

A Method to Automate the Calibration of a Multiple Fiducial Marker Setup

Douglas Coelho Braga de Oliveira, Felipe Andrade Caetano, Rodrigo Luis de Souza da Silva

Departamento de Ciência da Computação

Universidade Federal de Juiz de Fora

Juiz de Fora, Brasil

dcoelhobo@gmail.com, felipecaetano@ice.ufjf.br, rodrigoluis@ice.ufjf.br

Abstract—This work proposes a new method to automate the calibration of multiple fiducial markers distributed over an Augmented Reality scene, in order to obtain stable relations, mainly among markers that will never be visible in a same video frame. The main feature of the method is to maintain all relationships between markers in each video frame. These relationships will be updated during calibration in order to obtain an accurate result. During the execution, the referential (base) marker of the system may change when markers with smaller error appears in the scene. The application produces the output configuration file of a multiple marker augmented reality environment in the same format used by AVRLib and ARToolkit libraries.

Keywords—Automatic Calibration; Multimarker Setup;

Resumo—Este trabalho propõe um novo método para automatizar a calibração de múltiplos marcadores fiduciais distribuídos em um sistema de Realidade Aumentada, com objetivo de obter relações estáveis, principalmente entre marcadores que nunca estarão visíveis em uma mesma imagem do vídeo. A principal característica do método é preservar todas as relações entre os marcadores em cada quadro do vídeo. Estas relações serão atualizadas durante a calibração de modo a obter um resultado preciso. Durante a execução, o marcador referencial (base) do sistema pode ser alterado quando aparecer marcadores com menor erro. A aplicação gera como saída o arquivo de configuração de um ambiente de realidade aumentada com múltiplos marcadores no mesmo formato utilizado pelas bibliotecas AVRLib e ARToolkit.

Palavras-chave—Calibração Automática; Configuração Multimarcador;

I. INTRODUÇÃO

Em aplicações de Realidade Aumentada (RA) é comum o uso de múltiplos marcadores fiduciais calibrados, isto é, um sistema de marcadores em que as transformações rígidas entre todos os marcadores são previamente conhecidas. Tais sistemas oferecem algumas vantagens sobre os que utilizam apenas um marcador único, a principal vantagem diz respeito a visibilidade do sistema. Problemas como oclusão do marcador ou uma má iluminação em um sistema de um marcador singular pode ocasionar em falhas na projeção dos objetos virtuais. Em um sistema de múltiplos marcadores calibrados, desde que ao menos um marcador do sistema esteja visível em cena, a projeção dos objetos virtuais pode ocorrer adequadamente. Entretanto, para obter essa configuração, as transformações rígidas entre os marcadores devem ser calculadas.

A obtenção da relação entre os marcadores é bastante suscetível a erros. Existem vários problemas que fornecem erros de precisão indesejáveis, a começar pela própria

precisão da máquina que é limitada passando por possíveis problemas no sistema de captura como oclusões temporárias, iluminação irregular na cena, efeitos de reflexões do marcador etc. Outros problemas estão relacionados à correta captura da profundidade do marcador, o ângulo de inclinação e de rotação. Movimentações na câmera e perda de foco também podem causar erros de precisão.

Usualmente a criação de cenários que envolvem múltiplos marcadores é feita de forma manual, sendo necessário o uso de instrumentos para o cálculo de ângulos e distâncias entre os marcadores. Dependendo da complexidade do cenário, a obtenção das relações entre os marcadores que compõem a cena pode ser inviável.

O foco principal deste trabalho é obter resultados satisfatórios na criação automática de uma configuração de marcadores distribuídos em uma cena. O trabalho apresentará duas distribuições diferentes, uma ao longo de um cubo e outra em um tipo de cilindro hexagonal irregular (Fig. 1).

Este trabalho está organizado da seguinte forma: A seção II traz os trabalhos relacionados, na seção III é abordado o modelo implementado. Na sequência, a seção IV apresenta os resultados que foram obtidos nos experimentos realizados e a seção V conclui o presente trabalho.



Figura 1. Configurações de marcadores utilizadas neste trabalho

II. TRABALHOS RELACIONADOS

Em [1] foi proposto um método semi-automático de calibração, com base em imagens previamente tiradas e ajustamento de feixes perspectivos. Os experimentos obtiveram um tempo de calibração próximo a um segundo. Foi possível avaliar até quarenta fotos e escolher o melhor par para realização dos cálculos. Porém este método só é eficiente quando todos os marcadores se encontram próximos o suficiente para que a maioria deles estejam visíveis na mesma imagem como é revelado no trabalho de Klopschitz em [6]. Por este motivo o autor propôs um método semelhante aplicável a áreas maiores.

