas arestas (Simões, 2008).

Baseado na técnica de arestas, a amostragem de pontos consiste na seleção de pontos de controle 3D das arestas do objeto rastreado. O funcionamento da mesma, baseada na técnica de Drummond, pode ser observado na figura 25 (Simões, 2008).

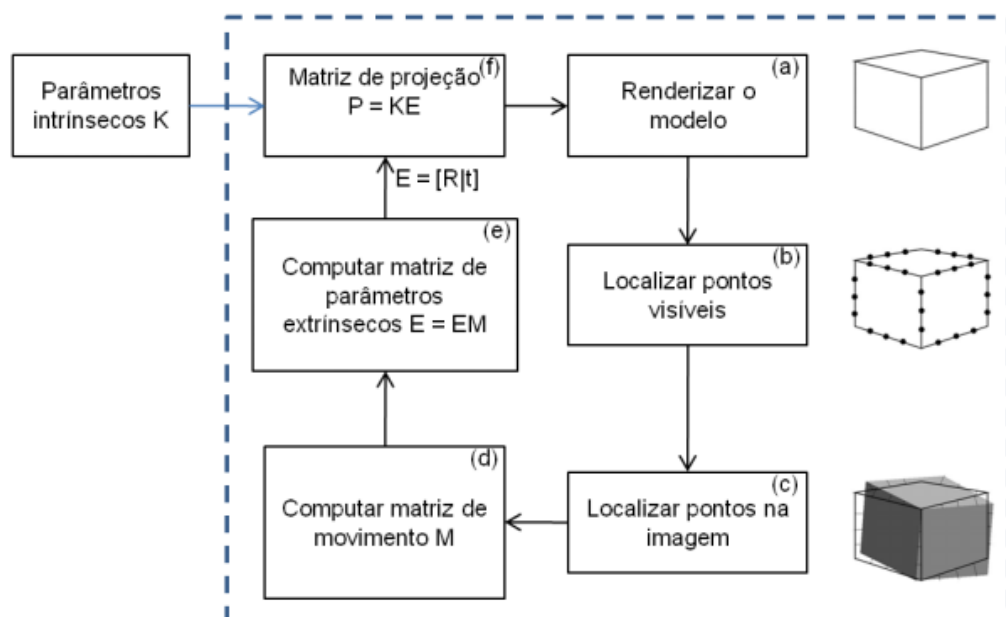


Figura 25: Funcionamento do algoritmo de amostragem de pontos (Simões, 2008).

Primeiramente é necessária a estimativa da pose inicial da câmera para realizar o rastreamento. A cada ciclo do algoritmo, é renderizado o modelo do objeto que se deseja rastrear utilizando o quadro anterior para estimar sua pose e extrair as arestas que estão visíveis. A partir das mesmas, os pontos das arestas são localizados e então correlacionados com os pontos do objeto. Após o correlacionamento dos pontos, é possível estimar o movimento relativo entre a pose anterior e a pose atual. Uma das formas de se estimar o movimento é através da técnica IRLS que utiliza vários parâmetros de minimização. A partir da estimativa dos movimentos é possível calcular a matriz de parâmetros extrínsecos que será utilizada no último passo (Simões, 2008).

3.1.2. MAR baseada na detecção da ponta dos dedos

Este método consiste no reconhecimento da mão através da ponta dos dedos, assumindo que os cinco dedos estão visíveis na cena. Para realizar o reconhecimento, o mesmo utiliza 3 passos: detecção da pele, detecção da ponta dos dedos e estimativa da pose da mão. A figura 22 mostra como é feito o reconhecimento da mão (Kato *et al*, 2011).

A detecção da pele é feita através de um modelo estatístico Gaussiano de aprendizagem. Tanto os tons de pele como os tons que não são de pele são modelados para que a variação de luminosidade não interfira tanto no processo (Kato *et al*, 2011).

A detecção da ponta dos dedos é feita escolhendo a região que contém a maior concentração de pixels com tons de pele, e através dessa região, é feito o contorno de toda região. Os vértices deste contorno são, então, escolhidos como candidatos a serem as pontas dos dedos. As regiões que contém tons que não são de pele também são utilizados para determinar cada dedo e somente as regiões que estão cercadas pelos dedos são utilizadas (Kato *et al*, 2011).

A estimativa da pose da mão é feita através da homografia da posição dos cinco dedos (Kato *et al*, 2011).

3.1.3. Detecção de objetos utilizando OpenCV

Técnica implementada no presente trabalho, a detecção de objetos utilizando OpenCV consiste no reconhecimento de características que contém alguma informação do objeto que se deseja detectar. Chamado de Haar Classifier (Classificador Haar), o mesmo implementa uma cascata de rejeição. O termo “Haar” foi utilizado por causa da característica

do classificador, que utiliza características haar-like wavelets, a qual consiste na adição e subtração de regiões retangulares da imagem antes de aplicar o threshold (Bradski e Kaebler, 2008).

O Classificador Haar utiliza dados supervisionados, ou seja, o classificador aprende através de rótulos o que deve ser reconhecido. Através desses rótulos, o classificador os busca nas imagens, que são apresentadas através de pacotes de histogramas e tamanhos iguais (Bradski e Kaebler, 2008).

