

# UMA TÉCNICA DE REALIDADE AUMENTADA SEM MARCADORES BASEADA EM ARESTAS COM FILTRO DE PARTÍCULAS

---

TRABALHO DE GRADUAÇÃO

**Aluno:** Tullio José de Souza Lucena (tjsl@cin.ufpe.br)

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

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

Recife, 29 de abril de 2013

## Resumo

Um dos maiores desafios da área de Realidade Aumentada é realizar o correto posicionamento dos elementos virtuais na cena real. Para isso, é necessário realizar o rastreamento da câmera ao longo da sequência de imagens capturadas. Este trabalho de graduação apresenta um estudo sobre técnicas de rastreamento de câmera baseado em arestas com a utilização de Filtro de Partículas para simulação da movimentação da câmera durante o rastreamento. Algumas técnicas disponíveis na literatura são apresentadas nesta monografia, com o propósito de encontrar os elementos mais relevantes para o adequado funcionamento destas técnicas. É apresentada uma implementação de uma técnica que utiliza etapas de projetos distintos da literatura para um resultado final que contemple os benefícios apresentados em cada técnica. Além disso, apresenta uma análise de qualidade do rastreamento obtido e velocidade de execução das configurações implementadas. Os resultados atingidos pela técnica resultam em erros de reprojeção próximos de 0,5 *pixel*, em uma taxa de aproximadamente 50 FPS.

## **Agradecimentos**

Agradeço primeiramente a Deus pela ótima vida que me foi dada e por sempre estar presente nas horas de maior desespero. Dedico este trabalho de graduação a meus pais (Cloves e Elizabeth), que me deram o melhor apoio que uma pessoa pode pedir, não me deixando desmotivar, sempre me apoiando nos estudos; sem eles não seria nada do que sou hoje. Em especial, gostaria de agradecer aos meus avós (Augusto, Isabel e Bernadete), pela experiência passada, principalmente pelos momentos em família que só eles conseguem proporcionar. Gostaria de agradecer aos meus tios, Arabela, Célia, Flávio, Degilene, Moisés e Marleide, pelo carinho e companheirismo compartilhado por todos esses anos de convivência. Não posso esquecer-me dos meus irmãos (Cloves e Pablo), primos (Adelia, Isabel, Giovanna, José, Moisarley, Bruno e Rodrigo), e os muitos amigos que sempre estão presentes no meu dia-a-dia, trazendo muitos momentos de felicidade. Preciso agradecer aos meus amigos da universidade (Arthur, Davi, David, Hugo M., Luiz, Pablo P. e Rafael L.); estas pessoas compartilharam de todos os momentos durante o curso, e foram de grande ajuda durante muitos momentos, não só relacionados às disciplinas, mas também abrindo oportunidades para meu crescimento pessoal. Gostaria de agradecer à equipe de Maratona de Programação que me ajudou no meu conhecimento prático em programação. Não sei como agradecer aos integrantes do VOXAR Labs (Veronica, Francisco, Lucas, Rafael, Artur, Mozart, Jorge, Mariana, e outros não menos importantes). Este ambiente de pesquisa me proporcionou um conhecimento incrível, trazendo um convívio com especialistas na minha área de trabalho e que sempre me ajudaram no meu crescimento intelectual e acadêmico. Gostaria de agradecer também à equipe do INdT (Daker, Hugo, Jesus, Lauro, Luciano e Marcelo) que está me proporcionando uma experiência inexplicável, me trazendo uma visão nova de um ambiente de trabalho.

Muito obrigado.

# Sumário

1.	Introdução .....	8
1.1.	Objetivo .....	8
1.2.	Estrutura do Documento .....	8
2.	Conceitos Básicos .....	10
2.1.	Realidade Aumentada .....	10
2.1.1.	Rastreamento .....	11
2.1.2.	Rastreamento baseado em Arestas .....	12
2.2.	Rastreamento da Câmera .....	13
2.2.1.	Parametrização da Câmera .....	15
2.3.	Filtros Bayesianos .....	15
2.3.1.	Filtro de Partículas .....	17
3.	Abordagens para Rastreamento baseado em Arestas com Filtro de Partículas .....	19
3.1.	Inicialização .....	19
3.2.	Propagação .....	20
3.3.	Avaliação .....	22
3.4.	Estimativa .....	25
3.5.	Reamostragem .....	25
4.	Desenvolvimento da Técnica de Rastreamento baseado em Arestas com Filtro de Partículas .....	27
4.1.	Inicialização .....	27
4.2.	Propagação .....	28
4.3.	Avaliação .....	28
4.4.	Estimativa .....	29
4.5.	Otimização .....	30
4.6.	Reamostragem .....	30
5.	Resultados .....	31
5.1.	Qualidade do Rastreamento .....	32
5.2.	Desempenho .....	36
5.3.	Resultados Adicionais .....	39
6.	Conclusão .....	41
6.1.	Trabalhos Futuros .....	41

Referências..... 43

## Índice de Figuras

Figura 1. Ambiente real aumentado com informações virtuais. ....	11
Figura 2. Técnicas de rastreamento com e sem marcadores. ....	12
Figura 3. Extração de arestas da imagem, utilizadas para o rastreamento. ....	13
Figura 4. Modelo de projeção em perspectiva. ....	13
Figura 5. Representação do Filtro Bayesiano. ....	16
Figura 6. Filtro de Partículas. ....	17
Figura 7. Diagrama das etapas que compõem o Filtro de Partículas. ....	19
Figura 8. Fluxograma de funcionamento da técnica de [11]. Figura extraída de [11]. ....	20
Figura 9. As poses das partículas (destacadas em verde). ....	21
Figura 10. Imagem base (esquerda), aplicação do filtro de Sobel nas direções $x$ e $y$ (meio) e mapa de distância (direita). ....	22
Figura 11. Função de distância comparando translações em $x$ e $y$ da tela. O ponto (0,0) corresponde à pose correta da câmera. Figura extraída de [10]. ....	23
Figura 12. Arestas do modelo e arestas da textura. Figura extraída de [13]. ....	24
Figura 13. Exemplo de arestas horizontais e verticais extraídas da imagem (cima), mapas de distância calculados com o filtro de Sobel na direção horizontal (esquerda) e vertical (direita). ....	24
Figura 14. Diagrama das etapas que compõem o rastreador baseado em arestas com Filtro de Partículas desenvolvido no presente projeto. ....	27
Figura 15. Mapas de distância. Horizontal (esquerda), vertical (meio), ambas direções (direita). ....	29
Figura 16. Classificação das arestas a partir do ângulo. ....	29
Figura 17. Sequência de <i>frames</i> durante a falha de rastreamento: resultado sem Filtro de Partículas (cima), resultado com Filtro de Partículas (baixo). ....	34
Figura 18. Resultado da criação do mapa de distâncias errado (esquerda) e corrigido (direita). ....	40

## Índice de Gráficos

Gráfico 1. Qualidade do rastreamento perante a quantidade de partículas.....	32
Gráfico 2. Qualidade do rastreamento ao longo da sua execução. ....	33
Gráfico 3. Qualidade do rastreamento perante a variação na avaliação do Filtro de Partículas.....	35
Gráfico 4. Qualidade do rastreamento perante o tipo de estimativa da pose da câmera.....	36
Gráfico 5. Velocidade de execução perante a quantidade de partículas.....	37
Gráfico 6. Tempo decorrido durante o algoritmo de minimização.....	38
Gráfico 7. Proporção de tempo entre as etapas do Filtro de Partículas. ....	39

# **1. Introdução**

A realidade aumentada sem marcadores se caracteriza pela inserção de elementos virtuais em cenas reais utilizando elementos já presentes na cena como arestas, ou texturas, para realizar o registro 3D dos objetos virtuais e reais sem a inserção de marcadores na cena. Mas, para que este posicionamento dos objetos virtuais seja realizado de forma correta, é necessário que a aplicação seja capaz de realizar o rastreamento da câmera que está filmando a cena real.

Abordagens estatísticas vêm sendo utilizadas para simulação da movimentação da câmera ao longo do tempo. Atualmente, a utilização da abordagem de Filtro de Partículas para realizar o rastreamento da câmera está sendo difundida na literatura, por se tratar de uma técnica bastante eficiente para este propósito.

## **1.1. Objetivo**

Neste contexto, este trabalho de graduação se propõe a fazer um levantamento das técnicas de rastreamento baseado em arestas disponíveis na literatura que utilizam a estrutura de Filtro de Partículas para o rastreamento da câmera. Após este estudo, pretende-se realizar uma implementação de uma técnica utilizando características relevantes das técnicas analisadas. Uma análise detalhada de cada parâmetro testado é feita para justificar a escolha das alternativas. Cabe ressaltar que o presente trabalho visa o desenvolvimento de uma técnica que melhore a qualidade do rastreamento e que execute em tempo real, a fim de viabilizar a sua utilização em aplicações de realidade aumentada.

## **1.2. Estrutura do Documento**

A estrutura desta monografia está conforme descrito na sequência. No Capítulo 2 são discutidos os principais conceitos sobre realidade aumentada, rastreamento da câmera e Filtros Bayesianos, incluindo o Filtro de Partículas. No Capítulo 3 são discutidas alternativas encontradas na literatura para o problema de rastreamento da câmera utilizando rastreamento baseado em arestas com Filtro de Partículas. No Capítulo 4 são apresentadas as alternativas escolhidas para serem desenvolvidas em cada etapa da técnica implementada neste trabalho de graduação. No Capítulo 5 são mostrados os resultados obtidos da avaliação de qualidade do rastreamento e



desempenho de execução da técnica, ressaltando as contribuições feitas por este trabalho. O Capítulo 6 expõe uma conclusão dos pontos discutidos e apresenta ideias para trabalhos futuros.

## **2. Conceitos Básicos**

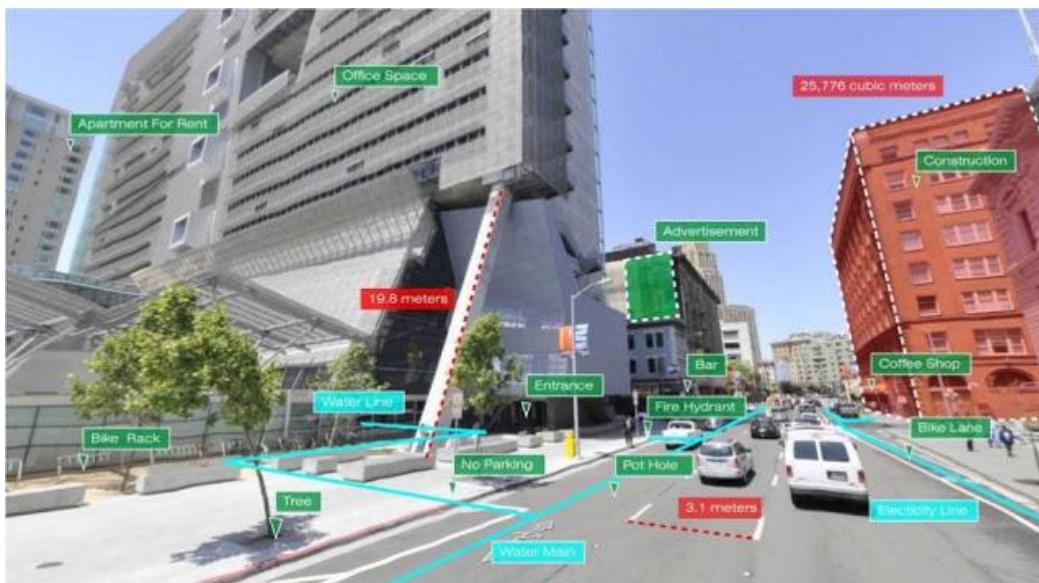
Este capítulo apresenta os principais fundamentos utilizados neste projeto. Primeiramente são mostrados os conceitos da realidade aumentada e as formas de rastreamento que podem ser utilizadas para identificar o posicionamento e orientação da câmera na cena. Na sequência, são discutidos os cálculos e representações para o rastreamento e parametrização da câmera, para um melhor entendimento dos termos utilizados ao longo do texto. Ao final do capítulo, é explicado o funcionamento dos Filtros Bayesianos e os Filtros de Partículas, que são ferramentas estatísticas que podem ser utilizadas para melhorar a informação temporal da pose da câmera durante o rastreamento.

