

REALIDADE AUMENTADA SEM MARCADORES BASEADA EM ARESTAS, UM ESTUDO DE CASO

TRABALHO DE GRADUAÇÃO

Aluno: Francisco Paulo Magalhães Simões (fpms@cin.ufpe.br)

Orientadora: Veronica Teichrieb (vt@cin.ufpe.br)

Co-orientadora: Judith Kelner (jk@cin.ufpe.br)

Recife, 26 de novembro de 2008

Resumo

Técnicas de realidade aumentada dependem da recuperação de informações da câmera virtual a cada quadro de uma seqüência de imagens para corretamente associar informações 3D às cenas reais. Este trabalho de graduação apresenta um estudo detalhado acerca das técnicas de realidade aumentada sem marcadores, que utilizam informações naturalmente presentes nas cenas para realizar o registro entre objetos virtuais e reais, focando especialmente nas técnicas baseadas em arestas. Este tipo de técnica é muito utilizado para o rastreamento de objetos poligonais sendo naturalmente estáveis a mudanças de iluminação. Além disto, esta monografia discute sobre os conceitos básicos da área de forma a permitir um melhor entendimento do leitor. Finalmente, é descrita a implementação e avaliação de uma técnica baseada na amostragem de pontos em arestas de um modelo 3D previamente gerado que se mostra capaz de recuperar as informações da câmera utilizadas para realizar a inserção dos objetos virtuais no mundo real. A técnica se baseia no algoritmo Moving Edges para encontrar as correspondências entre os pontos amostrados no objeto virtual e os pontos da imagem real. Foram utilizadas também a extração de arestas invisíveis e o minimizador não-linear Levenberg-Marquardt para tornar a técnica robusta a oclusões parciais e auto-oclusões. A técnica implementada atingiu resultados bastante precisos tanto na utilização de dados reais quanto em cenas sintéticas com um erro médio de 2 mm para o caso de teste de maior número de arestas e 15 mm para o caso de teste contendo menos arestas. Além disto, as aplicações elaboradas obtiveram taxas entre 15 e 30 fps denotando sua eficiência.

Agradecimentos

“O único lugar em que o sucesso vem antes do trabalho é no dicionário”

Albert Einstein

Motivado por esta frase épica alcanço mais um objetivo de vida através de muito trabalho, luta e determinação. Neste momento faz-se necessário um sincero reconhecimento às pessoas que contribuíram para minha formação intelectual, social e ética e um pedido de desculpas antecipado aos que por distração ou falha não aparecem neste texto apesar da importância que têm.

Primeiramente é importante agradecer aos meus pais Mary e Simões, e minha irmã Maryane, pessoas sem as quais não haveria a célula fundamental que me suporta desde que me entendo por gente. As lições de caráter e hombridade que recebi ao longo da caminhada e me fizeram ser o que sou hoje. MUITO OBRIGADO POR TUDO.

Também é importante agradecer à minha família e amigos próximos que sempre se fizeram presentes em minha vida, apoiando tanto nos bons quanto nos maus momentos, deixando a caminhada menos difícil.

Ainda com relação aos amigos, é importante destacar também aqueles que conviveram comigo durante estes 4 anos e meio no CIn, passando muitas vezes madrugadas inteiras trabalhando para entregar projetos que pareciam nunca acabar. Além de ótimos companheiros de trabalho, ganhei grandes amigos, pessoas com a qual poderei sempre contar e poderão sempre contar comigo como Thiago Lacerda (Xeroso), Fernando Kakimoto (Japa), Felipe Cavalcanti (Moxinho), André Guedes, Bruno Lins (Pigmeu), Rebeka Gomes (Beka), Henrique Seabra, Rílter Tavares, Ademir José (Tanga), Rodrigo Melo (Digão), David Aragão, Jesus Sanches e Filipe Cesar (Gravatá).

Outras pessoas muito importantes são os companheiros de trabalho, pessoas com as quais partilho meus momentos profissionais e por conta do tempo demandado por estas atividades muito de minha vida particular também. Aos amigos do GPRT/GRVM que me inspiram todo dia sendo ótimas pessoas e excelentes profissionais como João Grandão, Joma, Mouse, Curupa, Mozart, Márcio, Lucas, VT e Judith entre tantos outros que o curto espaço não me permite citar. Aos amigos de pesquisas no DEN, Departamento de Energia Nuclear da UFPE, que durante muito tempo compartilharam seu profissionalismo e amizade como Eric Ferreira, Sílvio Melo e Carlos Dantas.

Obrigado também aos mestres que tive ao longo da vida acadêmica, desde o segundo grau aos mestres do ciclo efetivamente acadêmico como Rogério Porto, Walter Lafaiete, Mauro Rego, Sílvio Melo, Augusto Sampaio, Fábio Silva, Sérgio Cavalcante, Carlos Ferraz, Veronica Teichrieb e Judith Kelner, as duas últimas sendo também orientadora e co-orientadora deste trabalho. Estas

figuras foram responsáveis diretamente por diversas qualidades profissionais adquiridas além de muitos se tornarem bons amigos.

Acredito que todos os sonhos podem ser alcançados com esforço, dedicação e um pouco de sorte. Acabo de realizar um sonho antigo e parto agora em busca de outros com a certeza de que pessoas maravilhosas continuarão no meu caminho e diversas outras surgirão complementando sempre os espaços nesta longa jornada chamada VIDA.

“Sonhar é a nobre arte de acreditar no amanhã”

Francisco Paulo Magalhães Simões

Sumário

1. Introdução	8
2. Conceitos básicos	9
2.1. Rastreamento da câmera.....	11
2.2. Realidade aumentada sem marcadores.....	13
3. MAR baseada em arestas	16
3.1. Extração explícita de arestas	17
3.2. Amostragem de pontos.....	20
4. Descrição da técnica.....	25
4.1. Amostragem de pontos.....	27
4.2. Teste de visibilidade.....	28
4.3. Casamento entre pontos com Moving Edges	29
4.4. Parametrização da pose	30
4.5. Estimativa da pose.....	31
4.5.1. Hipótese única.....	31
4.5.2. Múltiplas hipóteses.....	32
5. Estudo de caso.....	33
5.1. Implementação	34
5.2. Resultados	35
6. Conclusões	43
6.1. Trabalhos futuros.....	43
7. Referências	45

Índice de Figuras

Figura 1. Componentes formadores da RA segundo [7].	9
Figura 2. <i>Pipeline</i> típico de realidade aumentada com marcadores.	10
Figura 3. Os parâmetros extrínsecos definem a rotação e translação R, t responsáveis por levar o sistema de coordenadas de mundo O para o sistema de coordenadas da câmera C .	12
Figura 4. O ponto 3D M pode ser projetado no ponto 2D m através da aplicação da matriz de projeção P oriunda dos parâmetros intrínsecos e extrínsecos da câmera C .	13
Figura 5. Taxonomia das técnicas de MAR monocular.	14
Figura 6. Objeto poligonal rastreado a partir de suas arestas. Figura extraída de [32].	17
Figura 7. Arestas extraídas da cena exibidas na imagem da esquerda utilizadas para realizar o rastreamento do carrinho demonstrado na imagem da direita pela sobreposição do modelo ao carrinho (linhas cinzas). Figura extraída de [1].	18
Figura 8. <i>Pipeline</i> definido por Drummond [25].	20
Figura 9. Objeto a ser rastreado. Figura extraída de [25].	21
Figura 10. Rastreamento utilizando partes de mais de um objeto. Figura extraída de [25].	22
Figura 11. <i>Pipeline</i> definido por Wuest.	23
Figura 12. Aplicação da máscara gaussiana ao longo da normal à aresta projetada na busca de uma estimativa para o ponto amostrado na aresta.	24
Figura 13. Rastreamento realizado com sucesso através da utilização da técnica de Wuest em objetos complexos com oclusão. Figura extraída de [6].	25
Figura 14. <i>Pipeline</i> seguido pela técnica implementada nesta monografia.	25
Figura 15. Inicialização manual do rastreamento para testes com dados reais e sintéticos.	26
Figura 16. Distância entre pontos amostrados nas arestas (x, y) varia de forma a manter a quantidade de pontos proporcional ao tamanho da aresta projetada (tam, TAM) .	28
Figura 17. A aresta 1 é dividida nas arestas 1' e 2' devido à auto-occlusão após projeção.	29
Figura 18. A aresta vermelha é projetada e os pontos na cor azul são amostrados. Através do Moving Edges os pontos laranja são definidos como correspondentes.	30
Figura 19. Cubo rastreado no estudo de caso sendo manipulado pelas mãos do usuário.	33
Figura 20. Rastreamento de modelo sintético complexo.	34
Figura 21. Seqüência de rastreamento do cubo denotando a falha de rastreamento após movimento brusco do mesmo.	36
Figura 22. Cubo sendo rastreado mesmo com a existência de oclusão parcial.	36
Figura 23: Confusão do Moving Edges devido ao alto gradiente proporcionado pela proximidade dos dedos com o cubo.	36
Figura 24. Seqüência sem a utilização de múltiplas hipóteses.	37

Figura 25. Seqüência utilizando múltiplas hipóteses.	37
Figura 26. Duas seqüências sintéticas utilizadas para averiguar a precisão da técnica.....	37
Figura 27. Seqüência da fachada do prédio da UFAL.	38
Figura 28. Erro associado ao eixo X da seqüência frontal do prédio da UFAL.....	39
Figura 29. Erro associado ao eixo Y da seqüência frontal do prédio da UFAL.....	39
Figura 30. Erro associado ao eixo Z da seqüência frontal do prédio da UFAL.	39
Figura 31. Distribuição do erro total do rastreamento na seqüência frontal.....	40
Figura 32. Seqüência do prédio da UFAL completo.....	41
Figura 33. Erro associado ao eixo X da seqüência do prédio da UFAL completo.....	41
Figura 34. Erro associado ao eixo Y da seqüência do prédio da UFAL completo.....	42
Figura 35. Erro associado ao eixo Z da seqüência do prédio da UFAL completo.	42
Figura 36. Distribuição do erro total do rastreamento na seqüência do prédio da UFAL completo.....	42

1. Introdução

Atualmente, a realidade aumentada (RA) mostra-se como um ramo de pesquisa na vanguarda da ciência da computação. Associando conhecimentos de áreas afins como computação gráfica, processamento de imagens e visão computacional, a RA se propõe a realizar a inserção de elementos virtuais em cenas reais em tempo real, como forma de expandir as possibilidades e experiências dos usuários [1].

A realidade aumentada sem marcadores (do inglês *Markerless Augmented Reality*, ou MAR) se caracteriza por executar o rastreamento (*tracking*) da câmera, processo que permite a correta inserção dos elementos virtuais na cena real, utilizando informações naturalmente presentes como arestas [2], texturas [3] ou a própria estrutura da cena [4] sem a inserção de elementos intrusivos (chamados marcadores) ao ambiente.

Devido as suas perspectivas de aplicação nas mais diversas áreas, tais como construção civil, aviação e engenharias [5], as técnicas de MAR têm sido muito estudadas e discutidas tanto pela academia quanto pela indústria.

Neste contexto, este trabalho de graduação se propõe a fazer um levantamento abrangente acerca das técnicas de MAR monoculares, principalmente as baseadas em arestas para, a partir destas observações, realizar um estudo de caso através da implementação de uma técnica baseada na amostragem de pontos a ser usada no rastreamento de objetos com diferentes níveis de complexidade (um cubo real e um edifício 3D virtual). A partir deste estudo de caso será feita uma análise tanto de seu desempenho quanto de sua aplicabilidade.

A técnica escolhida para ser implementada neste trabalho foi uma variação da técnica proposta por Wuest [6] devido às características de tempo real adotadas pela técnica e seu desempenho lidando com situações adversas como o rastreamento de longas seqüências e tratamento de oclusões.

A estrutura da presente monografia segue conforme descrito. No capítulo 2 serão discutidos inicialmente os conceitos básicos de RA, para numa segunda parte apresentar a área de MAR bem como as suas técnicas. No capítulo 3 serão descritas as técnicas baseadas em arestas bem como as aplicações que se originam da utilização das mesmas. No capítulo 4 será detalhado e analisado o estudo de caso juntamente com os resultados obtidos, enquanto que no capítulo 5 serão discutidas as conclusões do trabalho e as previsões de trabalhos futuros.

2. Conceitos básicos

Técnicas de RA têm por objetivo realizar a correta inserção de elementos virtuais em cenas reais em tempo real, de modo a permitir uma melhor interação entre o usuário e o mundo real. Esta melhora ocorre principalmente pela possibilidade que o usuário tem de perceber informações adicionais sobre o ambiente real fornecidas pela interface de RA e aplicá-las na realização das tarefas. Para obter tais resultados são necessários três pilares [7]: rastreamento e registro, que são os processos e técnicas pelas quais o objeto virtual é corretamente posicionado sobre a cena real [8]; tecnologia de exibição, a qual define de que forma o objeto será mostrado ao usuário [8]; e renderização, que trata de como renderizar o objeto virtual de forma a possibilitar, em um mundo ideal, a completa homogeneização visual entre o real e o virtual [9]. Na Figura 1 são mostrados os componentes formadores de um sistema de RA.

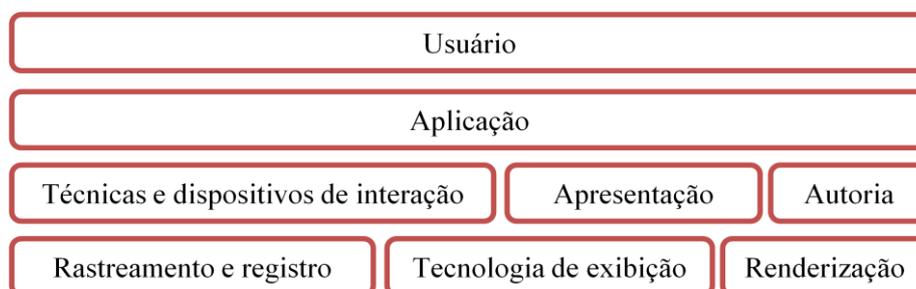


Figura 1. Componentes formadores da RA segundo [7].

Uma vez que a RA foca no melhoramento da interação do usuário com o mundo à sua volta, o nível mais alto de seus componentes formadores denota justamente o usuário. Enquanto isso, a utilização da RA por parte do usuário repousa sobre os ombros das aplicações, que tratam exatamente das formas de realizar tal interação.

De forma a possibilitar a construção de aplicações de RA, três são os estágios que precisam estar desenvolvidos: técnicas e dispositivos de interação, elementos responsáveis por definir de que forma a interação entre o usuário e a aplicação de RA acontecerá; apresentação, uma camada com a função de determinar o modo como a interface de RA será apresentada ao usuário; e as ferramentas de autoria, responsáveis por integrar as bases da RA.

Cabe ressaltar que os maiores desafios ainda se encontram nos estágios mais básicos do processo de construção de um sistema de RA, apesar de diversos avanços obtidos na área [10]. O pilar que denota os maiores desafios na área é o de rastreamento

e registro dos objetos virtuais, uma vez que os dois outros pilares advêm de áreas mais antigas e fundamentadas da ciência da computação como a computação gráfica, mostrando-se assim em estágio mais evoluído.