O Classificador Haar foi proposto por Paul Viola e depois estendido por Rainer Lienhart (Bradski e Kaebler, 2008).

Paul Viola utiliza um classificador baseado no método AdaBoost, porém organiza-o como uma cascata de rejeição por nós.

Em (Viola *et al.*, 2001) os autores contribuem com três “técnicas” para o reconhecimento de objetos que citaremos a seguir.

1 – A primeira contribuição é a representação da imagem de uma forma diferente, chamada integral image. A integral image pode ser computada por uma imagem através de algumas operações por pixels. A detecção utiliza três características semelhantes às funções baseadas em Haar. A figura 26 mostra as três características.

- 1- O valor da diferença do somatório dos pixels de dois retângulos que tem o mesmo tamanho e formato, além de serem adjacentes.
- 2- O valor da diferença do somatório dos pixels de dois retângulos com o somatório do retângulo central.
- 3- O valor da diferença do somatório de dois retângulos onde o segundo retângulo localiza-se na diagonal do primeiro.

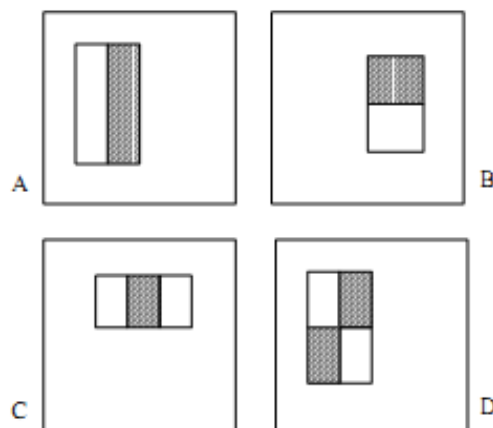


Figura 26: (A) e (B) característica 1 (C) característica 2 e (D) característica 3 (Viola *et al.*, 2001).

2 – A segunda contribuição é um método que constrói um classificador que seleciona um número pequeno de características utilizando AdaBoost.

O método utilizado é uma variação do método AdaBoost, que é usado para selecionar um pequeno conjunto de regiões retangulares da imagem e treinar o classificador.

Em sua forma original, o método AdaBoost de aprendizagem é utilizado para aumentar a performance de um algoritmo de aprendizagem simples.

Há a hipótese que um número pequeno de regiões retangulares possa ser combinado para formar um classificador efetivo. Esta hipótese foi levantada, pois computar toda imagem é um processo muito caro em questão de processamento, necessitando de máquinas com capacidade de processamento gráfico eficiente.

Para realizar o proposto na hipótese, o algoritmo de aprendizagem seleciona a região retangular que melhor separa o objeto de interesse do resto da imagem. Para realizar esta seleção, cada região retangular determina a função de classificação ótima por threshold, e através da mesma o número de classificações erradas é reduzida.

3 – A terceira contribuição é um método que combina classificadores complexos em uma estrutura cascade (cascata), a qual aumenta a velocidade do detector por focar somente em regiões da imagem que são promissoras.

O classificador pode rejeitar a maior parte das regiões retangulares da imagem que não contém o objeto de interesse enquanto detecta quase todas as regiões que contém.

A detecção do objeto é feito através de um processo em cascata. A avaliação positiva do primeiro classificador desencadeia a avaliação de um segundo classificador ajustado para atingir alta taxa de detecção, que caso seja obtido um resultado positivo, desencadeia a avaliação de um terceiro classificador e assim por diante. Caso algum classificador obtenha um resultado negativo, a região retangular é descartada. Para casos onde o objeto a ser detectado é raro, o método utilizado pode reduzir de forma considerável em termos de processamento, pois a maioria das regiões analisadas são obtidos resultados negativos. A figura 27 ilustra a avaliação baseada em face.

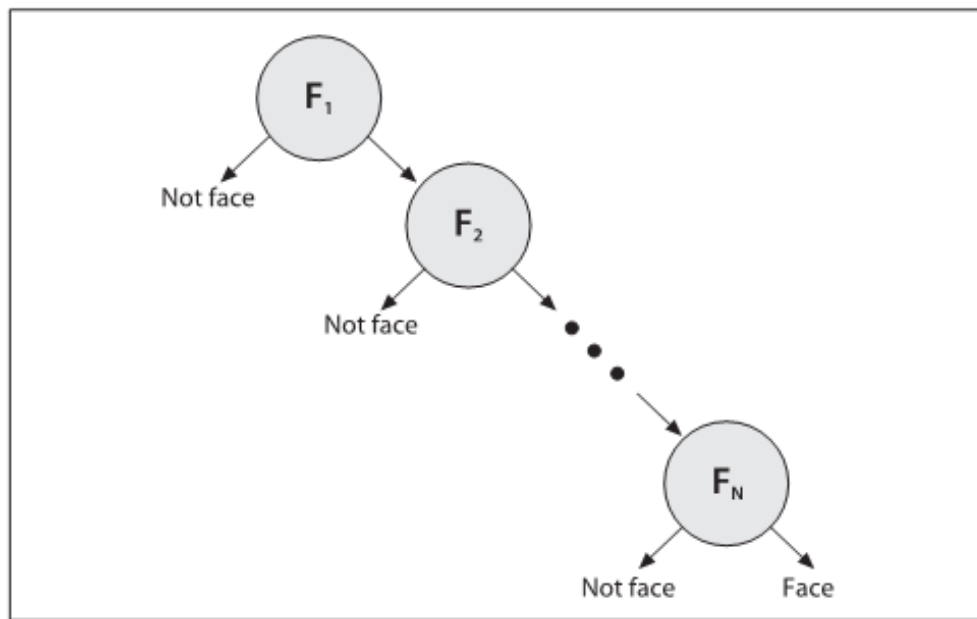


Figura 27: Avaliação das regiões retangulares de uma imagem (Bradski e Kaebler, 2008).