### **2.1. Realidade Aumentada**

Enquanto a realidade virtual tem como objetivo a imersão total em um ambiente 3D sintético gerado por computador, a realidade aumentada busca realizar a sobreposição de objetos virtuais em um ambiente real [1]. Segundo [2], três características são necessárias para que uma aplicação seja definida como realidade aumentada:

1. Combinar elementos reais e virtuais;
2. Ser interativo e em tempo real;
3. Ter registro em 3D.

O registro do conteúdo virtual sobreposto no ambiente real do usuário é a base das aplicações de realidade aumentada. O posicionamento e orientação adequados dos elementos virtuais permitem que o observador tenha a ilusão de que esta informação é inerente ao ambiente real, como ilustrado na Figura 1.



**Figura 1. Ambiente real aumentado com informações virtuais.**

Para isso, é necessária a estimativa da transformação do espaço virtual para o real, que é adquirida através do rastreamento da câmera. Neste contexto, a câmera representa a visão do observador.

### **2.1.1. Rastreamento**

Existem diversas técnicas para realizar o rastreamento da câmera. As aplicações de realidade aumentada geralmente são divididas em dois tipos, de acordo com o rastreamento empregado: realidade aumentada com marcadores e realidade aumentada sem marcadores. A diferença básica entre estas técnicas é a informação que é processada para definir a disposição dos elementos virtuais na cena. Algumas técnicas utilizam elementos artificiais inseridos no ambiente real para auxiliar o rastreamento, enquanto outras utilizam apenas elementos inerentes ao próprio objeto a ser rastreado.

A Figura 2 caracteriza quatro tipos de informações que podem ser utilizadas para realizar o rastreamento:

1. **Marcadores:** objetos artificiais adicionados ao mundo real. Eles possuem um padrão característico para serem diferenciados do restante da cena;
2. **Sensores:** aparelhos, como GPS e acelerômetro, que são utilizados para determinar o posicionamento, orientação e movimentação do observador.
3. **Textura:** pontos característicos de uma imagem, também conhecidos como *keypoints*, que são buscados na cena real. Estas características são próprias do objeto, sem a inserção de outros elementos para o seu rastreamento;

4. **Arestas:** são utilizados modelos sintéticos de objetos reais a serem rastreados e procura-se rastrear estes objetos a partir de mudanças acentuadas de coloração na imagem da cena. Estas mudanças representam as arestas do objeto a ser rastreado na cena.

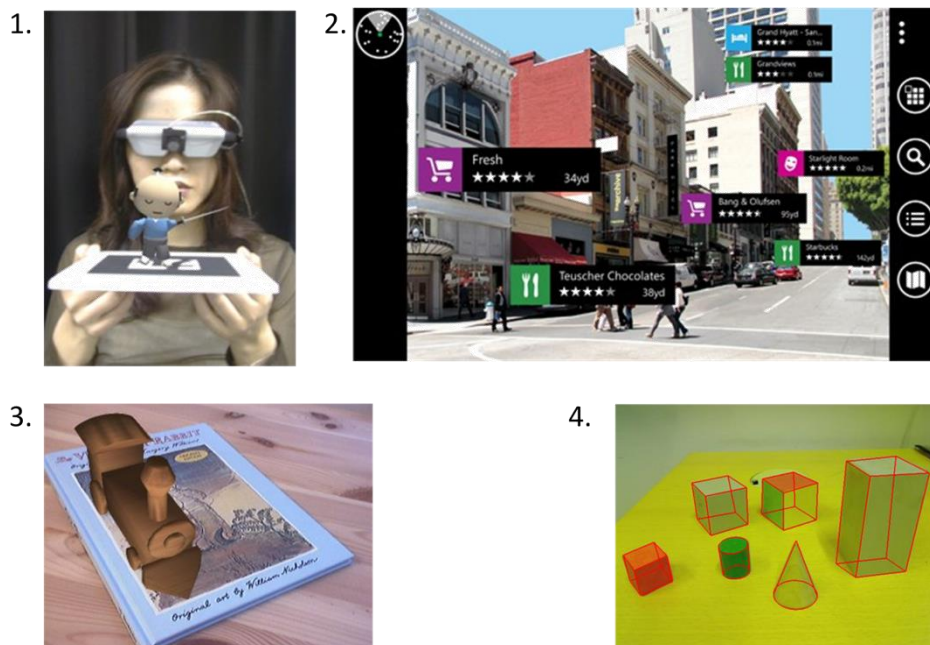


Figura 2. Técnicas de rastreamento com e sem marcadores.

## 2.1.2. Rastreamento baseado em Arestas

Este tipo de técnica foi uma das primeiras a serem utilizadas para rastreamento em aplicações de realidade aumentada sem marcadores, devido a sua eficiência na qualidade do rastreamento e relativa simplicidade de implementação [3].

Geralmente esta técnica é mais eficiente quando os objetos a serem rastreados não possuem muita textura, pois ela pode ser confundida com as arestas do objeto. Outro fator que pode influenciar no rastreamento é o plano de fundo (chamado de *background*) poluído, a presença de sombras na imagem, ou a movimentação rápida da câmera. Este último faz com que as arestas fiquem menos acentuadas, causando um efeito conhecido como *blur*, ou borrramento.

Para a realização do rastreamento é necessário que a aplicação tenha o conhecimento prévio das arestas presentes no objeto 3D a ser rastreado, também conhecido como modelo. A partir da imagem capturada pela câmera, um processamento da imagem é realizado para a obtenção de pontos de alto gradiente, com a maior chance de ser uma aresta. Na Figura 3 pode ser observada a aplicação do filtro de Sobel [4] resultando numa imagem onde os pontos mais claros tem o maior gradiente. A pose da

câmera é determinada quando estes pontos da imagem são corretamente relacionados com as arestas do modelo.

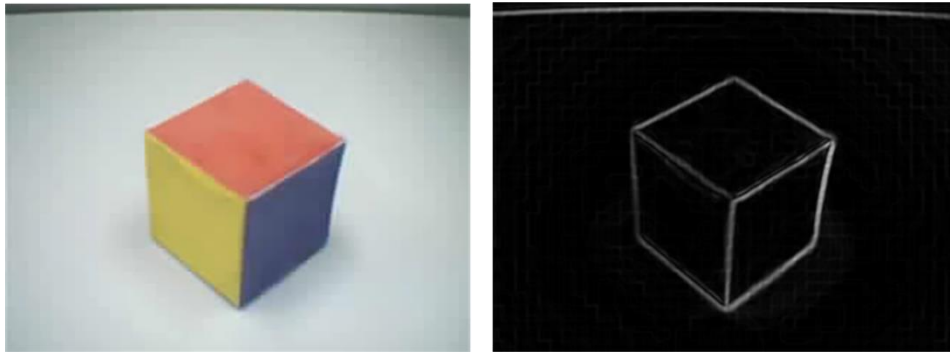


Figura 3. Extração de arestas da imagem, utilizadas para o rastreamento.

Neste trabalho, uma técnica de rastreamento baseado em arestas é utilizada para rastrear a cena real, permitindo o registro 3D dos objetos virtuais com a mesma. O Capítulo 4 apresenta em detalhes a técnica desenvolvida.

## 2.2. Rastreamento da Câmera

O rastreamento da câmera consiste em determinar o posicionamento e orientação da câmera que consiga correlacionar os pontos que estão no espaço 3D com seus respectivos pontos na imagem mostrada na tela [5]. A formação da imagem pode ser definida como a projeção do espaço 3D no plano da imagem, representada na Figura 4.

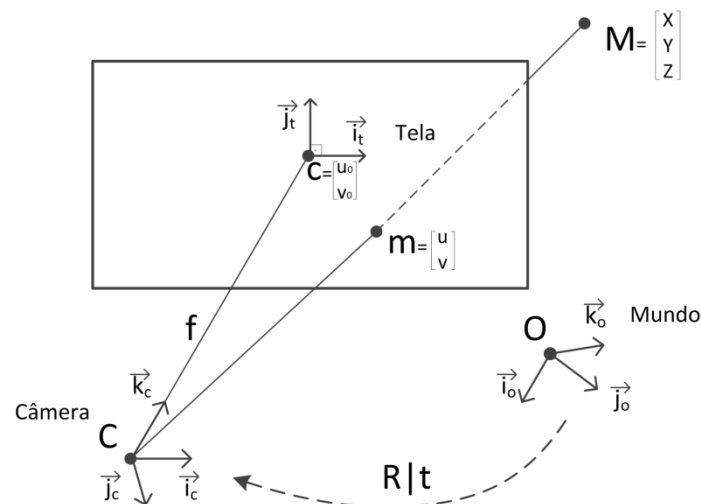


Figura 4. Modelo de projeção em perspectiva.

Na Figura 4 estão representados três sistemas de coordenadas: coordenadas de mundo  $(O, \vec{i}_o, \vec{j}_o, \vec{k}_o)$ , coordenadas de câmera  $(C, \vec{i}_c, \vec{j}_c, \vec{k}_c)$  e coordenadas de tela  $(c, \vec{i}_t, \vec{j}_t)$ . O ponto  $c$  é determinado pela intersecção entre o plano da tela e a reta

formada pela câmera ( $C$ ) e o eixo óptico da câmera ( $\overline{k_c}$ ). Dado um ponto 3D  $M = [X, Y, Z]^T$  em coordenadas de mundo, e sua projeção, em coordenadas de tela  $m = [u, v]^T$ , a relação entre estes pontos é dada por  $s\tilde{m} = P\tilde{M}$ , onde  $s$  é um fator de escala,  $\tilde{M} = [X, Y, Z, 1]^T$  e  $\tilde{m} = [u, v, 1]^T$  são os pontos  $M$  e  $m$  em coordenadas homogêneas [5] e  $P$  é uma matriz de projeção  $3 \times 4$ .

A matriz de projeção pode ser representada como a multiplicação de duas outras matrizes:  $P = K[R|t]$ , conhecidas como:

- **Parâmetros intrínsecos ( $K$ ):** é uma matriz  $3 \times 3$ , responsável pela calibração da câmera. Estes parâmetros são inerentes às características da câmera, logo não variam com a mudança do seu posicionamento e orientação. A matriz é definida por:

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

onde:

- $\alpha_x$  e  $\alpha_y$  são valores calculados à partir da distância focal ( $f$ ), e a quantidade de *pixels* nos eixos da tela ( $k_x$  e  $k_y$ ) são dados por  $k_x * f$  e  $k_y * f$ , respectivamente;
- $s$  representa o coeficiente de cisalhamento da câmera, determinado pelo ângulo entre o eixo óptico da câmera e o plano da tela;
- $u_0$  e  $v_0$  são as coordenadas da origem da base da tela, representadas pelo ponto  $c$  na Figura 4.
- **Parâmetros extrínsecos ( $[R|t]$ ):** é uma matriz  $3 \times 4$ , que representa a movimentação da câmera em relação ao sistema de coordenadas de mundo, também conhecida como pose da câmera. Esta matriz é constituída por duas submatrizes:

$$R|t = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}, \quad (2)$$

onde:

- $R$  é uma matriz  $3 \times 3$ , conhecida como matriz de rotação. Ela representa a orientação da câmera;
- $t$  é uma matriz  $3 \times 1$ , conhecida como matriz de translação. Ela representa a posição da câmera.

### 2.2.1. Parametrização da Câmera

De acordo com [6] e [7] a estimativa conjunta dos parâmetros intrínsecos e extrínsecos pode levar a inconsistências e falta de precisão na pose calculada da câmera. Devido a isto, a maioria das técnicas de rastreamento se preocupa apenas em calcular os parâmetros extrínsecos, utilizando câmeras calibradas (com parâmetros intrínsecos conhecidos).

Algoritmos de minimização de funções geralmente são utilizados para identificar os parâmetros da câmera. Este tipo de algoritmo é computacionalmente custoso, e este custo aumenta ainda mais com o aumento da quantidade de variáveis a serem encontradas. Logo a parametrização da câmera é realizada com o intuito de reduzir a quantidade de variáveis a serem utilizadas para a representação dos parâmetros da câmera. Com uma representação mais reduzida, os algoritmos de minimização de funções se tornam mais eficientes e tem o processamento reduzido.

Os parâmetros de translação são independentes, enquanto que a matriz de rotação tem três graus de liberdade [8]. Uma parametrização da matriz de rotação bastante utilizada na literatura é a fórmula de Rodrigues, por representar a matriz de rotação utilizando apenas três parâmetros.