Observando técnicas de rastreamento e registro existentes na literatura, pode-se dividir as técnicas aplicadas em RA em dois grupos fundamentais: as técnicas baseadas em marcadores e as técnicas sem marcadores.

Técnicas baseadas em marcadores utilizam informações adicionais às cenas reais, inseridas propositamente, para auxiliar e simplificar o processo de rastreamento da câmera [1]. Essas informações adicionais advêm de objetos sintéticos chamados marcadores que possibilitam uma associação entre suas características e a câmera agilizando assim o processo de rastreamento através de algoritmos de processamento de imagens. Para ilustrar o funcionamento geral destas técnicas, pode ser observado na Figura 2 o *pipeline* seguido pelo ARToolkit, que é a biblioteca mais utilizada para realizar o rastreamento de marcadores [11].

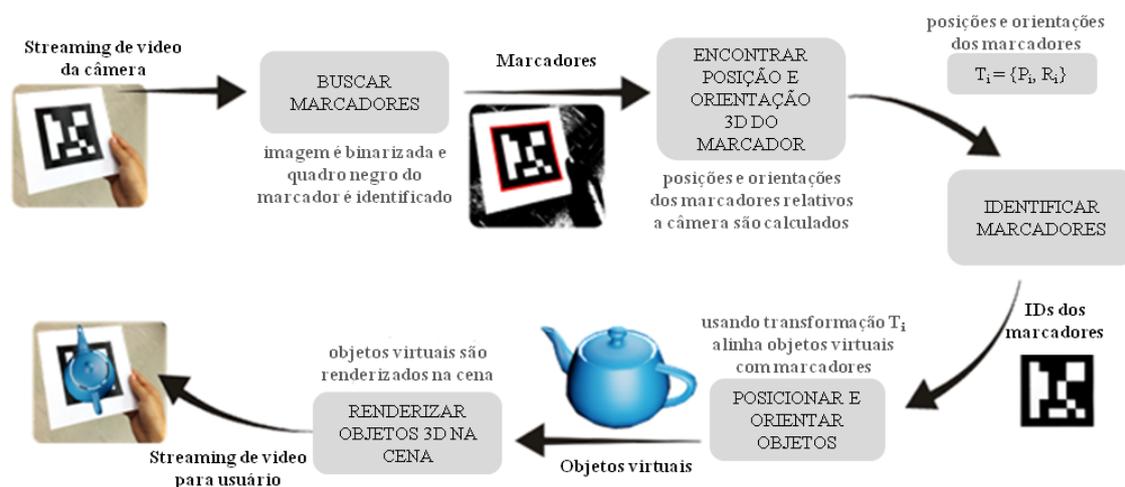


Figura 2. Pipeline típico de realidade aumentada com marcadores.

Apesar de acelerar o processo de rastreamento, as técnicas baseadas em marcadores possuem, entre outros, o ponto fraco de necessitar que o ambiente a ser rastreado seja modificado. Desta forma, essas técnicas possuem como cenários de utilização ambientes controlados onde a inserção dos marcadores se faz possível.

Por outro lado, as técnicas sem marcadores objetivam realizar o rastreamento da câmera através de informações inerentes às cenas, como a relação entre características da cena que se perpetuam ao longo das imagens capturadas pela câmera ou modelos 3D gerados a partir dos objetos presentes na cena, sem a inserção de nenhum objeto sintético. Devido a isto, técnicas sem marcadores mostram-se, apesar de mais

complexas, bastante promissoras devido à gama de aplicações a qual podem ser aplicadas [5][31]. Estas técnicas são o objeto de estudo deste trabalho de graduação e serão discutidas em detalhe na subseção 2.2.

2.1. Rastreamento da câmera

O rastreamento da câmera, aspecto fundamental no processo de rastreamento e registro, advém da recuperação de informações que descrevem corretamente uma câmera virtual para sua posterior utilização na inserção dos objetos virtuais na cena. A câmera virtual pode ser definida através de seus parâmetros intrínsecos e extrínsecos.

Parâmetros intrínsecos são aqueles parâmetros inerentes à câmera e que não dependem de sua movimentação ao longo do tempo. Tais parâmetros dependem exclusivamente de informações como distância focal ou ponto principal da câmera. A matriz que define os parâmetros intrínsecos da câmera, também conhecida como matriz de calibração da câmera é dada por:

$$k = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

onde tem-se que:

- α_x e α_y são proporcionais à distância focal com pesos k_x e k_y definidos pela quantidade de *pixels* por unidade de distância em cada direção. Desta forma α_x e α_y são definidos por $k_x * f$ e $k_y * f$, respectivamente;
- (u_0, v_0) são as coordenadas correspondentes ao ponto principal da câmera;
- s é o coeficiente de *skew* da câmera definindo a relação entre as direções dos eixos do *pixel* da câmera, sendo nulo nos casos em que as direções são perpendiculares.

Por sua característica intrínseca à câmera e não relacionamento com o movimento, diversas técnicas de RA se valem do prévio conhecimento acerca da câmera para garantir que os parâmetros intrínsecos são conhecidos. Diversas técnicas de RA simplificam ainda mais seu *pipeline* ao considerar os parâmetros intrínsecos da câmera fixos, além de conhecidos.

A matriz de parâmetros extrínsecos da câmera é dada em função de sua localização no espaço, correspondente à rotação e translação que deveriam ser feitas

para transformar o sistema de coordenadas de mundo para o sistema de coordenadas da câmera, como pode ser visto na Figura 3.

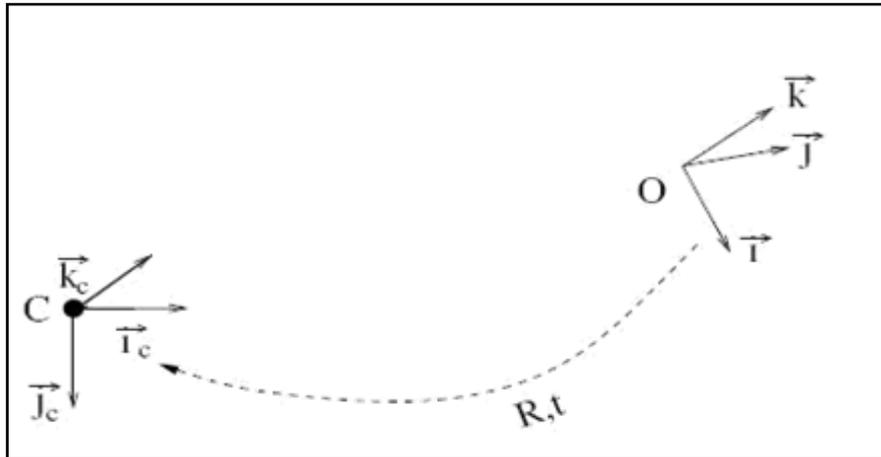


Figura 3. Os parâmetros extrínsecos definem a rotação e translação R, t responsáveis por levar o sistema de coordenadas de mundo O para o sistema de coordenadas da câmera C .

A rotação da câmera é dada por uma matriz 3×3 formada a partir de três vetores correspondentes às rotações dos eixos canônicos, enquanto a translação é definida por uma matriz 3×1 em que cada valor representa o valor da translação ao longo de um desses eixos. Através desta composição obtemos os parâmetros extrínsecos da forma $[R_{3 \times 3} | t_{3 \times 1}]$ que corresponde a uma matriz 3×4 .

Apesar de utilizar a rotação como uma matriz 3×3 na composição geradora dos parâmetros extrínsecos, outras formas de representação são necessárias para realizar partes do processo de rastreamento da câmera como, por exemplo no processo de minimização onde os dados estimados são idealmente representados por vetores. Três representações se mostram completas e satisfatórias na área de RA [12]: Ângulos de Euler, onde a rotação é dada pelo produto de três matrizes correspondentes cada uma à rotação em torno de um eixo canônico; Quaternions, na qual a rotação é representada por um número complexo caracterizado por um vetor de três dimensões e um escalar; e a representação através da fórmula de Rodrigues, na qual a rotação é definida por um vetor correspondente a um eixo de rotação e um ângulo de rotação em torno deste vetor. Neste trabalho em particular serão utilizadas tanto a representação padrão (matriz 3×3 quanto a representação sob a fórmula de Rodrigues).

Através da composição dos parâmetros intrínsecos e extrínsecos é obtida a matriz de pose da câmera virtual $P_{3 \times 4}$ dada por:

$$P_{3 \times 4} = K_{3 \times 3} * [R_{3 \times 3} | t_{3 \times 1}]_{3 \times 4}, \quad (2)$$

que pode ser aplicada aos pontos 3D de forma a gerar os pontos 2D pela projeção em perspectiva, como mostrado na Figura 4.

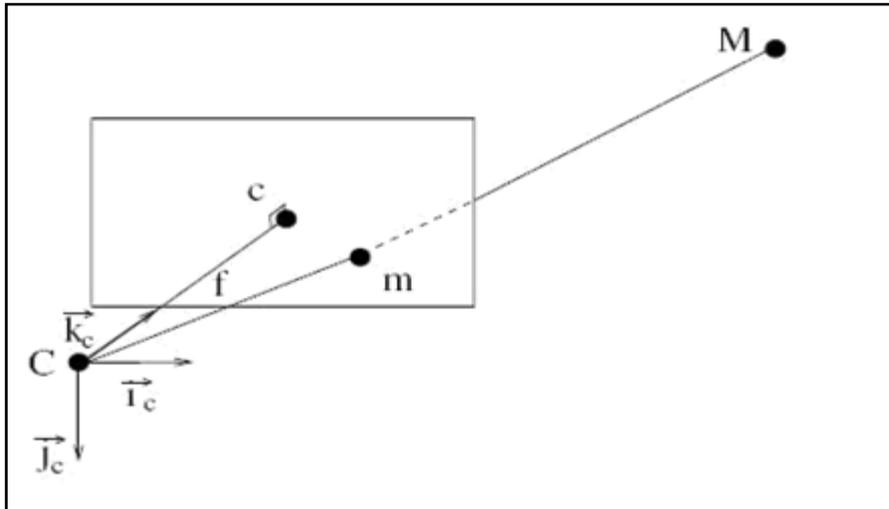


Figura 4. O ponto 3D M pode ser projetado no ponto 2D m através da aplicação da matriz de projeção P oriunda dos parâmetros intrínsecos e extrínsecos da câmera C.

A aplicação da pose $P_{3 \times 4}$ ao ponto 3D M se dá pela equação:

$$s\tilde{m}_{3 \times 1} = P_{3 \times 4} * \tilde{M}_{4 \times 1}, \quad (3)$$

onde $\tilde{m}_{3 \times 1}$ é o ponto $m[x \ y]^t$ escrito em coordenadas homogêneas, o ponto $\tilde{M}_{4 \times 1}$ corresponde ao ponto $M[x \ y \ z]^t$ também em coordenadas homogêneas e s é um fator de escala.

Deste modo, pela Equação 3, a matriz de pose P é considerada pela comunidade científica uma aproximação adequada da matriz de projeção em perspectiva, tanto do ponto de vista de uso quanto matemático uma vez que mantém as propriedades matemáticas em um modelo simples e correto [12].

2.2. Realidade aumentada sem marcadores

Observando as técnicas de MAR monocular, podem-se definir duas grandes vertentes segundo [1]. A primeira delas consiste nas técnicas SfM/SLAM, enquanto a segunda é definida como baseada em modelos (do inglês *Model Based*). A taxonomia de técnicas de MAR monocular pode ser vista na Figura 5.

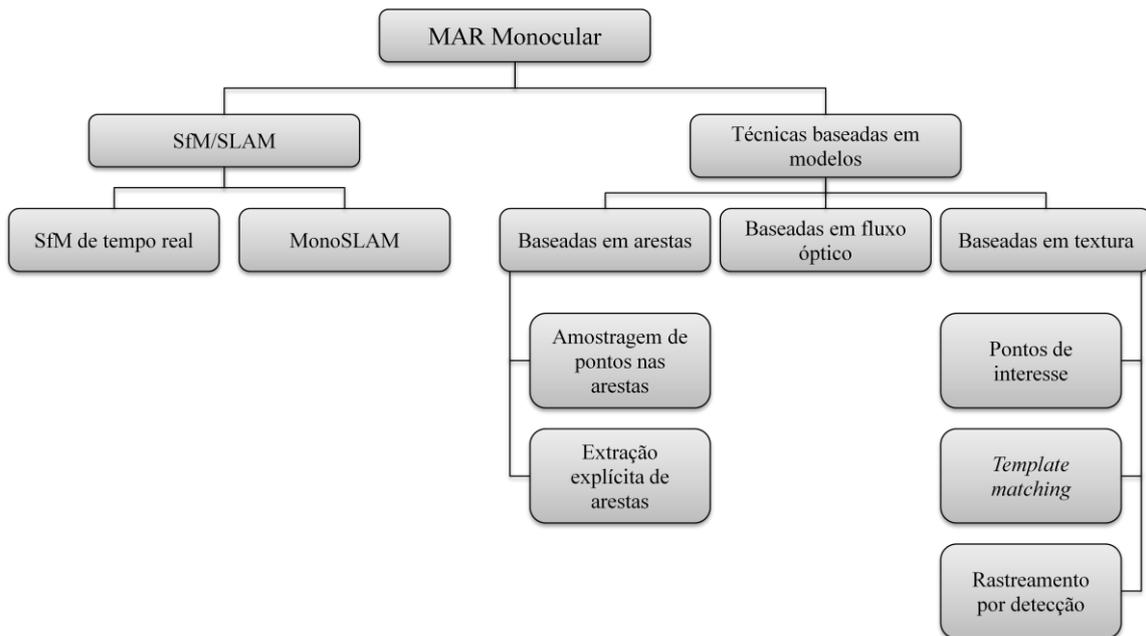


Figura 5. Taxonomia das técnicas de MAR monocular.

Técnicas SfM/SLAM têm como principal característica a recuperação tanto da pose da câmera quanto da estrutura da cena ao longo do rastreamento. Técnicas deste tipo têm a tendência de serem mais complexas do que as técnicas baseadas em modelos já que utilizam apenas as imagens capturadas da cena para recuperar a pose da câmera e a estrutura da cena, sem nenhuma ajuda de conhecimentos previamente adquiridos sobre a mesma.

Por ser uma técnica classicamente utilizada em visão computacional para realizar reconstrução 3D [4], SfM segue um *pipeline* sem preocupações acerca restrições de tempo real. Normalmente, o *pipeline* de SfM é composto das seguintes fases: extração de características, geração de hipóteses da pose da câmera, avaliação e refinamento da pose, auto-calibração, triangulação, estimativa de mapas densos de profundidade e geração do modelo 3D da superfície e do modelo 3D texturizado. Técnicas SfM produzem excelentes resultados em relação à malha final gerada, mas alguns algoritmos presentes em seu *pipeline* exigem muito tempo de processamento e são, portanto, inadequados para aplicações de tempo real. A fim de trabalhar em tempo real, o *pipeline* tradicional de SFM precisa de algumas modificações para eliminar gargalos e acelerar todo o processo. Isto minimiza os atrasos na reconstrução da cena, aproximando-as de reconstrução 3D de tempo real [13].