No trabalho desenvolvido em [9], foi proposto um método capaz de calcular as relações entre cada um dos marcadores, que tiveram seus sistemas de coordenadas independentes combinados em um espaço projetivo único. O sistema captura duas imagens aleatórias de posições diferentes, ambas as imagens devem ser obtidas tendo todos os marcadores visíveis para que o espaço seja projetado. Como nem sempre todos os marcadores poderão estar visíveis ao mesmo tempo, não importando o ângulo em que são capturadas as imagens, esta restrição torna-se indesejável.

Uma outra abordagem de calibração de marcadores distribuídos aleatoriamente na cena é sugerida em [2]. Neste trabalho o marcador usado como referência é alterado de acordo com sua qualidade em cada *frame* de vídeo. Diversos métodos são sugeridos como forma de definir a qualidade dos marcadores.

Em [8] foi proposta uma abordagem utilizando uma estrutura de grafos para interconectar marcadores que não são visíveis na cena ao mesmo tempo, mantendo o referencial fixo. Desta forma, sempre que existir um caminho no grafo que conecte-os por intermédio de outros marcadores do caminho, a relação entre eles pode ser obtida, acumulando-se as transformações do caminho. Uma desvantagem desta abordagem é que em algumas situações haverá a necessidade de termos um número desnecessário de cálculos para a obtenção do resultado final. Quando um marcador estiver conectado ao marcador base do sistema por vários outros marcadores, as transformações entre cada par de marcadores ligados no caminho devem ser multiplicadas umas com as outras, até chegar ao resultado final. A isso soma-se o custo de encontrar o menor caminho em um grafo.

O uso de marcadores fiduciais em aplicações de RA ainda é muito frequente por propiciar o registro preciso dos marcadores que comporão o cenário e exigir baixo poder de processamento se comparado a abordagens que utilizam características naturais da cena como os trabalhos propostos em [11][4][3][12].

III. MODELO PROPOSTO

No desenvolvimento do presente trabalho foi utilizada a biblioteca *AVRLib* [7]. Esta é uma biblioteca para criação de aplicações em RA desenvolvida em C/C++. Esta biblioteca faz uso dos recursos de visão computacional fornecidos pela biblioteca *ARToolKit* [5]. A *AVRLib* apresenta

interface de programação orientada a objetos e utiliza a biblioteca *OpenGL* como *API* gráfica.

A. Definições

Seja $M(S)$ o conjunto de marcadores registrados no sistema. Seja M_i a matriz de transformação de $i \in M(S)$ em relação à câmera e Cf_i o valor de confiança desta transformação. Seja M_{ib} a matriz de transformação de $i \in M(S)$ em relação ao marcador base $b \in M(S)$ e Cf_{ib} o valor de confiança da mesma. O marcador base do sistema sempre será representado pela letra “B” minúscula (b).

O fluxo geral do método proposto neste trabalho pode ser dividido em quatro etapas:

- 1) Detecção de marcadores e transformação da câmera
- 2) Seleção do marcador base
- 3) Estimativa da transformação de cada marcador visível com a base e vice-versa
- 4) Estimativa da transformação entre cada par de marcadores não visíveis

Estas etapas serão detalhadas a seguir.

B. Detecção de Marcadores e Transformação da Câmera

Esta etapa consiste em analisar cada *frame* de vídeo a procura dos marcadores previamente registrados no sistema e calcular para cada um deles a sua transformação em relação à câmera. A *AVRLib* executa esta tarefa de forma automática em seu *loop* principal. A biblioteca permite a criação de sistemas de marcadores específicos que são incorporados à mesma ao implementar a classe abstrata *avrSystemMarker*. Desta forma, foi criada uma classe para gerir os marcadores e estimar as transformações entre eles. A análise dos quadrados detectados pela biblioteca na imagem, que podem vir a ser marcadores, a transformação da câmera e as demais etapas da aplicação são todas realizadas nesta fase.

Uma outra notação será utilizada nesta subseção: $M(D)$ será o conjunto de quadrados detectados pela *AVRLib* em um *frame* do vídeo que podem vir a ser um dos marcadores de $M(S)$.

A análise dos elementos de $M(D)$ é feita pela simples comparação de identificadores, que foram atribuídos a cada elemento de $M(S)$ pela própria biblioteca. Quando o ID do marcador que está sendo procurado na imagem corresponder ao ID de um quadrado da imagem, este quadrado é o marcador procurado, logo está visível.