A matriz de rotação pode ser descrita a partir de um vetor 3D ( $\vec{w} = [w_x, w_y, w_z]^T$ ) e um ângulo ( $\theta = \|\vec{w}\|$ ), que representam uma rotação de  $\theta$  em torno do vetor  $\vec{w}$ . A fórmula de Rodrigues se dá pela seguinte equação:

$$R(\Omega) = I + \frac{\sin \theta}{\theta} \Omega + \frac{(1 - \cos \theta)}{\theta^2} \Omega^2, \quad (3)$$

onde:

- $\Omega = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$ ;
- $I$  é a matriz de identidade  $3 \times 3$ .

### 2.3. Filtros Bayesianos

Os Filtros Bayesianos são estruturas responsáveis por realizar a filtragem dos dados de um determinado problema de forma a obter uma estimativa suave e realizar a predição das informações relevantes ao longo do tempo. Para um melhor entendimento

de Filtros Bayesianos, segue a definição de alguns conceitos principais. Informações detalhadas podem ser encontradas em [9].

- **Filtragem:** é um processo responsável por extrair informações de um conjunto de dados, ao longo do tempo, utilizando uma métrica para avaliar sua relevância através de uma função de avaliação;
- **Predição:** é uma forma de estimativa *a priori*. É uma tentativa de inferir a importância das informações num instante de tempo no futuro, levando em conta os dados até o presente momento;
- **Suavização:** é uma forma de estimativa *a posteriori*. É uma maneira de determinar a melhor representação atual das informações, levando em conta os dados obtidos, o estado atual e os estados anteriores. Ela é responsável pela diminuição de ruído nos dados.

Segundo [9], os tipos de Filtros Bayesianos mais conhecidos são o Filtro de Kalman, Filtro de Kalman estendido e Filtro de Partículas (explicado em seguida na Subseção 2.3.1). O Filtro de Kalman e o Filtro de Kalman estendido são utilizados em problemas em que a função objetivo é bem estável, como o controle da movimentação de um robô equipado com sensores. A diferença básica entre eles é que o estendido pode tanto resolver problemas lineares, quanto não lineares, enquanto que o básico não consegue resolver problemas não lineares.

O diagrama ilustrado na Figura 5 descreve o modelo de representação do Filtro Bayesiano, seguindo as etapas de predição e suavização, apresentadas anteriormente. Além disso, mostra o fluxo que a informação segue para que o resultado final seja obtido.

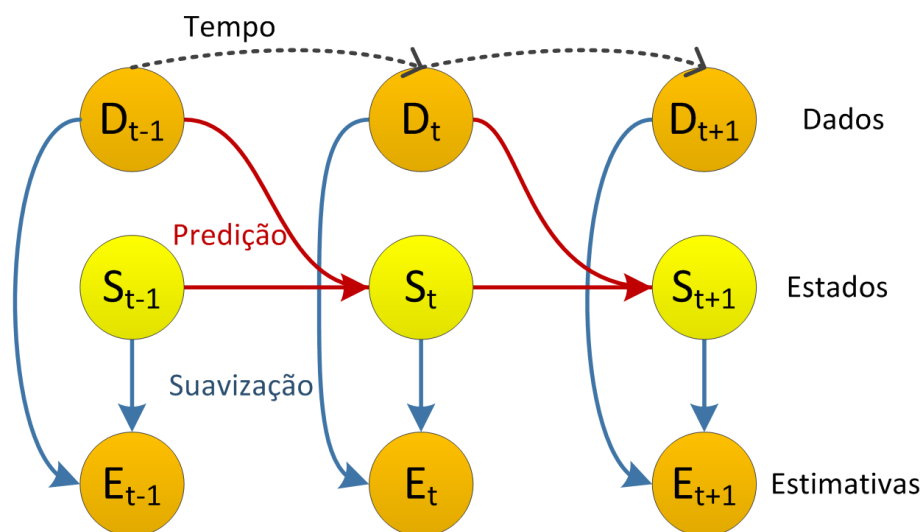


Figura 5. Representação do Filtro Bayesiano.



### 2.3.1. Filtro de Partículas

O Filtro de Partículas é uma especialização dos Filtros Bayesianos, em que cada estado é representado como um conjunto de amostras, cada amostra é também conhecida como partícula. O Filtro de Partículas representa uma estrutura mais complexa quando comparado com o Filtro de Kalman estendido, porque cada partícula pertence a uma distribuição de probabilidade [9]. Por sua vez, o Filtro de Partículas também consegue resolver tanto problemas lineares como não lineares, além de não necessitar de uma função objetivo tão estável para o seu funcionamento como o Filtro de Kalman e o Filtro de Kalman estendido. Esta técnica é muito utilizada quando a variação da função de avaliação, ao longo do tempo, é pequena [9]. O funcionamento desta técnica pode ser observado na Figura 6.

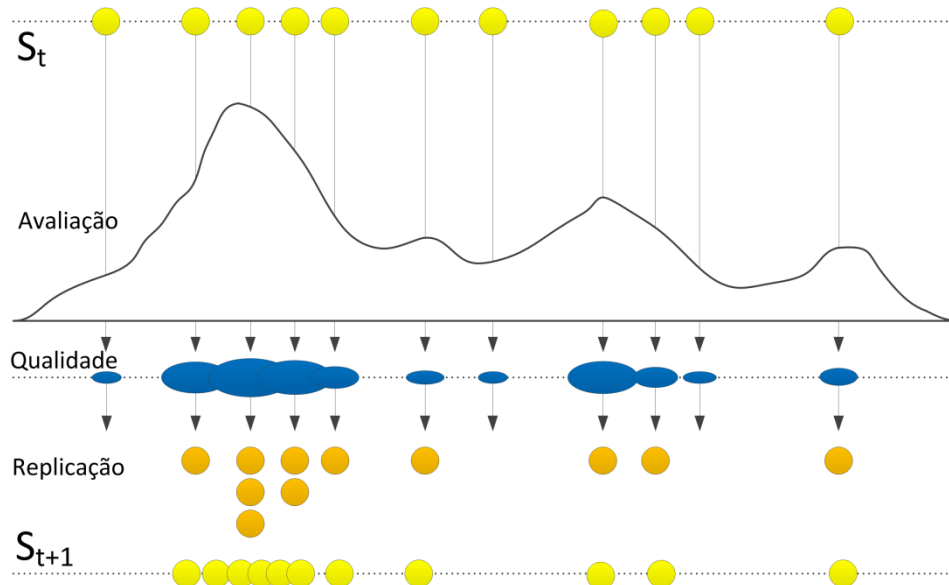


Figura 6. Filtro de Partículas.

Cada problema possui uma função de avaliação específica, que serve para avaliar a proximidade da solução esperada, também conhecida como função de avaliação. As partículas do instante de tempo atual ( $S_t$ ) passam por uma etapa de avaliação, para determinar a sua relevância. Na Figura 6 a função é aplicada em cada partícula, resultando no valor da avaliação; quanto mais alto o valor, mais relevante é a partícula.

A quantidade de partículas do próximo instante de tempo ( $S_{t+1}$ ) deve ser a mesma da quantidade inicial para que a dimensão do conjunto de partículas não se altere ao passar do tempo [9]. Para obter um melhor resultado, as partículas que tiveram a melhor avaliação devem ser mantidas, pois elas têm a maior probabilidade de serem

relevantes para o próximo instante de tempo. Para simular a variação dos dados do próximo instante de tempo, as partículas escolhidas sofrem um deslocamento, assim as partículas vão se adaptando à variação da entrada ao longo do tempo.

Filtros de Partículas têm um bom desempenho em situações que a variação dos parâmetros observados da entrada é regular, conseguindo boas aproximações da entrada. Além disso, são indicados em situações em que a função de avaliação seja convergente no ponto de mínimo erro, visto que as partículas tendem a manter o seu deslocamento ao longo do tempo e tendem a se concentrar neste ponto de mínimo erro [9].

Neste trabalho, a abordagem de Filtro de Partículas é utilizada para aumentar a precisão de um rastreador de objetos baseado em arestas. O Capítulo 4 apresenta em detalhes a técnica desenvolvida.

### 3. Abordagens para Rastreamento baseado em Arestas com Filtro de Partículas

Neste capítulo é explicado o funcionamento do Filtro de Partículas utilizando arestas para o rastreamento. São mostradas diferentes formas de rastreamento presentes na literatura, utilizando diferentes abordagens/informações. Para um melhor entendimento, o capítulo está dividido em subseções apresentando as etapas da técnica, que são executadas para realização do procedimento de rastreamento. A Figura 7 mostra a relação entre estas etapas.

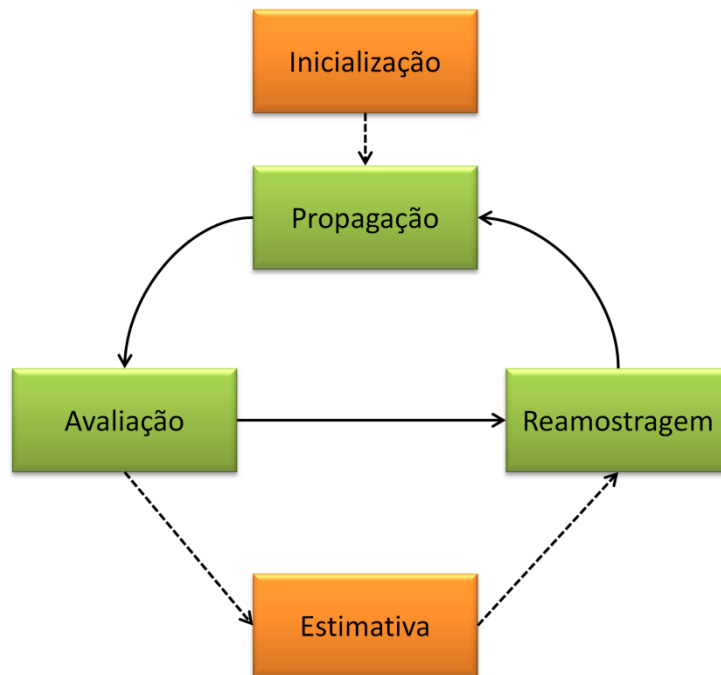


Figura 7. Diagrama das etapas que compõem o Filtro de Partículas.

#### 3.1. Inicialização

A primeira etapa a ser executada é a de inicialização do sistema, onde as informações a serem utilizadas ao longo do processo são criadas e armazenadas. Além de ser executada antes do início do rastreamento, pode ser preciso que esta etapa seja executada ao longo do rastreamento, sempre quando o mesmo é perdido, ou seja, quando o resultado do cálculo da pose fica distante do esperado.

No início do procedimento de inicialização, o Filtro de Partículas é formado pelo posicionamento das arestas do modelo a ser rastreado [10]. Inicialmente, a configuração do Filtro de Partículas é composta pela replicação da primeira partícula, que é criada a partir da pose do primeiro *frame*, que é fornecida pela aplicação como entrada. Além

disso, é preciso definir a quantidade de partículas a serem utilizadas ao longo do rastreamento, uma vez definida.

A técnica apresentada em [11] propõe a utilização da informação contida nas junções, que consistem basicamente em pontos de encontro entre arestas que são obtidos a partir do algoritmo conhecido como JUDOCA. Uma vez calculados os pontos das junções, várias imagens representando o modelo em diferentes poses e escalas são armazenadas em uma estrutura, que possibilite o mapeamento entre as imagens e as poses que originaram estas imagens. Com esta nova informação é possível identificar qual das imagens geradas mais se adequa à imagem real, obtendo uma pose aproximada da cena real. A Figura 8 ilustra o procedimento de inicialização descrito.

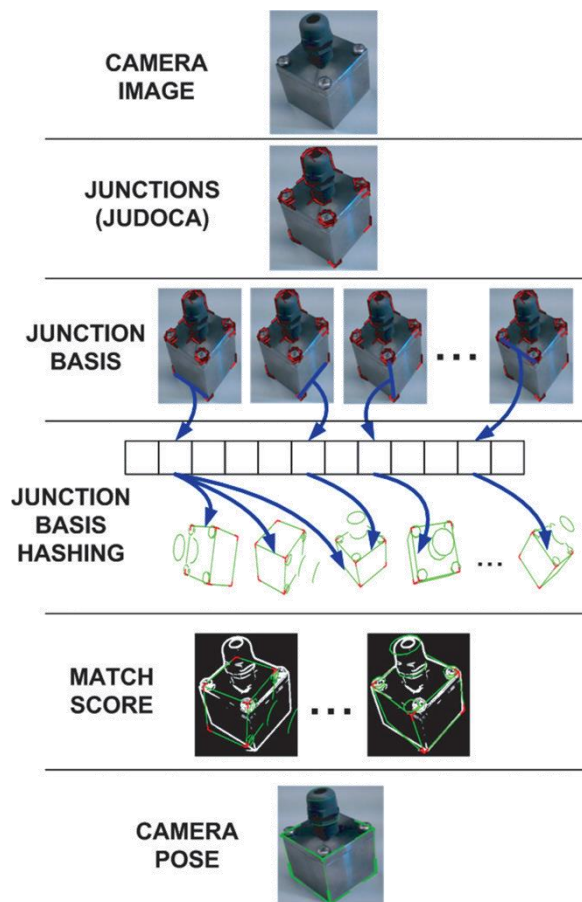


Figura 8. Fluxograma de funcionamento da técnica de [11]. Figura extraída de [11].