Os estágios de classificação são feitos através do treinamento dos classificadores com o algoritmo AdaBoost e o ajuste do threshold. A cada estágio é reduzido o número de falso-positivo e a taxa de detecção. Cada estágio é treinado adicionando características até que a taxa de detecção, fornecido pelo usuário, e falso-positivo são atendidas. Os mesmos são ordenados de acordo com a complexidade, onde o primeiro é o menos complexo, enquanto o último o mais complexo. Dessa forma, o processamento é minimizado ao rejeitar regiões mais simples de serem avaliadas.

Em (Lienhart *et al.*, 2002) é proposto uma melhoria: o conjunto de características é estendida por um conjunto de características rotacionadas em um ângulo de 45 graus, as quais podem ser computadas rapidamente e constantemente.

3.2. Prototipação

Para a implementação do protótipo foi utilizado a API OpenCV e OpenGL com a ferramenta Microsoft Visual Studio 2010 em linguagem C++.

Foi necessário utilizar a janela do OpenGL ao invés do OpenCV, pois na janela do OpenCV não foi possível desenhar um objeto 3D. Para exibir a imagem capturada pela câmera, a mesma é aplicada como textura em um objeto, no caso um retângulo, e a cada frame, a textura é atualizada. A partir da imagem capturada, a mesma é analisada e, caso o objeto de interesse esteja presente na cena, marca e projeta um objeto 3D sobre o mesmo. A figura 28 mostra o fluxo do protótipo.

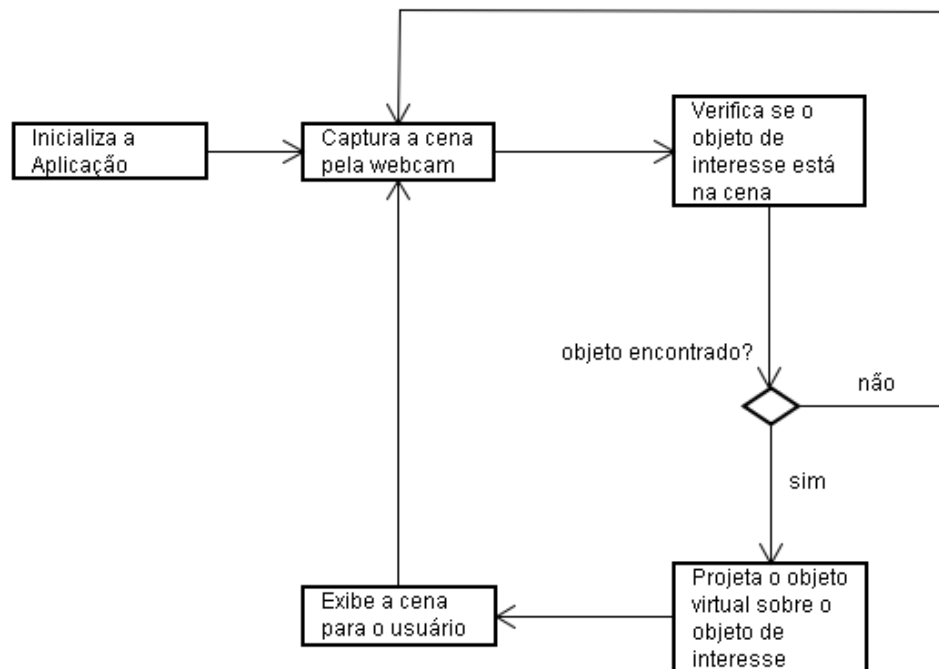


Figura 28: Fluxo do algoritmo do protótipo desenvolvido.

Na função principal, é necessário definir algumas características da janela que exibirá a imagem para o usuário. Primeiramente é definido o tipo de exibição através do comando `glutInitialDisplayMode()`. Após definir o tipo de exibição, é determinada a posição da janela no monitor (`glutInitWindowPosition()`), o tamanho da janela (`glutInitWindowSize()`) e então é criada a janela (`glutCreateWindow()`) (Manssour, 2005).

Através da função `glutDisplayFunc()` e `glutIdleFunc()` é definida a função que será chamada para atualizar a janela com a imagem desejada. A primeira função define uma função, passada por parâmetro, como função de call-back de exibição, que será chamada toda vez que a janela precisar ser redesenhada. A função definida normalmente é a mesma que exibe ou modela um objeto. A segunda função pode realizar tarefas ou animações contínuas, sendo chamada continuamente, enquanto nenhum evento é realizado (Manssour, 2005) (Kilgard, 1996).

