

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Calibração Automática de Múltiplos Marcadores em Ambiente de Realidade Aumentada

Felipe Andrade Caetano

JUIZ DE FORA
NOVEMBRO, 2011

Calibração Automática de Múltiplos Marcadores em Ambiente de Realidade Aumentada

FELIPE ANDRADE CAETANO

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Rodrigo Luis de Souza da Silva

JUIZ DE FORA

NOVEMBRO, 2011

CALIBRAÇÃO AUTOMÁTICA DE MÚLTIPLOS MARCADORES EM AMBIENTE DE REALIDADE AUMENTADA

Felipe Andrade Caetano

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Rodrigo Luis de Souza da Silva
Doutor em Engenharia

Marcelo Bernardes Vieira
Doutor em Ciência da Computação

Marcelo Caniato Renhe
Mestre em Engenharia de Sistemas e Computação

JUIZ DE FORA
30 DE NOVEMBRO, 2011

Aos meus pais.

Resumo

Este trabalho tem como objetivo desenvolver uma abordagem para sistemas de Realidade Aumentada que utilizam múltiplos marcadores para redundância de informação, visando manter a estabilidade e alinhamento da projeção mesmo quando haja oclusão de alguns marcadores. A abordagem proposta permite que tal cenário seja construído sem que seja necessária a etapa de calibragem das relações de rotação e translação entre os marcadores, normalmente adotada na pré-execução deste tipo de sistema. O método desenvolvido guarda as relações entre os marcadores visíveis e um marcador base, que é responsável por ditar as transformações do objeto virtual projetado. Serão propostos métodos para se escolher qual o marcador base, dentre os marcadores visíveis.

Palavras-chave: Realidade Aumentada, Múltiplos marcadores, Calibragem.

Abstract

This work's purpose is to develop an approach for Augmented Reality systems using multiple markers for redundancy, in order to maintain stability and alignment of the augmentation even when some markers are not fully visible. The proposed approach allows to build such scenario without needing of the calibration the relations of rotation and distance between the markers, usually adopted in the pre-execution of these systems. The method keeps the relations between the visible markers and a base marker which is responsible for dictating the transformation of the virtual object projected. Will be proposed methods to choose the base marker, within the visible markers.

Keywords: Augmented Reality, Multiple Markers, Calibration.

Agradecimentos

Amigos e familiares, de maneira geral, que contribuíram para a minha formação, não só acadêmica mas, como pessoa.

A CAPES pela disponibilização de acesso à inúmeros trabalhos pelo mundo, sem o qual esta monografia não seria a mesma.

Aos amigos do NRC e do DCC pelo apoio e pela paciência.

Aos amigos Mariana e Ivo pelas melhores lembranças e por ainda estarem presentes no meu dia-a-dia.

Pela iniciativa e ajuda que os professores Alessandra, Jairo e Rodrigo Luis têm fornecido para os alunos concluintes.

“Tudo isso está muito bem dito, mas devemos cultivar nosso jardim”.

Voltaire (Cândido ou O otimista)

Sumário

Lista de Figuras	7
Lista de Tabelas	9
1 Introdução	10
1.1 Problema a ser abordado	10
1.2 Objetivo	11
1.3 Estrutura do trabalho	11
2 Fundamentação Teórica	13
2.1 Realidade Aumentada	13
2.1.1 Dispositivos de rastreamento	13
2.1.2 Aplicações	20
2.1.3 Ambientes com Múltiplos Marcadores	23
2.2 ARToolkit	23
2.2.1 Funcionamento Básico	24
2.2.2 Código fonte	25
2.2.3 Estrutura de Dados	28
2.3 Álgebra de quatérnios	29
2.4 Trabalhos Relacionados	31
3 Modelo Proposto	34
3.1 Abordagem utilizada	34
3.1.1 Obtendo as relações entre os marcadores	34
3.2 Sistema multi marcador sem calibração simples	36
3.3 Sistema multi marcador sem calibração para vários marcadores	37
3.4 Métodos para escolha do marcador base	39
3.4.1 Por ordem de prioridade	39
3.4.2 Marcador com menor inclinação em relação à câmera	40
3.4.3 Marcador com maior acurácia	41
3.4.4 Marcador mais próximo da câmera	42
3.4.5 Marcador mais próximo da projeção	42
3.4.6 Marcador com maior resistência à mudança	44
3.5 Aplicações	44
3.6 Correção de Erro	44
3.7 Limitações	46
4 Resultados finais	47
4.1 Cálculo de Erro	47
4.2 Comparação entre os métodos	48
5 Conclusão e Trabalhos futuros	52
Referências Bibliográficas	53

Lista de Figuras

2.1	Esquema de funcionamento do dispositivo mecânico (Fonte: Sutherland (1968)).	14
2.2	Exemplo de marcador usado pelo Artoolkit (Fonte:Kato (2002)).	16
2.3	(a) Marcador visível com detecção correta dos vértices. (b) Efeitos da oclusão parcial na detecção dos marcadores.	17
2.4	(a) Cena em que um marcador é parcialmente coberto pela sombra de um objeto. (b) A segmentação da cena anterior. (c) A mesma cena, sem o objeto criando a sombra. (d) Segmentação do marcador sem sombra, usando o mesmo <i>threshold</i> do caso anterior.	18
2.5	Exemplo em que o Artoolkit confunde a tela de um celular com um marcador invisível na cena. Com o marcador invisível, a tela do celular era o que mais se aproximava do objetivo.	19
2.6	Realidade Aumentada aplicada à educação (Fonte:Kaufmann e Schmalstieg (2003)).	20
2.7	Paciente em tratamento contra fobia deixa aranhas virtuais se aproximarem de sua mão (Fonte: Juan et al (2005)).	22
2.8	RA aplicada à mecânica (Fonte: Dangelmaier et al (2005)).	22
2.9	Principais etapas de Visão Computacional do Artoolkit. (a) A imagem original. (b) Segmentação e Binarização da imagem. (c) Extração dos componentes conectados. (d) Extração dos contornos. (e) Extração de quinas. (f) Imagem virtual sobreposta. (Fonte:Kato (2002)).	24
2.10	Fluxograma básico das aplicações que utilizam o Artoolkit.	25
2.11	Exemplo de arquivo de Configuração para múltiplos marcadores (Fonte:Kato (2002)).	25
2.12	Sistema de Coordenadas do Artoolkit (Fonte:Kato (2002)).	27
2.13	Exemplo de configuração onde os marcadores são dispostos de maneira a ficar impossível exibir todos ao mesmo tempo (Fonte:Siltanen e Hakkarainen e Honkamaa (2007)).	33
3.1	Fluxograma geral do multimarcador simples.	37
3.2	Exemplo do sistema simples em execução: (a) O marcador onde é desejado projetar o cubo é detectado. (b) Marcador auxiliar encontrado, como ambos estão visíveis, as relações entre eles são computadas. (c) Com a oclusão do marcador principal, o sistema usa as relações previamente guardadas para tentar manter a projeção no mesmo lugar. (d) Ambos marcadores visíveis de novo, relações recalculadas. (e) Marcador ocluído novamente. (f) Projeção se mantém no lugar correto, mesmo de pontos de vista diferentes.	37
3.3	Fluxograma geral do multimarcador.	38
3.4	Configuração dos marcadores usados.	40
3.5	Marcador com menor inclinação projeta um segmento de reta menor na tela.	41
3.6	(a) Objeto inicialmente projetado sobre o marcador F. (b) Marcador F é ocluído e outro marcador se transforma em marcador base. (c) Marcador F é removido enquanto estava ocluído. (d) Objeto continua sendo projetado onde o marcador F se encontrava.	43

3.7	Exemplo de execução onde se há erro no resultado final.	45
4.1	Cálculo de erro baseado na distância entre os vértices.	47
4.2	Foto da configuração dos marcadores no primeiro vídeo.	49
4.3	Foto da configuração dos marcadores no segundo vídeo.	49
4.4	Foto da configuração dos marcadores no terceiro vídeo.	49
4.5	Foto da configuração dos marcadores no quarto vídeo.	50

Lista de Tabelas

2.1	Distância máxima para diferentes tamanhos de marcadores (Fonte:Kato (2002)).	19
4.1	Tabela de erro, em milímetros, para execução de quatro vídeos diferentes. .	50
4.2	Tabela de erro máximo, em milímetros, com correção de erro para quatro vídeos diferentes.	51

1 Introdução

Na área da Realidade Aumentada (RA) um dos principais problemas de se usar marcadores óticos está no fato de que a oclusão de um marcador, mesmo que de forma parcial, pode fazer com que o sistema deixe de reconhecer com exatidão qual o alinhamento do objeto virtual projetado. Uma técnica comumente usada para contornar este tipo de problema é a combinação de múltiplos marcadores gerando uma única projeção, ou seja, quando um dos marcadores não está mais inteiramente visível, outro marcador, este visível, e que tem as suas coordenadas conhecidas, é usado como referência para a projeção. O problema deste tipo de técnica é que para obter este efeito, todos os marcadores devem estar estáticos e precisam ter sua transformação em relação a um ponto global fixo no espaço conhecido antes da execução. A obtenção dessas transformações de forma automática, em tempo de execução e sem necessitar da interação do usuário é o foco deste trabalho.

No trabalho proposto, foi utilizada a biblioteca Kato (2002). O Artoolkit é uma biblioteca livre e de código fonte aberto voltada para a Realidade Aumentada, o que nos propicia uma maior liberdade para execução de testes e coleta de resultados. Também por isso, seu uso é amplamente difundido na comunidade acadêmica e, apesar de possuir falhas intrínsecas a sistemas de Realidade Aumentada baseados em rastreamento ótico, é na maioria das vezes uma solução robusta para desenvolvedores que desejam usar a RA sem que seja necessário refazer as complicadas etapas de reconhecimento e registro dos marcadores.

1.1 Problema a ser abordado

Como dito anteriormente, muitas aplicações que fazem uso de múltiplos marcadores devem passar pela etapa de calibração. Para a calibração, é necessário definir um plano no espaço e calcular, para cada marcador utilizado, qual a translação e rotação em \mathbb{R}^3 que este marcador está em relação à origem do plano definido. Para facilitar a tarefa, é comum adotar um dos marcadores como referência para os demais, eliminando assim a calibração

de um marcador na cena, como sugerido em Wang et al (2010).

A calibração manual é bastante efetiva quando se está lidando com um conjunto de marcadores que têm as transformações previamente conhecidas, por exemplo, quando impressos numa mesma folha plana ou utilizando uma base que têm seus ângulos conhecidos com precisão. Quando nenhuma dessas situações é encontrada, a calibração pode se tornar extremamente exaustiva por exigir um nível de detalhe que nem sempre é possível de se obter utilizando ferramentas manuais como réguas e transferidores, ou mesmo medidores precisos como trenas à laser ou ultrassônicas. A situação se agrava mais ainda quando a distância entre os marcadores é muito grande, pois, mesmo o menor dos erros na calibração resulta em objetos virtuais projetados em lugares bastante diferentes. Por exemplo, quando tenta-se calibrar manualmente vários marcadores dentro de uma mesma sala, é natural de se esperar erros na casa dos centímetros. Estes erros, apesar de parecerem pequenos em relação à distância total, são extremamente significativos para o cálculo final da pose e posição do objeto virtual.

1.2 Objetivo

O objetivo deste trabalho é desenvolver uma técnica que retire do usuário o trabalho manual de medir as relações entre os marcadores, quando o mesmo desejar criar uma aplicação de Realidade Aumentada em que isso seja necessário. O resultado final deve ser abrangente, robusto e leve, de forma a não haver limitações quanto ao ambiente e cenário de execução. É objetivo deste trabalho também analisar quais os melhores parâmetros do método, em relação às possíveis situações onde ele é aplicável e levantar as possíveis dificuldades em se implantar um sistema do tipo.

1.3 Estrutura do trabalho

Esta monografia será dividida em cinco capítulos, incluindo a introdução. No segundo capítulo, serão dadas as bases e os conceitos de Realidade Aumentada e da biblioteca utilizada, bem como suas respectivas limitações e variedades de uso. No terceiro capítulo, será apresentado com maior riqueza de detalhes como funciona o sistema proposto, propor

abordagens diferentes para sua implementação e compará-las no quarto capítulo, usando como parâmetro a precisão de cada método. O último capítulo traz as considerações finais sobre o trabalho e sugestões de trabalhos futuros.

2 Fundamentação Teórica

2.1 Realidade Aumentada

Azuma (1997) define a Realidade Aumentada como sendo uma variação da realidade virtual que busca uma combinação interativa, em tempo real e tridimensional entre o real e o virtual. Sendo mais específico, a Realidade Aumentada complementa o mundo real com objetos virtuais, gerados por computador, de tal forma que ambos pareçam coexistir de maneira plausível (Azuma et al, 2001).

Atualmente, a maioria dos sistemas de Realidade Aumentada funcionam à base de rastreamento ótico, ou seja, procuram por padrões físicos inseridos no ambiente real, chamados de marcadores, para cumprir com maior exatidão a tarefa de alinhar o objeto virtual criado com a cena, etapa conhecida como registro. Múltiplos objetos podem ser inseridos na cena usando um ou mais marcadores.

2.1.1 Dispositivos de rastreamento

Os sistemas de rastreamento não se restringem ao ótico podendo ser, por exemplo, mecânico, via ultrassom ou magnético. O rastreamento ótico é o mais viável na maioria dos casos por não depender de hardwares específicos e, por consequência, ser mais acessível que os outros.