## 3.2. Propagação

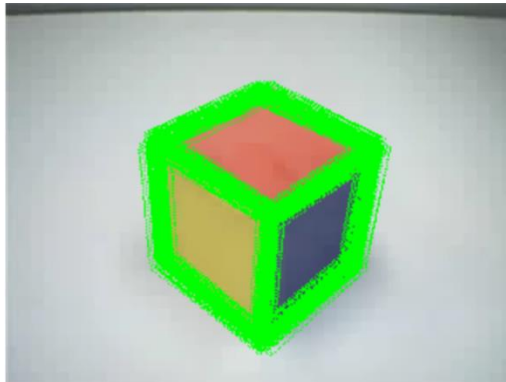
Esta etapa é responsável pela variação nas partículas, tentando prever a movimentação realizada pela câmera entre a imagem do instante de tempo anterior e a imagem atual. O descolamento que uma partícula sofre é representado por:

$$X_t = X_{t-1} * \exp(A_{t-1} + dW_t \sqrt{\Delta t}), \quad (4)$$

onde:

- $X_t$  e  $X_{t-1}$  representam a pose da partícula no instante de tempo atual e no instante de tempo anterior, respectivamente;
- $A_{t-1}$  é o fator de variação dinâmico da partícula, no instante de tempo anterior;
- $dW_t$  representa um ruído;
- $\Delta t$  é a variação de tempo;
- $\exp$  realiza o mapeamento para a base de Rodrigues.

Em [10] o fator  $dW_t$  é definido a partir de uma função gaussiana, que garante uma distribuição normal dos seus valores. Isso faz com que a variação fique centrada na pose inicial, e que os menores deslocamentos tenham uma maior probabilidade de serem adotados. Já o parâmetro  $A_{t-1}$  não é utilizado nesta técnica, fazendo com que a variação leve em conta apenas deslocamentos aleatórios, não considerando os deslocamentos realizados anteriormente. Este tipo de abordagem é chamada de passo aleatório. A Figura 9 ilustra as diferentes partículas geradas a partir do passo aleatório.



**Figura 9. As poses das partículas (destacadas em verde).**

A técnica descrita em [12] difere da anterior quanto ao parâmetro  $A_{t-1}$ . Neste trabalho este fator representa o vetor velocidade, que é calculado a partir da fórmula a seguir:

$$A_{t-1} = \lambda_a * \log\left(\frac{X_{t-1}}{X_{t-2}}\right), \quad (5)$$

onde:

- $\lambda_a$  é uma constante, que representa a relevância do vetor velocidade; valores mais altos são indicados quando a movimentação da câmera é mais regular;
- $X_{t-1}$  e  $X_{t-2}$  representam as poses da partícula no instante de tempo anterior e no instante de tempo antes do anterior, respectivamente.

Com a utilização do vetor de velocidade a etapa de propagação vai levar em conta o deslocamento que a partícula teve no instante anterior. Isso faz com que a propagação da partícula siga o deslocamento do modelo a ser rastreado.

### 3.3. Avaliação

A avaliação é a etapa responsável por calcular a qualidade da pose da partícula. Quanto mais precisa é a pose, mais correto será o posicionamento dos elementos virtuais na cena. O erro é o fator que determina a qualidade da partícula, representando a distância entre o resultado obtido e o resultado esperado, também conhecido como residual. O termo utilizado para calcular a qualidade da partícula é o *likelihood*, sendo o seu valor inversamente proporcional ao erro. Quanto maior o seu valor, mais precisa é a pose, assumindo valores no intervalo entre zero e um.

Na técnica descrita em [10], o erro é calculado a partir da menor distância entre amostras das arestas visíveis do modelo projetadas na tela, utilizando a pose da partícula e os pontos de arestas da imagem. As arestas são identificadas através do filtro de Sobel, aplicado nas direções  $x$  e  $y$  com o mesmo peso.

Um mapa de distância é construído a partir do resultado do filtro calculado para a imagem. Para isso, é preciso definir um limiar (*threshold*) para determinar os pontos que serão utilizados ou descartados neste mapa de distância.

A Figura 10 mostra os resultados da aplicação do filtro de Sobel na imagem base e o mapa de distância gerado. Quanto mais clara é a cor do *pixel* no mapa, menor é a distância para um ponto da aresta.

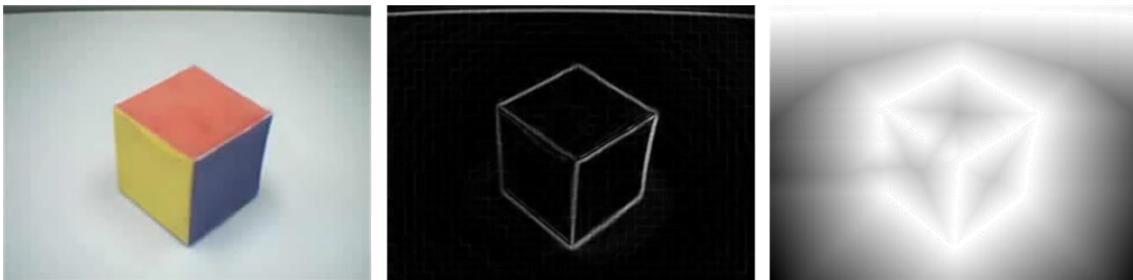


Figura 10. Imagem base (esquerda), aplicação do filtro de Sobel nas direções  $x$  e  $y$  (meio) e mapa de distância (direita).

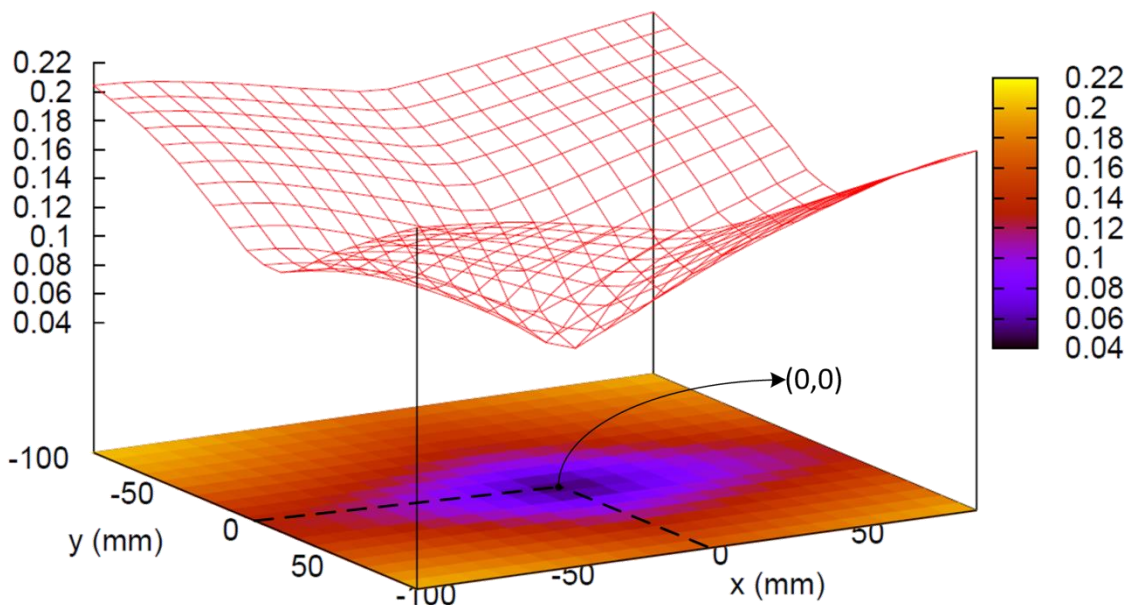
Após o cálculo do mapa de distância, amostras das arestas projetadas na tela são criadas para o cálculo do erro, que é definido como:

$$e_k = \frac{1}{N} \sum_{i=1}^N d_i, \quad (6)$$

onde:

- $e_k$  é o erro da partícula de índice  $k$ ;
- $N$  é a quantidade de amostras geradas;
- $d_i$  é o valor obtido do mapa de distância na posição da amostra de índice  $i$ .

Esta abordagem é bastante rápida comparada com outras técnicas, por não possuir uma correspondência entre o ponto projetado e o ponto de menor distância, mas isso faz com que o valor do erro seja aproximado. Apesar disto, foi observado em [10] que esta função pode ser satisfatória por ter um comportamento mostrado na Figura 11, que demonstra a presença de um mínimo centrado na pose correta e o comportamento de elevação do erro com translações nas direções  $x$  e  $y$  no plano da imagem.



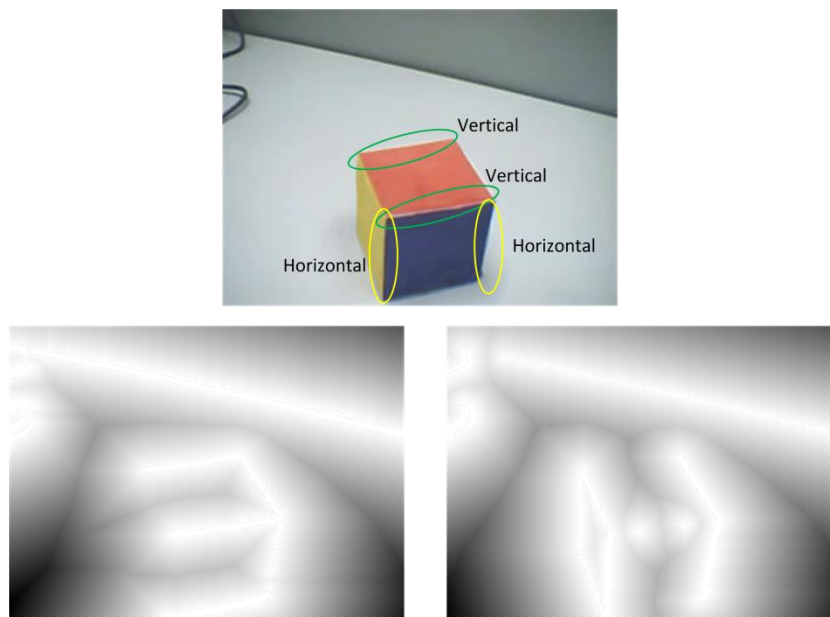
**Figura 11. Função de distância comparando translações em  $x$  e  $y$  da tela. O ponto  $(0,0)$  corresponde à pose correta da câmera. Figura extraída de [10].**

Em [13] é apresentada uma forma de utilizar a informação da textura para o rastreamento do modelo, que consiste em adicionar as arestas da textura às arestas do modelo a ser rastreado. Na Figura 12 pode ser observada a diferença entre as arestas do modelo e as da textura. Neste caso, a utilização apenas das arestas do modelo tornaria o rastreamento mais complexo, por se tratar de um objeto bastante texturizado e com poucas arestas.



**Figura 12. Arestas do modelo e arestas da textura. Figura extraída de [13].**

Para melhorar a correspondência entre os pontos amostrados das arestas visíveis projetadas na tela e os pontos das arestas, as arestas são classificadas em sua direção, vertical ou horizontal, também utilizando o filtro de Sobel. Quando for calculada a distância de um ponto, esta será tirada do mapa de distância com a direção da sua aresta de origem, como pode ser visto na Figura 13.



**Figura 13. Exemplo de arestas horizontais e verticais extraídas da imagem (cima), mapas de distância calculados com o filtro de Sobel na direção horizontal (esquerda) e vertical (direita).**

Como em [11] a representação das partículas é dada pelo posicionamento das juntas, a avaliação das partículas é dada pelo erro de reprojeção das arestas, ou seja, a soma da distância em *pixels* entre todos os pontos 3D das arestas projetados na tela e os seus respectivos pontos na imagem [5]. Esta abordagem mostra ser a melhor solução (das apresentadas aqui) para o problema de correspondência entre os pontos, mas



dependendo da pose da câmera a quantidade de juntas visíveis é pequena, fazendo com que a avaliação não seja tão precisa.