A transformação rígida da câmera é calculada através de duas funções da biblioteca, a *avrGetTransMat* e a *avrGetTransMatCont*, sendo que a última leva em consideração a matriz de transformação de um quadro anterior, em que o marcador estava visível. Ambas as funções retornam um valor de *erro* referente a transformação calculada, esse *erro* > 0 é usado na definição do valor de confiança da transformação dada na equação 1. O uso da matriz de um quadro anterior é sugerido em [8] e serve como forma de minimizar o erro da transformação. A Fig. 2 mostra o fluxograma desta etapa.

$$Cf_i = \frac{1}{erro} \quad (1)$$

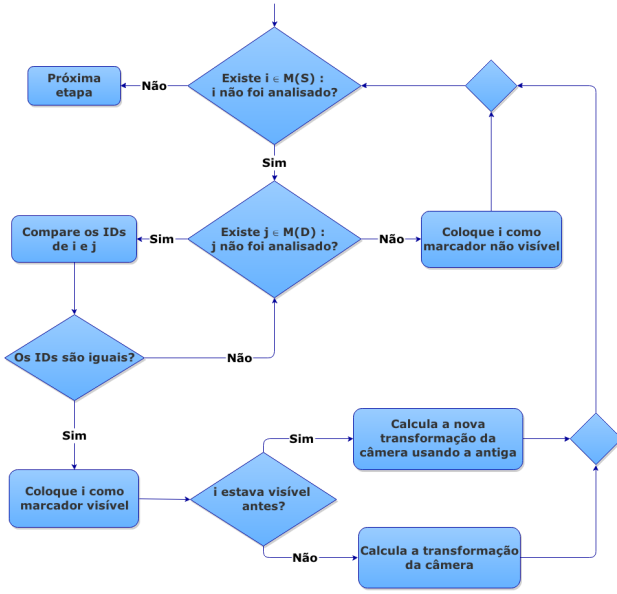


Figura 2. Análise dos marcadores e transformação da câmera

C. Seleção do Marcador Base

Sistemas de múltiplos marcadores possuem um referencial global, ou seja, a transformação de cada marcador que compõe o sistema é dada em relação a este referencial. Usar um dos marcadores do sistema como referência para os demais elimina ter de calcular a transformação para ele mesmo, como sugerido em [10]. É possível variar este marcador (chamado de *marcador base*) de acordo com a qualidade do mesmo na cena, i.e., se um marcador do sistema é o melhor com base em alguns critérios definidos, este pode ser usado como o referencial global do sistema. Em [2] são propostas algumas políticas de troca de marcador base.

Para efeito de comparação de resultados, serão utilizados três métodos de definição do melhor marcador da cena. O marcador base será substituído quando existir um marcador com qualidade $X\%$ superior a da base atual. Este valor de porcentagem é um parâmetro do sistema e para este trabalho foi definido como $X = 10$.

1) *Maior Confiança*: Como dito na subseção anterior, as funções que calculam a transformação rígida da câmera retornam um valor de erro, que representa a margem de erro desta matriz. Este valor é usado para obter a confiança desta transformação usando a relação $Conf \equiv \frac{1}{erro}$, onde $erro > 0$. Portanto este método define como o melhor marcador da cena o que tiver o maior valor de confiança.

2) *Menor Inclinação*: De acordo com [5] o rastreamento é afetado pelo ângulo de inclinação do marcador em relação à câmera. As duas configurações de marcadores, que são usadas neste trabalho, possuem vários marcadores inclinados, sendo assim, este método define o marcador menos inclinado em relação à câmera como o melhor da cena.

3) *Menor Inclinação e Erro Aceitável*: Este método sugere usar uma combinação dos dois métodos anteriores para definir o melhor marcador da cena. Deste modo, o

marcador base será o marcador que for o menos inclinado em relação a câmera, dentre os que possuem um erro pertencente ao intervalo $(0, 1)$.

D. Transformação entre Marcadores Visíveis

Quando mais de um marcador é visível, um deles é definido como base e a transformação Mib_t , $i \in M(S)$ e $i \neq b$, é obtida pela equação 2. O novo índice t nas matrizes indica o quadro de vídeo corrente.

$$Mib_t = Mb_t^{-1} \times Mi_t \quad (2)$$

De modo análogo, obtêm-se a transformação inversa, ou seja, a transformação da base b em relação a i , pela equação 3.

$$Mbi_t = Mi_t^{-1} \times Mb_t \quad (3)$$

O valor de confiança destas relações foi definido como sendo o inverso da média aritmética entre o erro da transformação da câmera de cada marcador. Pode-se escrever este cálculo como na equação 4.

$$Cf_{ib_t} = Cf_{bi_t} = \frac{2}{erro_{Mi} + erro_{Mb}} \quad (4)$$

As três equações anteriores são usadas para obter apenas a transformação e confiança do *frame* corrente, mas as informações que são armazenadas decorrem de todas as informações obtidas em *frames* anteriores. O motivo para isto é simples, o objetivo do método é obter as relações entre todos os marcadores, independente dos mesmos estarem ou não visíveis em um determinado *frame*. Este marcadores devem estar distribuídos na cena e não deve haver mudança na posição e orientação de um marcador em relação a outro, logo devem ficar estáticos. Isto implica que as relações não mudam com o tempo, porém isto não acontece na prática devido aos vários problemas com relação a precisão já discutidos na introdução deste trabalho. Em um *frame* t qualquer a transformação obtida pode estar muito próxima a real, porém em um outro *frame* t' a transformação pode conter grande imprecisão por fatores externos. Em vista disto, a abordagem utiliza o cálculo da média ponderada dos valores obtidos anteriormente, tanto para a transformação final quanto para a confiança correspondente.