Magnético

Dispositivos magnéticos são divididos em ativos e passivos. Dispositivos magnéticos ativos contam com uma fonte emissora de campo magnético próxima ao local do rastreamento. Um sensor magnético consegue calcular a posição e a orientação em relação à fonte emissora, calculando a variação do campo captado. Dispositivos magnéticos passivos funcionam usando o mesmo princípio, porém, ao invés de contar com uma fonte emissora local, usam o próprio campo magnético da Terra para se orientar; o mesmo princípio de

uma bússola. A vantagem deste tipo de dispositivo em relação ao rastreamento ótico é que ele é indiferente à variação da luz ambiente. A desvantagem, além de necessitar de um hardware específico, é que a qualidade do rastreamento pode ser afetada por ruídos provocados por outras fontes emissoras de campo magnético diferentes das utilizadas no sistema.

Mecânico

Rastreadores mecânicos calculam a projeção através de ligações físicas entre o observador e uma base. Por exemplo, o primeiro dispositivo de Realidade Aumentada criado por Sutherland (1968) utilizava um braço mecânico ligado à cabeça do usuário, de tal forma que os movimentos e a posição atual do usuário eram calculados capturando o movimento do braço mecânico (Figura 2.1). Rastreadores mecânicos são bastante eficientes, principalmente por não sofrerem com nenhum tipo de ruído externo, porém, limitam o movimento do usuário ao alcance do dispositivo físico (Azuma, 1997), necessitam de hardware específico e não são facilmente movidos para lugares diferentes.

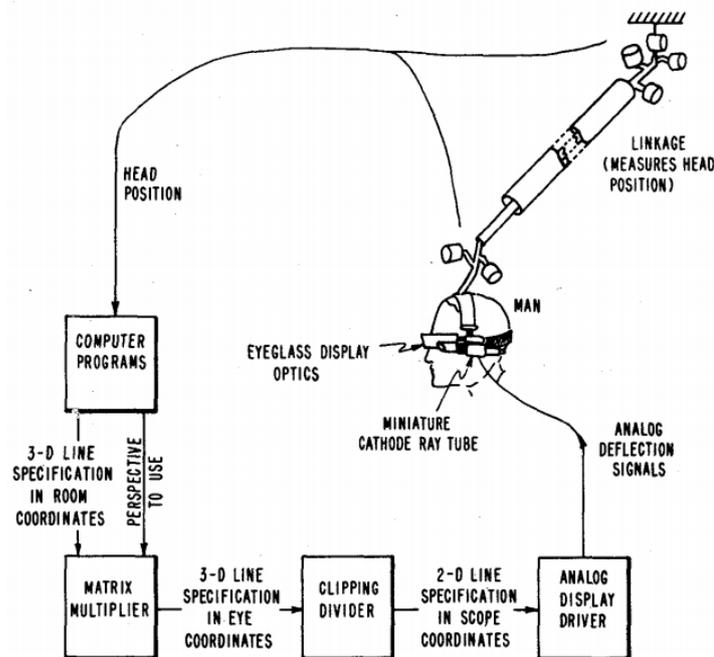


Figura 2.1: Esquema de funcionamento do dispositivo mecânico (Fonte: Sutherland (1968)).

Ultrassom

O sistema de rastreamento em ultrassom é baseado no cálculo da distância de um transmissor de pulsos ultrassônico para seus receptores. O atraso entre a emissão e o recebimento do pulso do transmissor para cada um dos receptores indica a distância entre eles. Como a comunicação entre um emissor e um receptor indica apenas a distância entre eles, e não a localização exata do receptor, é necessário que o sistema conte com 4 ou mais receptores para que, usando o processo de triangulação, seja possível computar a localização aproximada do emissor em \mathbb{R}^3 .

Um trabalho recente na área (Lukosiunas et al, 2011) concluiu que o sistema ultrassônico é extremamente preciso. Este trabalho indicou como qualidades em relação à RA baseada em rastreamento visual, menor esforço computacional e maior precisão.

A principal desvantagem deste método, além da necessidade de hardwares adicionais, é de que a precisão é inversamente proporcional à distância entre os receptores e o emissor, e o sistema é suscetível a ruídos externos como vento e obstruções físicas. Para aumentar a precisão, é comum usar mais emissores além dos 4 necessários. Lukosiunas et al (2011) revelou, experimentalmente, um limite máximo de 13 metros de distância entre o emissor e o receptor, não aconselhando o método para ambientes externos.

GPS

GPS (*Global Positioning System*) é um sistema de navegação por satélite desenvolvido e mantido pelo governo dos Estados Unidos da América cuja finalidade é o posicionamento geo-espacial em qualquer parte do planeta Terra ou próximo dele usando apenas um receptor.

O projeto básico se apoia num conjunto de 31 satélites equipados com relógios atômicos sincronizados orbitando em torno da Terra de tal forma que qualquer ponto do planeta tenha a cobertura de pelo menos 3 satélites a qualquer hora do dia. O receptor consegue, calculando o atraso entre a emissão de sinal de pelo menos 3 satélites (e usando como quarta distância a altura em relação à superfície da Terra), calcular a distância entre estes e a si mesmo, criando triangulação que localiza o receptor.

O sistema possui um controle avançado para correção de erros e é bem estável

contra influências externas como mau tempo e interferências. Em 2005, após a assinatura do Bill Clinton, ex-presidente americano, o sistema deixou de contar com o ”*Selective Availability*”, um sinal artificial que era produzido para diminuir a precisão dos equipamentos, e restringindo a maior precisão apenas para uso militar. O sistema agora conta uma precisão de aproximadamente 20 metros (Kowoma.de, 2009), sendo melhor utilizado na Realidade Aumentada em ambientes externos (Park e Park, 2010).

Ótico

Rastreamento ótico é talvez a forma de Realidade Aumentada mais utilizada hoje em dia. Seu baixo custo é a principal característica positiva desta técnica. Rastreamento ótico passivo usa apenas uma câmera convencional para procurar por padrões conhecidos na cena e, através de técnicas de Visão Computacional, conseguir estimar a posição e a pose do objeto aumentado. Normalmente, os padrões utilizados são marcadores fiduciais (Figura 2.2), porém, vários trabalhos na área (Simon e Berger, 2002) (Uematsu e Saito, 2005) têm avançado no uso de características naturais da cena para a tarefa.



Figura 2.2: Exemplo de marcador usado pelo Artoolkit (Fonte:Kato (2002)).

Apesar de bastante vantajoso e acessível no sentido financeiro, esta técnica oferece várias desvantagens sobre as demais.

A primeira delas diz respeito à oclusão dos marcadores, ou seja, quando um

marcador deixa de ser visível na cena, mesmo de forma parcial, seja porque saiu do enquadramento da câmera ou porque outro objeto está em sua frente. Nestas situações, o rastreamento pode deixar de ser preciso ou mesmo sequer captar a presença do marcador no local, resultando num efeito ruim que não condiz com as expectativas (Figura 2.3).

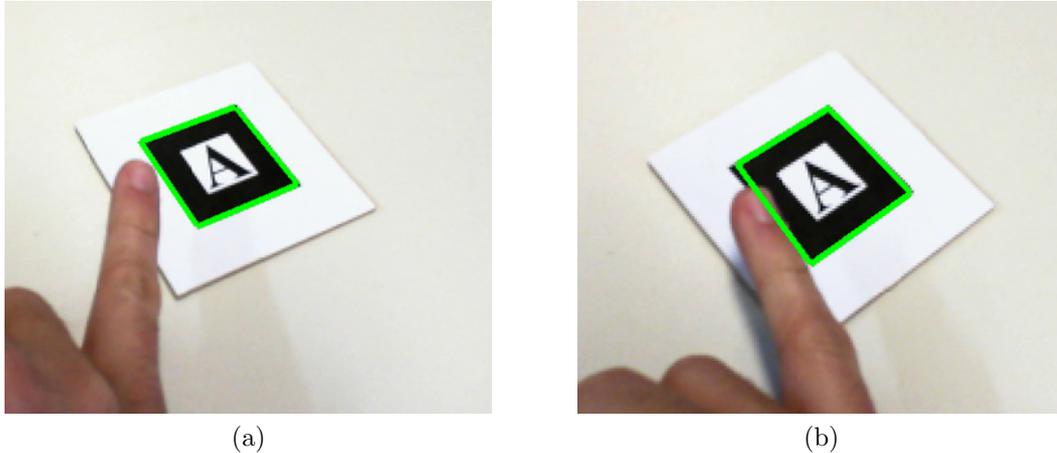


Figura 2.3: (a) Marcador visível com detecção correta dos vértices. (b) Efeitos da oclusão parcial na detecção dos marcadores.

Outro problema muito comum está associado à dependência de condições de iluminação favoráveis, geralmente bem iluminado e com poucas sombras. Um ambiente pouco iluminado ou com muitas sombras sobre os marcadores pode fazer com que o processo de segmentação que será apresentado mais à frente não funcione de maneira correta, se tornando assim uma tarefa complicada extrair a posição e orientação correta do marcador no espaço (Figura 2.4). Embora existam trabalhos como o de Wagner e Schmalstieg (2007), que tentam minimizar erros desse tipo, através de um processo que define automaticamente o melhor *threshold* para a segmentação da imagem, o problema de iluminação ainda não está completamente resolvido. Associado a este problema está o fato de que o reflexo de luzes sobre os marcadores também pode atrapalhar o resultado final, sendo então recomendado que se use material fosco para servir como material de impressão.

Além disso, existe um problema que diz respeito à confusão que alguns elementos na cena que não são marcadores, mas se parecem com marcadores, podem trazer no sistema (Figura 2.5). Esse erro não costuma aparecer nos trabalhos mais comuns onde os marcadores da cena estão quase sempre visíveis, mas é uma situação a ser tratada em

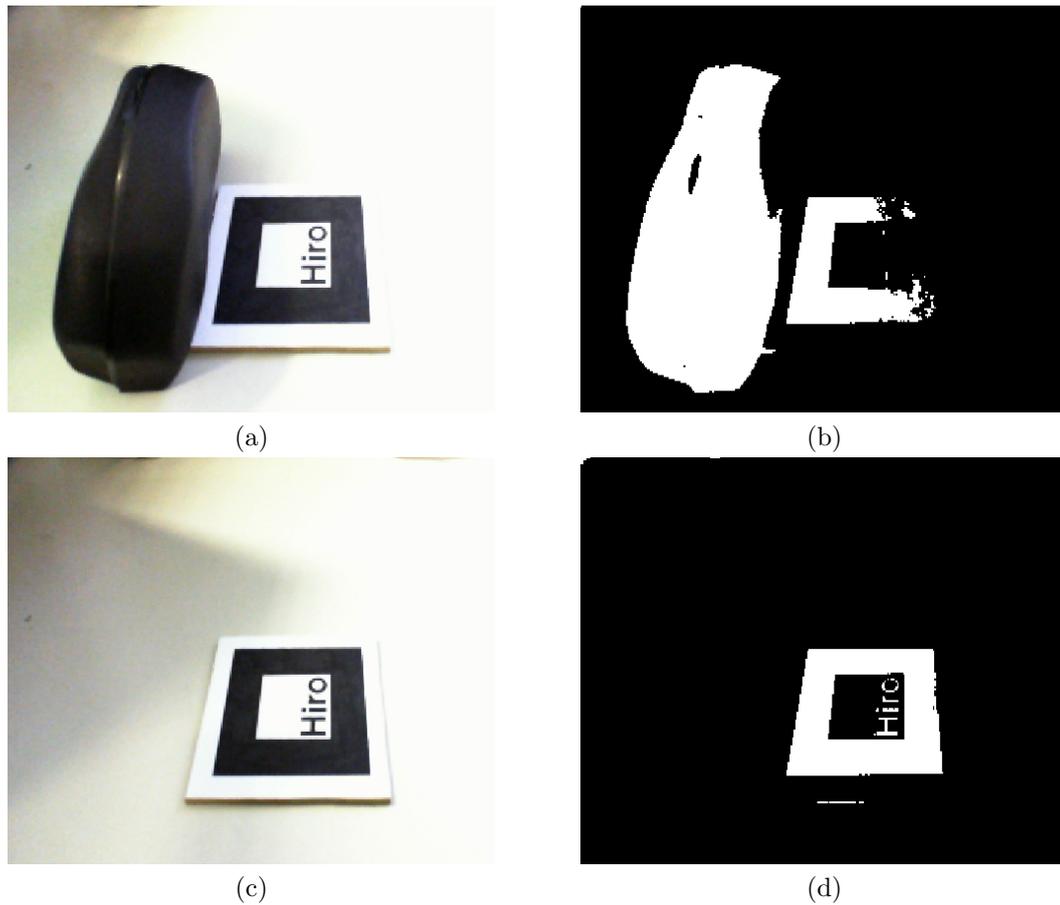


Figura 2.4: (a) Cena em que um marcador é parcialmente coberto pela sombra de um objeto. (b) A segmentação da cena anterior. (c) A mesma cena, sem o objeto criando a sombra. (d) Segmentação do marcador sem sombra, usando o mesmo *threshold* do caso anterior.

trabalhos como o aqui proposto, onde a oclusão dos marcadores é quase certa.