Desse modo, técnicas SFM podem ser divididas em duas categorias: SFM de tempo real, que ignora ou substitui algumas das fases do *pipeline* clássico de SFM a fim

de torná-lo de tempo real, mantendo ainda restrições acerca da robustez da técnica [14]; e MonoSLAM, do inglês *Monocular Simultaneous Localization and Mapping*, onde as imagens obtidas a partir de uma única câmera são processadas por um *framework* probabilístico que constrói um mapa 3D esparsos das *features* que representam o ambiente [15].

Para a utilização de técnicas baseadas em modelos, ao contrário das técnicas SFM, é necessária a aquisição de conhecimento prévio acerca do mundo real. Este conhecimento é guardado em um modelo 3D, geralmente um modelo CAD, posteriormente usado para estimar a pose da câmera [12].

As técnicas baseadas em modelos podem ser divididas em três tipos principais, dependendo basicamente de como são guardadas as informações nos modelos 3D. São elas: técnicas baseadas em arestas, tema de maior destaque nesta monografia onde a estimativa da pose da câmera é realizada pelo relacionamento entre o *wireframe* do modelo 3D e as informações das arestas da imagem do mundo real [16][30]; técnicas baseadas no fluxo óptico, onde as informações utilizadas para a estimativa da pose advém do movimento relativo entre os quadros da seqüência, e sendo por isso técnicas temporais [17]; e técnicas baseadas em texturas onde são utilizadas as informações de textura dos objetos presentes na cena para realizar o rastreamento [18].

Ainda considerando os três tipos e técnicas baseadas em modelos podem-se subdividir dois deles: as técnicas baseadas em arestas e as técnicas baseadas em textura.

As técnicas baseadas em arestas podem ser classificadas em técnicas baseadas na detecção explícita de arestas e técnicas baseadas na amostragem de pontos nas arestas, subdivisão que será tratada em detalhes no capítulo 3.

Técnicas baseadas em textura, por sua vez, podem ser divididas em três tipos de técnicas: *template matching*, que aplica um modelo de distorção em uma imagem de referência de forma a recuperar o movimento rígido do objeto rastreado [19]; técnicas de pontos de interesse, que levam em consideração o relacionamento entre essas *features* na estimativa da pose da câmera [3]; e rastreamento por detecção, onde *features* invariantes a escala, mudanças de iluminação e pontos de vista são extraídos da imagem a cada quadro, possibilitando assim a estimativa da pose [20].

As abordagens das técnicas de MAR baseadas em modelos devem ser avaliadas de acordo com sua aplicabilidade no cenário em questão. Técnicas baseadas em arestas têm sua utilização mais recomendada em situações em que o objeto a ser rastreado é poligonal ou possui contornos bem definidos, como será tratado no capítulo 3. Se os

objetos são texturizados, as técnicas baseadas em fluxo óptico possuem grande destaque, exceto em situações onde haja mudanças de iluminação ou grande deslocamento de câmera. Quando as técnicas baseadas em fluxo óptico não resolvem o problema para objetos texturizados, as técnicas baseadas em textura mostram-se, geralmente, como uma boa opção uma vez que são mais precisas.

Caso o objeto a ser rastreado seja planar, técnicas *template matching* demonstram bons resultados a um baixo custo computacional por se tratarem de técnicas globais; caso contrário, técnicas baseadas em pontos de interesse devem ser utilizadas. Técnicas de rastreamento por detecção, por sua vez, sofrem com *jitter*, efeito indesejado no rastreamento que ocorre devido à independência entre o quadro atual da cena e o anterior. Este problema é minimizado nas técnicas de *template matching* e pontos de interesse, já que são técnicas recursivas que estimam a pose atual da câmera levando em consideração a pose estimada no *frame* anterior.

Devido às qualidades e limitações das técnicas em cada cenário supracitado, existem diversos trabalhos que tratam da utilização de técnicas híbridas para o rastreamento, como em [29] onde é associada uma abordagem baseada em arestas à uma técnica de pontos de interesse de modo a permitir que a técnica se recupere de falhas no rastreamento.

3. MAR baseada em arestas

Assim como as demais técnicas baseadas em modelos, as técnicas baseadas em arestas necessitam guardar informações *offline* em um modelo 3D previamente adquirido de forma a possibilitar o rastreamento da câmera. Sua denominação advém do tipo de características utilizadas para guardar tais informações e realizar o rastreamento que se dá através das arestas do modelo. No cenário de MAR, as técnicas baseadas em arestas foram as primeiras técnicas desenvolvidas devido a sua alta eficiência computacional e relativa simplicidade de implementação.

Uma vez que utilizam as arestas dos objetos para realizar o rastreamento, as técnicas baseadas em arestas são muito aplicadas em cenários contendo objetos poligonais, ver Figura 6. Apesar desta preferência, tais técnicas também podem ser utilizadas em cenas com objetos de forma livre desde que obedecida a restrição da presença de gradientes fortes nos contornos dos objetos [28]. Além dessas aplicações, as técnicas baseadas em arestas podem, em geral, lidar com problemas dificilmente

solúveis por outras técnicas como mudanças bruscas de iluminação já que em tais situações as arestas geralmente são preservadas, até mesmo quando da presença de objetos especulares.

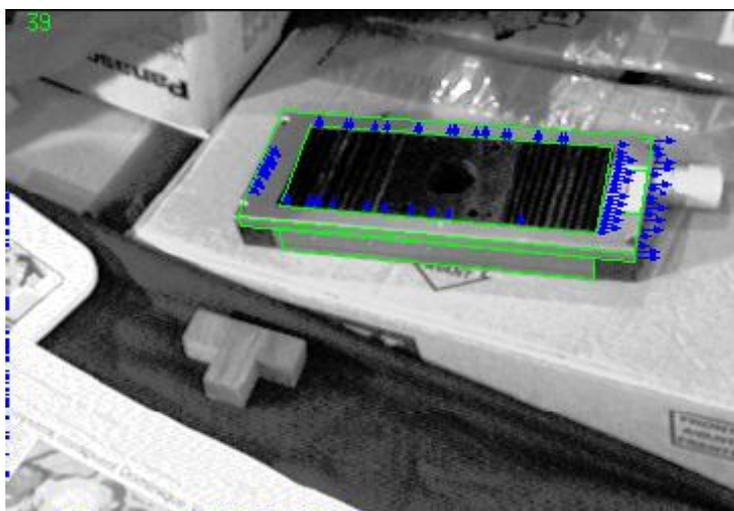


Figura 6. Objeto poligonal rastreado a partir de suas arestas. Figura extraída de [32].

Devido à utilização das arestas no rastreamento, as técnicas desta natureza possuem problemas em situações em que ocorrem movimentos bruscos de câmera. Nesses casos, a projeção do objeto na cena se afasta muito das arestas reais do objeto dificultando a correlação entre as arestas reais e virtuais. Outro fator que atrapalha bastante o rastreamento é a falsa correlação entre as arestas devido a um fundo poluído ou presença de sombras nas imagens.

Tomando-se como base as técnicas baseadas em arestas, tem-se duas classificações possíveis dependentes da forma como as características das arestas são utilizadas para o rastreamento: Técnicas que realizam a extração explícita das arestas, onde as arestas do modelo virtual são extraídas e relacionadas como um todo com as arestas presentes na cena real; e as técnicas baseadas na amostragem de pontos, que se utilizam de um relacionamento entre pontos extraídos nas arestas do modelo e pontos de forte gradiente extraídos da imagem da cena real.

3.1. Extração explícita de arestas

Técnicas baseadas na extração explícita de arestas têm a característica de utilizar a correspondência entre as arestas como um todo para realizar o rastreamento da câmera. Tal correspondência se dá entre as arestas extraídas na cena real e as arestas projetadas do modelo 3D utilizando-se a pose prevista para a cena (estimativa inicial).

Apesar de utilizar preferencialmente arestas poligonais como pode ser visto na Figura 7, técnicas baseadas na extração explícita de arestas podem também utilizar outras curvas paramétricas. Sendo assim, tais técnicas podem facilmente realizar o rastreamento de objetos poliédricos como tetraedros ou cilindros [16].

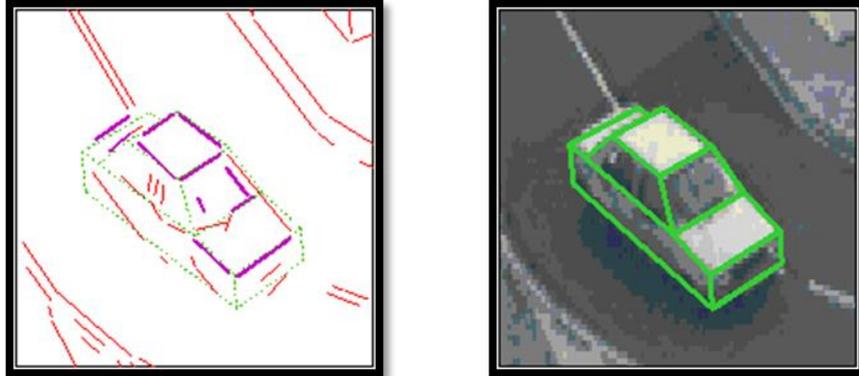


Figura 7. Arestas extraídas da cena exibidas na imagem da esquerda utilizadas para realizar o rastreamento do carrinho demonstrado na imagem da direita pela sobreposição do modelo ao carrinho (linhas cinzas). Figura extraída de [1].

Já que precisa extrair as arestas da cena, técnicas baseadas em extração explícita de arestas precisam de algoritmos de processamento de imagens rápidos e confiáveis para realizar esta tarefa. Algoritmos de detecção de arestas como Harris Corner Detector, extrator de Marr-Hildreth [21] ou algoritmos que se utilizam da transformada de Hough [22] são amplamente utilizados em diversas técnicas vistas na literatura. Técnicas mais rebuscadas de extração de arestas também se mostram interessantes sempre aliado ao cenário em questão [23].

Basicamente, as técnicas de extração de arestas se utilizam de uma abordagem estatística para realizar a correspondência entre as arestas projetadas do modelo e as arestas extraídas das cenas. Devido a isto, a técnica de distância de Mahalanobis mostrasse a mais utilizada pela comunidade [12]. A distância de Mahalanobis denota a diferença entre as arestas através da utilização da matriz de covariância dos dados descritores das mesmas. Por sua vez, a matriz de covariância mede o quanto cada descritor do espaço amostral varia em relação aos demais levando a comparação bastante equilibrada de todos os descritores utilizados.

Na literatura, diversos são os descritores utilizados para definir arestas: O ponto médio do segmento, o ponto definido mais a direita do segmento ou até mesmo sua orientação [24]. Uma vez que a distância de Mahalanobis é utilizada, são compostos vetores em que cada campo corresponde a um descritor de forma a melhor definir cada aresta. Deste modo, tem-se que a distância de Mahalanobis entre duas arestas i e v , uma

extraída da imagem e outra sendo a projeção de uma aresta do modelo respectivamente, dada pelo grau de similaridade entre seus vetores descritores X_i e X_v tem a forma:

$$d = (X_i - X_v)^t (\text{Cov})^{-1} (X_i - X_v)$$

onde X_i e X_v , são vetores de dimensão $m \times 1$ onde m corresponde a quantidade de descritores utilizados e Cov é a matriz de covariância dos dados com dimensões $m \times m$.

Uma vez encontrada a aresta do modelo com a menor distância de Mahalanobis para a aresta projetada, caso essa distância d seja menor do que um *threshold* definido previamente [23] passasse então ao processo de estimativa de pose.

O processo de estimativa da pose geralmente utiliza uma abordagem iterativa para minimizar a diferença entre a aresta projetada e sua aresta correspondente na imagem. Dependendo de uma pose inicial como estimativa, o processo de minimização ocorre iterando nesta pose e alterando seu valor de forma que as projeções das arestas utilizando tal pose tenham o somatório de suas distâncias em relação às arestas correspondentes (o erro) minimizado:

$$\text{erro} = \sum_j (X_i^j - X_v^j(P))^t (\text{Cov}^j)^{-1} (X_i^j - X_v^j(P))$$

onde as arestas M^j do modelo são projetadas na imagem através da pose P resultando nas arestas 2D $X_v^j(P)$ e a distância de mahalanobis de cada uma para suas aresta correspondente X_i^j é recalculada a cada iteração e incluída no erro daquela iteração.

Para realizar a minimização é utilizado o algoritmo Levenberg-Marquardt descrito em detalhe na subseção 4.5.

Além das técnicas que utilizam a correspondência entre as arestas projetadas e as arestas da cena, existe outro tipo de técnica que realiza a estimativa da pose a partir das correspondências entre as arestas projetadas e os gradientes da imagem [16]. Técnicas como estas são limitadas uma vez que não existem garantias de que as arestas do modelo serão correspondentes com os altos gradientes encontrados, além de precisarem de uma boa estimativa inicial para convergir em uma pose correta. Devido a estas restrições, técnicas deste tipo são comumente utilizadas para refinar a estimativa da pose, ao invés de realizarem efetivamente a estimativa.

Por serem adequadas ao tratamento de objetos poliédricos, as técnicas baseadas na extração explícita de arestas têm sido utilizadas no rastreamento de veículos e braços robóticos. No entanto, segundo Lepetit [12], elas tendem a ser substituídas por técnicas

baseadas na amostragem de pontos nas arestas devido à sua estabilidade e abordagem local do problema.

3.2. Amostragem de pontos

Técnicas baseadas na amostragem de pontos têm como característica fundamental a seleção de pontos de controle 3D ao longo das arestas do objeto rastreado. A partir destes pontos é feito um relacionamento dos mesmos com os pontos 2D extraídos das arestas a cada quadro da cena de forma a recuperar as informações de câmera, o que significa que neste tipo de técnica a pose é recuperada a partir do deslocamento 2D na cena dos pontos amostrados nas arestas do objeto.

Devido a sua baixa complexidade computacional, técnicas deste tipo foram as primeiras a obter resultados concretos de tempo real. As técnicas baseadas na amostragem de pontos seguem um *pipeline* simples e a partir deste sofrem modificações de forma a se tornarem mais robustas ou terem seu processo ainda mais acelerado. Um *pipeline* muito utilizado como base por diversos autores é o proposto por Drummond [25]. Este *pipeline* pode ser visto de forma simplificada na Figura 8.