A função `glutPostRedisplay()`, a cada iteração, fará com que o método definido para exibição seja chamado para redesenhar o plano da janela. A função `glutMainLoop()` é a responsável por inicializar a máquina de estados do OpenGL. Já a função `glViewport()` determina a área da janela criada que o OpenGL utilizará para exibir ou modelar objetos (Manssour, 2005) (Kilgard, 1996).

Foi definida uma função de inicialização que realiza as configurações necessárias para iniciar o aplicativo. A figura 28 mostra a função de inicialização.

```

140 int init(void)
141 {
142     cascade = ( CvHaarClassifierCascade* )cvLoad( "hand_cascade.xml", 0, 0, 0 );
143     storage = cvCreateMemStorage( 0 );
144     capture = cvCaptureFromCAM( 0 );
145
146     char cadena[25]="GestionMarcaAR\0";
147     char **argv;
148     int argc = 1;
149
150     argv=(char **)malloc(sizeof(char *));
151     *argv=(char *)malloc(sizeof(char)*10);
152     strcpy(*argv, cadena);
153
154     GLuint mTexture;
155     glGenTextures(1, &mTexture);
156     glBindTexture(GL_TEXTURE_2D, mTexture);
157     glEnable(GL_TEXTURE_2D);
158     glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
159     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
160     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
161     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
162     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
163     image = cvLoadImage("logo-compsi2.png");
164
165     glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
166     glClearColor(1.0f, 1.0f, 1.0f, 1.0f); //color blanco de fondo
167     glColor3f(0.0f,0.0f,0.0f);
168
169     return 0;
170 }

```

Figura 29: Função de inicialização.

Nesta função são feitas as seguintes configurações:

- Linha 142: carregar arquivo cascata através da função `cvLoad` (Bradski e Kaebler, 2008);
- Linha 143: alocação de memória para objetos dinâmicos através da função `cvCreateMemStorage` (Bradski e Kaebler, 2008);
- Linha 144: inicializa a câmera pela função `cvCaptureFromCAM` (Bradski e Kaebler, 2008);
- Linha 155: atrela uma textura a um número de identificação através da função `glGenTextures` (Kilgard, 1996);
- Linha 156: define a textura criada como sendo a textura corrente através da função `glBindTexture` (Kilgard, 1996);
- Linha 157: habilita o uso de texturas através da função `glEnable`;
- Linhas 159 à 162: atribui valores para o parâmetro de textura especificado e especifica onde será aplicado através da função `glTexParameteri` (Kilgard, 1996);

- Linha 165: especifica como os valores da textura são interpretados quando a mesma é aplicada em um objeto através da função `glTexEnvf` (Kilgard, 1996);
- Linha 166: limpa a janela através da função `glClearColor` (Massour, 2005);

Definido também uma função para exibir a imagem obtida pela câmera e renderizar o objeto sobre a região reconhecida, que no caso será a mão. Nesta função são realizados os seguintes procedimentos:

- define o modo de matriz através da função `glMatrixMode` e aplica as operações de matrizes subsequentes a mesma. O primeiro modo definido é o de projeção, ou seja, define que as alterações futuras afetarão a câmera (Kilgard, 1996) (Massour, 2005);
- inicia o sistema de coordenadas definida pelo comando `glMatrixMode`, substituindo a matriz corrente pela matriz identidade através da função `glLoadIdentity` (Kilgard, 1996);
- define os parâmetros da Projeção Perspectiva, atualizando a matriz da mesma através da função `gluPerspective` (Manssour, 2005);
- define o modo de matriz através da função `glMatrixMode` e aplica as operações de matrizes subsequentes a mesma. O primeiro modo definido é o de modelo, ou seja, define que as alterações futuras afetarão os objetos desenhados (Kilgard, 1996) (Massour, 2005);
- inicia o sistema de coordenadas definida pelo comando `glMatrixMode`, substituindo a matriz corrente pela matriz identidade através da função `glLoadIdentity` (Kilgard, 1996);
- determina a posição da câmera e para onde ela está apontada, ou seja, a posição do objeto através da função `gluLookAt` (Massour, 2005);
- captura os frames da câmera através da função `cvQueryFrame` (Bradski e Kaebler, 2008);
- rotaciona a imagem no eixo y através da função `cvFlip` (Bradski e Kaebler, 2008);
- faz a chamada da função que detecta o objeto e passa como parâmetro o frame capturado pela câmera;
- define o frame como textura em 2d através da função `glTexImage2D` (Kilgard, 1996);