Comparando a técnica de rastreamento ótico com marcadores fiduciais com os demais, é possível perceber também que os óticos levam séria desvantagem no que diz respeito à área de efeito. Supondo, por exemplo, que um arquiteto deseja projetar um prédio de tamanho real em uma área de construção para demonstrar aos seus clientes como ficará o resultado final, usando marcadores fiduciais. Para que os clientes pudessem ver o resultado de forma agradável, o marcador deveria ocupar uma área relativamente grande, visto que quanto maior a distância o observador fica do marcador, maior este precisa ser (Tabela 2.1). Casos como este podem ser contornados usando outras técnicas como as descritas anteriormente ou usando técnicas de rastreamento ótico alternativas como extração de características naturais da cena.

A última das desvantagens mais convencionais citadas aqui está no fato de que o



Figura 2.5: Exemplo em que o Artoolkit confunde a tela de um celular com um marcador invisível na cena. Com o marcador invisível, a tela do celular era o que mais se aproximava do objetivo.

Tabela 2.1: Distância máxima para diferentes tamanhos de marcadores (Fonte:Kato (2002)).

Tamanho Marcador (cm)	Distância Utilizável (cm)
6.99	40.64
8.89	63.5
10.80	86.36
18.72	127.00

processamento necessário para esta técnica pode ser relativamente maior que o das outras (Lukosiunas et al, 2011). Hoje em dia isso não chega a ser uma desvantagem impactante, uma vez que o avanço da tecnologia de hardware fornece velocidade suficiente para tarefas de RA simples, porém, um hardware mais avançado pode ser necessário para aplicações mais robustas.

Mistos

O uso de um dispositivo de rastreamento não exclui a possibilidade de que haja outro atuando ao mesmo tempo no mesmo sistema. De fato, vários trabalhos (Auer e Pinz, 1999) usam métodos mistos de rastreamento na Realidade Aumentada. O uso de tal técnica pode depender um maior gasto, porém a qualidade final é significativamente melhorada,

visto que as desvantagens de um método podem ser acobertadas pelas vantagens de outro e vice-versa.

2.1.2 Aplicações

A Realidade Aumentada possui aplicações em potencial para várias áreas. Além destes, existem várias outras áreas e vários outros exemplos, de modo que listá-los em sua totalidade seria dispendioso e não é o foco deste trabalho.

Educação

Na educação, como proposto por Kaufmann e Schmalstieg (2003), a Realidade Aumentada pode ser usada para auxiliar os jovens na compreensão e visualização de formas geométricas na terceira dimensão, visto que o uso do quadro em duas dimensões, como é feito hoje, nem sempre colabora para o entendimento do aluno do que realmente está acontecendo em uma dimensão superior. Observando a Figura 2.6 é possível perceber que ao manejar os objetos tridimensionais e explorá-los de vários ângulos, o aluno tem a possibilidade de compreender a geometria de maneira mais eficiente, especialmente em casos complexos como interseção de sólidos ou sólidos com muitas arestas.

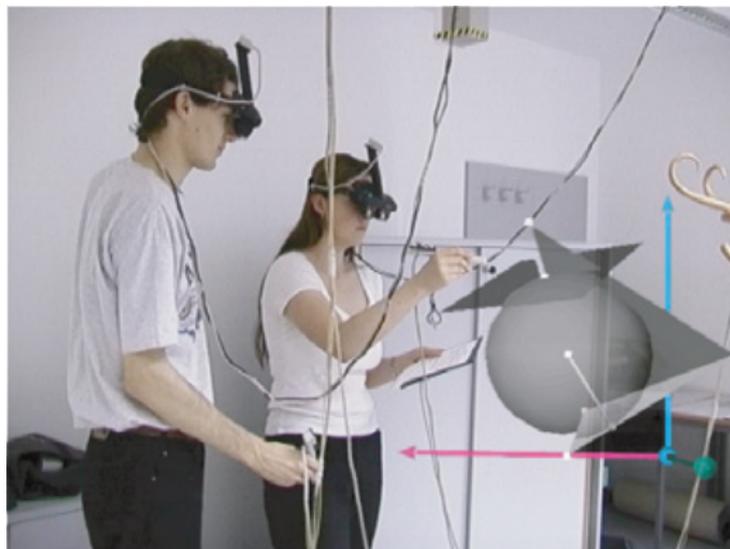


Figura 2.6: Realidade Aumentada aplicada à educação (Fonte:Kaufmann e Schmalstieg (2003)).

Entretenimento

A Realidade Aumentada tem uma gama extensa de aplicações na área de jogos e entretenimento. Apenas para exemplificar, Oda e Feiner (2010) propuseram em seu artigo dois jogos distintos, um de apenas um jogador em que este precisa guiar uma bola virtual através de um labirinto com obstáculos e outro em que mais de um jogador pode jogar, onde o objetivo é atirar bolas e derrubar os dominós do adversário. Jogos como este têm potencial de interatividade muito maior do que os jogos comuns baseados em monitor e *joysticks*.

Medicina

Como mostrado em Nicolau et al (2011), a Realidade Aumentada pode ser útil também no ramo da medicina, ajudando principalmente a tornar procedimentos cirúrgicos menos invasivos o possível. Nesses casos, o cirurgião não consegue ter uma visão perfeita do que está acontecendo e, por isso, a Realidade Aumentada pode ser usada como guia durante o procedimento. É conveniente ressaltar que a qualidade final do sistema de Realidade Aumentada envolvido deve exceder ao esperado das demais aplicações, uma vez que os riscos envolvidos são maiores e as consequências de um erro provocado por um sistema de RA mal calibrado podem ser fatais.

Psicologia

Juan et al (2005) propõe o uso da Realidade Aumentada para tratamento de fobias através da dessensibilização sistemática, técnica utilizada na terapia cognitiva-comportamental que consiste na exposição do paciente à vivência traumática (Figura 2.7). O usuário se sente mais confiante ao lidar com objetos virtuais e vai aos poucos se acostumando com a experiência, chegando a ponto de conseguir lidar com o objeto real pouco tempo depois. O autor revela ainda que os resultados excederam as expectativas, se tornando uma alternativa sólida e segura para este tipo de tratamento.

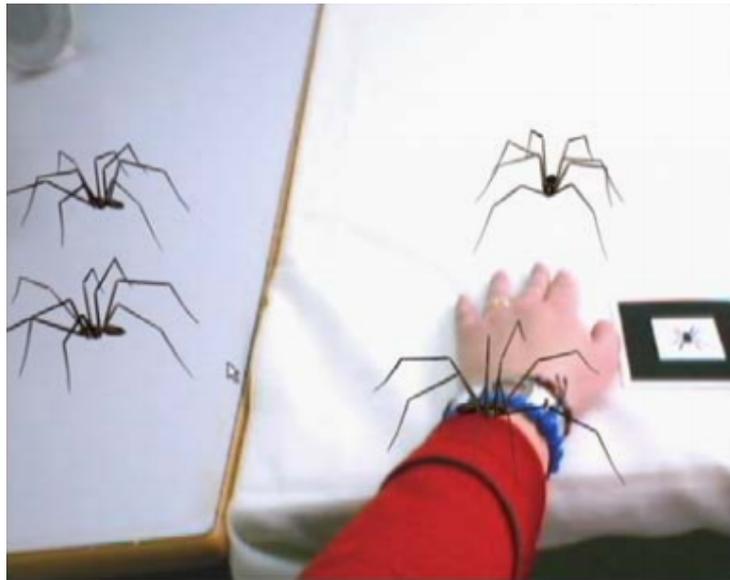


Figura 2.7: Paciente em tratamento contra fobia deixa aranhas virtuais se aproximarem de sua mão (Fonte: Juan et al (2005)).

Montagem e manutenção

Dangelmaier et al (2005) sugere em seu artigo o uso da Realidade Aumentada como auxílio à montagem e manutenção de equipamentos mecânicos e eletrônicos, usando como argumento que a velocidade e complexidade com que estes produtos estão sendo produzidos e miniaturizados torna mais difícil a compreensão de seu funcionamento e reparo, mesmo para as pessoas mais preparadas (Figura 2.8).



Figura 2.8: RA aplicada à mecânica (Fonte: Dangelmaier et al (2005)).

2.1.3 Ambientes com Múltiplos Marcadores

Ambientes com múltiplos marcadores podem ser usados, tanto com a intenção de se projetar um objeto em cada marcador em separado, ou para combinar vários marcadores em uma única projeção, sendo este último caso o foco do estudo apresentado.

Múltiplos marcadores para redundância de informação são apreciados em sistemas de Realidade Aumentada que operam em ambientes onde se tenha muita oclusão ou quando a área a ser utilizada é grande demais para o uso de um único marcador. Considere como exemplo prático do uso de redundância de marcadores a situação apresentada em Kalkusch et al (2002). Os autores construíram um sistema de localização, usando Realidade Aumentada dentro de um edifício. A intenção era que o usuário, usando óculos especiais para Realidade Aumentada, conseguisse saber em qual posição exata está dentro do prédio e como chegar em seu destino. Para tal, os autores interligaram vários marcadores nos corredores a um único sistema guia e calcularam as relações entre eles usando a planta do prédio.

Situação semelhante é encontrada em Baratoff e Neubeck e Regenbrecht (2002), onde um engenheiro necessita usar a RA para visualizar da Fluidodinâmica Computacional (CFD) em bancos de passageiros do *Airbus*. Usar um único marcador para a situação limita a área de abrangência possível, levando então os autores a usar múltiplos marcadores para a tarefa.

2.2 ARToolkit

Como dito anteriormente, ARToolkit (Kato, 2002) é uma biblioteca livre de Realidade Aumentada e de código fonte aberto, voltado para a utilização da técnica de rastreamento ótico usando marcadores fiduciais planos para realizar a estimativa de pose e posição. O Artoolkit usa as bibliotecas *DSVideoLib*, para captura do vídeo e *OpenGL / GLUT* para renderização.

2.2.1 Funcionamento Básico

Seu funcionamento básico é baseado nos seguintes passos, sendo que os passos 2 e 3 podem ser subdivididos nas etapas ilustradas pela Figura 2.9:

1. A câmera captura o vídeo do mundo real e envia para o computador;
2. Software procura por padrões com formas quadriláteras em cada quadro do vídeo;
3. Se um quadrilátero é encontrado, o software usa cálculos matemáticos para inferir a posição da câmera em relação ao mesmo;
4. Uma vez que a posição da câmera é conhecida, um modelo gráfico é desenhado sobre o marcador;
5. O objeto projetado é sobreposto ao vídeo para que pareça estar sobre o marcador;
6. O resultado final é mostrado de volta no dispositivo de saída.

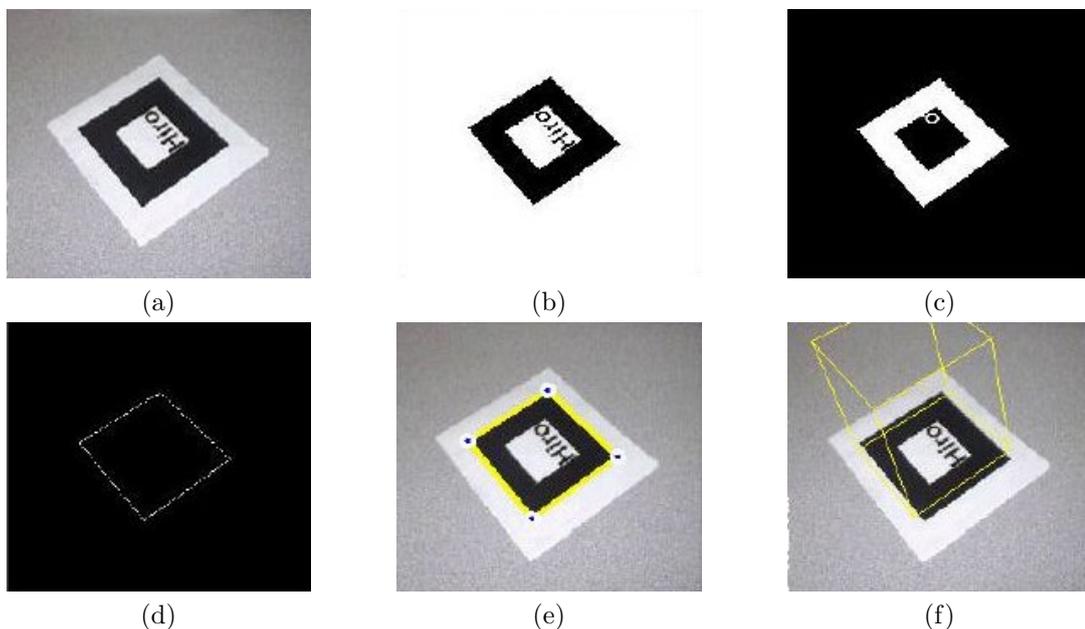


Figura 2.9: Principais etapas de Visão Computacional do Artoolkit. (a) A imagem original. (b) Segmentação e Binarização da imagem. (c) Extração dos componentes conectados. (d) Extração dos contornos. (e) Extração de quinas. (f) Imagem virtual sobreposta. (Fonte:Kato (2002)).