### 3.4. Estimativa

A etapa de estimativa representa a fase de filtragem das partículas, que neste caso consiste em determinar a melhor representação da pose da câmera no instante de tempo atual levando em conta apenas as informações contidas nas partículas. Nas técnicas de [10], [11], [12] e [13] a pose é calculada a partir da média ponderada do conjunto de partículas, em que o peso é dado por:

$$w_k = \frac{l_k}{\sum_{i=1}^N l_i}, \quad (7)$$

onde:

- $w_k$  é o peso da partícula de índice  $k$ ;
- $l_k$  é o *likelihood* da partícula de índice  $k$ ;
- $N$  é a quantidade de partículas presentes no filtro.

### 3.5. Reamostragem

A reamostragem é a etapa responsável por determinar a configuração inicial das partículas que será utilizada no próximo instante de tempo, levando em conta as partículas atuais. Apesar da mudança na configuração do Filtro de Partículas, a quantidade de partículas deve permanecer a mesma durante todo o processo de rastreamento.

Em [10] o próximo conjunto de partículas é definido pela replicação das partículas de maior peso, que foi calculado na etapa de estimativa de pose. Isso faz com que as partículas que obtiveram o melhor resultado estejam mais presentes no próximo instante de tempo. O cálculo para determinar a quantidade de cada partícula da próxima iteração é:

$$Q_i = N * w_i, \quad (8)$$

onde:

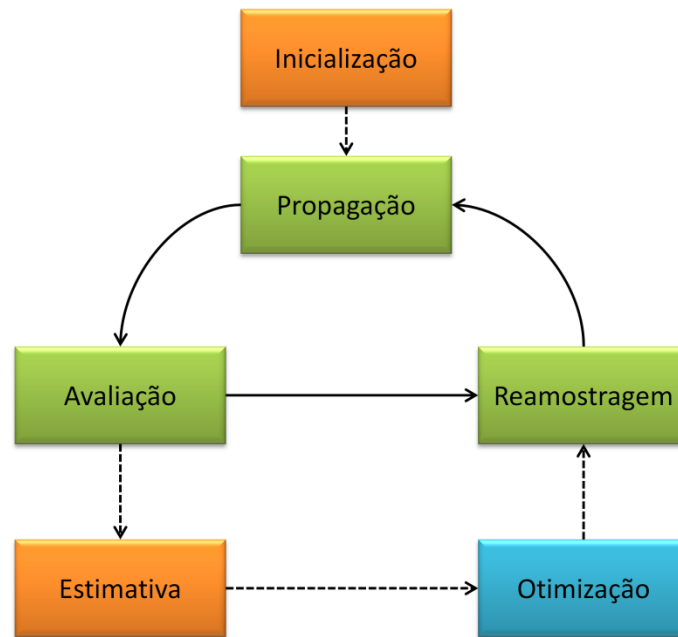
- $Q_i$  é a quantidade de réplicas da partícula de índice  $i$  que vão estar presentes no próximo instante de tempo.

Em [12] é utilizada uma heurística para realizar uma reinicialização do filtro de partículas, em que a reinicialização é feita quando a distância entre as partículas é

menor que um valor definido. Isso é realizado com a finalidade de evitar a estagnação do filtro em um mínimo local. Para realizar a reinicialização é necessário identificar a nova pose da câmera, que é obtida a partir da busca de pontos chave, chamados de *keypoints*, na imagem através do algoritmo conhecido como SURF (*Speeded Up Robust Features*) [14].

## 4. Desenvolvimento da Técnica de Rastreamento baseado em Arestas com Filtro de Partículas

Com base nas abordagens da literatura apresentadas no Capítulo 3, foi desenvolvida uma técnica de rastreamento por arestas com Filtro de Partículas utilizando algumas das técnicas apresentadas. Além disso, uma nova etapa de otimização foi adicionada à técnica para minimizar o erro da pose estimada e consequentemente melhorar o resultado do rastreamento. O diagrama da Figura 14. mostra as etapas que foram implementadas e o fluxo de execução da técnica.



**Figura 14. Diagrama das etapas que compõem o rastreador baseado em arestas com Filtro de Partículas desenvolvido no presente projeto.**

Para o desenvolvimento foi utilizada a linguagem de programação C++, além de bibliotecas gráficas, matemáticas e de processamento de imagens (OpenGL [15], VXL [16], OpenCV [17] e DevIL [18]), para a utilização de funções básicas disponíveis. A escolha das bibliotecas ocorreu pela popularidade de sua utilização na área e por serem bibliotecas de código aberto, permitindo sua utilização e modificação do código utilizado na biblioteca.

### 4.1. Inicialização

Para inicializar o rastreador, o sistema recebe como entrada as informações da câmera correspondentes aos parâmetros intrínsecos e à pose da primeira imagem obtida

da câmera. A pose inicial é replicada no Filtro de Partículas até que seja completada a quantidade de partículas a serem utilizadas durante todo o processo assim como apresentado o processo de [10], presente na Subseção 3.1. O modelo 3D do objeto a ser rastreado durante o processo, com a informação dos pontos 3D dos vértices e as arestas formadas por estes pontos, também é requerido como entrada durante a inicialização do sistema.

## 4.2. Propagação

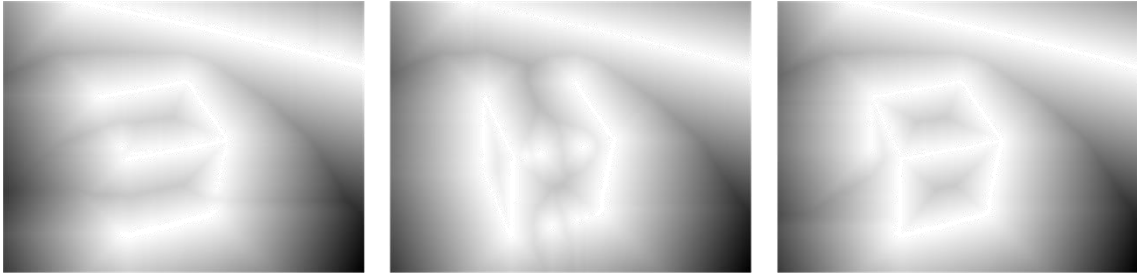
A propagação do Filtro de Partículas consiste no passo aleatório descrito na Subseção 3.2. Para isso, a matriz de rotação é parametrizada pela fórmula de Rodrigues para garantir a independência entre os parâmetros. O vetor de translação recebe o ruído diretamente, sem parametrização. A função utilizada para gerar o ruído é uma função gaussiana centrada em zero, com variância constante de 0,0001, como proposto por [10].

## 4.3. Avaliação

Foram implementadas duas abordagens para a etapa de avaliação das partículas. As duas utilizam informação de mapa de distância, gerado a partir da aplicação do filtro de Sobel na imagem obtida pela câmera.

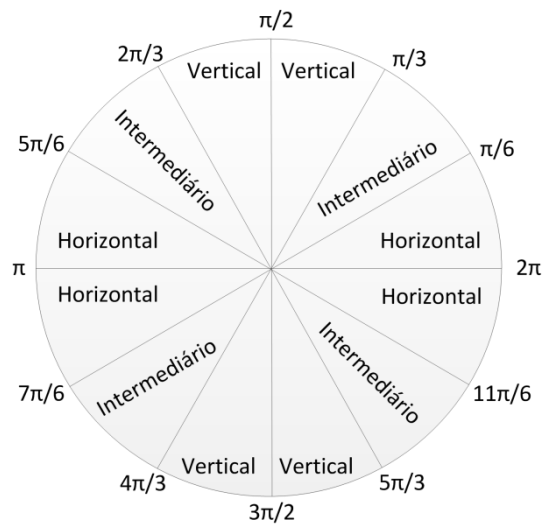
A primeira abordagem aplica um filtro de Sobel nas direções horizontal e vertical. O erro é calculado como a soma dos valores obtidos do mapa de distância na posição da projeção dos pontos 3D amostrados das arestas visíveis do modelo, utilizando a pose da partícula. Para identificar as arestas visíveis é realizado um teste de visibilidade, em que um modelo é desenhado em um *buffer* de OpenGL e é verificado quais arestas do modelo estão presentes nesta imagem virtual.

A segunda abordagem desenvolvida utiliza três mapas de distância, um deles gerado pela aplicação do filtro de Sobel na direção horizontal, o segundo na vertical e o terceiro nas duas direções. Os três mapas de distância podem ser observados na Figura 15.



**Figura 15. Mapas de distância. Horizontal (esquerda), vertical (meio), ambas direções (direita).**

Para o cálculo do erro, os pontos das arestas projetadas do modelo que tem a direção horizontal são avaliados com base no mapa de distância horizontal, as arestas verticais com o mapa de distância vertical e as intermediárias são calculadas com o mapa de distância nas duas direções. Para determinar o sentido das arestas, foi calculado o ângulo que a aresta forma com o vetor  $(1,0)$ . A partir deste valor, foi possível classificar a aresta pelo setor em que ela se encontra, como pode ser observado na Figura 16.



**Figura 16. Classificação das arestas a partir do ângulo.**

#### 4.4. Estimativa

Dois alternativas para o cálculo da pose da câmera foram analisadas neste trabalho. A primeira consiste na forma apresentada em [10], com a utilização da média ponderada das poses das partículas, com o peso proporcional ao *likelihood* de cada partícula. A segunda utiliza a moda do conjunto de partículas, que consiste na pose com menor erro, ou seja, a de melhor *likelihood*.

## 4.5. Otimização

Em [19] é apresentada uma etapa de otimização que consistem em utilizar um algoritmo de minimização de funções, chamado Levenberg-Marquardt [20], com a finalidade de minimizar o erro de reprojeção da pose da partícula. Este algoritmo necessita de um conjunto de parâmetros iniciais para realizar a minimização, logo a pose estimada é utilizada como entrada. Esta etapa foi acrescentada no fluxo de execução do Filtro de Partículas para melhorar a qualidade da pose calculada.

Um problema identificado no uso deste algoritmo é o tempo de execução, visto que este tipo de algoritmo geralmente é mais complexo e acaba ocupando uma boa parcela do tempo do processamento do rastreador. Por outro lado, a qualidade do resultado obtido é bastante significativa, justificando a utilização deste tipo de alternativa.

## 4.6. Reamostragem

Esta etapa é responsável por selecionar o conjunto de partículas que vão ser utilizadas no próximo instante de tempo. O conjunto é dividido em duas partes iguais, um dos conjuntos são utilizadas apenas réplicas da pose estimada, pelo fato de ter passado pelo processo de otimização esta pose tem uma maior chance de ter uma influência positiva no próximo instante de tempo. Na outra metade vão ser escolhidas as partículas do conjunto atual, de forma que as que tiveram a melhor avaliação (maior *likelihood*) tem maior probabilidade de serem escolhidas.

## 5. Resultados

A técnica desenvolvida foi avaliada de acordo com alguns critérios, para verificar sua adequabilidade para o rastreamento de objetos em aplicações de realidade aumentada. Os resultados coletados que são apresentados na sequência foram adquiridos em um computador *desktop*, com o processador Core i7-3770 3,4 GHz com 4GB memória RAM e placa de vídeo GeForce GTS 450. Os testes foram realizados a partir de uma filmagem de 500 *frames*, com resolução de 320 *pixels* por 240 *pixels*, em formato RGB, em um ambiente real. O modelo a ser rastreado é um cubo e o objeto virtual a ser inserido na cena é a sua representação em *wireframe* (arestas de cor vermelha).