- projeta um retângulo e aplica a textura no mesmo, que exibe a imagem capturada pela câmera. O retângulo é feito através da função `glBegin` que delimita os vértices de uma ou mais primitivas (Kilgard, 1996);
- antes de desenhar o objeto 3D sobre o objeto reconhecido no frame capturado é feito a translação do mesmo através da função `glTranslatef`, o qual recebe como parâmetro a posição do objeto reconhecido (Massour, 2005);
- faz a chamada da função que desenha o objeto 3D, que no caso será um cubo;
- executa todo e qualquer comando OpenGL que não foi executado através do método `glFlush` (Massour, 2005);

```

64 void detectObject( IplImage *img )
65 {
66     int i;
67
68
69     CvSeq *hand = cvHaarDetectObjects(
70         img,
71         cascade,
72         storage,
73         1.1,
74         3,
75         0 ,
76         cvSize( 40, 40 ) );
77
78
79     for( i = 0 ; i < ( hand ? hand->total : 0 ) ; i++ ) {
80         CvRect *r = ( CvRect* )cvGetSeqElem( hand, i );
81         cvRectangle( img,
82             cvPoint( r->x, r->y ),
83             cvPoint( r->x + r->width, r->y + r->height ),
84             CV_RGB( 255, 255, 255 ), 1, 8, 0 );
85
86         x = ((float)r->x-270) / -318.75;
87
88         y = ((float)r->y-270) / -231.25;
89     }
90 }

```

Figura 30: Função responsável por detectar o objeto de interesse.

A função que detecta o objeto no frame capturado pela câmera, ilustrada pela figura 29, realiza os seguintes procedimentos:

- Linha 69: chama o método `cvHaarDetectObject` para detectar o objeto de interesse. Este método recebe como parâmetro uma imagem, um arquivo cascata, um buffer, o fator de escala, vizinhos mínimos, flag e valor mínimo. A imagem será frame capturado em escala de cinza, o arquivo cascata será o mesmo carregado anteriormente, a alocação de memória será a mesma feita

anteriormente, o buffer será a alocação de memória feita anteriormente, o fator de escala será um valor que determinará o valor do salto entre cada escala, o parâmetro vizinhos mínimos determina a quantidade de sobreposição de detecção necessária para validar a detecção do objeto, o parâmetro flag determina a forma de detecção do objeto e o valor mínimo determina a menor região que será verificada se existe o objeto de interesse (Bradski e Kaebler, 2008);

- Linha 79: para cada objeto encontrado, é desenhado um retângulo em volta, indicando que o mesmo foi reconhecido, e a sua posição no eixo x e y são armazenados para projetar o objeto 3D;

Para desenhar o objeto 3D sobre o objeto de interesse detectado foi criado um método que utiliza as seguintes funções:

- `glTranslatef`: função responsável por transladar o objeto. A mesma multiplica a matriz atual por uma matriz de translação com os valores passados por parâmetro (Massour, 2005);
- `glRotatef`: função responsável por rotacionar o objeto. A mesma multiplica a matriz atual por uma matriz de rotação utilizando os valores passados por parâmetro (Massour, 2005);
- `glBegin`: delimita os vértices de uma ou mais primitivas. No presente projeto foi utilizado um cubo como objeto 3D de projeção sobre o objeto de interesse detectado na cena (Kilgard, 1996);
- `glTexImage2d`: define uma textura em 2d e mapeia a mesma para o objeto que está habilitado a receber a textura (Kilgard, 1996);

No capítulo 4 serão apresentados os resultados obtidos a partir da implementação do protótipo.

CAPÍTULO 4 - RESULTADOS OBTIDOS

Através da avaliação do protótipo com usuários, foi possível detectar o objeto de interesse, no caso a mão fechada, e realizar a projeção do objeto 3D sobre a mão detectada como é mostrado na figura 30. Quando há detecção, o objeto 3D é projetado sobre o objeto de interesse reconhecido e um quadrado com bordas brancas é feito em volta do objeto de interesse.

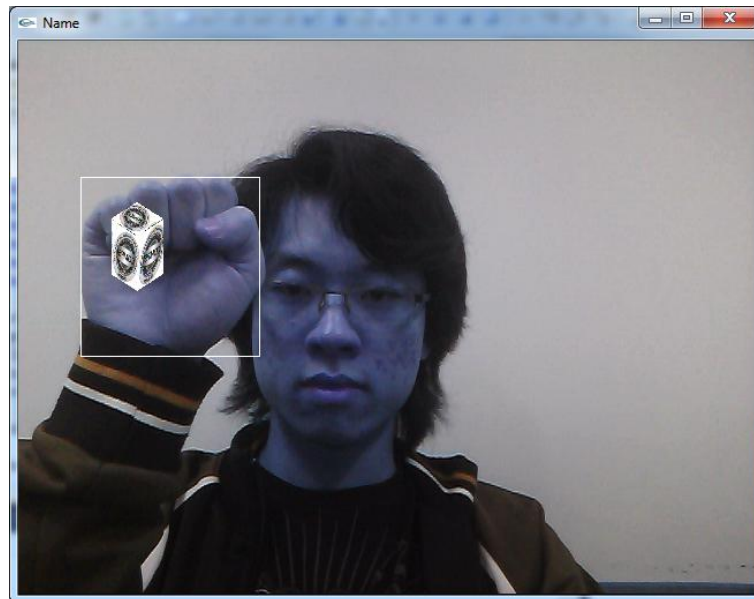


Figura 31: Reconhecimento da mão e projeção do objeto 3D sobre a mesma.

Na figura 31 pode-se observar que algumas áreas da cena real são reconhecidas como objeto de interesse, que no caso é a mão fechada, porém essas mesmas áreas reconhecidas não são o objeto de interesse.



Figura 32: Reconhecimento de áreas não desejadas.