2.2.2 Código fonte

O código fonte das aplicações que utilizam o ARToolkit pode ser dividido em 6 principais etapas (Figura 2.10) :

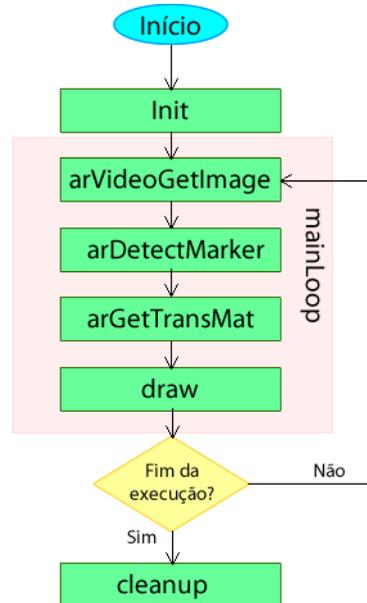


Figura 2.10: Fluxograma básico das aplicações que utilizam o Artoolkit.

init

Função que inicializa a captura do vídeo (*arParamLoad* e *arVideoOpen*), lê os parâmetros iniciais da aplicação e os arquivos de configuração. É nesta etapa (*read_ObjData*) que o Artoolkit lê o arquivo que normalmente configura quais marcadores serão usados na aplicação, quais são suas dimensões e, caso o usuário deseje usar múltiplos marcadores para redundância, qual é a relação entre eles como exemplificado pela Figura 2.11.

```

#the number of patterns to be recognized
6
#marker 1
Data/multi/patt.a
40.0
0.0 0.0
1.0000 0.0000 0.0000 -100.0000
0.0000 1.0000 0.0000 50.0000
0.0000 0.0000 1.0000 0.0000
...
  
```

Arquivo que define o marcador

Tamanho do marcador e origem das coordenadas

Transformação em relação a um ponto global

Figura 2.11: Exemplo de arquivo de Configuração para múltiplos marcadores (Fonte:Kato (2002)).

arVideoGetImage

arVideoGetImage é a primeira função dentro do *mainLoop* (laço principal do Artoolkit). Sua função principal é capturar um quadro do vídeo, usando as configurações intrínsecas de calibragem da câmera (cisalhamento, distorção radial, centro da imagem, etc). Neste momento, a imagem capturada já é exibida na tela.

arDetectMarker

Esta função procura por formas quadriláteras que contém os padrões corretos dos marcadores na imagem do vídeo, usando um valor de *threshold* para segmentação previamente configurado e retornando a quantidade de marcadores encontrados. Em seguida, compara os fatores de confiança de cada forma encontrada, a fim de casá-las com os padrões dos marcadores previamente carregados.

arGetTransMat

Esta é, talvez, a função mais importante do Artoolkit. Ela usa as informações sobre os padrões detectados anteriormente e calcula a transformação da câmera em relação a cada um deles. Além disso, a função retorna também um valor real representando a acurácia daquela projeção. Mais tarde será mostrado como esse valor será importante para a implementação.

A função pode ainda existir na forma de *ArGetTransMatCont*, que é semelhante à anterior, porém muito mais eficiente, pois leva em conta nos cálculos, não apenas a extração de características de imagem corrente, mas também os dados obtidos no quadro anterior. Quando um marcador é visto pela primeira vez ou reaparece na cena, a função usada deve ser, obrigatoriamente, a *arGetTransMat*.

A função *arGetTransMat* gera, para cada objeto virtual que se deseja projetar, uma matriz que transforma o espaço da câmera em espaço do marcador, ou seja, nos dá a posição do marcador no sistema de coordenadas da câmera. A matriz de transformação do Artoolkit é semelhante à do OpenGL e é representado por uma matriz 3×4 . Sendo esta composta por uma primeira matriz 3×3 que representa as rotações, e outra matriz

3×1 que representa as translações.

$$\begin{bmatrix} X1 & Y1 & Z1 & DX \\ X2 & Y2 & Z2 & DY \\ X3 & Y3 & Z3 & DZ \end{bmatrix}$$

Para interpretar a matriz de transformação gerada, é necessário entender o sistema de coordenadas usado pelo Artoolkit, exemplificado na figura 2.12. Nele é possível perceber que o eixo Z do sistema de coordenadas do marcador é virado para face de cima do mesmo e em direção distinta ao eixo Z da câmera.

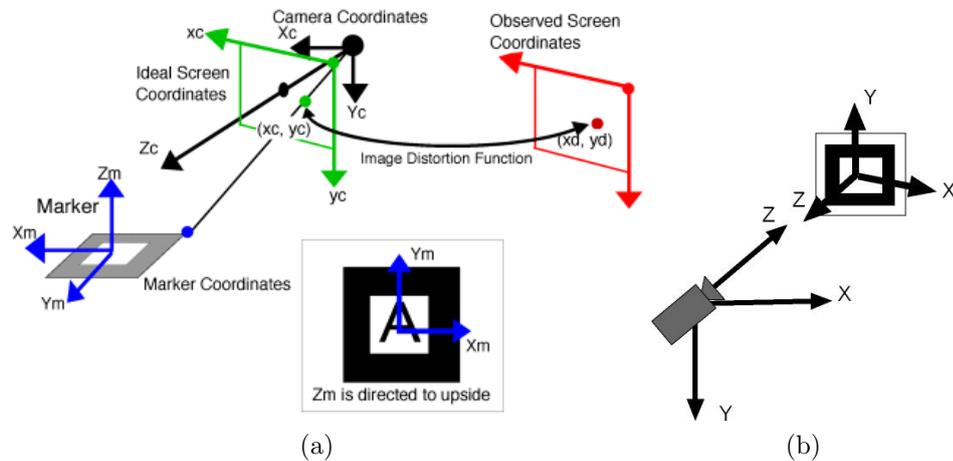


Figura 2.12: Sistema de Coordenadas do Artoolkit (Fonte:Kato (2002)).

draw

Sobrepõe a imagem do quadro atual com os objetos virtuais gerados, usando as matrizes de transformação geradas na etapa anterior.

O Artoolkit é comumente usado para abstrair dos softwares de Realidade Aumentada as etapas de detecção e alinhamento, ou seja, a maioria dos softwares tem um foco maior em o quê e como algo é projetado na cena e não no alinhamento e posicionamento deste. Desta forma, o foco da maioria das aplicações está na função *draw*, onde a pose e posição já foram estimados pela biblioteca.

cleanup

Fecha a captura de vídeo e limpa as configurações de contexto do OpenGL e GLUT.

2.2.3 Estrutura de Dados

Quando está lidando com ambientes de múltiplos marcadores, o Artoolkit trabalha com a seguinte estrutura para representar cada marcador.

```
1 typedef struct {
2     char      name [256];
3     int       id;
4     int       visible;
5     int       collide;
6     double    trans [3] [4];
7     double    marker_width;
8     double    marker_center [2];
9     double    acuracia;
10 } ObjectData_T;
```

- O primeiro valor indica um nome atribuído ao marcador pelo usuário. Não é fundamental para o cálculo final e serve apenas caso o usuário deseje identificar cada marcador por um nome.
- O segundo valor indica um número de identificação único gerado pelo ARtoolkit para cada marcador.
- O campo *visible* indica se o marcador se encontra visível naquele quadro.
- O campo *collide* é um *flag* para identificar se o marcador se encontra suficientemente próximo a outro.
- O campo *trans* representa a transformação do marcador em relação à câmera.
- O campo *marker_width* representa a largura do marcador.
- O campo *marker_center* indica quais as coordenadas X e Y da origem do marcador.
- O campo *acuracia* foi introduzido neste trabalho, para que fosse possível ter acesso à qualidade do rastreamento em etapas diferentes do código. O objetivo é fazer com

que este guarde o valor de acurácia retornado pela função *arGetTransMat* como exposto na subseção 2.2.2.

2.3 Álgebra de quatérnios

Quatérnios são encarados como uma generalização no espaço tridimensional do que os números complexos representam no espaço bidimensional. Sua estrutura algébrica foi proposta por William Rowan Hamilton em 1843 e Shoemake (1985) propôs seu uso na Computação Gráfica. Os quatérnios representam uma forma algébrica simples e elegante para representar rotações no espaço. As definições a seguir foram inspiradas no trabalho de Galo e Tozzi (1997).

Um quatérnio q é representado pela forma

$$q = a + bi + cj + dk$$

Onde o escalar a representa a parte real e o vetor $\vec{v} = (b, c, d)$ de três componentes representa a parte imaginária. É possível também representá-los pela forma compacta

$$q = (a, \vec{v})$$

Os componentes i , j e k possuem propriedades semelhantes à do i dos complexos, ou seja

$$i^2 = j^2 = k^2 = -1$$

Porém, como se está lidando com multiplicação de imaginários de natureza diferente, é necessário definir

$$ij = k$$

$$ji = -k$$

A operação de soma entre quatérnios é dada como uma soma de vetores de quatro dimensões.

$$q_1 + q_2 = (a_1, \vec{v}_1) + (a_2, \vec{v}_2) = (a_1 + a_2, \vec{v}_1 + \vec{v}_2)$$

A operação de multiplicação de um quatérnio por um escalar também é dado como o esperado para um vetor de quatro dimensões.

$$nq = n(a, \vec{v}) = (na, n\vec{v})$$

O produto entre dois quatérnios, talvez a parte mais relevante em sua aplicação em rotações, é dado por:

$$q_1 q_2 = (a_1 + b_1 \mathbf{i} + c_1 \mathbf{j} + d_1 \mathbf{k})(a_2 + b_2 \mathbf{i} + c_2 \mathbf{j} + d_2 \mathbf{k})$$

$$q_1 q_2 = a_1 a_2 + a_1 b_2 \mathbf{i} + a_1 c_2 \mathbf{j} + a_1 d_2 \mathbf{k}$$

$$+ b_1 a_2 + b_1 b_2 \mathbf{i} + b_1 c_2 \mathbf{j} + b_1 d_2 \mathbf{k}$$

$$+ c_1 a_2 + c_1 b_2 \mathbf{i} + c_1 c_2 \mathbf{j} + c_1 d_2 \mathbf{k}$$

$$+ d_1 a_2 + d_1 b_2 \mathbf{i} + d_1 c_2 \mathbf{j} + d_1 d_2 \mathbf{k}$$

Simplificando os termos:

$$q_1 q_2 = (a_1, \vec{v}_1)(a_2, \vec{v}_2) = (a_1 a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2)$$

O conjugado de um quatérnio q é representado por \bar{q} e é dado por

$$\bar{q} = (a, -\vec{v})$$

Para realizar rotações sobre um ponto $p(x, y, z)$ usando quatérnios, segue-se os seguintes passos:

- Representar p como o quatérnio $q_p = (0, \vec{p})$, com a parte real nula e a parte imaginária correspondendo ao vetor que é dado pelas coordenadas do ponto;
- Definir o eixo sobre o qual a rotação será realizada, representado pelo vetor unitário \vec{n} e qual será o ângulo de rotação;

- Representar a rotação no quatérnio $q_r = (\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\vec{n})$;
- Realizar a operação $R(q_r, q_p) = q_r q_p \overline{q_r}$;
- O resultado será um quatérnio com a parte real nula e a parte imaginária dará o ponto rotacionado.

A composição de rotações é dada de forma natural pela álgebra dos quatérnios. Supondo que se tenha dois quatérnios q_1 e q_2 unitários, representando duas rotações distintas é possível se obter um único quatérnio q_3 que represente o resultado da aplicação das duas rotações de uma forma unificada, multiplicando q_1 por q_2 .

A interpolação de orientações também pode ser facilmente determinada usando-se a representação dos quatérnios. Dado q_1 e q_2 representando duas rotações distintas e supondo que é desejado obter q_3 de forma que este represente uma transição suave da primeira rotação para a segunda, faz-se:

$$q_3 = q_1 \frac{\sin(1-u)\theta}{\sin\theta} + q_2 \frac{\sin(u\theta)}{\sin\theta}$$

Onde θ pode ser obtido através do arco co-seno do produto escalar entre q_1 e q_2 e com parâmetro $u \in [0, 1]$ representando a posição do quatérnio q_3 no caminho entre q_1 e q_2 . Esta técnica, em específico, será utilizada neste trabalho para trabalhar com a correção de erros de rotação, que será mostrado futuramente.

2.4 Trabalhos Relacionados

Siltanen e Hakkarainen e Honkamaa (2007) propõem um método semelhante ao deste trabalho para utilização de múltiplos marcadores sem a calibração. Na execução, um marcador central serve de referência para todos os outros. O cálculo da relação entre os marcadores se dá de maneira semelhante, porém como o marcador base é sempre o mesmo, o autor propõe o uso de uma estrutura de grafos para interconectar marcadores que não são visíveis na cena ao mesmo tempo. Desta forma, se houver um caminho no grafo que os conecte através de outros marcadores, a relação entre ambos pode ser conhecida, sabendo as relações de cada aresta que conecta os marcadores do caminho. Um lado negativo deste

tipo de abordagem é que em determinadas situações, a quantidade necessária de cálculo para se chegar ao resultado final pode ser proibitiva. Quando um marcador é conectado com a base por vários outros marcadores, a relação entre cada um dos marcadores, ligados par a par, devem ser conhecidas e multiplicadas uma com a outra, até que se chegue no resultado final, somando a isso o custo de se encontrar o menor caminho dentro do grafo. A abordagem proposta aqui permite que o marcador base varie de acordo com o campo de visão da câmera, de forma que o próximo marcador a ser eleito como base, com certeza era visível pela base anterior, fazendo com que descobrir a transformação entre eles seja algo direto. O autor afirma que o resultado final exibido é bastante satisfatório.