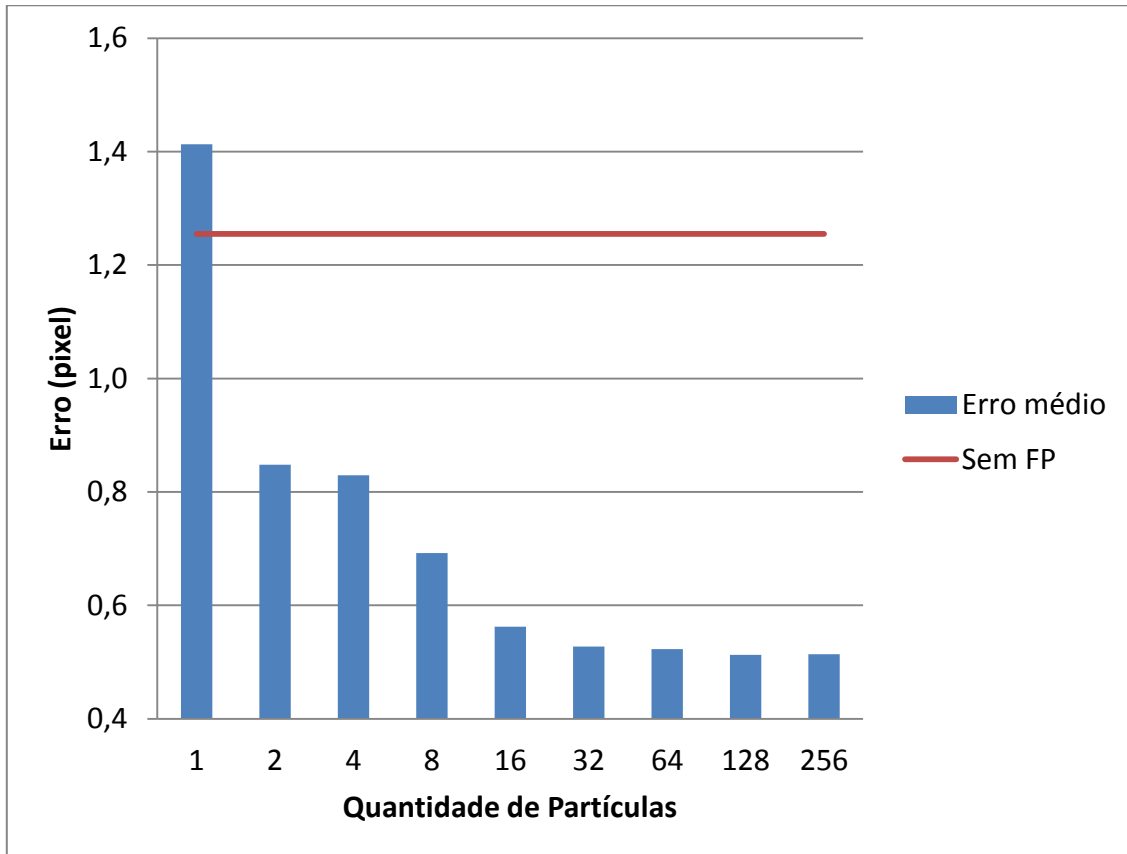
Para a comparação dos resultados obtidos pelo Filtro de Partículas desenvolvido neste trabalho, foi necessária a implementação da técnica descrita em [19]. Esta técnica serve de base de comparação, como uma técnica de rastreamento baseado em arestas sem Filtro de Partículas, enquanto a técnica deste projeto utiliza o Filtro de Partículas.

A métrica utilizada para a medição da qualidade do rastreamento foi a do erro de reprojeção, obtido do retorno do algoritmo de minimização de Levenberg-Marquardt. O valor desta métrica representa a distância média em *pixels* entre os pontos projetados na tela e os seus respectivos pontos na imagem obtida da câmera. Segundo [21], valores de erro de reprojeção médio maiores que 2,5 por ponto são considerados perda de rastreamento. Este parâmetro será utilizado como erro máximo aceitável durante as análises.

Para os testes de velocidade de execução (desempenho) do processo completo, é analisada a quantidade de quadros que são processados por segundo, do inglês *frames per second* – FPS. Quanto maior a quantidade de FPS, maior a rapidez do algoritmo. Para que a técnica seja aplicável em tempo real, deve ter uma taxa média de 30 FPS, que é a taxa aproximada de informação que os olhos humanos conseguem processar. Para a comparação do desempenho das etapas do algoritmo é utilizado o tempo médio de processamento, em milissegundos.

## 5.1. Qualidade do Rastreamento

Os dados presentes no Gráfico 1 mostram o erro médio calculado para variações da quantidade de partículas utilizadas pelo Filtro de Partículas, durante o decorrer de todo o vídeo base.



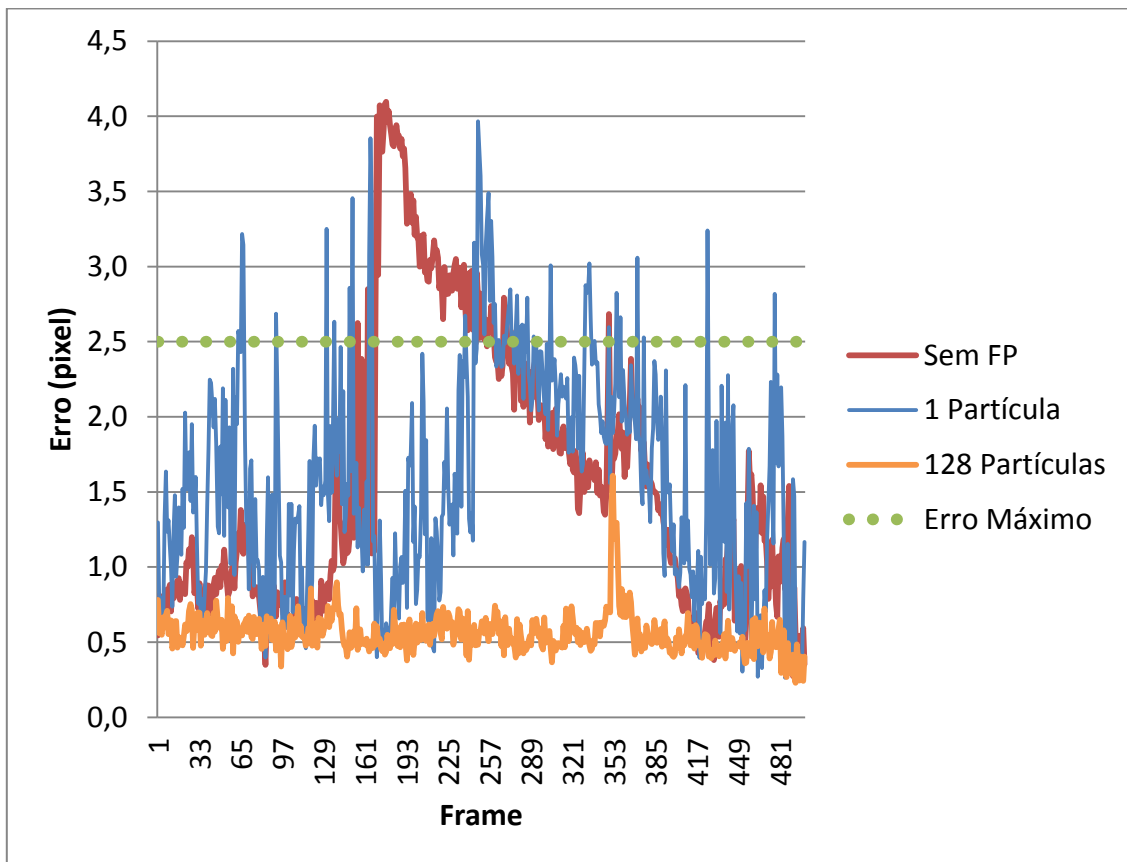
**Gráfico 1. Qualidade do rastreamento perante a quantidade de partículas.**

Pode ser observado que a única configuração que obteve um erro médio maior do que o algoritmo sem Filtro de Partículas foi a que utilizava uma partícula. Além disso, o erro médio diminuiu com o aumento da quantidade de partículas e estabiliza com valores maiores que 128 partículas. Este comportamento é esperado, uma vez que o erro obtido com esta quantidade de partículas é relativamente pequeno (próximo de 0,5 *pixel*) e quanto mais partículas utilizadas, maior a probabilidade de conseguir uma pose mais correta.

O Gráfico 2 ilustra o valor do erro, ao longo do rastreamento, para configurações com 1 partícula, 128 partículas e sem Filtro de Partículas. É possível observar que a configuração com 1 partícula varia bastante o erro durante o processo, confirmando uma instabilidade no rastreamento. Este comportamento é explicado pelo fato de utilizar



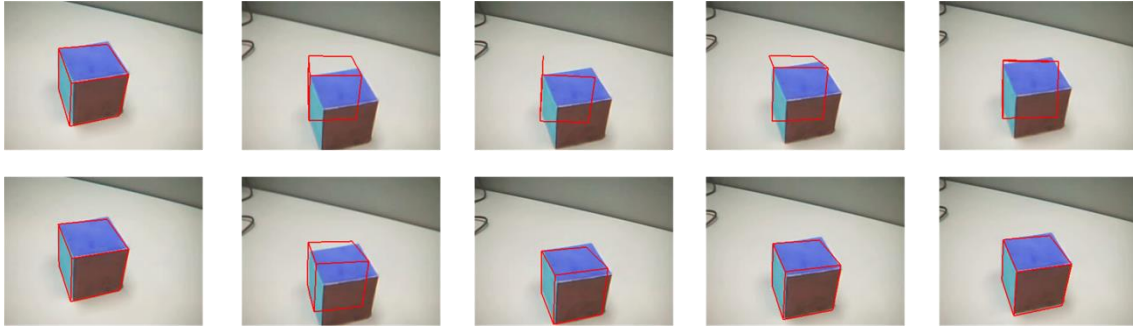
apenas uma hipótese para a pose, fazendo com que o poder de previsão do Filtro de Partículas seja limitado.



**Gráfico 2. Qualidade do rastreamento ao longo da sua execução.**

Além disso, devido à oclusão de uma aresta do cubo, a implementação sem Filtro de Partículas tem um erro bastante elevado por volta do *frame* 180, com um valor aproximado de 4 *pixels*, bem acima do erro máximo de 2,5 *pixels*. Enquanto a configuração com 128 partículas tem o valor máximo abaixo de 2 *pixels* durante todo o rastreamento.

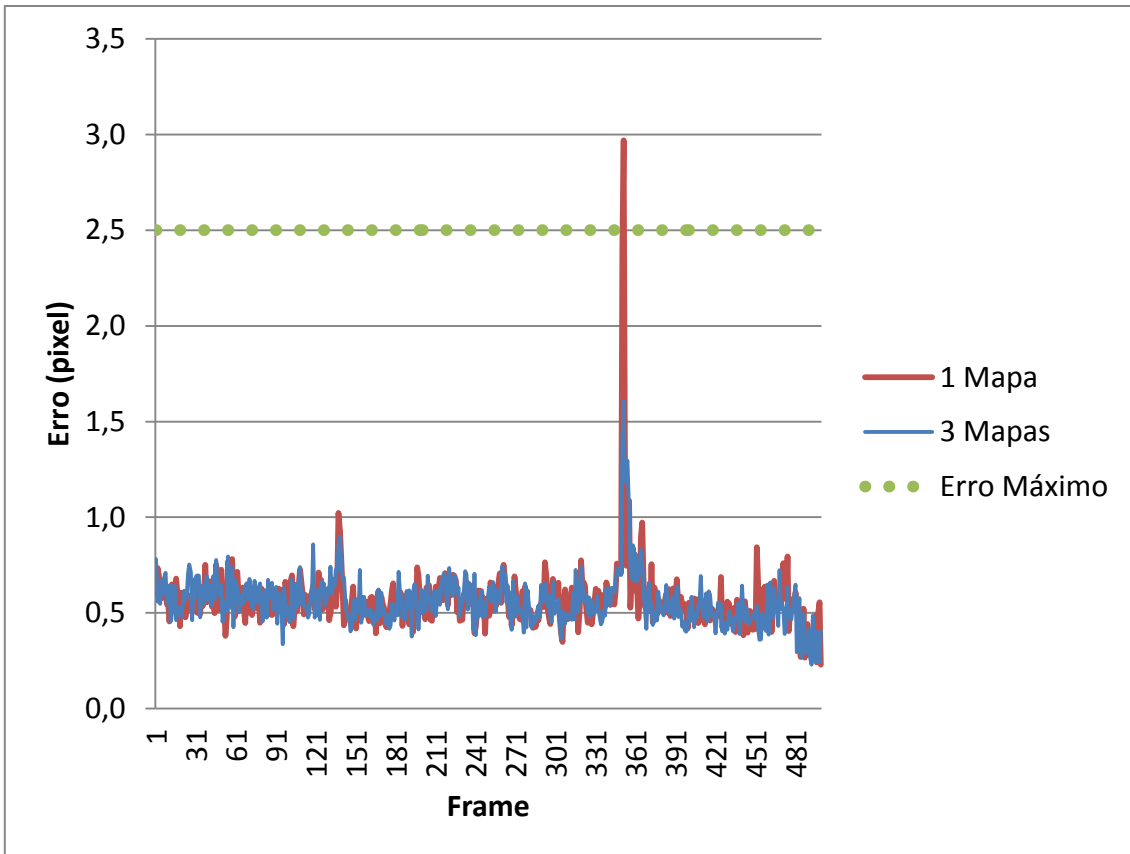
O maior valor obtido para o erro para a configuração de 128 partículas ocorre por volta do *frame* 350 de aproximadamente 1,6 *pixels*, como pode ser observado no Gráfico 2. Este comportamento é explicado com o fato de ocorrer um deslocamento muito rápido da câmera nesta sequência de *frames*. Esta sequência foi separada para ser realizada uma análise visual da falha do rastreamento da técnica sem e com a utilização do Filtro de Partículas. O resultado pode ser observado na Figura 17, com algumas imagens obtidas durante o rastreamento.