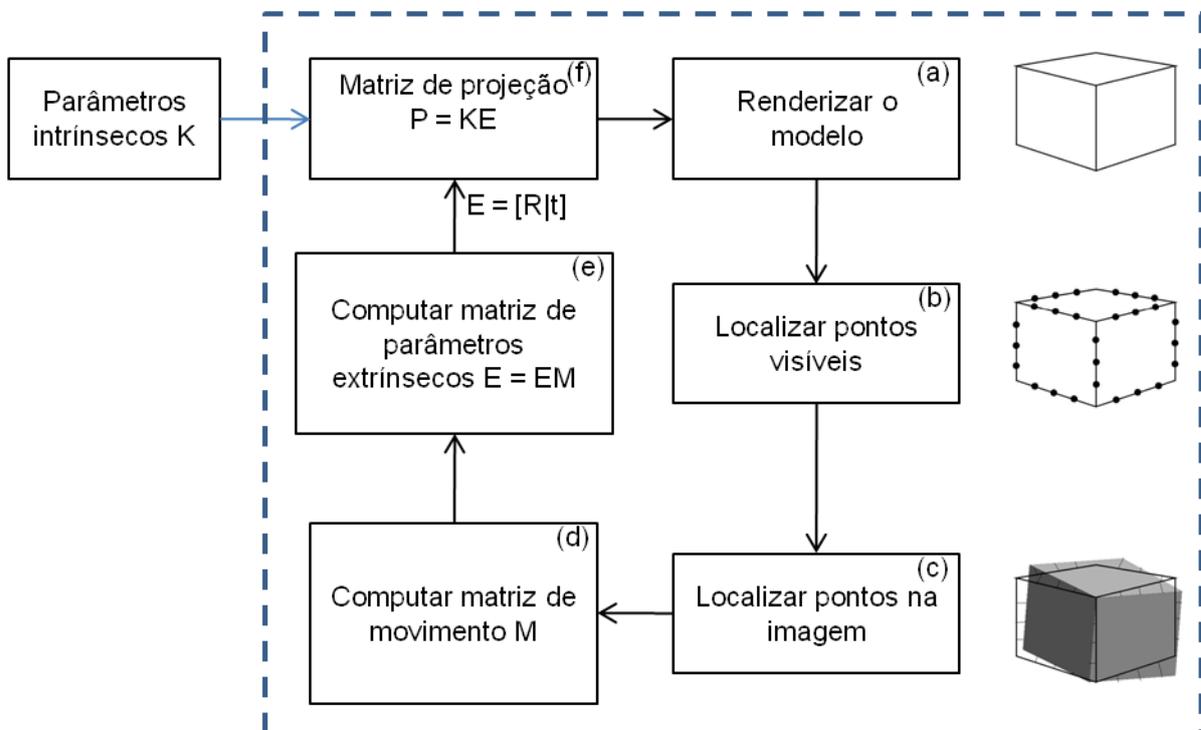


Figura 8. *Pipeline* definido por Drummond [25].

Primeiramente, por se tratar de uma técnica recursiva, uma estimativa inicial da pose da câmera se faz necessária para possibilitar o início do rastreamento. Esta

estimativa pode ser dada tanto por processos automáticos quanto simplesmente por um conhecimento prévio da pose da câmera no início do rastreamento.

A cada iteração da técnica, o modelo 3D que serve de base para o rastreamento é renderizado (a) utilizando a pose estimada no quadro anterior de forma a possibilitar a extração das arestas visíveis. Esta extração pode ser feita utilizando diversas abordagens, desde abordagens em *software* [6] às abordagens em GPUs (do inglês *Graphics Processing Unit*) [27]. No trabalho de Drummond, a renderização é realizada utilizando uma árvore de espaço de partição binária para obter a extração das arestas visíveis (b). Desta forma, a técnica passa a utilizar apenas os pontos visíveis para realizar o rastreamento da câmera, fato que permite sua utilização em tempo real com objetos bastante complexos. A Figura 9 ilustra o objeto 3D a ser rastreado (lado esquerdo) e as arestas visíveis extraídas do modelo (lado direito).

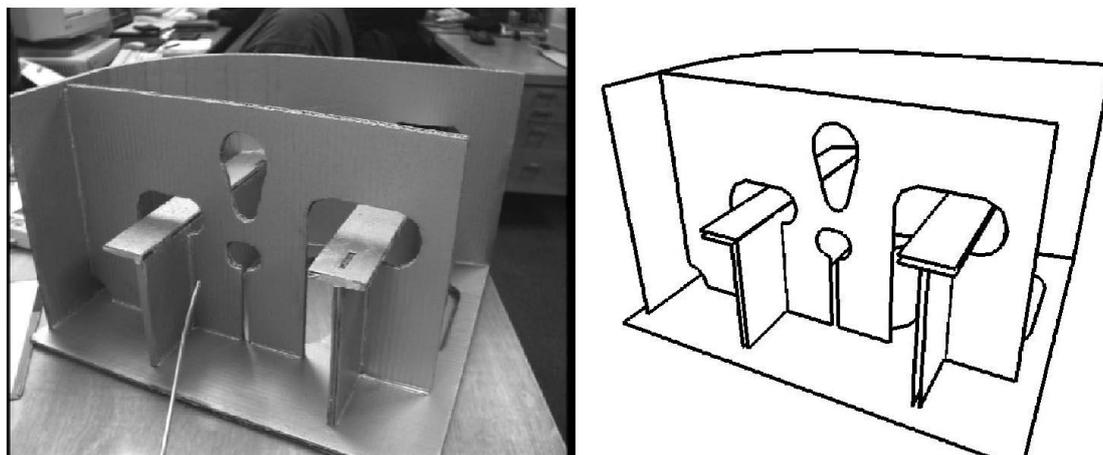


Figura 9. Objeto a ser rastreado. Figura extraída de [25].

Após a identificação dos pontos visíveis, os mesmos precisam ser correlacionados aos pontos de contorno da imagem. Para isto, é feita uma busca ao longo da normal da aresta encontrando, dentro de uma amostragem de geralmente 400 pontos segundo o autor, o ponto correspondente na imagem (c).

Depois de correlacionados, os pontos de controle e os pontos presentes na imagem servirão de base para a estimativa do movimento relativo (d) entre a pose anterior (utilizada para projetar os pontos de controle) e a pose atual (aquela que define os pontos da imagem atual), a partir de um método baseado em álgebra de Lie. Esta estimativa se dá através da utilização da técnica IRLS (*Iterative Reweighted Least Squares*) que funciona de forma mais robusta que a abordagem padrão de mínimos quadrados, pois utiliza vários critérios adicionais à minimização como, por exemplo,

um *m-estimator* que permite que em cada iteração o resultado seja refinado, dando mais ênfase a correspondências corretas e menos ênfase a correspondências erradas (*outliers*).

Basicamente, Drummond define que o movimento relativo pode ser utilizado para encontrar a matriz euclidiana de parâmetros extrínsecos (e) que determina a pose atual após ser composta com os parâmetros intrínsecos da câmera no último passo do algoritmo (f). Após a estimativa da pose atual, a matriz euclidiana é sempre guardada para ser utilizada nas estimativas futuras.

A técnica definida por Drummond obtém resultados coerentes e robustos, se mantendo estáveis mesmo em imagens com ruídos ou com fundo poluído. Além disto, esta técnica se mostra bastante precisa, sendo utilizada inclusive no rastreamento de braços articulados e cenas com diversos objetos complexos, como pode ser observado na Figura 10.

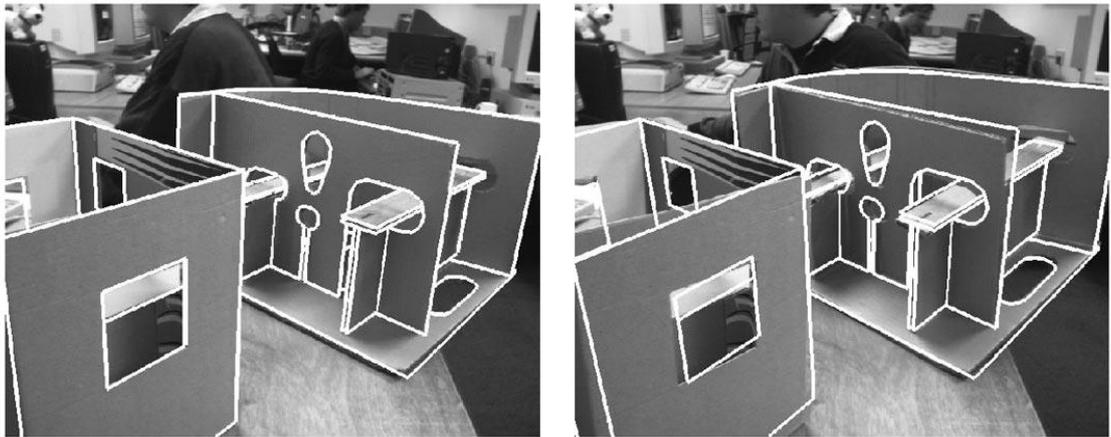


Figura 10. Rastreamento utilizando partes de mais de um objeto. Figura extraída de [25].

Uma variação interessante do *pipeline* de Drummond é descrita por Wuest [6]. Neste trabalho são feitas simplificações e otimizações do trabalho de Drummond de forma a deixá-lo robusto ainda que eficiente permitindo sua utilização em máquinas mais simples que as *workstations* do trabalho de Drummond. Além disto é proposta uma abordagem adaptativa denotando resultados bastante estáveis mesmo em seqüências longas que a princípio são problemáticas pela recursividade presente neste tipo de técnica. A técnica a ser discutida tem o *pipeline* definido na Figura 11.

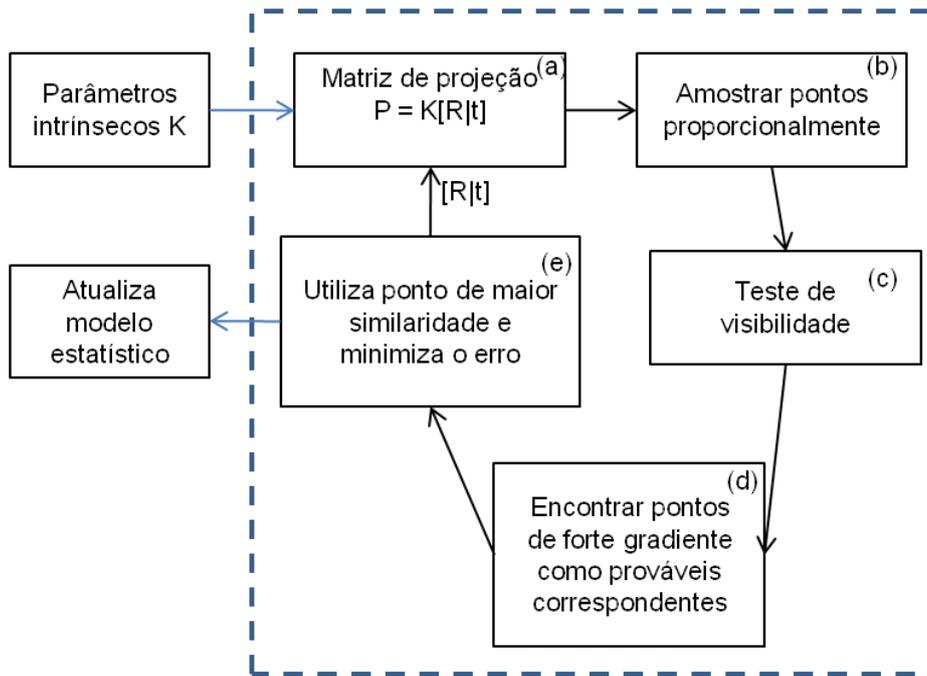


Figura 11. Pipeline definido por Wuest.

Segundo o fluxo seguido por Wuest, primeiramente são extraídas as arestas do objeto virtual. Utilizando então a pose anterior, no primeiro quadro são utilizados marcadores para encontrar a pose inicial, é estimada uma medida da quantidade de pontos que devem ser amostrados na aresta de forma a ser utilizada no *pipeline* uma quantidade de pontos proporcional ao tamanho da aresta projetada (b). Após definir a quantidade de pontos a serem amostrados, é então realizado um teste de visibilidade utilizando a placa gráfica para selecionar apenas os pontos visíveis na cena de forma a minimizar a influência negativa de tais pontos criando assim um vetor V_i de pontos visíveis (c).

Para cada ponto de V_i é realizada sua projeção na cena e é feita uma busca na imagem de forma a encontrar pontos de forte gradiente que possam ser utilizados como correspondentes do ponto amostrado (d). Esta busca ocorre posicionando uma máscara gaussiana bidimensional ao longo da perpendicular à aresta que contém o ponto. A máscara gaussiana por sua vez é escolhida dentro um conjunto de 180 possíveis variações pré-computadas correspondentes a 180 ângulos possíveis de forma a eliminar o impacto computacional da geração destas máscaras a cada quadro da seqüência. A aplicação da máscara gaussiana pode ser observada na Figura 12.

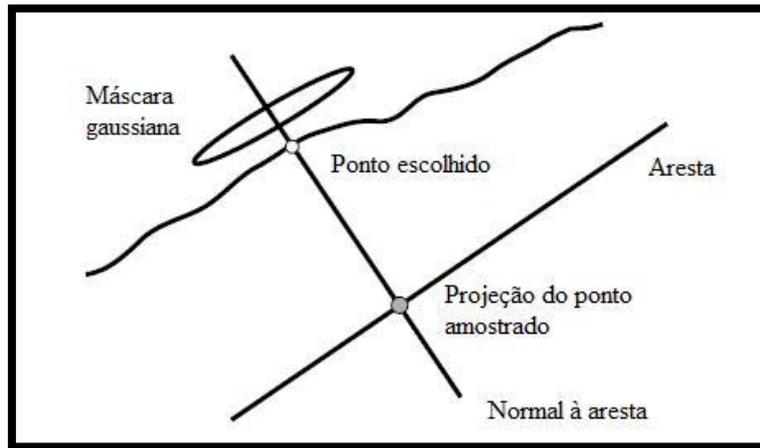


Figura 12. Aplicação da máscara gaussiana ao longo da normal à aresta projetada na busca de uma estimativa para o ponto amostrado na aresta.

Após encontrar as correspondências entre os pontos amostrados na aresta 3D e os pontos de forte gradiente pertencentes à imagem, o algoritmo realiza então a estimativa da pose com base em uma abordagem de múltiplas hipóteses adaptativa (e) utilizando um algoritmo de minimização não-linear chamado Levenberg-Marquardt [26]. O termo múltiplas hipóteses é utilizado uma vez que são guardados mais de um ponto como correspondente do ponto amostrado e a decisão de qual utilizar é postergada de forma a utilizar o ponto de gradiente mais forte que se encontra mais próximo a projeção do ponto a cada iteração na minimização da pose, veja seção 4.5.2 para mais detalhes da utilização de múltiplas hipóteses. O termo adaptativa diz respeito a utilização de um modelo estatístico simples utilizado para guardar as informações temporais do rastreamento e que é atualizado a cada minimização através da distribuição de probabilidades gaussiana [6].

A minimização, que se dá através do algoritmo Levenberg-Marquardt, corresponde a um processo iterativo que vai, a partir de uma estimativa inicial, convergindo para a solução que minimiza o erro do sistema, neste caso o erro de reprojeção (comparação da projeção do ponto 3D amostrado e seu correspondente na imagem) retornando como solução a estimativa da pose da câmera. Para mais detalhes observar seção 4.5.

Na Figura 13 pode ser observado o resultado da aplicação da técnica definida por Wuest para o rastreamento de um objeto complexo mostrando-se robusta a grandes oclusões graças a utilização da abordagem adaptativa.