Kotake e Uchiyama e Yamamoto (2004) propõem um método de calibragem combinando um conjunto de marcadores planos com a técnica de ajustamento de feixes perspectivos (*bundle adjustment*), porém, apesar do método não exigir que o usuário meça as distâncias e as rotações dos marcadores, ele exige que se tenha algum conhecimento prévio sobre seu arranjo, por exemplo, quando os marcadores estão em um mesmo plano ou quando estão em planos diferentes mas são co-normais. Os resultados apresentados pelos autores são bastante satisfatórios.

Uematsu e Saito (2005) desenvolveram um método que combina os sistema de coordenadas independentes de todos os marcadores em um espaço projetivo único, e a partir desse, consegue calcular as relações entre cada um dos marcadores. Para construir o espaço projetivo, o sistema captura duas imagens aleatórias de posições diferentes do sistema em uma pré-etapa que, segundo os autores, dura cerca de 30 segundos. Ambas as imagens devem ser tiradas com todos os marcadores visíveis para que o espaço possa ser projetado. Isso pode ser um impedimento, já que em alguns casos, (Figura 2.13, por exemplo) nem todos os marcadores são visíveis por duas tomadas de cena ou mesmo por uma única tomada. Além disso, este processo pode ser um inconveniente, pois não é desejável esperar o sistema executar a etapa de pré-execução.

Baratoff e Neubeck e Regenbrecht (2002) propuseram um método semi-automático de calibração, baseado também em imagens previamente tiradas e ajustamento de feixes perspectivos. O método funciona de maneira semelhante ao supracitado, transformando as coordenadas de cada marcador em uma só coordenada global, e então, calculando a

relação entre cada um deles. O autor obteve experimentalmente um tempo de calibração igual a aproximadamente um segundo, onde seria possível avaliar até quarenta fotos e escolher o melhor par para realizar os cálculos. Klopschitz e Schmalstieg (2007) revela um problema no método anterior que já foi citado aqui: ele só é eficiente quando todos os marcadores se encontram razoavelmente próximos, a ponto de ser possível que todos ou quase todos saiam na mesma foto. Por isso, o autor propôs um método semelhante para grandes áreas, onde uma câmera navega pelo espaço capturando os marcadores e características naturais como quinas presentes e guarda as relações em uma base. Wang et al (2010) também trabalhou de forma semelhante a Baratoff e Neubeck e Regenbrecht (2002), porém, ao invés de computar as transformações entre os marcadores até a base de forma "gulosa", utilizou o algoritmo de *Floyd-Warshall*, usando como peso entre as arestas um valor de erro pré-calculado em função do ângulo e distância dos marcadores.

Todos os trabalhos encontrados que propõem um sistema semelhante ao desenvolvido aqui têm em comum o fato de usarem métodos complexos ou etapas de pré-execução. Cálculos mais simples aumentam a performance da aplicação e são mais facilmente executáveis em computadores mais lentos ou em sistemas embarcados como *smartphones* ou *tablets*.

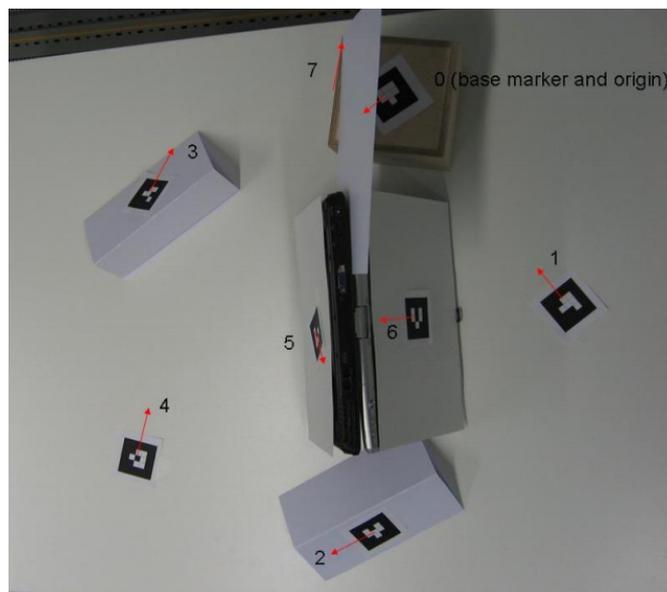


Figura 2.13: Exemplo de configuração onde os marcadores são dispostos de maneira a ficar impossível exibir todos ao mesmo tempo (Fonte: Siltanen e Hakkarainen e Honkamaa (2007)).

3 Modelo Proposto

O modelo proposto se baseia na ideia de que é possível extrair as relações entre dois marcadores distintos em tempo de execução. Tendo as relações previamente armazenadas em algum momento onde ambos marcadores são visíveis ao mesmo tempo, é possível inferir a posição de um marcador invisível na cena, caso o outro continue visível.

3.1 Abordagem utilizada

3.1.1 Obtendo as relações entre os marcadores

Como dito anteriormente, o ARtoolkit tem como resultado final uma matriz de transformação, seguindo os mesmos modelos das matrizes do OpenGL, que transporta o espaço da câmera em espaço do marcador. Dado que dois marcadores A e B estão presentes na cena e são detectados, e considerando que Ma e Mb simbolizam, respectivamente, as matrizes de transformação dos marcadores A e B, é possível calcular uma matriz Mba que diz a relação entre eles usando:

$$Mb \times Mba = Ma$$

$$Mba = Mb^{-1} \times Ma$$

Neste caso, o resultado nos dá qual a transformação necessária para levar B até A. Para calcular a transformação que leva A até B é necessário trocar as variáveis.

$$Mab = Ma^{-1} \times Mb$$

Dado que em um primeiro instante de tempo é possível visualizar os dois marcadores e calcular a relação entre eles e tendo em mente que a relação entre os marcadores teoricamente não varia de acordo com o ponto de vista da câmera, é possível inferir a posição de um marcador que não é mais visível na cena usando a matriz de

posição e pose do marcador visível. Usando ainda o exemplo dos marcadores A e B, suponha que num instante de tempo t pode-se calcular e guardar a matriz Mb_{a_t} , que representa a transformação de A_t no espaço de B_t e que num próximo instante de tempo $t + 1$ seguinte, A_{t+1} deixe de ser conhecido. É possível estimar a posição de A usando:

$$Ma_{t+1} = Mb_{t+1} \times Mb_t^{-1} \times Ma_t$$

ou

$$Ma_{t+1} = Mb_{t+1} \times Mba_t$$

Na prática, quando são calculadas as transformações de um marcador no espaço de outro, é comum que o resultado varie com o tempo e com a posição da câmera. Isso acontece porque o registro, mesmo quando feito com qualidade, é suscetível a erros e, de fato, guardar apenas a primeira transformação calculada não é uma boa estratégia. Por isso, além de guardar a relação absoluta do último quadro em que os dois marcadores foram visíveis, existem abordagens como a proposta por Siltanen e Hakkarainen e Honkamaa (2007) que leva em conta os dados dos quadros anteriores, calculando a média ponderada entre eles usando a acurácia de cada marcador retornada pelo Artoolkit como peso. Esta abordagem, apesar de ser mais eficiente que simplesmente guardar uma nova relação quadro a quadro, ainda está suscetível a falhas quando, por exemplo, um marcador não está visível e é confundido com um objeto ou outro marcador na cena por muito tempo. Apesar do peso na média ser baixo porque a acurácia provavelmente será, o fato da exposição ter sido prolongada afeta diretamente o resultado da transformação. Além do mais, se o objeto confundido com um marcador ocluso estiver em uma posição relativa extremamente diferente do marcador original, por exemplo, a vários metros de distância, mesmo que a acurácia provavelmente seja baixa, o fato da nova e falsa distância ser gigantesca, se comparado com a original, faz com que o baixo peso na média seja compensado e o erro consiga afetar diretamente o cálculo final de relação entre os marcadores. O próprio Siltanen e Hakkarainen e Honkamaa (2007) nos alerta sobre isso e estabelece em sua estratégia um valor máximo de erro para que a relação entre os marcadores naquele quadro seja levado em conta.

A alternativa proposta neste trabalho é a de, ao invés de se usar todos os dados coletados de dois marcadores através de todos os quadros desde o início da execução, usarmos, de forma absoluta, apenas os dados de quando ambos estão com o menor erro possível. Ou seja, supondo que se quer as melhores relações entre o marcador A e o marcador B, deve-se, a cada novo quadro f , se A e B estiverem visíveis, calcular $T = Erro(A, f) * Erro(B, f)$. Se T é menor já registrado, então a relação entre os marcadores é calculada e o novo T é guardado, caso contrário, os dados são simplesmente descartados.

Tal abordagem, além de ser igualmente simples, poupa o sistema de precisar de realizar os cálculos extras de relação entre os marcadores a cada novo quadro pior que algum quadro já registrado. Na abordagem em que se considera o erro apenas como peso da média, mesmo que dados com um peso infinitamente baixo sejam obtidos, todo o cálculo de inversão e multiplicação de matrizes, irrelevante para o resultado final, é feito. Um ponto negativo desta abordagem é que ela exige que os marcadores permaneçam estáticos. Esta restrição evita que o sistema use, durante os cálculos, uma matriz de transformação antiga, de quando os marcadores estavam em configurações diferentes e com erro de detecção menor.

3.2 Sistema multi marcador sem calibração simples

Usando o conceito apresentado anteriormente, é possível construir um sistema simples, com apenas dois marcadores (Figura 3.1). Quando o marcador principal está visível, este projeta o objeto virtual normalmente e, se o outro marcador também estiver visível, a matriz de relação para este é guardada para ser usada futuramente. Quando o marcador principal deixa de ser visto, o sistema multiplica a matriz gerada pelo Artoolkit para o segundo marcador pela matriz que foi guardada previamente, conseguindo assim, manter a projeção sobre o primeiro marcador (Figura 3.2).

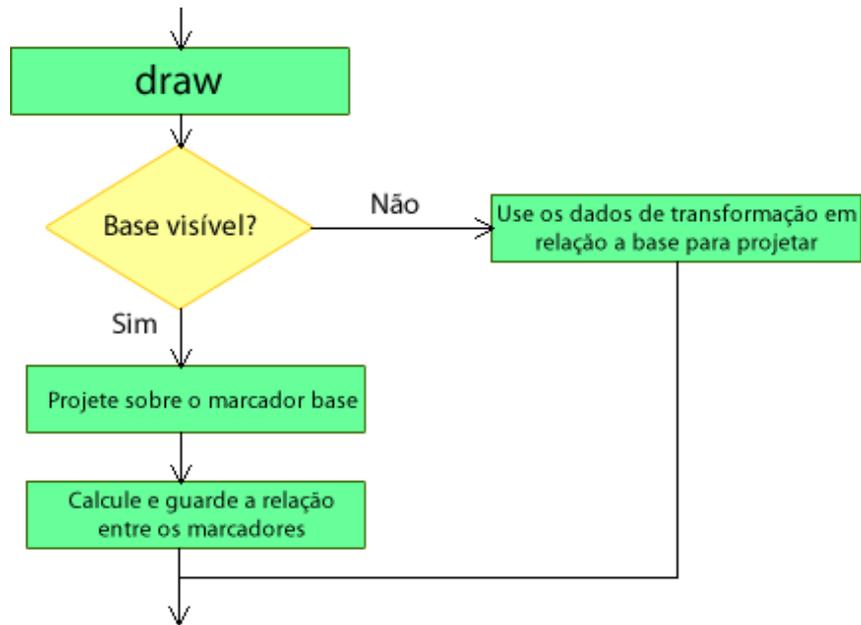


Figura 3.1: Fluxograma geral do multimarcador simples.

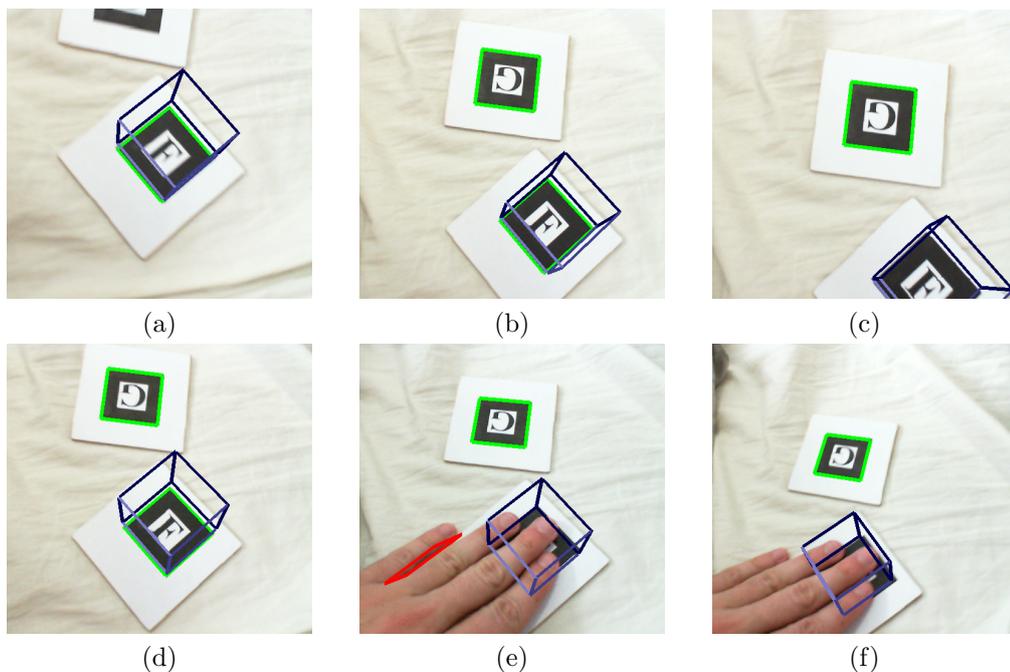


Figura 3.2: Exemplo do sistema simples em execução: (a) O marcador onde é desejado projetar o cubo é detectado. (b) Marcador auxiliar encontrado, como ambos estão visíveis, as relações entre eles são computadas. (c) Com a oclusão do marcador principal, o sistema usa as relações previamente guardadas para tentar manter a projeção no mesmo lugar. (d) Ambos marcadores visíveis de novo, relações recalculadas. (e) Marcador oculto novamente. (f) Projeção se mantém no lugar correto, mesmo de pontos de vista diferentes.