A confiança final é a média ponderada entre os valores de confiança nos frames $t - 1$ e t . O peso de $Cf_{ib_{t-1}}$ é dado pelo número de vezes que a transformação final foi atualizada até o frame $t - 1$ (*upd*) e o peso de Cf_{ib_t} é 1, visto que ela é válida apenas no frame corrente (Eq. 5).

$$Cf_{ib} = Cf_{bi} = \frac{upd * Cf_{ib_{t-1}} + Cf_{ib_t}}{upd + 1} \quad (5)$$

O mesmo se aplica ao cálculo da matriz de transformação final. A equação 6 mostra este cálculo para Mib . Para a matriz Mbi o cálculo é exatamente o mesmo bastando-se trocar de posição os índices i e b .

$$Mib = \frac{upd * Mib_{t-1} + Cf_{ib_t} * Mib_t}{upd + 1} \quad (6)$$

O fluxo de execução desta etapa é simplificado no fluxograma da Fig. 3. Em linhas gerais, percorre-se todo

o conjunto $M(S)$ calculando as transformações M_{ib_t} e M_{bi_t} , $\forall i$ visível. Se é a primeira vez que estas matrizes são calculadas, as próprias já são as matrizes finais para este *frame*, senão calcula-se a média. Em ambos casos, os cálculos só serão realizadas se $C_{f_{ib_t}}^{-1}, C_{f_{bi_t}}^{-1} \in (0, 1)$, que é o intervalo de erro aceitável para o método. Ao final o contador de atualizações da transformação será atualizado.

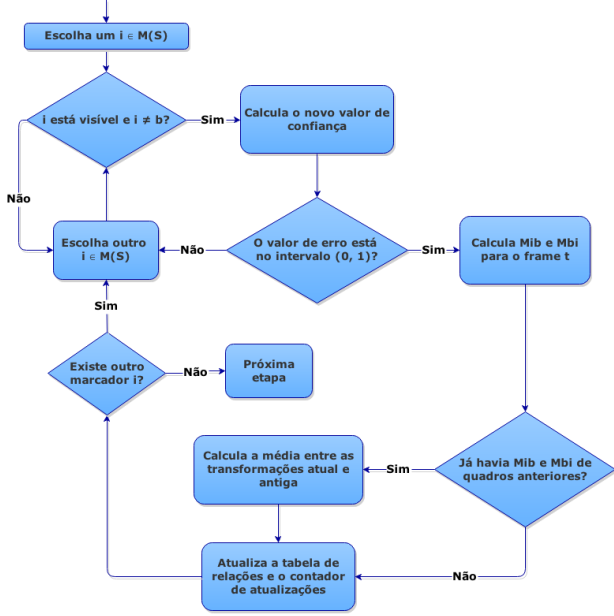


Figura 3. Transformação entre marcadores visíveis

E. Transformação entre Marcadores Não Visíveis

Nos exemplos de configurações ilustrados na Fig. 1, existem pares de marcadores que nunca estarão visíveis numa mesma imagem do vídeo. Portanto é preciso uma maneira diferente de estimar tais relações.

Tomando o exemplo do cubo, um par de marcadores que jamais serão vistos juntos na mesma imagem são os marcadores 1 e 6. Pode-se observar que é possível visualizar, por exemplo, os marcadores 1 e 2 em um *frame* e os marcadores 2 e 6 em outro. Então, em um dado momento, o sistema poderá ter calculado as matrizes M_{12} , M_{21} , M_{26} e M_{62} . Analisando então a construção dessas matrizes observa-se que é possível obter a relação M_{16} e M_{61} usando o marcador 2 como intermediário.

Ao aplicarmos a matriz M_{12} no espaço vetorial do marcador 1 obtemos o referencial de 2. Do mesmo modo, aplicando a matriz M_{26} no espaço de 2 encontramos o referencial de 6. Então é possível obter a relação M_{16} aplicando ambas transformações em sequência. O processo é análogo para obter M_{61} , mas usa-se as matrizes M_{62} e M_{21} . A demonstração formal deste processo encontra-se a seguir.