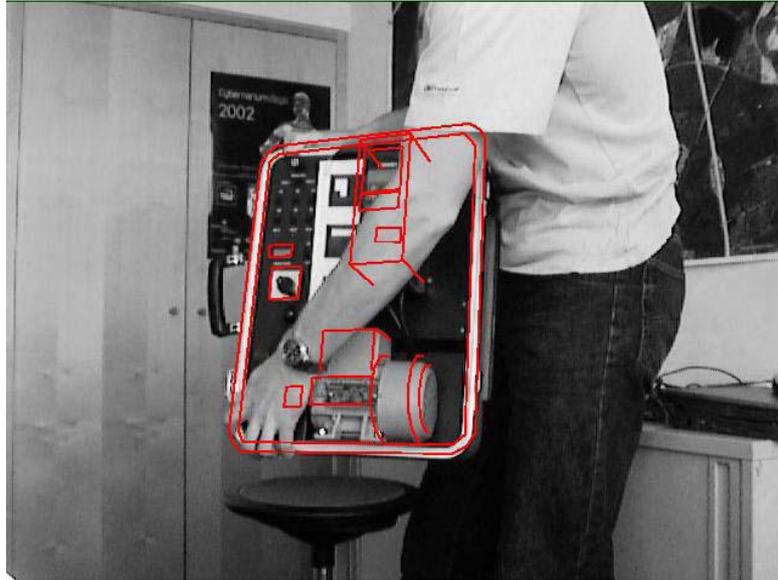


Figura 13. Rastreamento realizado com sucesso através da utilização da técnica de Wuest em objetos complexos com oclusão. Figura extraída de [6].

4. Descrição da técnica

Como mencionado anteriormente a técnica implementada neste trabalho é uma técnica recursiva baseada na amostragem de pontos nas arestas de um modelo 3D previamente construído que serve para guardar as informações 3D a serem relacionadas com informações da cena a cada quadro de forma a possibilitar o rastreamento da câmera e a com isso implementar realidade aumentada sem marcadores. Esta técnica pode ser considerada uma variação da técnica proposta por Wuest [6] e seu *pipeline* é demonstrado na Figura 14.

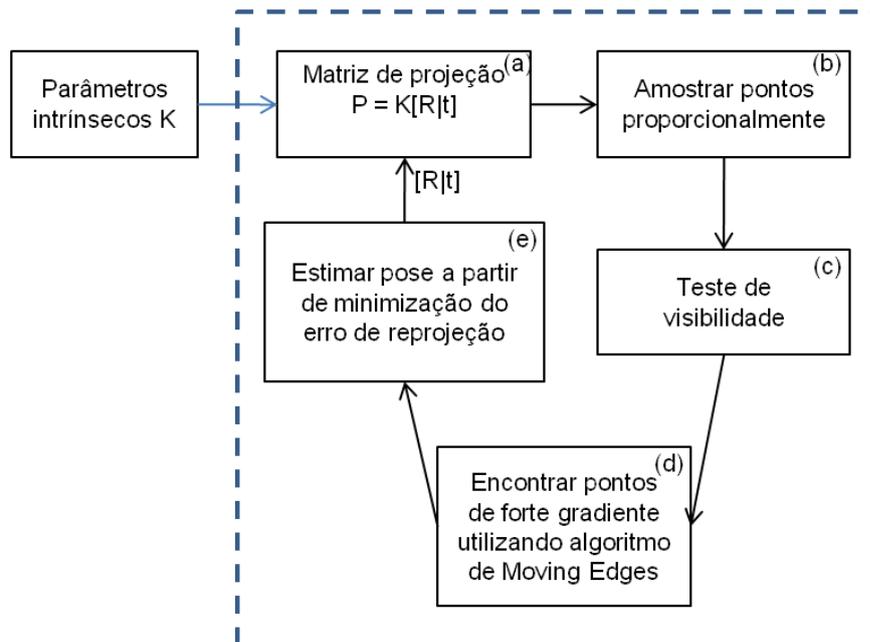


Figura 14. *Pipeline* seguido pela técnica implementada nesta monografia.

De forma a obter o rastreamento do objeto 3D na cena real, primeiramente faz-se necessário o conhecimento prévio da pose inicial da câmera (a). Isto implica ter um prévio conhecimento dos parâmetros intrínsecos e extrínsecos da câmera obtidos numa fase *offline* do rastreamento. Já que a técnica implementada tem a precondição de possuir parâmetros intrínsecos fixos e conhecidos, o problema inicial passa a ser a descoberta dos parâmetros extrínsecos. Neste trabalho não houve abordagem de forma a tratar o problema da inicialização automática e a mesma foi realizada manualmente posicionando a câmera de forma a haver um enquadramento entre um objeto reprojetoado utilizando a pose considerada inicial e o objeto presente na cena a ser utilizado no rastreamento como pode ser visto na Figura 15 onde em (a) a câmera é manualmente deslocada até que seja corretamente localizado o modelo 3D renderizado sobre o objeto 3D real e em (b) onde a cena é sintética e o objeto é inicialmente renderizado utilizando a pose definida como inicial.



Figura 15. Inicialização manual do rastreamento para testes com dados reais e sintéticos.

Ao longo da fase *online* do algoritmo, primeiramente se faz necessário realizar a amostragem dos pontos das arestas do modelo 3D previamente construído em coordenadas de mundo. Essa amostragem é feita de forma a deixar os pontos proporcionalmente espaçados de acordo com o tamanho das projeções das arestas (b), utilizando sempre a pose do quadro anterior que na primeira iteração é a estimativa inicial da pose, para melhorar a qualidade do rastreamento (ver seção 4.1).

Uma vez concluída a amostragem, é necessário realizar a extração das arestas invisíveis (c) de forma a acelerar o processo de estimativa da pose e reduzir a presença de *outliers*, que são correspondências errôneas que prejudicam o algoritmo de estimativa dos parâmetros da câmera. A extração das arestas invisíveis é realizada em hardware gráfico para deixar o processo mais rápido e preciso também utilizando a pose

do quadro anterior para realizar sua operação, como pode ser visto detalhadamente na seção 4.2.

Conhecidos os pontos do objeto 3D que são visíveis utilizando a projeção a partir da pose do quadro anterior, é então necessário encontrar os casamentos entre os pontos amostrados no objeto 3D e a imagem da cena rastreada (d). Este passo é feito utilizando o algoritmo Moving Edges tanto para utilização de uma única hipótese quanto para utilização de múltiplas hipóteses (ver seção 4.3). A última etapa da técnica descrita e aplicada neste trabalho tem por objetivo realizar a estimativa dos parâmetros de câmera (e) através da utilização dos casamentos encontrados pelo Moving Edges. Como pode ser visto mais detalhadamente na seção 4.5, esta estimativa se dá através da minimização da distância entre o ponto 3D projetado com a pose atual e o ponto correspondente na imagem sendo por isso chamada minimização do erro de reprojeção. Esta minimização se dá através da utilização de um estimador não linear chamado Levenberg-Marquardt já mencionado anteriormente.

4.1. Amostragem de pontos

Para realizar a amostragem de forma mais equilibrada, faz-se importante a utilização de um coeficiente de amostragem aplicado na amostragem das arestas projetadas. Este coeficiente é responsável por permitir que a amostragem se dê nas arestas mantendo fixa a quantidade de pontos por unidade de comprimento das arestas projetadas, obedecendo à equação:

$$\text{qntPontos} = \text{tamanhoProj} \times \text{coefDeAmostragem} \quad (1)$$

Realizada desta forma, a amostragem de pontos se mostra uniforme, equilibrando a influência das arestas de acordo com seu tamanho projetado. Deste modo, é considerada a real importância dos pontos projetados na imagem e não apenas do modelo 3D virtual. Uma aresta virtual grande pode ter uma projeção de tamanho pequeno, o que implica em menos pontos amostrados, enquanto uma aresta pequena pode ter um grande comprimento, contribuindo com mais pontos para a técnica (ver Figura 1).

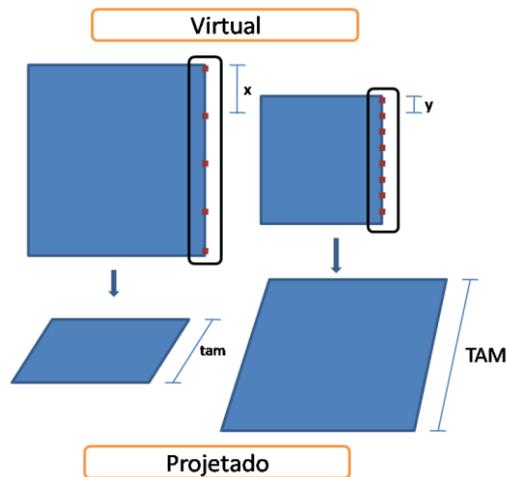


Figura 16. Distância entre pontos amostrados nas arestas (x , y) varia de forma a manter a quantidade de pontos proporcional ao tamanho da aresta projetada (tam , TAM).

4.2. Teste de visibilidade

Nas técnicas de amostragem de pontos, a auto-occlusão, que ocorre quando partes do objeto ocludem partes do próprio objeto, representa um aspecto importante na qualidade do rastreamento. Uma vez que pontos amostrados não possuem correspondentes na imagem do quadro rastreado por serem ocluídos, eles se tornam *outliers* que irão influenciar negativamente a estimativa da pose.

Existem várias formas de minimizar a influência de outliers, como a utilização de técnicas estatísticas como o RANSAC (RANDOM SAMPLE CONSENSUS) ou estimadores robustos como o tukey m-estimator [12]. Porém antes de minimizar a influência dos mesmos, a eliminação de arestas invisíveis se mostra fundamental uma vez que já realiza a eliminação de *outliers* antes mesmo do processo de estimativa iniciar.

Como forma de realizar a eliminação dos pontos ocluídos na imagem foi utilizada uma abordagem em hardware gráfico de forma a deixar o processo menos custoso computacionalmente.

Para que isto fosse possível foi utilizada a extensão de OpenGL chamada `GL_ARB_OCCLUSION_QUERY` de forma a realizar o teste de visibilidade de cada ponto amostrado na aresta. O algoritmo implementado se comporta de acordo com o *pipeline* a seguir.

Primeiramente, se faz necessária a projeção de todo o objeto virtual através do uso da pose anterior, uma vez que a mesma pode ser considerada uma boa estimativa para o quadro atual. Após esta projeção é realizada a projeção de cada ponto

individualmente para que as *queries* sejam avaliadas em hardware gráfico e guardem o estado de visibilidade do ponto.

Após esta avaliação de todos os pontos, as arestas são varridas uma a uma e, enquanto os pontos são visíveis, a aresta vai sendo validada. A partir do momento em que algum ponto da aresta não é visível a mesma é considerada concluída e é adicionada à lista de arestas.

Uma vez que a oclusão de parte da aresta não implica necessariamente na invisibilidade de todo o resto da aresta, os pontos seguintes ainda são avaliados e caso sejam visíveis, outra aresta é criada de modo semelhante ao ocorrido inicialmente (ver Figura 17).

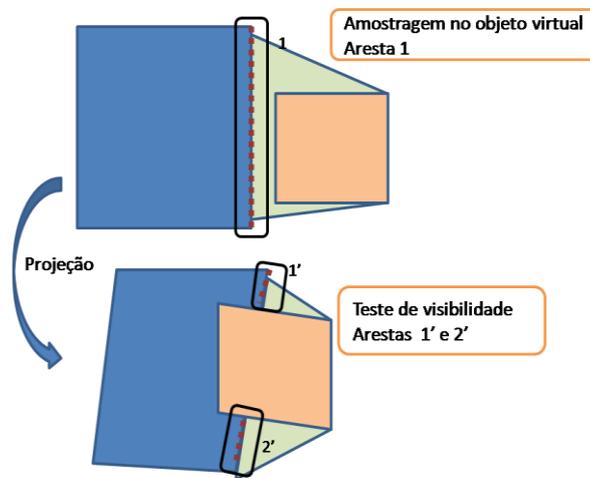


Figura 17. A aresta 1 é dividida nas arestas 1' e 2' devido à auto-occlusão após projeção

Deste modo, uma única aresta pode dar origem a várias arestas de forma a apenas utilizar os pontos visíveis para a estimativa da pose, minimizando assim o problema de auto-occlusão sem prejudicar o rastreamento.

4.3. Casamento entre pontos com Moving Edges

Uma vez concluída a fase de eliminação das arestas invisíveis o sistema está apto a iniciar a fase de casamento. Nesta fase os pontos amostrados no objeto virtual 3D são correlacionados com pontos da imagem de forma a obter uma estimativa para sua posição na cena.

A correlação entre pontos do objeto virtual e pontos da imagem é realizada a partir do algoritmo Moving Edges. Este algoritmo realiza a busca através do *pipeline* descrito a seguir.

Inicialmente a aresta é projetada na cena utilizando a pose anterior como estimativa e os pontos são amostrados. Após esta projeção o algoritmo Moving Edges realiza uma busca na reta que passa pelo ponto amostrado e é perpendicular à aresta projetada, de forma a encontrar pontos de forte gradiente [32].

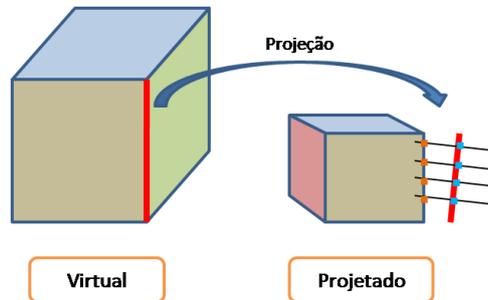


Figura 18. A aresta vermelha é projetada e os pontos na cor azul são amostrados. Através do Moving Edges os pontos laranja são definidos como correspondentes.

Como resultado da busca perpendicular por pontos de forte gradiente, o algoritmo determina os pontos de gradiente mais alto que um limiar definido na implementação do algoritmo mais próximos da projeção da aresta (onde se lê “mais próximos” leia “pertencentes ao intervalo de busca predefinido na implementação”) como possibilidades para o casamento do ponto virtual a ser utilizado na fase de estimativa da pose. O limiar utilizado neste trabalho e o intervalo de busca serão descritos na subseção xx.

Dependendo da abordagem escolhida para a estimativa de pose, com hipótese única ou múltiplas hipóteses o algoritmo de casamento retorna o ponto de gradiente mais forte entre os mais próximos da projeção do ponto virtual ou uma quantidade predefinida de pontos (quantidade de hipóteses) próximos a projeção de tal ponto pertencentes à reta de busca perpendicular a aresta. As utilizações de hipótese única ou múltiplas hipóteses serão detalhadas na próxima subseção.

4.4. Parametrização da pose

Para a técnica descrita neste trabalho, foram considerados arranjos em que os parâmetros intrínsecos da câmera são fixos e conhecidos, deixando como problema a ser resolvido a estimativa dos parâmetros extrínsecos a cada quadro.

Uma vez que os parâmetros extrínsecos da câmera podem ser definidos como a composição $[R|t]$ onde R representa a matriz de rotação 3×3 e t representa a matriz de translação 3×1 foi utilizada na minimização a representação da rotação pela fórmula de

Rodrigues de modo a reduzir o número de variáveis a serem minimizadas (inicialmente doze, sendo nove da rotação e três da translação).

