UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

RASTREAMENTO DE OBJETOS 3D EM IMAGENS RGB-D USANDO OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

JOSÉ GUEDES DOS SANTOS JÚNIOR

RECIFE, FEVEREIRO/2018

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

RASTREAMENTO DE OBJETOS 3D EM IMAGENS RGB-D USANDO OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

JOSÉ GUEDES DOS SANTOS JÚNIOR

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade Federal Rural de Pernambuco como requisito parcial para conclusão do curso.

Orientador: Prof. Wilson Rosa de Oliveira Júnior, PhD

Coorientador: Prof. João Paulo Silva do Monte Lima, PhD

RECIFE, FEVEREIRO/2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

S237r Santos Júnior, José Guedes dos.
Rastreamento de objetos 3D em imagens RGB-D usando otimização por enxame de partículas / José Guedes dos Santos Júnior. – Recife, 2018.
77 f.: il.

Orientador(a): Wilson Rosa de Oliveira Júnior.
Coorientador(a): João Paulo Silva do Monte Lima.
Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE, 2018.
Inclui referências e apêndice(s).

1. Rastreamento 3D sem marcadores 2. Imagens RGB-D 3. Otimização por exame de partículas 4. Processamento em GPU I. Oliveira Júnior, Wilson Rosa de, orient. II. Lima, João Paulo Silva do Monte, coorient. III. Título

CDD 004

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO DE MESTRADO ACADÊMICO DE

JOSÉ GUEDES DOS SANTOS JÚNIOR

RASTREAMENTO DE OBJETOS 3D EM IMAGENS RGB-D USANDO OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A comissão examinadora, composta pelos professores abaixo, sob a presidência do primeiro, considera o candidato José Guedes dos Santos Júnior _____.

Orientador:

_____
Prof. Wilson Rosa de Oliveira Júnior, PhD
Universidade Federal Rural de Pernambuco

Banca Examinadora:

_____
Prof. Wilson Rosa de Oliveira Júnior, PhD
Universidade Federal Rural de Pernambuco

_____
Prof. Péricles Barbosa Cunha de Miranda, PhD
Universidade Federal Rural de Pernambuco

_____
Prof. João Marcelo Xavier Natário Teixeira, PhD
Universidade Federal de Pernambuco

DEDICATÓRIA

Dedico este trabalho a Maria B.

AGRADECIMENTOS

Agradeço a minha esposa Kaline e a meu filho Heitor pelo apoio, compreensão, companheirismo e carinho ao longo de todas as etapas deste curso. Obrigado, vocês me motivam demais.

Agradeço a toda minha família, em especial a minha mãe Maria Bernardete (Dona Beta) e as minhas quatro irmãs (Rosy, Diu, Gê e Jeane) por tudo que vivemos juntos, pelo carinho e dedicação de sempre.

Agradeço a todos que compõem o PPGIA da UFRPE pelo acolhimento, aos professores: foram aulas admiráveis! E particularmente ao professor Wilson Rosa pela oportunidade de cursar o mestrado.

Agradeço ao professor João Paulo Lima pelo seu compromisso e seriedade com este trabalho, pela disponibilidade e enorme paciência em me responder todas as vezes que precisei (acreditem, foram muitas). Nesse curto intervalo de tempo você proporcionou um importante aprendizado.

Agradeço aos amigos Jefferson Azevedo e Cícero Renan pelas dicas e por todo o apoio que me deram no Recife, e ao amigo Sérgio Fideles por me receber, pela convivência e conversas incríveis, depois desses dias minha mente nunca mais voltará ao seu tamanho original.

“A realidade apenas se forma na memória; as flores que hoje me mostram pela primeira vez não me parecem verdadeiras flores.”

Marcel Proust

RESUMO

O termo Realidade Aumentada é usado para especificar os sistemas que possuem a tecnologia de inserir objetos virtuais em cenas reais, permitindo assim aumentar a quantidade de informações presentes no ambiente real original. No resultado final de uma cena filmada com Realidade Aumentada, o grau de naturalidade dessa inserção não está relacionado apenas com a qualidade da renderização dos objetos virtuais, mas também com a precisão com que se conhece a pose dos objetos reais em relação à câmera ao longo da filmagem, isto é, depende também da qualidade do rastreamento desses objetos.

Marcadores artificiais, além de facilitar, podem aumentar a qualidade do rastreamento de objetos, porém em algumas situações nem sempre é possível ou desejável inserir manualmente marcadores na cena que vai ser rastreada. A solução adotada tem sido usar características presentes naturalmente nos objetos pertencentes à cena, esse tipo de rastreamento é chamado de rastreamento sem marcadores. Algumas técnicas de rastreamento sem marcadores usam o conhecimento prévio dos objetos que serão rastreados, isso é feito a partir da obtenção antecipada de modelos virtuais desses objetos. Existem diversos métodos de rastreamento a partir de modelos, alguns deles usam algoritmos de busca e otimização como o filtro de partículas ou a otimização por enxame de partículas para avaliar conjuntos de poses candidatas durante o rastreamento, estes métodos têm mostrado resultados muito bons.