Sejam $A, B, C \in M(S)$ tal que os marcadores A e C em nenhum momento estejam visíveis juntos. As matrizes M_{AB} , M_{BC} , M_{CB} e M_{BA} são obtidas da equação 2. Multiplicando M_{BC} por M_{AB} e, M_{BA} por M_{CB} , na

ordem em que foram citadas, visto que a multiplicação de matrizes não é uma operação comutativa, obtém-se:

$$M_{BC} \times M_{AB} = (M_C^{-1} \times M_B) \times (M_B^{-1} \times M_A)$$

$$M_{BA} \times M_{CB} = (M_A^{-1} \times M_B) \times (M_B^{-1} \times M_C)$$

A multiplicação de matrizes, entretanto, é associativa, logo podemos reescrever as equações acima da forma:

$$M_{BC} \times M_{AB} = M_C^{-1} \times (M_B \times M_B^{-1}) \times M_A$$

$$M_{BA} \times M_{CB} = M_A^{-1} \times (M_B \times M_B^{-1}) \times M_C$$

Considerando que o produto $M_B \times M_B^{-1}$ resulta na matriz Identidade I , obtemos:

$$M_{BC} \times M_{AB} = M_C^{-1} \times M_A = M_{AC}$$

$$M_{BA} \times M_{CB} = M_A^{-1} \times M_C = M_{CA}$$

Durante a execução do sistema, a estimativa da transformação entre A e C não se dará apenas por um marcador intermediário, mas por todos os marcadores que servirem como intermediários. Isto é, $\forall k \in M(S), k \neq A, k \neq C$, se existe as matrizes M_{Ak} e M_{kC} e estas tem o valor de confiança definido no intervalo $(0, 1)$, então k é usado como intermediário. Deste modo as equações 5 e 6 também são válidas nesta etapa da calibração.

Existe um critério para que esta etapa seja executada. Tomando novamente o cubo como exemplo, é possível visualizar em algum *frame* os marcadores 1, 2 e 3, não mais que isso. Entretanto, em um outro *frame* futuro, o marcador 4 poderá ser visto junto ao 1, não havendo necessidade de ter sido calculado a relação entre 1 e 4 usando um intermediário. Como este método utiliza mais operações, a propagação de erro tende a ser maior, então é conveniente evitá-lo quando possível.

Para que este método seja executado, foi definido um número mínimo de atualizações para todas as relações necessárias ao cálculo. Desta forma, para calcular a relação entre os marcadores 1 e 4 por este método, todas as relações entre estes dois e os marcadores que lhes servem como intermediário, devem ter sido atualizadas ao menos N vezes. Com isso ganha-se um tempo para que os outros marcadores apareçam e, mesmo quando não aparecerem, garante que as relações tenham um valor de confiança elevado. O fluxo de execução pode ser conferido no fluxograma da Fig. 4.

F. Visualização da calibração

Na *callback* de renderização as relações são exibidas no vídeo por meio de quadrados verdes que contornam o marcador que teve sua relação com a base calculada. Sobre o marcador base são desenhados os eixos Ox e Oy e seu contorno é definido pela cor azul. A Fig. 5 mostra um momento da execução para o cubo onde todas as relações já haviam sido calculadas.

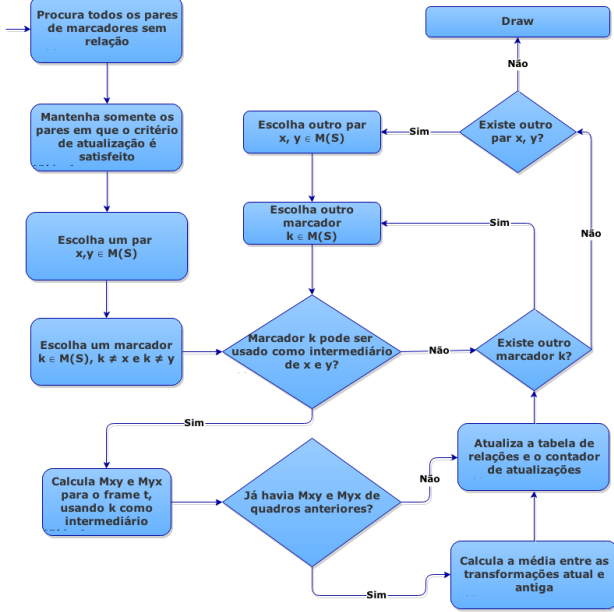


Figura 4. Transformação entre marcadores não visíveis

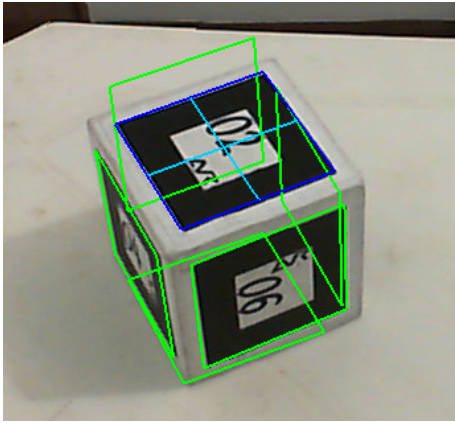


Figura 5. Quadro da execução do sistema para o cubo com todas relações calculadas

IV. RESULTADOS

Para os experimentos desta seção foi utilizado um notebook com processador Intel Core i5 2410m 2ª geração, 6GB HDR3 e placa de vídeo integrada Intel Graphic 3000. A câmera é uma Microsoft LifeCam 720p. Como visto anteriormente, as configurações de marcadores testadas estão ilustradas na Fig. 1. Todos os testes foram executados numa faixa entre 27fps e 30fps. As subseções seguintes apresentam a métrica de erro utilizada e os testes para cada configuração.