Utilizando a fórmula de Rodrigues, como mencionado na seção 2.1, a rotação é representada por um vetor 3×1 que define o eixo de rotação e seu módulo representa o ângulo de rotação. Esta abordagem reduz o número de variáveis de doze para seis sem, no entanto, reduzir a precisão ou qualidade da rotação.

Além da representação de Rodrigues ser coerente, sua inversa também é necessária, pois para a geração da matriz de projeção é preciso conhecer a rotação na forma de matriz 3×3 :

$$P_{3 \times 4} = K_{3 \times 3} * [R_{3 \times 3} | t_{3 \times 1}]_{3 \times 4} \quad (3)$$

4.5. Estimativa da pose

Através da utilização dos casamentos encontrados através do Moving Edges torna-se possível realizar a estimativa da pose da câmera.

Esta estimativa se dá utilizando a técnica de minimização não-linear Levenberg-Marquardt, usando como critério para a minimização o erro associado à distância euclidiana entre a projeção do ponto virtual e o ponto correspondente ao mesmo. Por tratar-se de uma técnica recursiva, o Levenberg-Marquardt necessita de uma boa estimativa inicial para funcionar corretamente. Ao longo da seqüência, a cada quadro, é utilizada a pose do quadro anterior como estimativa inicial para a pose do quadro rastreado pois se considera um deslocamento pequeno entre os quadros da seqüência.

Dependendo da escolha do ponto correspondente para utilizar na minimização têm-se duas abordagens possíveis: hipótese única, que utiliza o ponto de gradiente mais forte, ou múltiplas hipóteses, na qual se tem um número máximo de pontos de forte gradiente e a cada iteração do algoritmo se utiliza aquele mais próximo da projeção do ponto virtual casado.

4.5.1. Hipótese única

Quando se escolhe trabalhar com uma única hipótese para casar com o ponto virtual, o algoritmo faz a busca utilizando o Moving Edges retornando apenas o ponto de mais forte gradiente encontrado na busca perpendicular à aresta projetada como casamento do ponto virtual. Esta abordagem implica em problemas quando da

aproximação de objetos de forte gradiente às arestas do objeto utilizado no rastreamento como será discutido na seção de Estudos de caso.

Como pode ser visto na equação seguinte, o erro de reprojeção minimizado a cada quadro e a cada iteração do Levenberg-Marquardt é dado por:

$$r = \sum_{i=0}^n (\text{dist}(P * M_i, m_i)) \quad (4)$$

onde P é a matriz de projeção, M_i é o ponto virtual 3D amostrado na aresta do objeto virtual, m_i é o ponto correspondente na imagem encontrado pelo Moving Edges e dist é a função que computa a distância euclidiana entre $P * M_i$ e m_i .

A minimização ocorre então a cada iteração do Levenberg-Marquardt modificando a pose até que algum critério de parada do algoritmo seja atingido. Estes critérios podem ser número máximo de interações, erro relativo entre uma estimativa e outra ou erro total. Um equilíbrio entre estes parâmetros define a qualidade do rastreamento.

4.5.2. Múltiplas hipóteses

De forma a permitir uma minimização mais equilibrada pode ser utilizada a abordagem de múltiplas hipóteses onde vários pontos de forte gradiente são passados como casamento para o ponto virtual 3D projetado e a escolha de qual utilizar na minimização muda a cada iteração do Levenberg-Marquardt.

Esta mudança se dá utilizando a cada iteração o ponto mais próximo do ponto virtual projetado, levando a um modelo em que o erro de reprojeção sempre é o menor possível considerando as hipóteses utilizadas. Este fator denota um algoritmo menos sensível a aproximação de objetos de forte gradiente ao objeto rastreado uma vez que mesmo tendo gradiente mais forte, a projeção da aresta na maioria dos casos está mais próxima do objeto rastreado que de outros objetos próximos.

A fórmula para o erro a ser minimizado passa a ser:

$$r = \sum_{i=0}^n \left(\text{dist}(P * M_i, \text{mindist}(m_i^j, P * M_i)) \right) \quad (5)$$

onde $\text{mindist}(m_i^j, P * M_i)$ determina a escolha do ponto m_i^j que está mais próximo da projeção $P * M_i$.

5. Estudo de caso

De forma a validar a técnica implementada neste trabalho fez-se necessária a elaboração de um estudo de caso detalhado acerca do seu funcionamento. Foram testadas três aplicações de rastreamento, uma utilizando um objeto simples numa seqüência de dados reais e outras duas seqüências sintéticas de um objeto complexo formado por muitas arestas. A matriz de parâmetros intrínsecos da câmera utilizada nas aplicações foi:

$$K = \begin{bmatrix} 350.47574 & 0.00000 & 158.25000 \\ 0.00000 & 363.04709 & 120.75000 \\ 0.00000 & 0.00000 & 1.00000 \end{bmatrix}$$

A seqüência mais simples de dados reais foi capturada realizando a movimentação de um cubo, com suas faces pintadas de cores diferentes denotando diversos níveis de mudança de gradiente com relação ao meio. Sua manipulação ocorreu com as mãos do usuário do sistema e foram feitos testes comparando a qualidade do rastreamento com múltiplas hipóteses e com uma hipótese única. O modelo utilizado no estudo de caso pode ser visto na Figura 19.

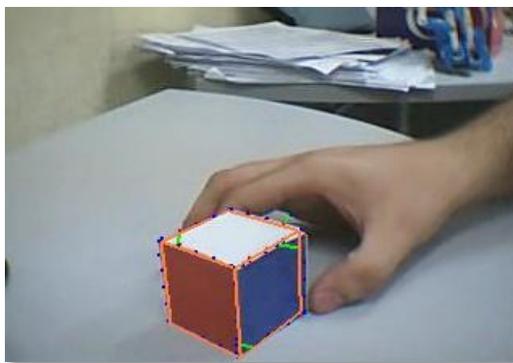


Figura 19. Cubo rastreado no estudo de caso sendo manipulado pelas mãos do usuário.

As seqüências sintéticas se deram através do rastreamento do modelo virtual de um prédio da Universidade Federal de Alagoas – UFAL; uma delas mostra o rastreamento da fachada do prédio e a outra rastreamento o prédio inteiro. Estes casos serviram para averiguar a precisão e eficiência da técnica quando aplicada a um objeto com muitas arestas, onde ocorrem muitas auto-ocluções e a extração de arestas visíveis se torna um gargalo. O modelo virtual do prédio pode ser visto na Figura 20.

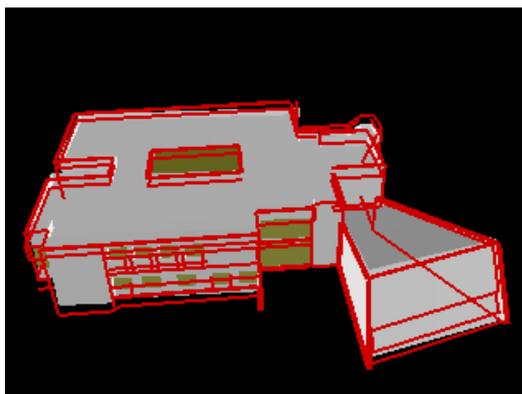


Figura 20. Rastreamento de modelo sintético complexo.

5.1. Implementação

Para o desenvolvimento dos estudos de caso utilizou-se bibliotecas desenvolvidas em C/C++, tanto de computação gráfica, no caso a OpenGL [34], quanto de visão computacional, como a VISP (*Visual Servoing Platform*) [35] e a VXL (*Vision “something” Libraries*) [36], de forma a evitar o trabalho de reimplementar funções básicas destas áreas como operações e dados matriciais que são fornecidas por estas bibliotecas.

A OpenGL foi utilizada na elaboração de operações de projeções e visualizações de resultados, bem como na realização do teste de visibilidade do algoritmo. O teste de visibilidade ocorreu utilizando *queries* oriundas de uma extensão chamada ARB_OCCLUSION_QUERY e foi implementada de forma a tratar o teste isoladamente para cada ponto amostrado do objeto virtual.

Ambas as bibliotecas de visão utilizadas são open source permitindo sua utilização e modificações no código das mesmas. Esta possibilidade foi de grande importância uma vez que elas provêm funções básicas que precisam ser complementadas de forma a realizar funcionalidades mais complexas.

Basicamente, a arquitetura da técnica foi implementada utilizando as funções e estruturas de dados da VXL, a saber os módulos vcl, vnl, vil e vgl. Estes quatro módulos são responsáveis, respectivamente, pelas seguintes funções: o núcleo da biblioteca com funções equivalentes às de C++ com tipos básicos, estruturas de dados e funções sobre estas estruturas; funções e estruturas básicas de álgebra linear e vetorial como matrizes, operações e sistemas lineares; funções e estruturas utilizadas no processamento de imagens de um modo geral; e, por último, as funções de câmera de

forma a realizar as transformações de matrizes de rotação, transformando da representação matricial para a representação de Rodrigues e vice-versa.

A VISP, por sua vez, foi utilizada na implementação do algoritmo Moving Edges, uma vez que já possui uma implementação do mesmo. Apesar disto, foram necessárias diversas modificações nas funções responsáveis pelo Moving Edges de forma a possibilitar a utilização de múltiplas hipóteses pela técnica. As funções que sofreram alterações em seu funcionamento básico foram as correspondentes às classes vpMeEdge e vpMeSite.

Como foram utilizadas na implementação as linguagens C/C++, a aplicação foi desenvolvida utilizando a IDE Visual Studio 2005, desenvolvida pela Microsoft. A aplicação foi executada em um computador *desktop*, com processador Core 2 duo 6600 2.41GHz da Intel tendo 2 GB de memória RAM e placa de vídeo NVidia GeForce 7900 GTX.

5.2. Resultados

No primeiro caso analisado neste trabalho têm-se um cubo sendo movimentado sobre uma mesa plana através da utilização das mãos. Esta aplicação pode ser executada a uma taxa de 30 quadros por segundo (do inglês *frames per second* – fps). Neste teste, à medida que o cubo real é rastreado, um cubo em *wireframe* (arestas em cor laranja) é renderizado sobre o mesmo de forma a possibilitar uma análise visual coerente acerca da precisão do rastreamento.

Por se tratar de uma técnica de tempo real, o ideal era que o *wireframe* se mantivesse durante todo o tempo alinhado com o cubo real. Porém, apesar de mostrar uma boa taxa de fps, a técnica sofreu um pouco com movimentos bruscos do objeto devido à sua característica de recursividade. Uma vez que a estimativa da próxima pose depende da anterior, quando há um movimento brusco a pose atual se afasta muito da anterior e o registro do objeto 3D com sua projeção passa a sucumbir devido às falhas de rastreamento, como pode ser visto na Figura 21.

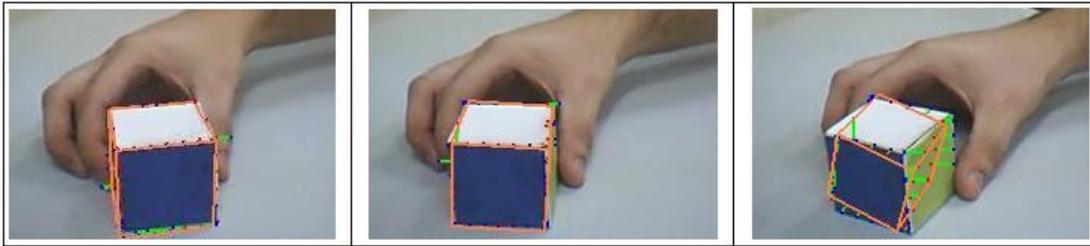


Figura 21. Seqüência de rastreamento do cubo denotando a falha de rastreamento após movimento brusco do mesmo.

Além disto, mesmo não utilizando estimadores robustos, a técnica se mostrou robusta a oclusões parciais resultantes da manipulação do objeto. A Figura 22 ilustra a corretude do rastreamento quando da oclusão de uma aresta do cubo.

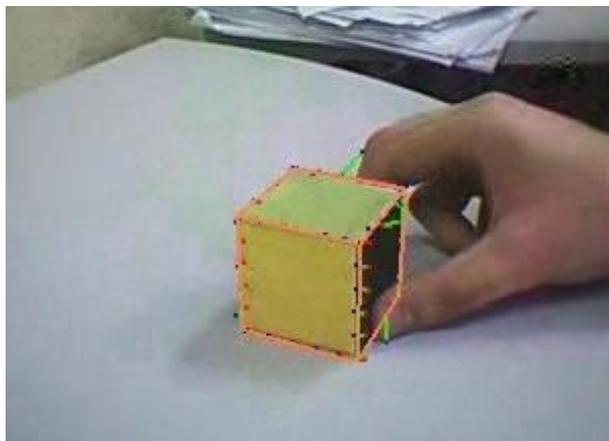


Figura 22. Cubo sendo rastreado mesmo com a existência de oclusão parcial.

A Figura 23 ilustra o problema da utilização de uma única hipótese. Devido a esta utilização, o alto gradiente proporcionado pelos dedos do usuário faz com que alguns casamentos sejam realizados entre o objeto e os dedos, ocasionando a presença de falsas correspondências, como pode ser observado pelas linhas (verde) que ligam os pontos amostrados no objeto virtual (vermelho) aos pontos ditos correspondentes (azul).

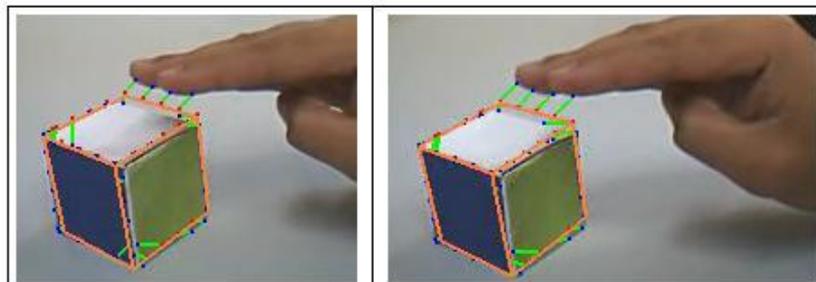


Figura 23: Confusão do Moving Edges devido ao alto gradiente proporcionado pela proximidade dos dedos com o cubo.

Através da utilização de múltiplas hipóteses problemas como estes são minimizados uma vez que a estimativa utiliza a ponderação de várias hipóteses para

realizar o rastreamento ao invés de usar apenas o maior gradiente, provendo uma maior estabilidade à técnica.

Como pode ser visto na Figura 24 quando não são utilizadas múltiplas hipóteses o rastreamento sofre falhas bastante acentuadas devido a aproximação de um objeto de forte gradiente. Na Figura 25 pode ser percebida a estabilidade mesmo quando ocorre a aproximação do objeto.