Ao filmar uma cena com uma câmera digital comum, há sempre a perda de informações, pois, além da amostragem e quantização dos pontos, a representação geométrica de um objeto real no plano de imagem da câmera a cada quadro capturado é sempre em 2D. Contudo, a partir de sensores RGB-D é possível construir nuvens de pontos 3D de uma cena, permitindo assim obter uma representação mais fiel dos pontos pertencentes aos objetos do mundo real. Dessa forma, novas técnicas de rastreamento de objetos 3D que usam características extraídas de nuvens de pontos 3D, antes inacessíveis em imagens 2D, têm sido desenvolvidas, proporcionando algoritmos de rastreamento sem marcadores e com 6 graus de liberdade mais precisos.

Com o objetivo de contribuir com as pesquisas atuais relacionadas ao rastreamento sem marcadores de objetos 3D genéricos e com 6 graus de liberdade, este trabalho propõe o uso de otimização por enxame de partículas para lidar com múltiplas hipóteses de pose durante o rastreamento *top-down* a partir de imagens RGB-D e baseado em modelos. O

processamento em GPU foi utilizado no intuito de aprimorar o tempo de execução. A realização de uma série de experimentos revelou uma melhora na precisão obtida pelo método de rastreamento proposto em comparação com outras técnicas baseadas em otimização do estado da arte.

Palavras-chave: rastreamento 3D sem marcadores; imagens RGB-D; otimização por enxame de partículas; processamento em GPU.

ABSTRACT

The term Augmented Reality is used to specify the systems that have a technology of inserting virtual objects in real scenes, allowing an increase in the amount of information present in the former real environment. In the final result of an Augmented Reality scene footage, the degree of naturalness of this insertion is related not only to the rendering quality of the virtual objects, but also to the accuracy with which the pose of the real objects in relation to the camera is known during the footage, that is, it also depends on the quality of the tracking of these objects.

Artificial markers can facilitate and increase the quality of object tracking, however in some situations it is not always possible or desirable to manually insert markers in the scene to be tracked. The solution adopted has been to use features present naturally in the objects belonging to the scene, this type of tracking is called markerless tracking. Some markerless tracking techniques use prior knowledge of the objects to be tracked, this is done by obtaining virtual models of those objects in advance. There are several model-based tracking methods, some of which use search and optimization algorithms such as particle filter or particle swarm optimization to evaluate sets of candidate poses during tracking, these methods have shown very good results.

When capturing a scene with a common digital camera, there is always an information loss, since – besides point sampling and quantization – the geometric representation of a real object in the camera image plane in each captured frame is always in 2D. However, by using RGB-D sensors it is possible to build 3D point clouds of a scene, allowing to obtain a more accurate representation of the points that belong to real world objects. This way, new 3D object tracking techniques that use features extracted from 3D points clouds, previously inaccessible in 2D images, have been developed, allowing more precise 6 degree of freedom markerless 3D tracking algorithms.

In order to contribute with current research related to 6 degree of freedom markerless 3D generic object tracking algorithms, this work proposes the use of particle swarm optimization to handle multiple pose hypotheses during top-down model-based tracking from RGB-D images. GPU processing was utilized with the aim of improving execution time. A series of experiments were performed, which revealed an improvement in accuracy obtained

by the proposed tracking method in comparison to other state of the art optimization-based techniques.

Keywords: markerless 3D tracking; RGB-D images; particle swarm optimization; GPU processing.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1. Introdução

Na Realidade Aumentada (RA) o *rastreamento 3D* consiste em determinar a pose da câmera com seis graus de liberdade (*degree of freedom* – DOF) em relação à cena a cada quadro a partir de informações extraídas do ambiente. Dessa forma é possível determinar a posição de objetos em relação à câmera. De modo geral, uma vez que o rastreamento proporciona que sistemas conheçam a posição de objetos em uma cena, é possível usar essa técnica para realizar diversas tarefas, tais como: a interação de robôs com objetos pertencentes ao ambiente, a inserção precisa de objetos virtuais em imagens de ambientes reais, melhorar a interação homem máquina, dentre outras [1].

Um exemplo de uma aplicação prática e comum do uso de rastreamento em RA é na inserção de informações adicionais às imagens capturadas em transmissões esportivas. Na Figura 1.1 as informações sobre uma partida de futebol americano, tais como a posição da linha de descida (em amarelo), o número de descidas e a quantidade de jardas restantes para a primeira descida (em um cone vermelho e azul), são corretamente inseridas sobre a imagem do campo durante a transmissão, aparentando fazer parte da cena.