**Figura 17. Sequência de *frames* durante a falha de rastreamento: resultado sem Filtro de Partículas (cima), resultado com Filtro de Partículas (baixo).**

A técnica que utiliza o Filtro de Partículas resulta em um erro menor na pose calculada e consegue corrigir a pose em pouco tempo, o que não ocorre com a outra técnica sem Filtro de Partículas. Este comportamento é explicado pela característica temporal do Filtro de Partículas e pelo fato de criar várias hipóteses para a pose da câmera.

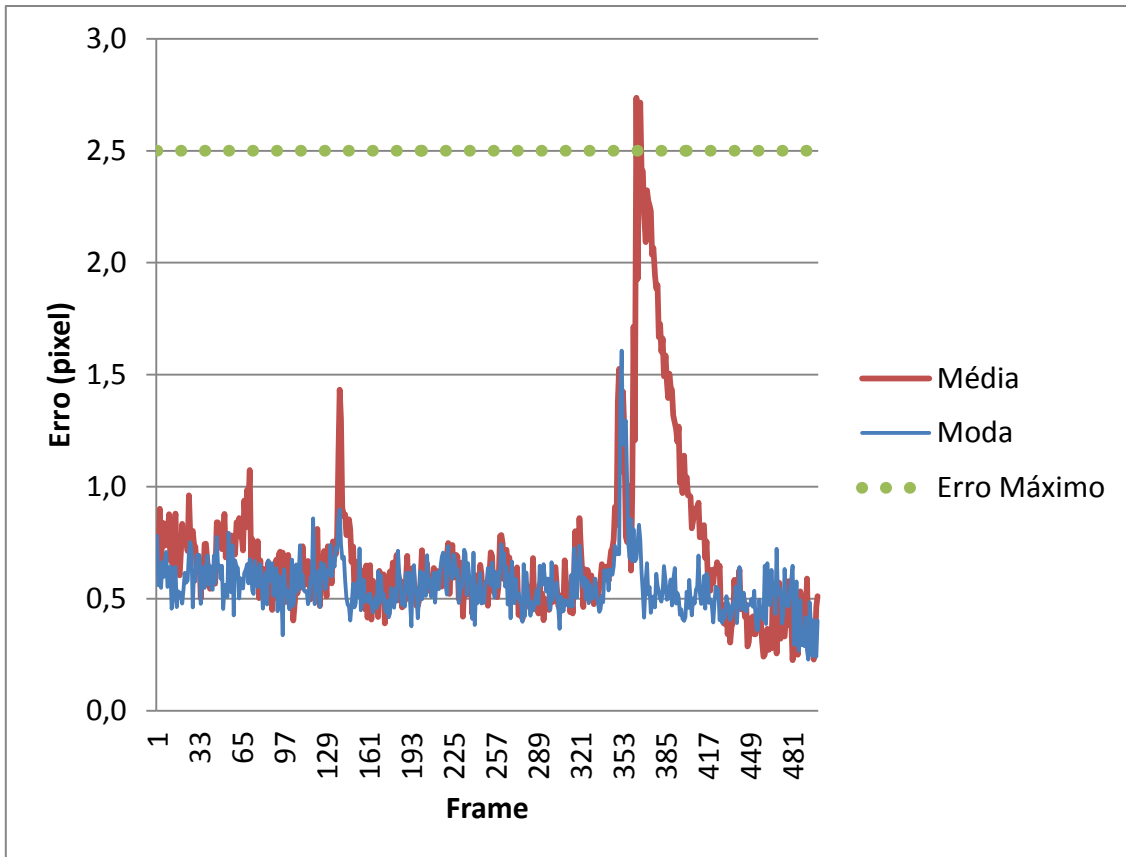
Outra configuração testada é a forma de avaliação utilizada para o Filtro de Partículas. Para isso foi utilizada uma configuração com 128 partículas, variando apenas a forma de avaliação das partículas, uma utilizando 1 mapa de distância, e outra utilizando 3 mapas de distância (nas direções  $x$ ,  $y$  e em ambas). O resultado do erro ao longo do rastreamento pode ser observado no Gráfico 3.



**Gráfico 3. Qualidade do rastreamento perante a variação na avaliação do Filtro de Partículas.**

Os resultados são semelhantes, a não ser pelo erro durante a movimentação brusca da câmera mencionada anteriormente, em que a utilização dos 3 mapas faz com que o erro associado à pose da câmera calculada seja menor, enquanto a que utiliza apenas um mapa tenha um erro superior ao máximo. O menor erro é justificado pela melhor correspondência entre as arestas projetadas e as presentes na imagem, como foi explicado na Subseção 4.3.

A última configuração avaliada é o tipo de estimativa de pose da câmera. Duas formas foram analisadas, uma utilizando a média da pose das partículas e a outra utilizando a moda da pose das partículas. O valor do erro durante o rastreamento pode ser observado no Gráfico 4.

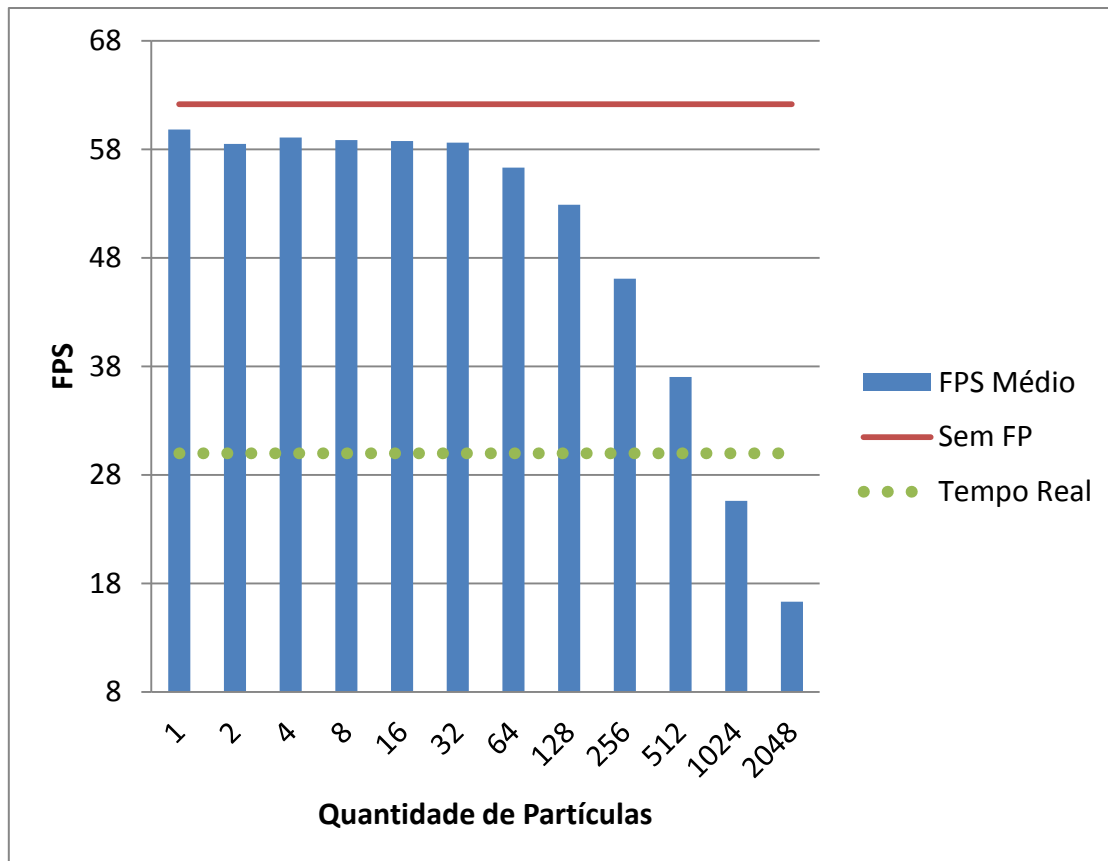


**Gráfico 4. Qualidade do rastreamento perante o tipo de estimativa da pose da câmera.**

De modo geral, a utilização da moda se mostrou mais eficiente, quando comparada com a média das partículas. Outro fato a ser observado na configuração que utiliza a média é que a variação brusca do *frame* 350 gerou um erro superior ao máximo, além de resultar em uma elevação do erro nos *frames* seguintes. Os dados mostram que a utilização da moda ao invés da média resultou em uma melhora de cerca de 19% no caso de teste utilizado. Este comportamento pode ser explicado pelo fato da média ter uma velocidade de resposta mais lenta que a moda, devido a influência de poses de maior erro na estimativa da pose da câmera.

## 5.2. Desempenho

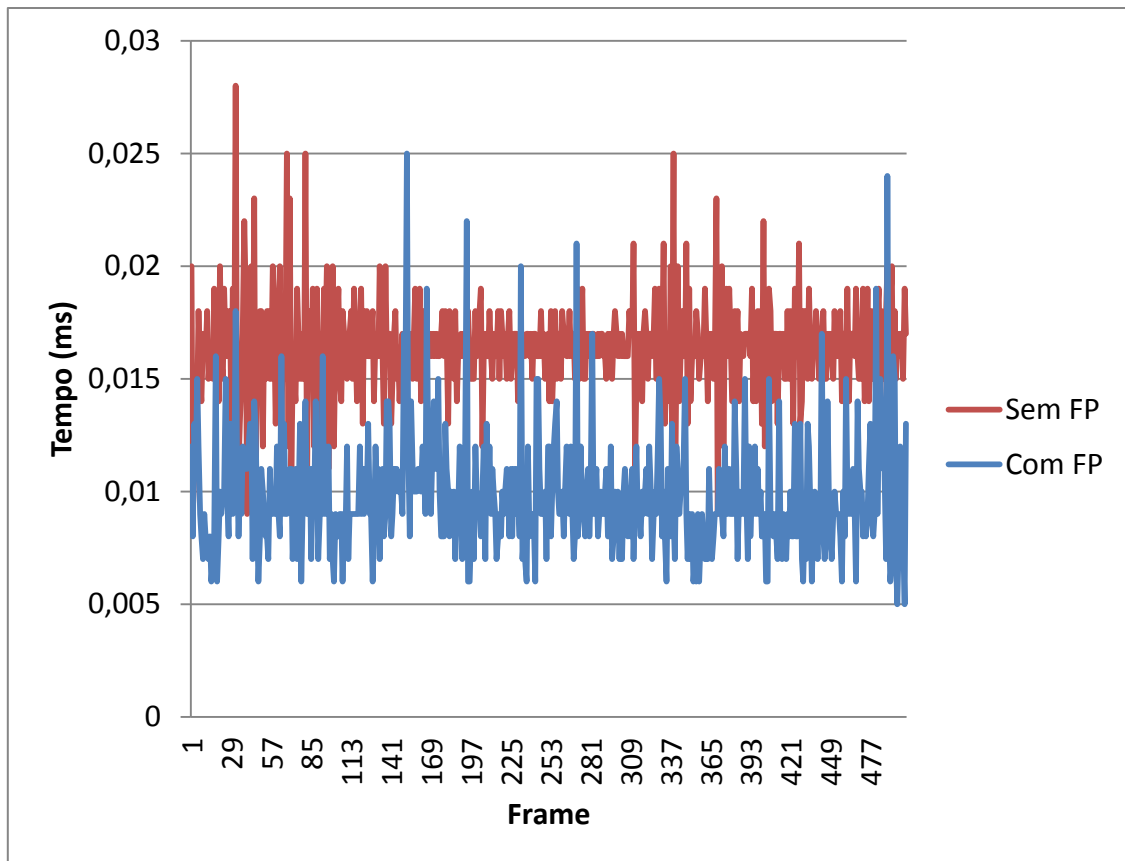
Além da análise da qualidade do rastreamento, foi realizada uma análise do tempo de execução da técnica de rastreamento baseado em arestas com Filtro de Partículas, variando a quantidade de partículas. O resultado do rastreamento sem Filtro de Partículas e os resultados obtidos utilizando diferentes quantidades de partículas podem ser observados no Gráfico 5.