Isto ocorre pois, de acordo com (Bradski e Kaebler, 2008), para criar um bom classificador é necessário de 1.000 a 10.000 imagens de alta qualidade. O termo alta qualidade refere-se à imagem ter somente o objeto que se deseja detectar. O alto número de imagens necessárias para um bom classificador torna-o difícil de conseguir, porém com poucas imagens podemos verificar que o classificador ainda reconhece bem o objeto de interesse.

O método utilizado, apesar da baixa qualidade do classificador, consegue acompanhar a mão em movimento, porém movimentos não muito rápidos e bruscos. A figura 32 mostra o reconhecimento da mão enquanto o usuário realiza o movimento. A variação de iluminação também prejudica o reconhecimento da mão como é mostrado na figura 33.

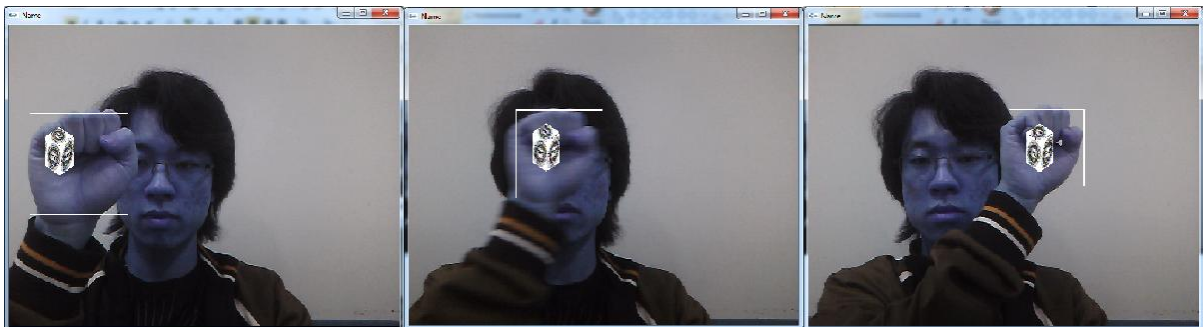


Figura 33: Reconhecimento da mão em movimento.

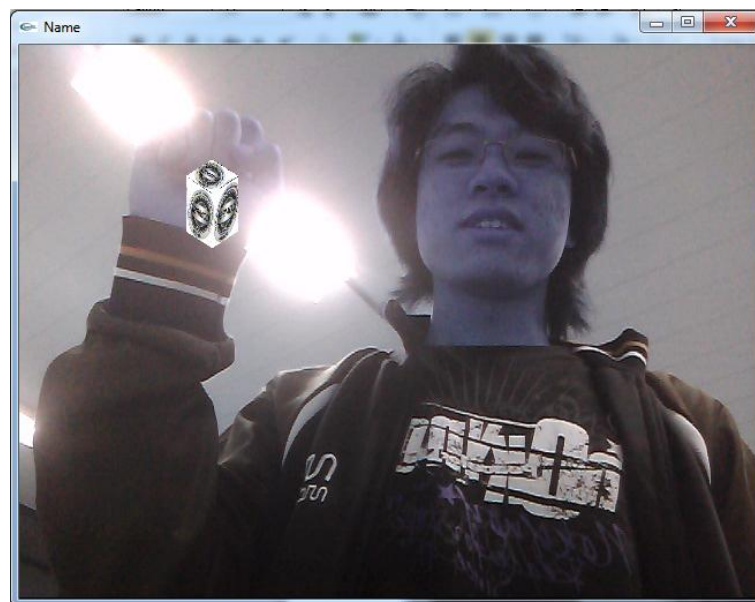


Figura 34: A alta variação de iluminação prejudica o reconhecimento da mão.

Na figura 34 pode-se verificar que o método consegue reconhecer mais de um objeto na cena, porém a projeção do objeto 3D ocorre somente em um dos objetos. Isto ocorre pois no protótipo é feito somente a projeção de um objeto ao invés de projetar na mesma quantidade das mãos. Pode-se verificar também que na figura 35, dependendo da pose da

mão, a mesma não é reconhecida. O motivo de a mão não ter sido reconhecida é que o classificador não contém um modelo da mão naquela posição.



Figura 35: Reconhecimento de mais de um objeto de interesse na cena.

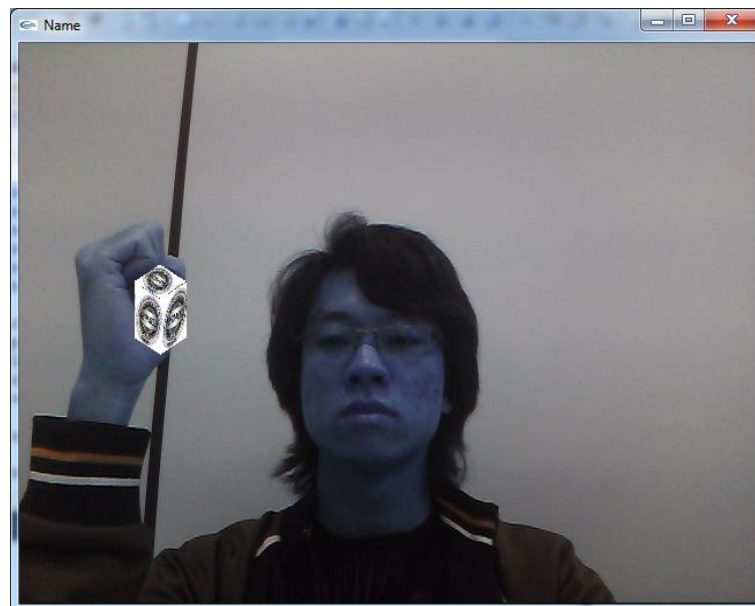


Figura 36: Não reconhecimento da pose da mão.