3.3 Sistema multi marcador sem calibração para vários marcadores

Para construir um sistema que aceite vários marcadores ao invés de apenas dois, deve-se antes de tudo definir qual será a política de substituição do marcador base. Neste trabalho,

será chamado de marcador base aquele marcador que vai controlar a cena no momento, ou seja, obrigatoriamente o marcador base deve estar visível e a sua escolha deve ser feita de maneira cuidadosa, para que o resultado final seja o melhor possível. A cada novo quadro do vídeo, a qualidade de cada marcador visível é calculada segundo um critério a fim de definir qual deles controlará a cena. Na seção a seguir serão propostos alguns métodos de escolha do marcador base. A figura 3.3 descreve como é o funcionamento do sistema que, inicialmente, procura avaliar todos os marcadores visíveis na cena, sobre algum aspecto. Após eleger qual dos marcadores é o melhor, um teste simples confere se este é igual ao marcador base do último quadro. Caso seja, as relações entre a base e todos os outros marcadores visíveis são recalculados e o objeto é projetado usando o marcador base. Caso não seja, para evitar que o marcador base alterne repetitivamente entre os quadros quando a qualidade de dois marcadores é muito próxima, definiu-se que o melhor marcador da cena só será definido como nova base quando superar a qualidade do atual em pelo menos 10% ou 5%.

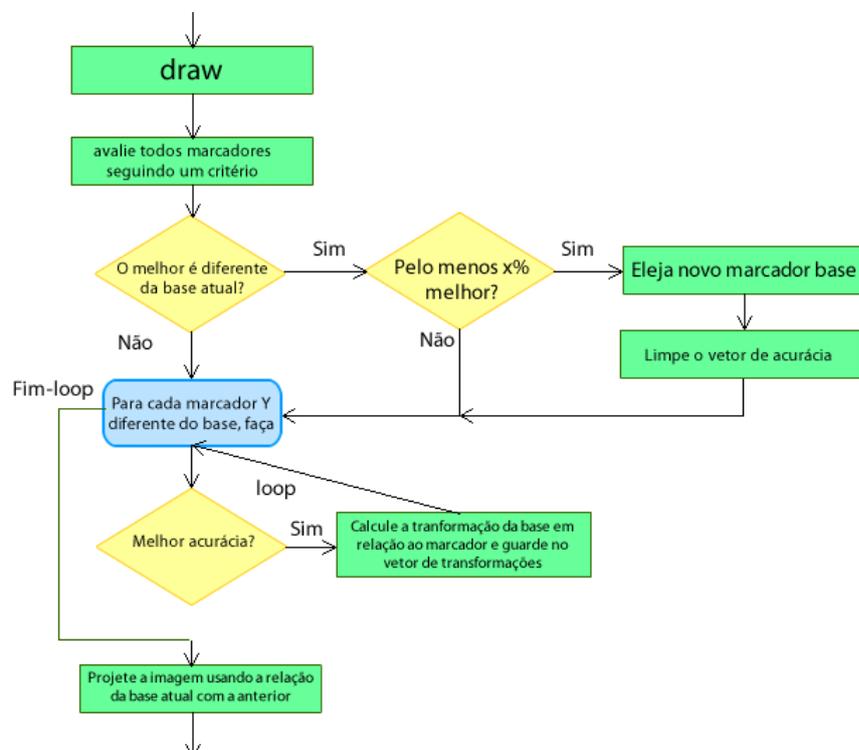


Figura 3.3: Fluxograma geral do multimarcador.

3.4 Métodos para escolha do marcador base

Serão discutidas neste capítulo algumas políticas de substituição de marcadores base. Para todos os efeitos, $ArTrans^n$ será considerado como sendo a transformação dada pelo Artoolkit em um quadro específico e que nos dá a matriz que leva ao centro de coordenadas do marcador n . $Transf^n$ será a matriz calculada a cada quadro e que representa a relação entre o marcador n e a base anterior. C será o conjunto de todos os marcadores utilizados na aplicação.

3.4.1 Por ordem de prioridade

O usuário define previamente qual a ordem de preferência a ser usada. Esta abordagem é interessante em casos específicos, onde o usuário, por uma construção peculiar do sistema, já conhece qual a ordem mais interessante para a seleção da base ou quando os marcadores se encontram com boa qualidade e suficientemente próximos a ponto de não fazer diferença qual marcador será selecionado.

Por exemplo, supondo que a configuração inicial dos marcadores seja em linha reta e que o usuário deseja passar por todos os marcadores de uma ponta da linha à outra. Conhecendo a ordem em que eles estão alinhados, é possível criar uma ordem de prioridade que facilite o trabalho do sistema, reduzindo o esforço computacional necessário para tal.

Desta forma, o cálculo de escolha de marcador base é bastante simplificado, resumindo-se a apenas encontrar o primeiro marcador que está visível na cena. No caso deste trabalho, para se definir qual a prioridade dos marcadores, o usuário deve editar o arquivo do Artoolkit que define quais marcadores estão cadastrados para usar no sistema. Alterando a ordem em que eles aparecem no arquivo, automaticamente a ordem de prioridade é alterada, já que o Artoolkit busca a configuração dos marcadores na mesma ordem em que está no arquivo. No exemplo da figura 3.4, o marcador A tem prioridade sobre o B que tem prioridade sobre o C. Note que, diferentemente do exemplo de arquivo de calibração mostrado na figura 2.11, este não apresenta a matriz de transformação entre cada marcador e um ponto global, já que o intuito deste trabalho é justamente abstrair a necessidade de tal dado.

```

1 #the number of patterns to be recognized
2 3
3 #pattern 1
4 a
5 Data/multi/patt.a
6 40.0
7 0.0 0.0
8 #pattern 2
9 b
10 Data/multi/patt.b
11 40.0
12 0.0 0.0
13 #pattern 3
14 c
15 Data/multi/patt.c
16 40.0
17 0.0 0.0

```

Figura 3.4: Configuração dos marcadores usados.

3.4.2 Marcador com menor inclinação em relação à câmera

Segundo Kato (2002), o rastreamento é afetado pelo ângulo entre o marcador e a câmera. Quanto mais inclinado o marcador está em relação à câmera, menos visível é o centro do marcador e, logo, a estimativa de pose se torna menos confiável.

Desta maneira, é possível calcular, dentre os marcadores visíveis, qual possui o menor ângulo de inclinação e usá-lo como próxima base. Para tal, será usada a matriz de transformação do marcador para calcular a posição de dois pontos, $P1(0,0,0)$ e $P2(0,0,100)$ no espaço da câmera. Estes dois pontos juntos traçam um segmento de reta sobre o eixo Z do marcador e, quando projetados este sobre a tela, pode-se comparar o quão inclinado o marcador está, a partir da distância entre os dois pontos projetados. Quanto mais inclinado, mais a distância se aproximará de 100 e quanto menos inclinado, mais a distância se aproximará de 0. Isto é o suficiente para dizer qual marcador está com menor inclinação sem precisar de medir os ângulos de cada um (Figura 3.5).

$$\{\forall i | i \in C\} \Rightarrow A_i = \sqrt{\sum_{n=1}^2 ((ArTrans^i \times P1)_n - (ArTrans^i \times P2)_n)^2}$$

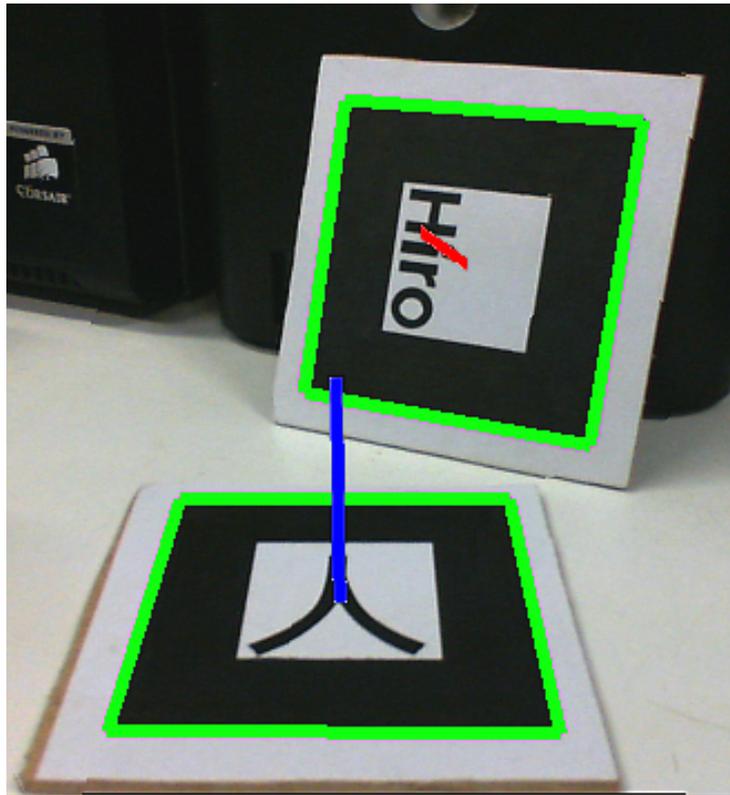


Figura 3.5: Marcador com menor inclinação projeta um segmento de reta menor na tela.

3.4.3 Marcador com maior acurácia

A parte principal do Artoolkit, caracterizado pela função *ArGetTransMat* (Ou *ArGetTransMatCont*, como já explicado), que computa a posição da câmera em função dos marcadores detectados, trabalha tentando, de maneira iterativa, minimizar uma variável de erro.

Quando o máximo de iterações setadas pelo usuário é atingido, se a variável de erro não atingiu um valor aceitável, também configurável pelo usuário, a função retorna que o marcador não pode ser detectado.

Esta função, quando consegue detectar o marcador, retorna um valor que representa a confiança que aquela detecção representa. A confiança é calculada, basicamente usando $Conf \equiv \frac{1}{erro}$ onde a *erro* é a variável que foi minimizada no processo iterativo. Por padrão, as aplicações do Artoolkit não guardam este dado, por isso, é preciso fazer leves mudanças no código para que este seja guardado junto à estrutura de dados dos marcadores.

3.4.4 Marcador mais próximo da câmera

Selecionar como base o marcador mais próximo da câmera pode ser uma boa alternativa para reduzir erros e melhorar a qualidade da projeção, principalmente quando se pretende montar um ambiente onde os marcadores estão bem espalhados e longe um do outro e da câmera.

Na tabela 2.1 é possível notar que quanto maior o marcador, maior é distância em relação à câmera em que ele pode ser utilizado. Esta tabela foi construída posicionando os marcadores de diferentes tamanhos em perpendicular com a câmera e os movendo para longe, até que o Artoolkit não conseguisse mais distingui-lo. Com esta limitação, é desejável que, em ambientes grandes, ao se utilizar um sistema de múltiplos marcadores, não se escolha como base um marcador que está próximo do limiar de reconhecimento, para evitar trocas excessivas de base toda vez que este marcador alterar seu estado de indetectável para detectável.

Para calcular a distância do marcador para a câmera, basta buscar na matriz de transformação a coluna mais à direita, responsável pela translação. Com a matriz em mãos, é possível obter a distância euclidiana fazendo:

$$\{\forall i | i \in C\} \Rightarrow D_i = \sqrt{\sum_{n=1}^3 (ArTrans_{n,4})^2}$$

3.4.5 Marcador mais próximo da projeção

É comum notar nas aplicações de RA que usam marcadores fiduciais que quanto mais longe um objeto projetado fica do seu marcador, pior é a qualidade final do resultado. Isso é bastante óbvio, pensando que as operações de rotação tem um efeito de deslocamento muito maior quando o objeto está mais distante da origem e que é praticamente inevitável que não hajam ruídos na detecção da pose e posição do marcador. Quanto mais a projeção é afastada do marcador, maior será a amplificação do erro, resultando em uma projeção tremida e que não transmite a sensação de realidade desejada.

Selecionar como marcador base o mais próximo da projeção diminui a incidência deste tipo de erro e é aplicável em situações onde há remoção dos marcadores da cena,

principalmente o marcador central, onde o objeto virtual é inicialmente projetado (Figura 3.6, por exemplo).