A. Métrica de Erro

Para calcular o erro de projeção foram seguidas as propostas sugeridas nos trabalhos de Siltanen et al [8] e Uematsu et al [9]. Estes trabalhos utilizaram a média das distâncias, em milímetros, entre os vértices do cubo projetado sobre o marcador, usando os próprios dados deste marcador e, os vértices do cubo projetado usando

a transformação estimada que fora calculada durante a execução dos trabalhos (Fig. 6). Porém, foi necessária uma modificação neste cálculo para o presente trabalho, visto que todas as relações são armazenadas e que o referencial do sistema pode variar.

Desta forma, calcula-se normalmente a média das distâncias entre os vértices para cada marcador visível junto a base corrente. Após, uma segunda média é calculada referente ao número de marcadores visíveis. Este processo é realizado para cada marcador que em algum momento assumiu o papel de referencial do sistema. Esse cálculo é repetido em cada *frame* e então encontramos o valor do erro de projeção máximo para cada base durante toda a execução do sistema.

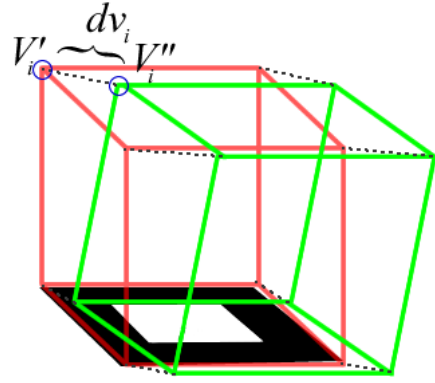


Figura 6. Ilustração do cálculo do erro de projeção para um marcador

B. Resultados Obtidos

O método implementado tem como saída arquivos de configuração de um ambiente de realidade aumentada multimarcador. Uma aplicação, chamada *DragonTest*, foi criada a fim de testar os arquivos gerados pelo sistema. Esta aplicação realiza a renderização de um dragão sobre um plano.

Para o cubo, a base do sistema gerado é o marcador 1. O resultado pode ser conferido na Fig. 7. É possível notar uma leve alteração visual na projeção quando a base é coberta (Fig. 7.b).

Para a segunda configuração de marcadores, a base do sistema é o marcador 2, entretanto, a projeção foi realizada sobre o marcador 1 a fim de manter o plano na horizontal. Para obter este efeito, a matriz de transformação do marcador 2 em relação à câmera foi multiplicada pela matriz M_{12} gravada em arquivo. O resultado é a matriz de transformação estimada do marcador 1 em relação à câmera.

$$M_2 \times M_{12} = M_2 \times (M_2^{-1} \times M_1) = M_{1_{est}} \quad (7)$$

Assim, aplicando a transformação $M_{1_{est}}$ é possível estimar a posição do marcador 1 na cena em relação à câmera e, com isso, realizar a projeção sobre ele. O resultado pode ser verificado na Fig. 8. Novamente, nota-se que a oclusão do marcador base acarreta em uma alteração na projeção que, apesar de ainda ser aceitável, é mais visível que a alteração sofrida no cubo. Um resultado

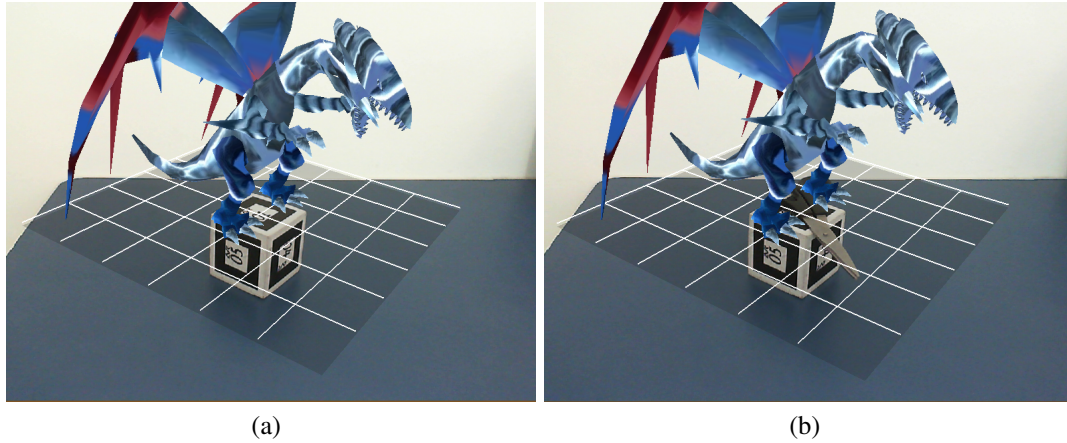


Figura 7. Execução da aplicação DragonTest para o marcador cúbico. Em (a) o marcador base está visível. Em (b) a base e outro marcador, antes visíveis, são cobertos o que acarreta em uma pequena alteração na projeção.