No capítulo 5 serão apresentados as conclusões obtidas através da implementação e validação do protótipo criado.

CAPÍTULO 5 - CONCLUSÕES

Com a metodologia proposta era esperado:

- Entender o funcionamento da técnica baseada em aresta utilizando a ferramenta OpenCV para a implementação do protótipo com a linguagem C++.
- A validação do protótipo criado, reconhecendo objetos sem marcadores recebidos por um dispositivo de entrada, que no caso será uma webcam.
- Aumentar a interação do usuário com o objeto apresentados em tela sem a utilização de marcadores.

Para a implementação, foi utilizado a ferramenta OpenCV para adquirir os frames da câmera e o OpenGL para a exibição do frame capturado e a projeção do objeto 3D na cena real. Os mesmos são apresentados através do monitor. Foi utilizado o Microsoft Visual Studio 2010 e a linguagem C++ para a implementação.

A técnica de reconhecimento de objetos baseadas em classificador em cascata foi implementada com sucesso e mostrou-se robusta a movimentos suaves e variação de luminosidade mesmo com um classificador de baixa qualidade. Porém, o classificador reconhece áreas da cena como objeto de interesse sendo que as mesmas áreas não são o objeto de interesse, causando desconforto para o usuário. Mostrou-se instável a alta variação de iluminação e movimentos rápidos.

Porém, no protótipo criado no presente projeto possui a limitação de não projetar mais de um objeto virtual na mesma cena. Outra limitação é a questão da variação de luminosidade e movimentos bruscos, sendo necessário equipamentos de uma qualidade melhor do que os usados para a criação do protótipo.

Através da técnica implementada, pode-se notar uma maior interação do usuário com o objeto projetado em tela. Com a ação do usuário em exibir e movimentar a mão, o protótipo reage exibindo e movimentando o objeto virtual de acordo com os movimentos da mão, havendo interação entre o objeto virtual e o real sem a necessidade de outros objetos ou equipamento de intermédio.

REFERÊNCIAS

- ARToolKit. **Feature List, ARToolKit.** Disponível em: <
<http://www.hitl.washington.edu/artoolkit/documentation/features.htm> >. Acesso em 17 de
 Maio de 2011.
- Azuma, Ronald. **A Survey of Augmented Reality.** Presence: Teleoperators and Virtual
 Environments, v. 6, n. 4, Hughes Research Laboratories, August 1997, p. 355-385.
- Botega, Leonardo *et al.* **Realidade Virtual: Histórico, Conceitos e Dispositivos.** Aplicações
 de Realidade Virtual e Aumentada, Pré-Simpósio SVR2009: XI Simpósio de Realidade
 Virtual e Aumentada, 2009, Porto Alegre – RS, p. 08-30.
- Bradski e Kaebler, Gary e Adrian. **Learning OpenCV Computer Vision with OpenCV
 Library.** United States of America: O'Reilly Media, Inc., 2008.
- BUCCIOLI, Arthur Augusto Bastos ; ZORZAL, E. R. ; KIRNER, Claudio . **Usando
 Realidade Virtual e Aumentada na Visualização da Simulação de Sistemas de
 Automação Industrial.** In: SVR2006 - VIII Symposium on Virtual Reality, 2006, Belém-
 PA, 2006. Disponível em: <<http://www.realidadeaugmentada.com.br/artigos/16585.pdf>>
- Bueno, Márcio Augusto Silva *et al.* **Meta3D++ – Visualização de Informações em
 Realidade Aumentada.** WRA 2005 – II Workshop de Realidade Aumentada, 2005.
 Disponível em: <<http://www.ckirner.com/download/anais/WRA2005-Anais/WRA2005-25-55.pdf>>
- Gorodnichy, D. **Video Based Framework for Face Recognition in Video.** Second Workshop
 on Face Processing in Video (FPiV'05) in Proceedings of Second Canadian Conference on
 Computer and Robot Vision (CRV'05). pp. 330-338. Victoria, British Columbia, Canada,
 2005. National Research Council Canada, 2005.
- Kato, Haruhisa *et al.* **A Marker-Less Augmented Reality Based on Fast Fingertip
 Detection for Smart Phones.** IEEE International Conference on Consumer Electronics
 (ICCE), 2011.
- Kilgard, Mark. **The OpenGL Utility Toolkit (GLUT) Programming Interface API Version
 3.** Silicon Graphics, Inc., 1996. Disponível em:
 <<http://www.opengl.org/resources/libraries/glut/spec3/spec3.html>>
- Kirner, Claudio *et al.* **Fundamentos de Realidade Virtual e Aumentada.** Realidade Virtual e
 Aumentada: Uma abordagem tecnológica, Livro do Pré-Simpósio X Symposium on Virtual
 and Augmented Reality, 2008, João Pessoa – PB, p. 01-20.
- Kirner, Claudio *et al.* **Fundamentos de Realidade Virtual e Aumentada.** Realidade Virtual e
 Aumentada: Conceitos, Projetos e Aplicações, Livro do Pré-Simpósio IX Symposium on
 Virtual and Augmented Reality, 2007, Petrópolis – RJ, p. 02-21.
- Kodak. Disponível em: <
http://store.kodak.com.br/store/ekconsus/pt_BR/pd/C%C3%A2mera_TOUCH/productID.22432900>
- Lienhart, Rainer *et al.* **An Extended Set of Haar-like Features for Rapid Object Detection.**
 Intel Labs, Intel Corporation, Santa Clara, 2002. Disponível em:
 <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1038171>