**Gráfico 5. Velocidade de execução perante a quantidade de partículas.**

Os resultados mostram que o tempo de execução para a configuração de 128 partículas é bastante satisfatório quando comparado com a restrição de 30 FPS para aplicações de tempo real. Além disso, para uma quantidade de partículas elevada, a técnica proposta tem um bom desempenho, conseguindo ser executado em tempo real com quantidades superiores a 512 partículas.

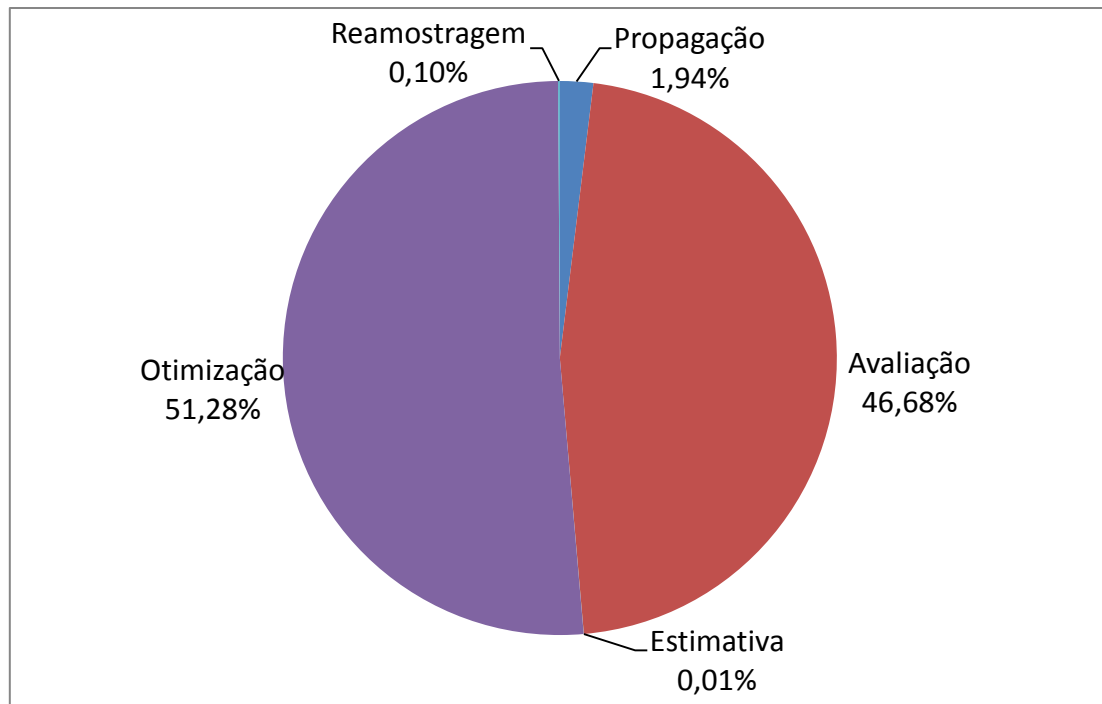
O algoritmo utilizado para a etapa de otimização da pose utiliza uma pose inicial que, quanto mais próxima da pose correta mais rápido o algoritmo vai ser executado. O Gráfico 6 mostra o tempo de execução do algoritmo de minimização da pose sem a utilização do Filtro de Partículas e utilizando uma configuração com 128 partículas, durante o rastreamento da pose ao longo dos *frames*.



**Gráfico 6. Tempo decorrido durante o algoritmo de minimização.**

Na configuração adotada a técnica com Filtro de Partículas diminuiu o tempo de execução do algoritmo de minimização, representada pela etapa de otimização, em média de aproximadamente 40%, durante o decorrer do rastreamento. Esta diferença é resultado da escolha de uma melhor pose para iniciar o algoritmo de minimização, em que quanto mais próxima a pose está da resposta esperada mais rápido o algoritmo de minimização vai ser executado.

A distribuição do tempo de cada fase da técnica com Filtro de Partículas pode ser observada no Gráfico 7.

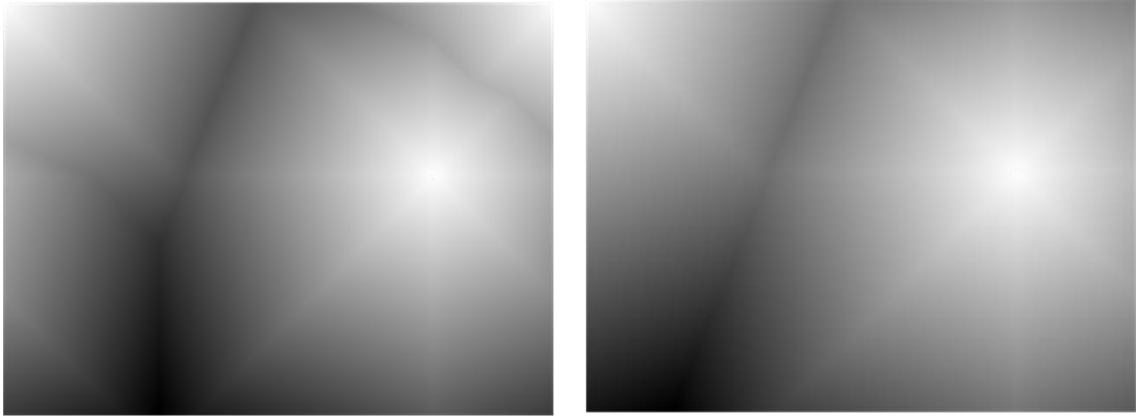


**Gráfico 7. Proporção de tempo entre as etapas do Filtro de Partículas.**

É possível observar que a etapa de otimização é a mais custosa, mostrando a relevância da redução do tempo de execução mencionada no Gráfico 6. A etapa de avaliação também tem uma grande parcela no tempo de execução do algoritmo. Este tempo é resultado do processamento dos 3 mapas de distância e do cálculo do erro de reprojeção de todas as partículas.

### **5.3. Resultados Adicionais**

Durante o desenvolvimento do projeto, foi identificado um problema na função de criação de mapa de distâncias (`vil_distance_transform`) da biblioteca gráfica VXL. Esta função, por ser parte de uma biblioteca de código aberto, pode ser analisada e identificou-se o problema. A correção da função foi enviada para a equipe de desenvolvedores da VXL, que confirmaram o problema e inseriram a modificação que está presente na versão atual da biblioteca (versão 1.17). A Figura 18 mostra o resultado da função quando passados como entrada os pontos (0,0) e (250,100), antes e após a correção do problema.



**Figura 18. Resultado da criação do mapa de distâncias errado (esquerda) e corrigido (direita).**



## 6. Conclusão

A técnica de rastreamento de arestas utilizando Filtro de Partículas se mostrou bastante interessante, tanto com relação ao valor do erro obtido, como à velocidade de processamento. A combinação das técnicas de [10], [11], [13] e [19], além de variações na avaliação das partículas e na estimativa da pose foi o diferencial para o resultado que foi obtido.

A configuração do Filtro de Partículas com 128 partículas, utilizando 3 mapas de distância e com a estimativa sendo a moda das partículas teve uma taxa de atualização média superior a 50 FPS e com o erro médio de aproximadamente 0,5 *pixel*. Por outro lado, o algoritmo sem Filtro de Partículas resulta em um erro médio de 1,25 *pixel*.

A contribuição feita para a biblioteca VXL teve um retorno positivo da comunidade de desenvolvimento, por se tratar de uma função importante e por ter sido proposta uma solução para o problema encontrado.

### 6.1. Trabalhos Futuros

Com relação à etapa de propagação das partículas, pretende-se investigar a proposta realizada por [12] de utilizar o vetor de velocidade. Esta abordagem pode ser uma boa estratégia para problemas decorrentes de variações/movimentações bruscas na câmera.

A utilização da informação de juntas, proposta por [11] mostra ser muito eficiente para relacionar a imagem obtida com a pose da câmera. Esta pode ser uma solução para o problema da necessidade de uma pose inicial para o rastreamento, assim como uma alternativa para o caso do rastreamento ser perdido ao longo do processo.

A criação de uma aplicação de realidade aumentada que utilize a técnica proposta serviria como um caso de testes real para a técnica desenvolvida. Isso faria com que novos problemas pudessem ser descobertos e assim melhorar ainda mais a técnica.

A etapa de avaliação é um dos gargalos da técnica apresentada, representando quase metade do tempo de processamento. Uma alternativa para melhorar o desempenho desta etapa é utilizar processamento paralelo em placa gráfica (GPU – *Graphics Processing Unit*) [22], uma vez que o cálculo da avaliação de cada partícula é

independente das demais partículas sendo otimizável em arquiteturas paralelas como a GPU.

## Referências

- [1] P. Milgram and F. Kishino, “A Taxonomy of Mixed Reality Visual Displays”, *IEICE Transactions on Information and Systems*, vol. E77-D, no. 12, pp. 1321–1329, 1994.
- [2] R. T. Azuma, “A Survey of Augmented Reality”, *Presence: Teleoperators and Virtual Environments*, pp. 355–385, 1997.
- [3] V. Lepetit and P. Fua, “Monocular Model-Based 3D Tracking of Rigid Objects : A Survey”, *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2004.
- [4] H. Scharr, “Optimal Second Order Derivative Filter Families for Transparent Motion Estimation”, *15th European Signal Processing Conference*, Poznan, Poland, pp. 302–306, 2007.
- [5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, no. June. Cambridge, 2000, pp. 1–240.
- [6] V. Teichrieb, J. Lima, E Apolinário, T. Farias, M. Bueno, J. Kelner and I. Santos, “A Survey of Online Monocular Markerless Augmented Reality”, *International Journal of Modeling and Simulation for the Petroleum Industry*, vol. 1, no. 1, pp. 1–7, 2007.
- [7] J. Lima, F. Simoes, L. Figueiredo and J. Kelner, “Model based markerless 3D tracking applied to augmented reality”, *SBC Journal on 3D Interactive Systems*, vol. 1, no. 1, pp. 2–15, 2010.
- [8] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, 1993.
- [9] Z. Chen, “Bayesian Filtering : From Kalman Filters to Particle Filters, and Beyond”, *Statistics*, pp. 1–69, 2003.
- [10] C. Teulière, E. Marchand, and L. Eck, “Using multiple hypothesis in model-based tracking”, *Robotics and Automation (ICRA)*, pp. 4559–4565, 2010.
- [11] H. Álvarez, I. Aguinaga, and D. Borro, “Providing guidance for maintenance operations using automatic markerless Augmented Reality system”, *10th IEEE International Symposium on Mixed and Augmented Reality*, Basel, Switzerland, pp. 181–190, Oct. 2011.
- [12] C. Choi and H. I. Christensen, “Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge

- features”, *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, Mar. 2012.
- [13] T. Mörwald, M. Zillich, and M. Vincze, “Edge tracking of textured objects with a recursive particle filter,” *19th International Conference on Computer Graphics and Vision*, Pilsen, Czech Republic, pp. 96–103, 2009.
- [14] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, “SURF: Speeded Up Robust Features”, *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.
- [15] OpenGL, “Biblioteca de processamento gráfico”. Disponível em: <http://www.opengl.org/>.
- [16] VXL, “Biblioteca de visão computacional”. Disponível em: <http://vxl.sourceforge.net/>.
- [17] OpenCV, “Biblioteca de visão computacional”. Disponível em: <http://opencv.org/>.
- [18] DevIL, “Biblioteca de processamento de imagens”. Disponível em: <http://openil.sourceforge.net/>.
- [19] F. Simões, “Realidade Aumentada Sem Marcadores Baseada em Arestas, um Estudo de Caso”, *Trabalho de Graduação, Centro de Informática - UFPE*, 2008.
- [20] K. Levenberg, “A Method for the Solution of Certain Non-linear Problems in Least Squares”, *Quarterly of Applied Mathematics*, pp. 164–168, Jul. 1944.
- [21] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab, “A dataset and evaluation methodology for template-based tracking algorithms”, *8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 145–151, Oct. 2009.
- [22] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, J. Phillips “GPU Computing”, *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008.