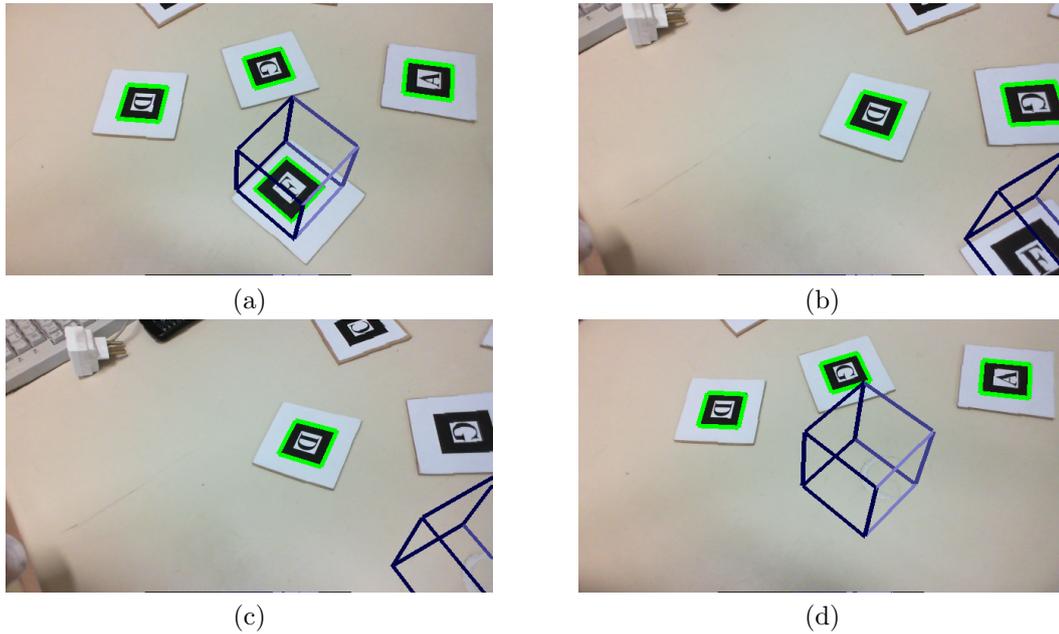


Figura 3.6: (a) Objeto inicialmente projetado sobre o marcador F. (b) Marcador F é ocluído e outro marcador se transforma em marcador base. (c) Marcador F é removido enquanto estava ocluído. (d) Objeto continua sendo projetado onde o marcador F se encontrava.

Para descobrir qual marcador está mais próximo do objeto projetado, inicialmente a matriz de transformação final é calculada como se o marcador base do quadro anterior não tivesse sido alterado, multiplicando a matriz de pose/posição dada pelo ArToolkit naquele quadro em específico pela matriz de transformação que foi calculada para o marcador base atual quando houve a última mudança de base. Tendo isso em mãos, um método simples para cálculo da distância Euclidiana é operado, como proposto no método anterior a este, porém, usando desta vez a matriz de transformação final e a de cada marcador diferente da base.

$$M = ArTrans^{base} \times Transf^{base}$$

$$\{\forall i | i \in C, i \neq base\} \Rightarrow D_i = \sqrt{\sum_{n=1}^3 (ArTrans_{n,4}^i - M_{n,4})^2}$$

3.4.6 Marcador com maior resistência à mudança

Como explicado no subcapítulo 2.1.1, a oclusão parcial dos marcadores pode fazer com que a estimativa do marcador não seja tão precisa quanto desejado. Se um marcador base sofre esse tipo de problema, seria interessante eleger um novo marcador base o mais cedo possível, mesmo que o Artoolkit ainda aponte aquele marcador como visível.

Com base nisso e imaginando uma situação em que, por exemplo, existem pessoas ou qualquer outro tipo de objeto passando ao redor dos marcadores, de tal maneira que a oclusão parcial será com certeza um problema, é possível adotar a política de rejeitar como base o marcador que tiver sofrido mais mudanças do que os outros, comparando-o com a sua estimativa de pose passada. Seja, $OldArTrans^i$ a matriz de transformação do marcador i no quadro anterior, tem-se:

$$\{\forall i | i \in C\} \Rightarrow R_i = \|ArTrans^i - ArOldTrans^i\|_2$$

3.5 Aplicações

O modelo proposto pode ser aplicado em qualquer circunstância onde a redundância de marcadores é benéfica, seja porque estão muito distantes; como é o caso abordado por Kalkusch et al (2002) e já comentado aqui; seja porque os marcadores ficam em posições que são difíceis de mensurar, como é o caso abordado por Baratoff e Neubeck e Regenbrecht (2002) ou mesmo porque os marcadores ficam em posições onde há perigo de medir distâncias manualmente, por exemplo, marcadores que ficam em lados opostos de uma rua.

3.6 Correção de Erro

Como é natural de se esperar, a proposta desenvolvida neste trabalho, por ser baseado em um sistema de rastreamento com características apenas visuais e também por conta dos erros numéricos intrínsecos, está suscetível à cálculos incorretos. Por exemplo, na Figura 3.7, o cubo inicialmente é projetado sobre o marcador **F** e, como é proposto, deveria

continuar sendo projetado mesmo quando ocluso, usando o marcador **C**. Porém, é visível que o cubo não se alinha perfeitamente com o lugar onde supostamente estaria, sobre o marcador original.

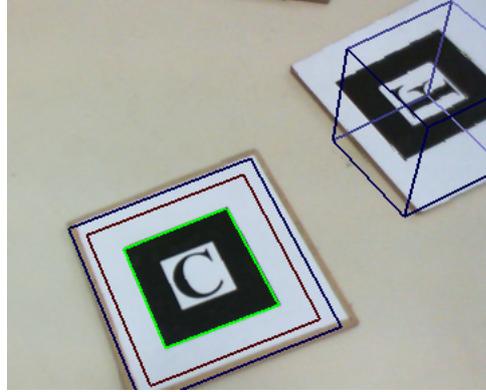


Figura 3.7: Exemplo de execução onde se há erro no resultado final.

A situação se agrava quando o tempo de execução é prolongado, fazendo com que o erro registrado se acumule com o tempo. Em outras palavras, quando o objeto é mal projetado e a base é trocada, a nova base segue projetando o objeto com o erro acumulado pela base anterior fazendo com que, após algum tempo de execução, o erro de torne grande e limitando o tempo de execução com a sensação desejada a poucos segundos.

Diante disso, foi criada uma função que corrige o erro sempre que o marcador principal, onde o objeto é inicialmente projetado, estiver visível. Tal função foi desenvolvida com a intenção de fazer com que o objeto virtual retorne ao seu marcador de origem e deve agir de forma suave, para que este não se mova bruscamente de um lugar no espaço para outro. Por isso e devido à facilidade de representar interpolações de rotação com quatérnios, optou-se por estes à representação matricial na correção dos erros de rotação, corrigindo o erro de translação através de uma interpolação linear simples.

O cálculo da correção usa as funções *arUtilMat2QuatPos* e *arUtilQuatPos2Mat* do Artoolkit que fazem, respectivamente, a transformação da representação matricial para quatérnios e vice-versa. Ainda usando as definições de $ArTrans^n$ e $Transf^n$ dadas na Seção 3.4, obtém-se inicialmente a matriz da projeção calculada de forma indireta fazendo:

$$M = ArTrans^{base} \times Transf^{base}$$

E em seguida, é gerada uma representação intermediária entre M e $ArTrans^P$, tal que P é o ID do marcador principal, onde o objeto é inicialmente projetado. Para garantir que a interpolação se dará pelo caminho mais curto, ao realizar o cálculo do cosseno do ângulo entre os dois quatérnios, deve-se garantir que este é positivo, indicando que o ângulo θ entre os dois quatérnios está entre -90° e 90° . Caso contrário, é necessário negar um dos quatérnios calculando seu conjugado e realizar os cálculos de interpolação novamente.

3.7 Limitações

Uma das limitações do sistema é que para calcular as transformações de um marcador para outro é necessário que exista pelo menos um quadro em que ambos estão visíveis. Nos testes executados isso não chegou a ser uma limitação decisiva já que a aplicação rodou quase sempre em uma taxa de 30 *fps*, dando tempo suficiente para que os marcadores visíveis pudessem aparecer em alguns quadros e tivessem suas relações calculadas. Como características naturais da cena não foram exploradas, é intuitivo imaginar que para funcionar, é necessário ter pelo menos um marcador visível.

O sistema ainda conta com uma limitação que obriga que os marcadores precisem ser previamente conhecidos, junto com a largura de cada um. Criar novos marcadores antes da execução é uma tarefa relativamente simples, usando o conjunto de utilitários que vêm junto com ARtoolkit. Mas, é possível modificar o projeto, de forma a aceitar novos marcadores em tempo de execução, como sugerido por Siltanen e Hakkarainen e Honkamaa (2007).

4 Resultados finais

4.1 Cálculo de Erro

Para cálculo de erro de projeção, foram seguidas as propostas sugeridas por Uematsu e Saito (2005) e Siltanen e Hakkarainen e Honkamaa (2007). Nestas propostas, a qualidade é representada pela média das distâncias, em milímetros, entre os vértices do cubo projetado sobre o marcador usando os dados do próprio marcador e outro cubo projetado sobre ele (Figura 4.1), mas usando os marcadores auxiliares. Neste caso, supondo $n \in C$ o marcador onde é desejado projetar o cubo inicialmente, e V_i as coordenadas do vértice i de um cubo projetado sobre a origem, tem-se:

$$M = ArTrans^{base} \times Transf^{base}$$

$$e = \frac{1}{8} \sum_{i=1}^8 \|Transf^n \times V_i - M \times V_i\|_2$$

Ou seja, a posição em coordenadas de câmera de cada aresta são calculadas, aplicando-as tanto à posição do marcador obtida diretamente quanto indiretamente e são computadas a diferença entre elas. O valor final é obtido realizando a média das distâncias de cada aresta.

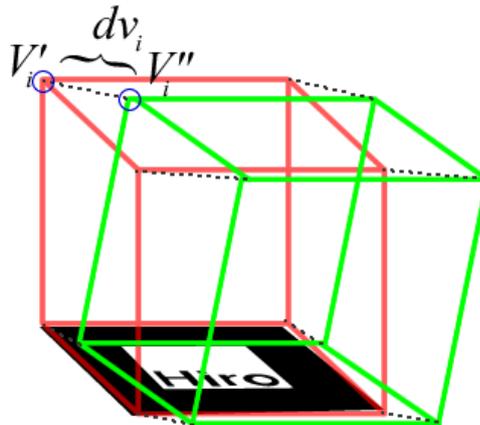


Figura 4.1: Cálculo de erro baseado na distância entre os vértices.

4.2 Comparação entre os métodos

Para comparar os métodos de selecionar os marcadores base, foram montados quatro vídeos de um minuto cada, usando configurações diferentes de marcadores, onde um cubo é projetado sobre um marcador central, sendo que este fica visível inicialmente e ocluído em alguns outros trechos. Ao final do vídeo, o marcador central volta a ser visível, para que seja possível calcular qual foi o erro final. Cada um dos vídeos é executado pelo menos uma vez usando os métodos de seleção de base propostos e o resultado final guardado para comparação. Em cada vídeo, foram usados oito marcadores que já vêm com o Artookit e foram desenvolvidos para rodar exemplos de múltiplos marcadores. Além disso, cada marcador foi impresso em folha de papel ofício e aderido à um pedaço de placa MDF (*Medium Density Fiberboard*) para que a flexibilidade do material onde foi impresso não atrapalhasse a detecção. Os testes foram executados em um Notebook *Dell XPS L401X* com processador *Intel® Core™ i5 M460 2.53GHz*, 4 GB de memória *RAM* em um sistema operacional *Windows 7 Home Premium 64 bits* com placa de vídeo *Nvidia® GeForce® GT 420M* com 1 GB de memória dedicada. Os vídeos foram gravados usando uma *Webcam Microsoft Lifecam Cinema*, na resolução de 640x360 e 30fps.

Vídeo 1

Os marcadores foram dispostos a uma distância relativamente curta entre eles, em um mesmo plano (Figura 4.2).

Vídeo 2

Os marcadores foram dispostos a uma distância relativamente curta entre eles, em um mesmo plano e o marcador central foi coberto no meio do vídeo (Figura 4.3).

Vídeo 3

Os marcadores foram dispostos a uma distância relativamente curta entre eles, alternando a inclinação de cada um (Figura 4.4).

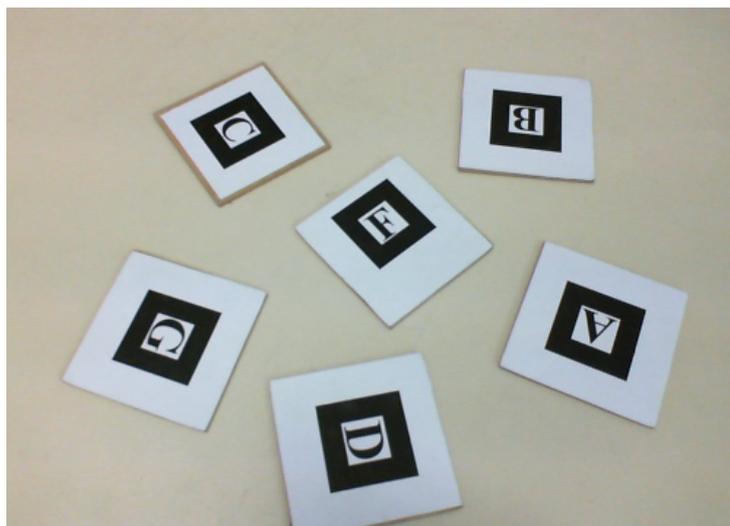


Figura 4.2: Foto da configuração dos marcadores no primeiro vídeo.