- Manresa, Cristina *et al.* **Hand Tracking and Gesture Recognition for Human-Computer** Electronic Letters on Computer Vision and Image Analysis 5. Interaction. Electronic Letters on Computer Vision and Image Analysis, 2005, p. 96-104.
- Manssour, Isabel Harb. **Introdução à OpenGL.** 2005. Disponível em: <<http://www.inf.pucrs.br/~manssour/OpenGL/Tutorial.html>>
- Mellor, J. P. **Enhanced Reality Visualization in a Surgical Environment.** A.I. Technical Report No. 1544, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1995. Disponível em: <<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.3364>>
- Platonov e Langer, Juri e Marion. **Automatic Contour model creation out of polygonal CAD models for markerless Aumented Reality.** Metaio GmbH, 2007. Disponível em: <<http://www.computer.org/portal/web/csdl/doi/10.1109/ISMAR.2007.4538829> >. Acesso em 05 de Maio de 2011.
- Simões, Francisco P. M.. **Realidade Aumentada Sem Marcadores Baseadas em Arestas, um estudo de caso.** Trabalho de Graduação, Universidade Federal de Pernambuco, 2008, Recife - PE. Disponível em: < <http://www.cin.ufpe.br/~tg/2008-2/fpms.pdf> >. Acesso em 21 de Novembro de 2010.
- Simões, Francisco P. M. *et al.* **Realidade Aumentada sem Marcadores Baseada na Amostragem de Pontos em Arestas.** Grupo de Pesquisa em Realidade Virtual e Multimídia, Universidade Federal de Pernambuco, 2008, Recife - PE. Disponível em: < <http://www2.fc.unesp.br/wrva/artigos/50429.pdf> >. Acesso em 21 de Novembro de 2010.
- Stricker, Didier *et al.* **Real-time and Markerless Vision-Based Tracking for Outdoor Augmented Reality Applications.** In: Proceedings of the International Symposium on Augmented Reality, 2001. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.1333>>
- Teichrieb, Veronica *et al.* **A Survey of Online Markerless Augmented Reality.** **International Journal of Modeling and Simulation for the Petroleum Industry**, VOL 1, NO. 1, 2007. Disponível em: <http://rpcmod.ganer.ex-br.com/revista/articles/1.pdf>
- Viola, Paul *et al.* **Rapid Object Detection using a Boosted Cascade of Simple Features.** ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2001.
- Zhou, Shaohua *et al.* **Probabilistic recognition of human faces from video.** Computer Vision and Image Understanding 91. Center for Automation Research. Department of Electrical and Computer Engineering. University of Maryland, 2003, p. 214-245.
- ZORZAL, E. R.; BUCCIOLI, Arthur Augusto Bastos; KIRNER, Cláudio. **Usando Realidade Aumentada no Desenvolvimento de Quebra-cabeças Educacionais.** In: SVR2006 - VIII Symposium on Virtual Reality, 2006, Belém-PA. Proceedings of VIII Symposium on Virtual Reality. Belém, PA : Editora CESUPA, 2006. p. 221-232. Disponível em: <<http://www.realidadeaumentada.com.br/artigos/16457.pdf>>. Acesso em 07 de Maio de 2011.
- ZORZAL, E. R. ; BUCCIOLI, Arthur Augusto Bastos ; KIRNER, Claudio . **O uso da Realidade Aumentada no Aprendizado Musical.** In: WARV - WORKSHOP DE APLICAÇÕES DE REALIDADE VIRTUAL, 2005, Uberlândia-MG, 2005. Disponível em: < http://realidadeaumentada.com.br/artigos/WARV_Music.pdf >. Acesso em 07 de Maio de 2011.

ZORZAL, E. R. ; BUCCIOLI, Arthur Augusto Bastos ; KIRNER, Claudio . **Desenvolvimento de Jogos em Ambiente de Realidade Aumentada**. In: SBGAMES2005 - SIMPÓSIO BRASILEIRO DE JOGOS PARA COMPUTADOR E ENTRENIMENTO DIGITAL, 2005, São Paulo - SP. WJOGOS2005 - IV Workshop Brasileiro de Jogos e Entretenimento Digital, 2005. p. 152-161. Disponível em: <<http://www.realidadeaumentada.com.br/artigos/14534.pdf>>