interessante é que a projeção se manteve possivelmente próxima ao marcador 1 mesmo quando o cilindro foi rotacionado, exibindo os marcadores opostos à ele e sem a base estar presente.

C. Análise dos Dados

Na subseção IV-A foi explicada a métrica de erro usada neste trabalho. Aplicamos esta métrica a cada uma das configurações de marcadores, trabalhadas aqui, para cada método de seleção de base. Os dados gerados estão organizados nas Tabelas I e II, sendo a primeira referente ao marcador cúbico e a segunda ao cilindro hexagonal irregular. Estas tabelas mostram o erro máximo de projeção, em milímetros, encontrado durante a execução do sistema para cada marcador que em algum momento foi base do sistema.

É preciso levar em consideração alguns fatores:

- Só é possível calcular o erro de projeção das matrizes de transformação entre a base corrente e os marcadores visíveis no momento. Tomando o cubo como exemplo, isto implica que nos quadros onde o marcador 1 foi a base, a relação entre 1 e 6 não foi levada em consideração no cálculo do erro em nenhum destes quadros.
- O valor do erro refere-se a apenas um quadro de toda a execução: o quadro onde a base corrente obteve o seu maior erro de projeção.

É possível perceber que para o cubo, em todos os métodos de seleção, não houve grande variação do erro

para cada marcador. É visível também que o método de *Maior Confiança* se sobressaiu aos demais obtendo, em sua maioria, erros menores. Entretanto a diferença dos valores não é suficientemente grande para afirmar com exatidão que esta proposta de seleção de base seja a melhor para esta configuração. Em relação ao marcador hexagonal, é visível a grande variação do erro para cada base. Percebe-se que os resultados obtidos nos métodos onde a inclinação em relação à câmera é levada em consideração, são consideravelmente melhores que o método de seleção de *Maior Confiança*. Os erros de cada marcador são, em sua maioria, menores nestes métodos que consideram a inclinação. A variação do erro, interna a cada método, também é muito menor nestes métodos de seleção.

V. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou um novo método para obter as relações entre marcadores em um cenário com múltiplos marcadores em uma cena. Este método é potencialmente útil em cenários onde existam pares de marcadores jamais visíveis juntos numa mesma cena. Duas configurações foram usadas como exemplo prático para os experimentos. Foi proposto que o referencial do sistema possa variar para aumentar a confiança nas relações e métodos para determinar o melhor marcador da cena nestas configurações. Foi apresentada ainda uma métrica de erro para avaliar os resultados obtidos.

Os resultados experimentais mostraram que, para uma configuração com vários marcadores em diferentes inclina-

Tabela I
ERRO DE PROJEÇÃO MÁXIMO DE CADA MARCADOR BASE EM CADA UM DOS MÉTODOS DE SELEÇÃO PARA O MARCADOR CÚBICO

Base	1	2	3
Maior Confiança	65.91	73.28	62.81
Menor Inclinação	81.77	60.19	77.61
Inclinação + Erro	102.18	70.71	83.46
Base	4	5	6
Maior Confiança	68.59	62.49	52.26
Menor Inclinação	77.07	89.95	41.46
Inclinação + Erro	77.65	62.26	91.22

Tabela II
ERRO DE PROJEÇÃO MÁXIMO, EM MILÍMETROS, DE CADA BASE EM CADA MÉTODO DE SELEÇÃO COM MARCADOR HEXAGONAL

Base	1	2	3	4
Maior Confiança	1308.94	768.80	35.17	26.46
Menor Inclinação	109.13	552.04	55.03	53.01
Inclinação + Erro	61.14	38.30	59.91	527.91
Base	5	6	7	8
Maior Confiança	1009.43	27.16	305.86	129.25
Menor Inclinação	63.45	32.48	53.51	16.12
Inclinação + Erro	745.22	22.51	237.85	35.42