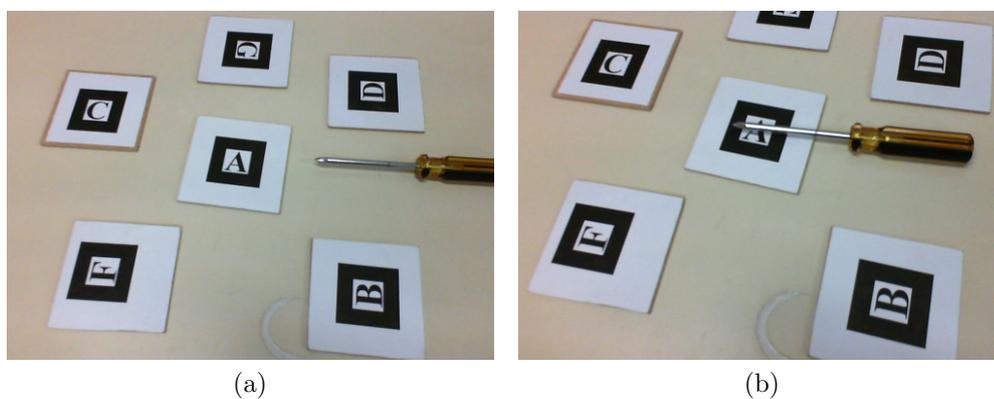


Figura 4.3: Foto da configuração dos marcadores no segundo vídeo.

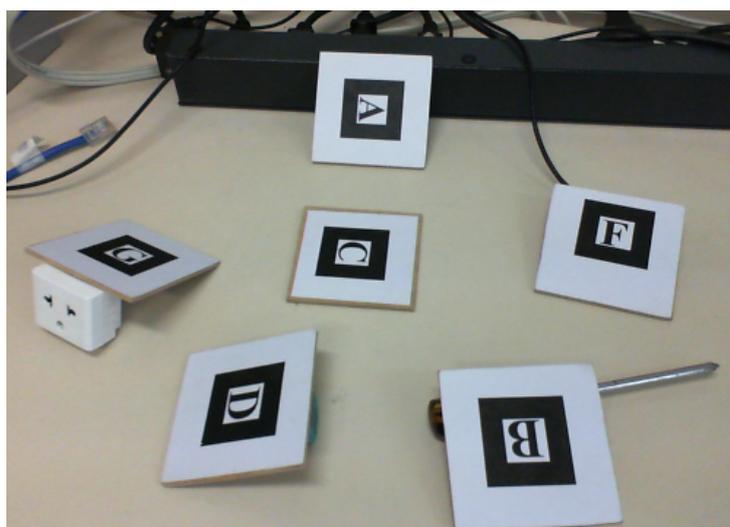


Figura 4.4: Foto da configuração dos marcadores no terceiro vídeo.

Vídeo 4

Os marcadores foram dispostos a uma distância relativamente longa entre eles, alternando a inclinação de cada um (Figura 4.5).

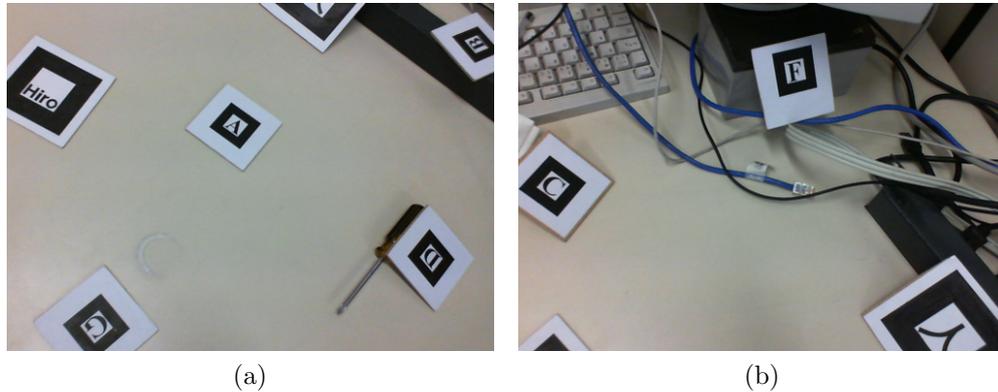


Figura 4.5: Foto da configuração dos marcadores no quarto vídeo.

Na Tabela 4.1 são comparados os resultados finais de cada método, usando o erro de cada um como comparação, sem a correção de erro durante a execução. É possível perceber claramente que a proposta de selecionar o marcador mais próximo da projeção se sobressaiu às demais, tendo sido a técnica com menor erro em dois casos e ficando à altura das demais em outros dois. Alguns métodos tiveram o erro extremamente acentuado, principalmente porque durante a execução exploraram qualidades locais dos marcadores, ao invés de uma abordagem a nível global. Por exemplo, o marcador com menor ângulo foi escolhido como base, mesmo que este esteja muito distante da projeção, o erro natural que se obtém ao se analisar um marcador distante é amplificado, impactando em muito no resultado final.

Tabela 4.1: Tabela de erro, em milímetros, para execução de quatro vídeos diferentes.

	Vídeo 1	Vídeo 2	Vídeo 3	Vídeo 4
Acurácia	157.942474	135.752650	94.689654	2711.850343
Menor inclinação	53.403829	39.068601	258.659012	513.166828
Prioridade	160.298768	23.004197	260.716111	329.239257
Mais Próximo Câmera	165.698957	48.099470	132.429314	958.415439
Próximo da Projeção	170.009281	23.779001	48.015252	276.283990
Resistente à Mudança	253.007655	161.706441	241.503088	679.898225

A Tabela 4.2 mostra o erro máximo obtido durante a execução de cada um dos métodos com os quatro vídeos diferentes, executando o método proposto de correção de

erro. Como o erro ao final da execução está intimamente ligado ao fato do marcador central estar visível e não à qualidade de cada um dos métodos, não faz sentido mostrar o resultado do erro final (já que, por exemplo, se o marcador principal estiver visível nos últimos segundos do vídeo, provavelmente o erro final será 0), e sim qual foi o maior valor que este alcançou. Observando a tabela, pode-se concluir que o método de seleção por marcador mais próximo da projeção também se sobressaiu em relação ao demais.

Tabela 4.2: Tabela de erro máximo, em milímetros, com correção de erro para quatro vídeos diferentes.

	Vídeo 1	Vídeo 2	Vídeo 3	Vídeo 4
Acurácia	68.579869	80.786609	62.958569	772.678263
Menor inclinação	76.986180	58.279673	36.210793	539.686455
Prioridade	39.891589	58.279673	62.958569	257.855705
Mais Próximo Câmera	40.697909	60.326894	44.801940	723.480250
Próximo da Projeção	35.909568	44.907186	44.801940	207.734911
Resistente à Mudança	39.285883	65.288721	44.801940	212.996589

5 Conclusão e Trabalhos futuros

O presente trabalho cumpriu o objetivo a que se propunha, construindo, de fato, um sistema que não necessitasse de calibração no cenário especificado. A maioria dos problemas constatados ficam a cargo, principalmente, dos erros de rastreamento do *Artoolkit*, ficando assim como ideia para trabalho futuro, a tentativa de modificar o modo como o *Artoolkit* faz a leitura dos marcadores a fim de otimizar a abordagem de marcadores múltiplos. Usar outras bibliotecas diferentes como *ArtoolkitPlus* ou *ArtoolkitPro* (comercial) também pode ser uma alternativa para resolver tais problemas.

Nos resultados experimentais, ficou bastante claro que a qualidade final está diretamente ligada à distância entre os marcadores, sendo então, esta abordagem não recomendada quando se pretende criar uma aplicação em RA para ambientes externos, a menos que seja possível o uso de marcadores com tamanho maior do que os marcadores de quatro centímetros utilizados nos testes. Esta dependência da qualidade em relação à distância já era esperada, visto que, como dito anteriormente, quanto maior a distância para o marcador, pior é a qualidade do rastreamento.

Foi observado também que o melhor dos cenários se dá quando o marcador escolhido como alvo da projeção está mais ao centro em relação aos outros e é visível por maior tempo durante a execução. Logo, a proposta aqui elaborada é melhor recomendada em situações onde o marcador principal será visível na maior parte do tempo. Isto não implica, necessariamente, a visibilidade durante todo o tempo, mantendo a proposta deste trabalho.

Como ideias para trabalhos futuros, pode-se tentar sanar algumas limitações, já expostas na seção 3.7, como aceitar marcadores que não foram previamente registrados antes da execução, usar o reconhecimento de características naturais da cena durante o processo de inferência da posição dos marcadores e introduzir a opção de usar múltiplos métodos para seleção, seja cada um com um peso ou como critério de desempate.

Referências Bibliográficas

- Auer, T.; Pinz, A. **The integration of optical and magnetic tracking for multi-user augmented reality**, 1999.
- Azuma, R. A survey of augmented reality. **Teleoperators and Virtual Environments**, v.6, p. 355–385, 1997.
- Azuma, R.; Bailiot, Y.; Behringer, R.; Feiner, S.; Julier, S. ; MacIntyre, B. Recent advances in augmented reality. **IEEE Computer Graphics and Applications**, v.21, p. 34–47, 2001.
- Baratoff, G.; Neubeck, A. ; Regenbrecht, H. **Interactive multi-marker calibration for augmented reality applications**. In: Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, p. 107–, Washington, DC, USA, 2002. IEEE Computer Society.
- Dangelmaier, W.; Fischer, M.; Gausemeier, J.; Grafe, M.; Matysczok, C. ; Mueck, B. Virtual and augmented reality support for discrete manufacturing system simulation. **Computers in Industry**, v.56, n.4, p. 371 – 383, 2005.
- Galo, M.; Tozzi, C. A representação de matizes de rota e o uso de quatérnions em ciências geodésicas. **Série em Ciências Geodésicas**, 2001.
- Juan, M. C.; Alca, M.; Monserrat, C.; Botella, C.; Ba, R. M. ; Guerrero, B. Using augmented reality to treat phobias. **IEEE Computer Graphics and Applications**, v.25, p. 31–37, 2005.
- Kalkusch, M.; Lidy, T.; Knapp, M.; Reitmayr, G.; Kaufmann, H. ; Schmalstieg, D. **Structured visual markers for indoor pathfinding**, 2002.
- Kato, H. **ARToolKit 2.33 Documentation (Alpha Version)**. Human Interface Technology Laboratory, University of Washington, <http://www.hitl.washington.edu/artoolkit/documentation/>, 2005. [Online; acessado em 21-Outubro-2011].
- Kaufmann, H.; Schmalstieg, D. Mathematics and geometry education with collaborative augmented reality. **Computers & Graphics**, v.27, n.3, p. 339 – 345, 2003.
- Klopschitz, M.; Schmalstieg, D. **Automatic reconstruction of wide-area fiducial marker models**. In: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07, p. 1–4, Washington, DC, USA, 2007. IEEE Computer Society.
- Kotake, D.; Uchiyama, S. ; Yamamoto, H. **A marker calibration method utilizing a priori knowledge on marker arrangement**. In: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '04, p. 89–98, Washington, DC, USA, 2004. IEEE Computer Society.

- Kowoma.de. **Sources of errors in gps**, April 2009.
- Lukosiunas, E.; Bulbenkiene, V.; Andziulis, A.; Pasviestis, L. ; Artemciukas, E. **An ultrasonic tracking method for augmented reality**. In: 17th International Conference on Information and Software Technologies, Kaunas, Lithuania, April 2011.
- Nicolau, S.; Soler, L.; Mutter, D. ; Marescaux, J. Augmented reality in laparoscopic surgical oncology. **Surgical Oncology**, v.20, n.3, p. 189 – 201, 2011.
- Oda, O.; Feiner, S. **Rolling and shooting : two augmented reality games**. In: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, p. 3041 – 3044, New York, NY, USA, 2010. ACM.
- Park, J.; Park, J. **3dof tracking accuracy improvement for outdoor augmented reality**. In: ISMAR, p. 263–264, 2010.
- Shoemake, K. **Animating rotation with quaternion curves**. In: Proceedings of the 12th annual conference on Computer graphics and interactive techniques, SIGGRAPH '85, p. 245–254, New York, NY, USA, 1985. ACM.
- Siltanen; Hakkarainen ; Honkamaa. **Automatic marker field calibration**. In: Virtual Reality International Conference (VRIC2007), p. 261–267, 2007.
- Simon, G.; Berger, M.-O. Reconstructing while registering: A novel approach for markerless augmented reality. **Mixed and Augmented Reality, IEEE / ACM International Symposium on**, v.0, p. 285, 2002.
- Sutherland, I. **A head-mounted three dimensional display**. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I, AFIPS '68 (Fall, part I), p. 757–764, New York, NY, USA, 1968. ACM.
- Uematsu, Y.; Saito, H. **Vision based registration for augmented reality using multi-planes in arbitrary position and pose by moving uncalibrated camera**. In: Proceedings of the 2005 international conference on Augmented tele-existence, p. 111–119, 2005.
- Uematsu, Y.; Saito, H. **Ar registration by merging multiple planar markers at arbitrary positions and poses via projective space**. In: Proceedings of the 2005 international conference on Augmented tele-existence, ICAT '05, p. 48–55, New York, NY, USA, 2005. ACM.
- Wagner, D.; Schmalstieg, D. **ARToolKitPlus for Pose Tracking on Mobile Devices ARToolKit**, p. 139–146. Citeseer, 2007.
- Wang, L.; Springer, M.; Heibel, T. H. ; Navab, N. **Floyd-warshall all-pair shortest path for accurate multi-marker calibration**. In: ISMAR, p. 277–278. IEEE, 2010.