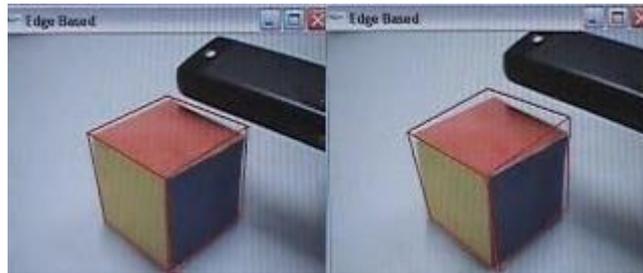


Figura 24. Seqüência sem a utilização de múltiplas hipóteses.

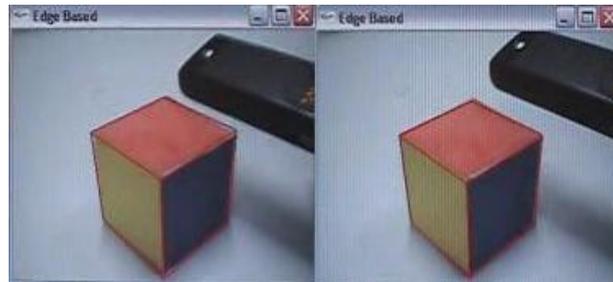


Figura 25. Seqüência utilizando múltiplas hipóteses.

Através das seqüências sintéticas podem ser observados aspectos com relação à robustez da técnica de um ponto de vista estatístico, ao contrário da seqüência de dados reais onde não existia um “*Ground Truth*” para averiguar o quão preciso o rastreamento estava sendo realizado. Foram utilizadas duas seqüências sintéticas diferentes, uma utilizando o modelo completo de um prédio da UFAL e outra utilizando apenas a fachada frontal do prédio, como pode ser visto na Figura 26.

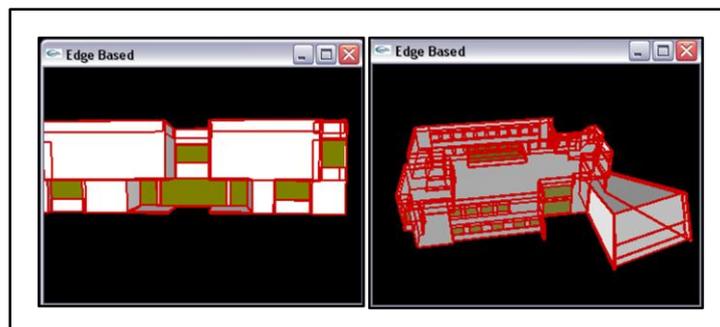


Figura 26. Duas seqüências sintéticas utilizadas para averiguar a precisão da técnica.

A seqüência utilizando apenas a fachada do prédio da UFAL demonstrou certa instabilidade uma vez que possui poucas arestas a serem rastreadas e um movimento não muito suave em alguns momentos. As figuras exibidas abaixo retratam o erro de rastreamento com relação ao posicionamento do centro da câmera para a seqüência exibida na Figura 27, tanto com respeito aos três eixos canônicos quanto ao posicionamento comparado como um todo através da distância euclidiana entre o centro obtido e o centro esperado (*Ground Truth*).

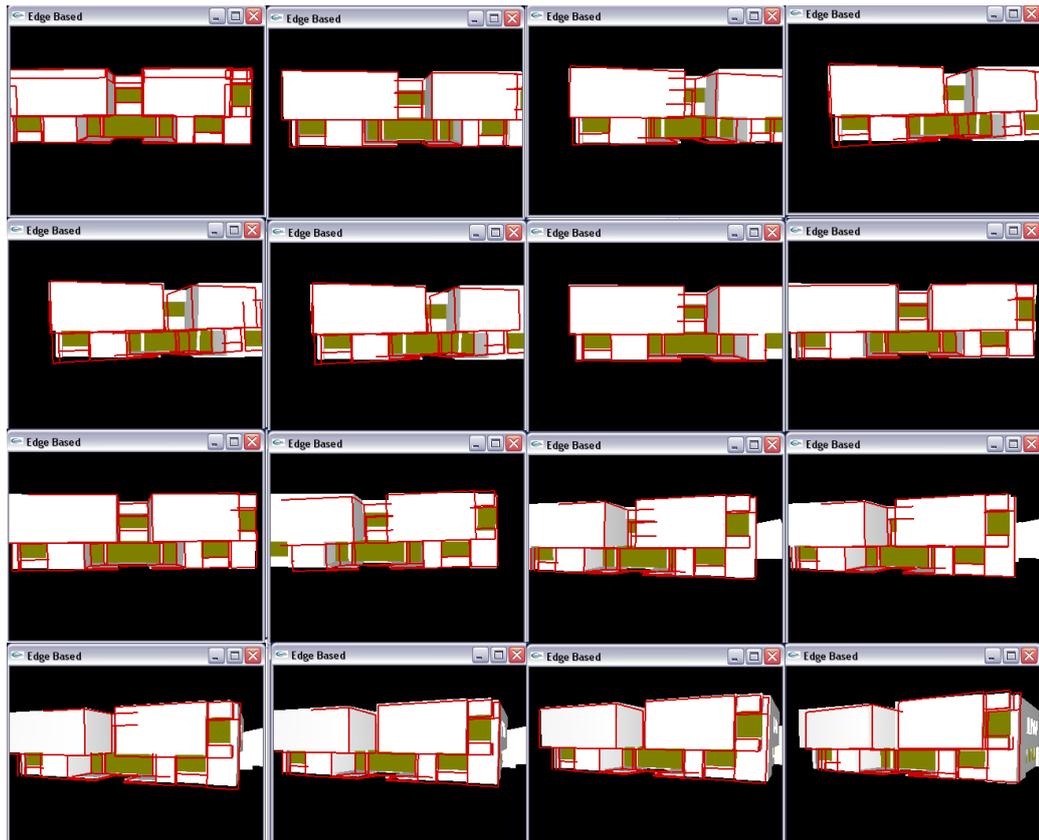


Figura 27. Seqüência da fachada do prédio da UFAL.

Nas figuras 28, 29 e 30 tem-se o erro associado a cada eixo canônico ao longo do rastreamento da seqüência frontal do prédio da UFAL, denotando boa qualidade do rastreamento ao longo dos 145 quadros da seqüência. É importante notar que a dimensão dos dados se dá em milímetros devido aos parâmetros intrínsecos da câmera utilizada e que a falha ocorrida no eixo Z entre os quadros 13 e 49 se deve a não recuperação de uma falha ocorrida no quadro 13 devido a um movimento brusco, pois a câmera permanece parada até o quadro 49, quadro em que a técnica volta a se recuperar.

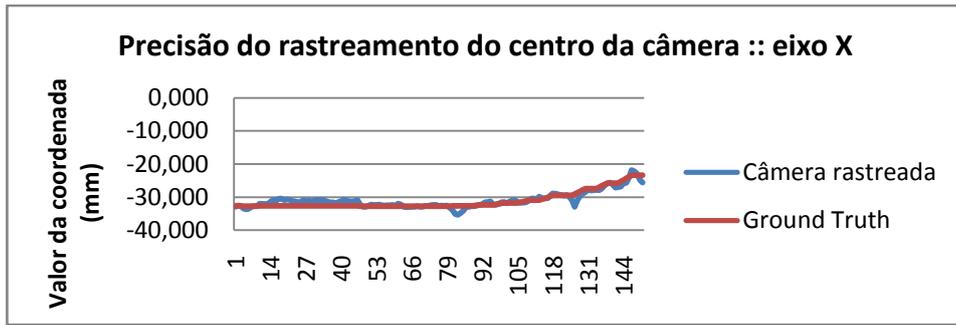


Figura 28. Erro associado ao eixo X da seqüência frontal do prédio da UFAL.



Figura 29. Erro associado ao eixo Y da seqüência frontal do prédio da UFAL.



Figura 30. Erro associado ao eixo Z da seqüência frontal do prédio da UFAL.

No gráfico seguinte é possível observar uma manutenção do erro em um valor acima da média devido a manutenção da câmera parada em uma pose errônea a partir do quadro 13 principalmente no eixo Z. O erro médio desta seqüência atingiu o valor de 15 mm, valor considerado pequeno para este tipo de aplicação [3].

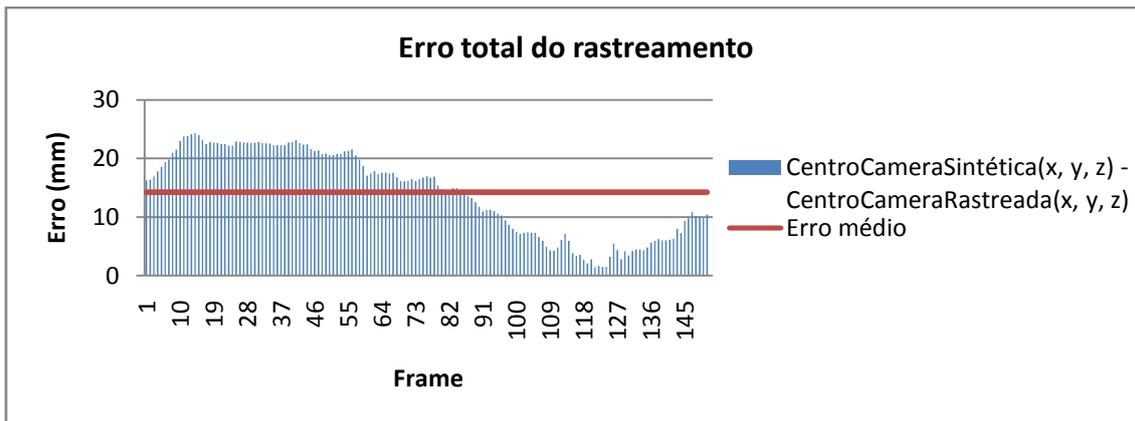


Figura 31. Distribuição do erro total do rastreamento na seqüência frontal.

Na Figura 32 pode ser vista a seqüência do prédio da UFAL completo sendo rastreado ao longo de 469 quadros. Os quadros presentes na figura foram amostrados ordenadamente para transparecer o rastreamento. É importante perceber que o rastreamento se manteve coerente ao longo de toda seqüência de quadros e poucos foram os erros apresentados devido a alta quantidade de arestas que facilitam o rastreamento apesar de influenciar negativamente no desempenho.

Nas figuras 31, 32 e 33 tem-se o erro associado a cada eixo canônico ao longo do rastreamento da seqüência do prédio da UFAL completo. Vale salientar que neste caso o erro se distribuiu uniformemente ao longo de todos os eixos canônicos.

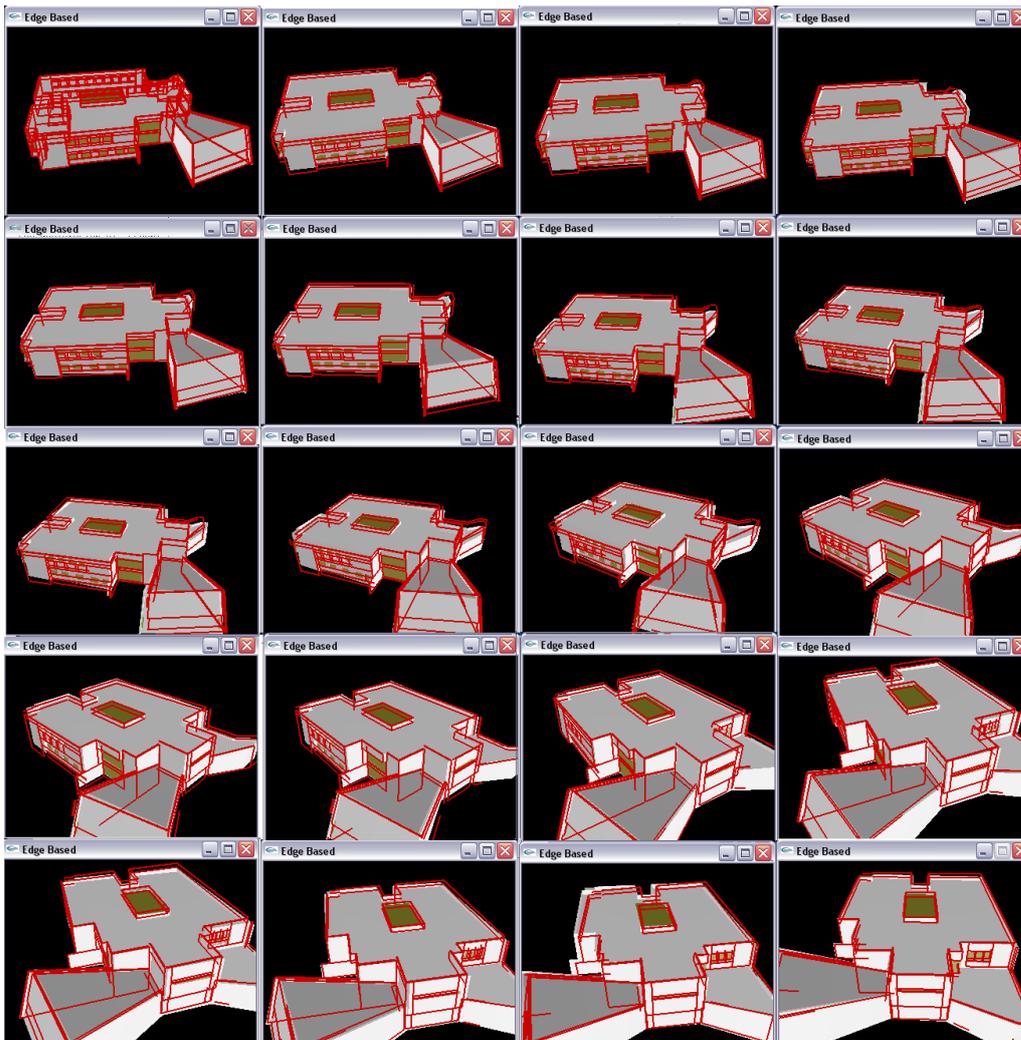


Figura 32. Sequência do prédio da UFAL completo.

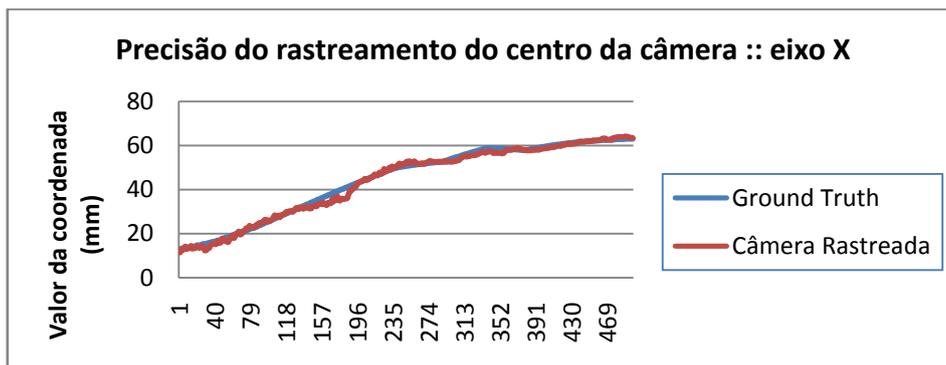


Figura 33. Erro associado ao eixo X da sequência do prédio da UFAL completo.