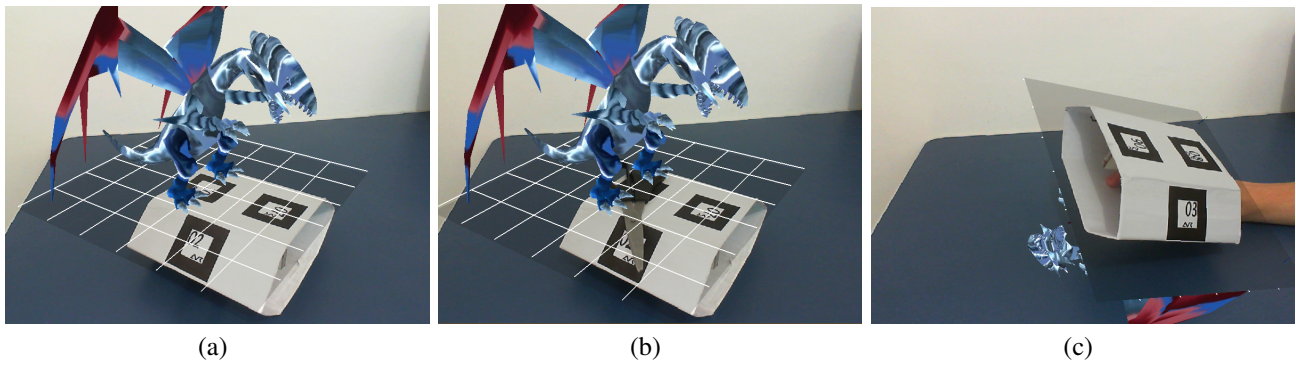


Figura 8. Execução da aplicação DragonTest para o marcador hexagonal. Em (a) a base do sistema está visível. Em (b) a base e o marcador onde a projeção foi aplicada estão cobertos. Em (c) o cilindro é rotacionado até que não apareça nenhum marcador do mesmo plano de visão da base.

ções, usar como base o marcador que estiver com a menor inclinação em relação à câmera pode diminuir o erro da projeção. O resultado dos arquivos gerados, confirmam que as relações calculadas pelo sistema desenvolvido neste trabalho estão estáveis, atingindo assim seu objetivo.

O processo de geração automática dos marcadores ainda não é prático, pois exige que as relações entre os marcadores atualmente visíveis sejam estabilizadas com um grande número de atualizações. Movimentos bruscos, erros no rastreamento, entre outros fatores, geram uma relação errônea que interfere na obtenção de um bom resultado final.

Sugere-se como trabalhos futuros a implementação de métodos que reconheçam relações entre marcadores de forma mais precisa no sistema, impedindo atualizações desnecessárias que possam propagar erros. A criação de filtros capazes de rejeitar configurações imprecisas durante a etapa de atualização também podem ajudar na obtenção de configurações estáveis mais rapidamente.

REFERÊNCIAS

- [1] G. Baratoff, A. Neubeck, and H. Regenbrecht. Interactive multi-marker calibration for augmented reality applications. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 107–, Washington, DC, USA, 2002. IEEE Computer Society.
- [2] F. A. Caetano, D. C. B. Oliveira, and R. L. S. Silva. Automatic calibration of multiple markers in an augmented reality environment. *Workshop de Realidade Virtual e Aumentada (WRVA)*, pages 48–53, 2013.
- [3] Z. Chen and X. Li. Markless tracking based on natural feature for augmented reality. In *Educational and Information Technology (ICEIT), 2010 International Conference on*, volume 2, pages V2–126–V2–129, Sept 2010.
- [4] M. Kanbara, N. Yokoya, and H. Takemura. A stereo vision-based augmented reality system with marker and natural feature tracking. In *Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM'01)*, VSMM '01, pages 455–, Washington, DC, USA, 2001. IEEE Computer Society.
- [5] H. Kato. Artoolkit 2.33 documentation (alpha version), 2005. [Online; acessado em 21-Outubro-2011].
- [6] M. Klopschitz and D. Schmalstieg. 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*, pages 1–4, Washington, DC, USA, 2007. IEEE Computer Society.
- [7] D. C. B. Oliveira, F. A. Caetano, and R. L. S. Silva. Avrlib - an object oriented augmented reality library. *Workshop de Realidade Virtual e Aumentada (WRVA)*, pages 54–59, 2013.
- [8] Siltanen, Hakkarainen, and Honkamaa. Automatic marker field calibration. In *Virtual Reality International Conference (VRIC2007)*, pages 261–267, 2007.
- [9] Y. Uematsu and H. Saito. 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*, pages 48–55, New York, NY, USA, 2005. ACM.
- [10] L. Wang, M. Springer, T. H. Heibel, and N. Navab. Floyd-warshall all-pair shortest path for accurate multi-marker calibration. In *ISMAR*, pages 277–278. IEEE, 2010.
- [11] M. L. Yuan, S. K. Ong, and A. Y. C. Nee. Registration using natural features for augmented reality systems. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):569–580, July 2006.
- [12] C. Zhao, Z. Yu, Q. Tian, and Z. Dalin. Augmented reality for planar scene via affine transform based natural features tracking. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 5085–5089, June 2008.