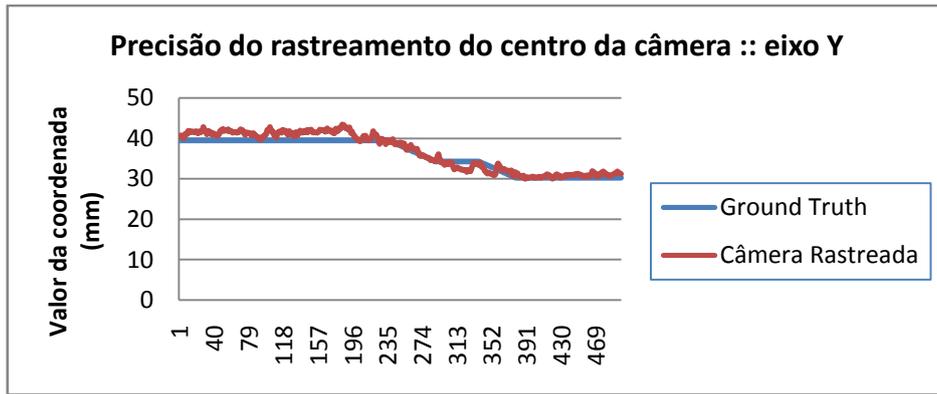


Figura 34. Erro associado ao eixo Y da seqüência do prédio da UFAL completo.

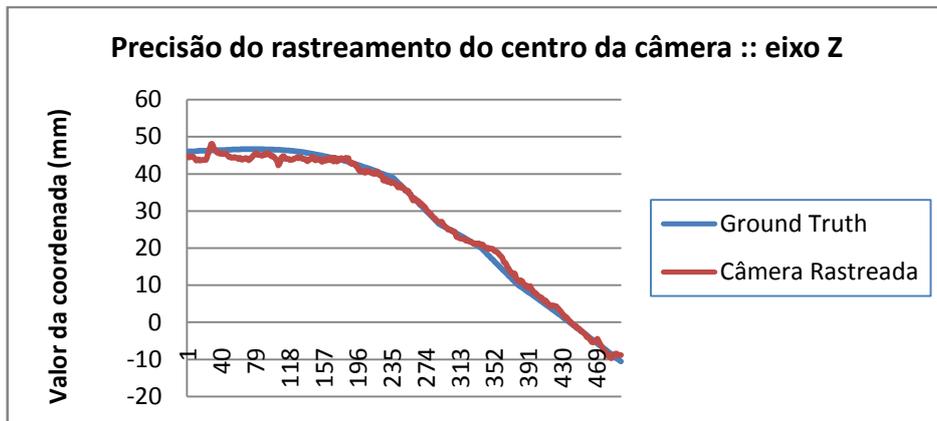


Figura 35. Erro associado ao eixo Z da seqüência do prédio da UFAL completo.

Na Figura 36 pode ser observado o erro total do rastreamento do prédio completo da UFAL. Este erro se mantém muito pequeno ao longo da seqüência inteira atingindo um pico de 6.5 mm e uma média de 2.1 mm. Os erros pequenos se devem a complexidade do modelo que possui muitas arestas que compensam entre as falhas existentes.

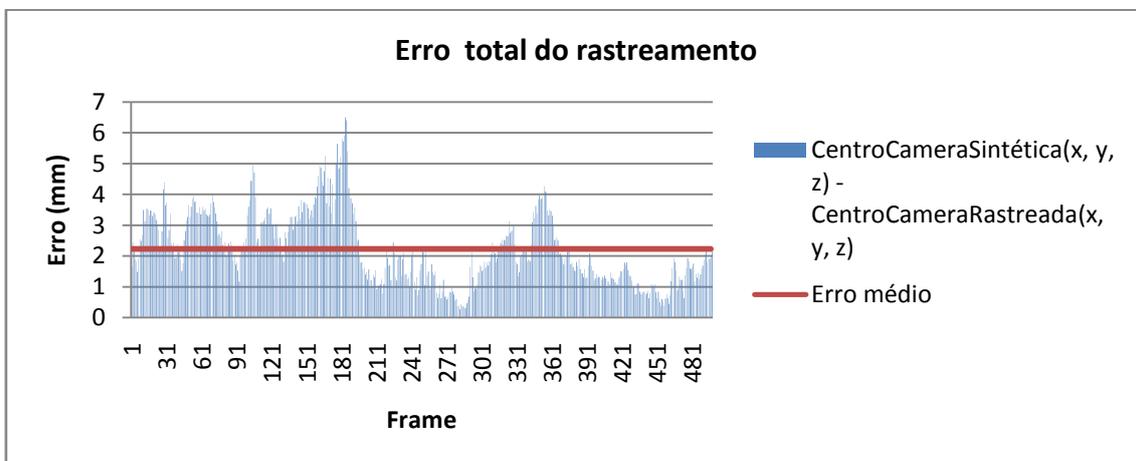


Figura 36. Distribuição do erro total do rastreamento na seqüência do prédio da UFAL completo.

6. Conclusões

A técnica de rastreamento 3D baseada na amostragem de pontos nas arestas foi implementada com êxito e mostrou tanto uma boa taxa de quadros por segundo como robustez a oclusões parciais e aproximações de objetos de forte gradiente. Porém, apesar destes resultados, a técnica ainda se mostra instável em situações onde são realizados movimentos menos suaves, uma vez que não utiliza informações temporais acerca da manutenção das condições do rastreamento (abordagem estatística).

A seqüência utilizando o cubo real atingiu uma taxa de 30 fps enquanto a fachada do prédio atingiu uma taxa de 25 fps. A seqüência do prédio completo devido a grande quantidade de arestas teve um desempenho abaixo das demais ficando na média de 15 fps o que denota a necessidade de realizar otimizações no algoritmo principalmente na fase de extração de arestas visíveis que é o gargalo da aplicação.

A seqüência da fachada do prédio da UFAL demonstrou maior instabilidade que a seqüência do prédio completo devido à quantidade inferior de arestas e a sua concentração em torno de uma superfície próxima a um plano enquanto o modelo do prédio completo possuía uma profundidade maior e mais bem definida. Esta diferença pode ser percebida ao se comparar o erro médio que ficou na casa dos 15 mm para a seqüência da fachada enquanto a seqüência do prédio completo teve média na casa dos 2 mm.

6.1. Trabalhos futuros

Como proposto em [6], a utilização de uma abordagem adaptativa em que um modelo estatístico acumula as informações do rastreamento ao longo da seqüência deverá permitir um rastreamento mais estável e robusto, principalmente no que diz respeito à movimentação do objeto.

Ainda no sentido de melhorar o rastreamento, é interessante utilizar estimadores robustos como o tukey m-estimator para realizar a minimização do efeito negativo ocasionado pela presença de *outliers*. Estes estimadores aplicam uma função que atribui pesos aos pontos estimados, de forma a atribuir pesos maiores a pontos que pertencem à faixa aceitável de variância com relação ao erro e pesos menores a pontos que fogem do padrão. Esta faixa de aceitabilidade é definida pelo desvio-padrão do erro.

Outro problema importante a ser endereçado é o processo de inicialização que, atualmente, é feito de forma manual posicionando a câmera numa pose inicial

conhecida, impedindo com isso sua utilização em situações não usuais. Alguns trabalhos na literatura já apontam na direção de inicialização automática, como por exemplo em [33] onde é feita uma esqueletização da imagem inicial de forma a utilizar este esqueleto para recuperar a pose do objeto tanto no quadro inicial quanto nos demais quadros da seqüência.

Além destas abordagens, a utilização associada de técnicas com enfoques diferentes aponta um caminho de novas perspectivas na área de MAR. Uma vez que cada técnica se propõe a resolver problemas específicos, associar técnicas de modo a obter rastreamentos ainda mais robustos utilizando qualidades complementares e minimizando problemas às vezes intratáveis por técnicas isoladas devido às suas características parece ser uma alternativa promissora. Em [29] é proposta uma técnica híbrida entre uma técnica baseada em textura e uma técnica baseada em arestas, provendo uma nova técnica mais robusta.

Ainda sobre o aspecto de combinações de técnicas e incremento de qualidades também se mostra interessante a elaboração de frameworks que proporcionem uma aceleração na implementação das técnicas e facilitem a co-utilização das mesmas. Com isto, pode-se prover um ambiente comum de maior produtividade ao desenvolvedor e elevar assim o nível de trabalho da “elaboração dos fundamentos” para a “elaboração das paredes”, numa metáfora à construção civil. Este trabalho de graduação é parte de um projeto maior que visa a definição e implementação de um framework de MAR que fornece suporte ao desenvolvimento de soluções baseadas tanto em técnicas baseadas em modelos como SfM/SLAM. A integração da técnica aqui implementada neste framework é a primeira atividade futura a ser realizada no contexto deste trabalho.

7. Referências

- [1] V. Teichrieb et al., “A Survey of Online Monocular Markerless Augmented Reality”, *International Journal of Modeling and Simulation for the Petroleum Industry*, 2007, vol. 1, no. 1, pp. 1–7.
- [2] E. Marchand et al., “Robust real-time visual tracking using a 2D-3D model-based approach”, *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp 262-268.
- [3] L. Vacchetti, V. Lepetit and P. Fua, “Stable Real-Time 3D Tracking Using Online and Offline Information”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 10, pp. 1385-1391.
- [4] M. Pollefeys, “Self-calibration and metric 3D reconstruction from uncalibrated image sequences”, *Ph.D. Thesis, ESAT-PSI, Katholieke Universiteit Leuven*, 1999.
- [5] N. Navab, “Developing Killer Apps for Industrial Augmented Reality”, *IEEE Computer Graphics and Applications*, 2004, vol. 24, no. 3, pp. 16-20.
- [6] H. Wuest, F. Vial and D. Stricker, “Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality”, *Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR2005)*, 2005, pp. 62-69.
- [7] O. Bimber, R. Raskar, “Spatial augmented reality, merging real and virtual worlds”, Ed. A. K. Peters Ltd, 2005, ISBN: 1-56881-230-2.
- [8] R. T. Azuma, “A Survey of Augmented Reality”, *Presence: Teleoperators and Virtual Environments*, 1997, vol. 6, no. 4, MIT Press, Cambridge, p. 355-385.
- [9] S. Pessoa, E. L. Apolinário, G. S. Moura, J. P. S. M. Lima, M. A. S. Bueno, V. Teichrieb, J. Kelner, “Illumination Techniques for Photorealistic Rendering in Augmented Reality”, *Symposium on Virtual and Augmented Reality*, 2008, Gráfica e Copiadora Nacional, p. 223-232.
- [10] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, "Recent advances in augmented reality," *Computer Graphics and Applications, IEEE*, 2001 vol.21, no.6, pp.34-47.
- [11] H. Kato, M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on* , 1999, vol., no., pp.85-94, 1999

- [12] V. Lepetit, P. Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: a Survey”, *Foundations and Trends in Computer Graphics and Vision*, 2004, vol. 1, no. 1, pp. 1-89.
- [13] S. Gomes Neto et al., “Experiences on the implementation of a 3d reconstruction pipeline”, *International Journal of Modeling and Simulation for the Petroleum Industry*, 2008, vol. 2, n. 1, pp. 7–15.
- [14] D. Nistér, “Preemptive RANSAC for live structure and motion estimation”, *Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 2003 Springer-Verlag New York, Inc., Secaucus, NJ, USA, vol. 16, n. 5, pp. 321–329. ISSN 0932-8092.
- [15] A. J. Davison, N. D. Molton, “Monoslam: Real-time single camera slam”. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, IEEE Computer Society, Washington, DC, USA, vol. 29, n. 6, pp. 1052–1067. ISSN 0162-8828.
- [16] A. Comport et al., “Real-time markerless tracking for augmented reality: the virtual visual servoing framework”, *IEEE TVCG*, 2006, vol. 12, pp. 615-628.
- [17] S. Basu, I. Essa, A. Pentland, “Motion regularization for model-based head tracking”, 1996, Disponível em: <citeseer.ist.psu.edu/article/basu96motion.html>.
- [18] M. Uenohara, T. Kanade, “Vision-based object registration for realtime image overlay”, *Journal of Cognitive Neuroscience*, 1991, vol. 3, n. 1, pp. 71–86.
- [19] F. Jurie, M. Dhome, “A simple and efficient template matching algorithm”. 2001, pp. 544–549. Disponível em: <citeseer.ist.psu.edu/jurie01simple.html>
- [20] I. Skrypnik, D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features", *Third IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2004*, 2004, vol. 2, no. 5, pp. 110-119.
- [21] D.G. Lowe, “A Robust Model-Based Motion Tracking through the Integration of Search and Estimation”, *Int'l J. Computer Vision*, 1992, vol. 8, no. 2, pp. 113-122.
- [22] P. Wunsch, G. Hirzinger, “A Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion”, *Proc. 1997 Int'l Conf. Robotics and Automation*, 1997, pp. 2868-2873.

- [23] D. Koller, K. Daniilidis, H. Nagel, “Model-based object tracking in monocular image sequences of road traffic scenes”, *IJCV*, 1993 vol. 10, n. 3, pp. 257–281. Disponível em: <citeseer.ist.psu.edu/koller93modelbased.html>.
- [24] R. Deriche, O. Faugeras, “Tracking line segments,” *Image and Vision Computing*, 1990, vol. 8, pp. 271–270.
- [25] T. Drummond, R. Cipolla, “Real-Time Visual Tracking of Complex Structures”, *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 2002, Vol. 24, No. 27, pp. 932-926.
- [26] K. Levenberg, “A Method for the Solution of Certain Non-linear Problems in Least Squares”, *Quarterly of Applied Mathematics*, 1944, vol. 2, no. 2, pp. 164-168
- [27] P. Mitchell et al., “GPU Accelerated Real-Time 3D Tracking for Humanoid Locomotion and Stair Climbing”, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 463-469.
- [28] G. Simon, M. Berger, “A two-stage robust statistical method for temporal registration from features of various type”, *Sixth International Conference on Computer Vision (ICCV'98)*, 1998, pp. 261-269.
- [29] L. Vachetti, V. Lepetit and P.Fua, “Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking”, *Proceedings of the Third IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2004, pp. 48-57.
- [30] J. Barandiarán, D. Borro, “Edge-Based Markerless 3D Tracking of Rigid Objects”, *17th International Conference on Artificial Reality and Telexistence*, 2007.
- [31] G. Reitmayr, T. W. Drummond, “Going out: Robust Model-based Tracking for Outdoor Augmented Reality”, *Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'06)*, 2006, pp.109-118.
- [32] P. Bouthemy, “A maximum likelihood framework for determining moving edges”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, vol. 11, no. 5, pp 499-511.
- [33] G. Blasko, P. Fua, “Real-Time 3D Object Recognition for Automatic Tracker Initialization”, *International Symposium on Augmented Reality*, Nova York, NY, 2001.

- [34] D. Shreiner, M. Woo, J. Neider, T. Davis, “OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL”, 2005, 5th Edition.
- [35] F. Spindler, A. Saunier. “Getting started with ViSP-2.4.0”. 2007, Disponível em:
<<http://www.irisa.fr/lagadic/visp/publication.html>>
- [36] <<http://vxl.sourceforge.net/#docs>>