Figura 1.1 – Exemplo prático do uso de rastreamento em RA para inserção de informações nas imagens de uma transmissão esportiva. Imagem retirada de [2]**.**

Diferentes abordagens viabilizam a realização do rastreamento de objetos, estas por sua vez podendo ser classificadas em dois grandes grupos: aquelas que usam marcadores na cena e aquelas que não usam [3]. Existem diversas situações em que não é possível ou desejável a inserção de marcadores em uma cena. Para esse tipo de ocasião o rastreamento pode ser feito a partir de características naturais do ambiente ou objetos pertencentes a ele. Muitas dessas características podem ser extraídas de imagens em cores obtidas por câmeras

monoculares convencionais, porém trabalhos recentes [4][5][6][7][8] passaram a introduzir o uso de imagens de sensores RGB-D no rastreamento de objetos.

A partir do uso sensores RGB-D é possível não apenas obter as imagens em cores comuns (aquelas em RGB), mas também uma imagem que registra a profundidade de cada ponto visível da cena em relação à câmera no instante da captura e em tempo real. O uso de imagens RGB-D pode melhorar o rastreamento de objetos, pois permite criar nuvens de pontos 3D que correspondem à parte visível da cena em cada quadro e com isso possibilita a extração e uso de características geométricas antes inacessíveis, tais como coordenadas 3D, curvatura de superfícies, vetores normais, dentre outras.

Existem ainda técnicas de rastreamento que são classificadas segundo o modo como cada pose do objeto é encontrada ao longo da trajetória. Abordagens que extraem as características da imagem para, a partir delas, determinar a pose do objeto rastreado são conhecidas como *bottom-up*. Já nas abordagens *top-down* o problema do rastreamento de objetos 3D é tratado avaliando várias hipóteses de pose a partir de informações da imagem capturada da cena com o objetivo de definir aquela que melhor se aproxima da pose real do objeto a ser rastreado naquele instante. Existem diversas técnicas usadas para avaliar um conjunto de hipóteses de pose. Duas delas, que serão mencionadas neste trabalho, são o filtro de partículas (*Particle Filter* – PF) [6][8] e a otimização por enxame de partículas (*Particle Swarm Optimization* – PSO) [4][9][10][11].

1.1. Definição do Problema de Pesquisa

Em abordagens *top-down* o rastreamento tem sido explorado como um problema de otimização em que o intuito principal é aperfeiçoar a avaliação de um conjunto de hipóteses de pose a partir de características naturais do objeto e da cena. A partir desse contexto, o problema desta dissertação pode ser expresso pela seguinte questão: qual o desempenho e precisão do rastreamento sem marcadores de objetos 3D genéricos e com 6-DOF baseado no uso de PSO como método de avaliação de conjuntos de hipóteses de pose a partir de imagens RGB-D em relação a outras técnicas presentes no estado da arte?

As seguintes hipóteses serão avaliadas no decorrer desta dissertação:

- H1: O PSO pode ser usado como método de otimização em problemas de rastreamento sem marcadores de objetos 3D genéricos em abordagens *top-down*;

- H2: O rastreamento sem marcadores de objetos 3D genéricos usando o método proposto nesta dissertação é tão preciso quanto outros métodos de rastreamento de mesma natureza presentes no estado da arte.

1.2. Objetivos da Pesquisa

O objetivo geral deste trabalho é propor e avaliar o uso do PSO como método de otimização de múltiplas hipóteses de pose durante o rastreamento sem marcadores de objetos 3D genéricos e com 6-DOF baseado em modelos e a partir de imagens RGB-D.

Os objetivos específicos deste trabalho são os seguintes:

- Verificar os métodos de rastreamento sem marcadores relacionados ao tema da pesquisa presentes no estado da arte;
- Pesquisar e definir uma representação matemática do problema de rastreamento, para que o mesmo possa ser traduzido para um PSO;
- Criar um protótipo da técnica de rastreamento sugerida com a finalidade de definir e verificar parâmetros, analisando a viabilidade da proposta através dos resultados encontrados;
- Identificar qual conjunto de parâmetros do PSO melhor se adapta ao problema de rastreamento;
- Implementar a técnica de rastreamento proposta de acordo com viabilidade dos parâmetros definidos no protótipo;
- Avaliar o método proposto através de um conjunto de experimentos, bem como comparar o desempenho da abordagem sugerida em relação às técnicas encontradas na literatura.

1.3. Estrutura da Dissertação

Este trabalho foi escrito em seis capítulos: o Capítulo 2 aborda os conceitos matemáticos utilizados no desenvolvimento da técnica de rastreamento; o Capítulo 3 faz uma breve exposição de como é realizado o rastreamento sem marcadores de objetos 3D baseado em modelos; o Capítulo 4 apresenta e explica em detalhes o método de rastreamento

proposto; o Capítulo 5 expõe a metodologia dos experimentos realizados com o intuito de avaliar a técnica proposta, além de apresentar e discutir os resultados obtidos nesses experimentos; o Capítulo 6 apresenta as considerações finais, contribuições e trabalhos futuros desta dissertação.

2. Fundamentos Matemáticos

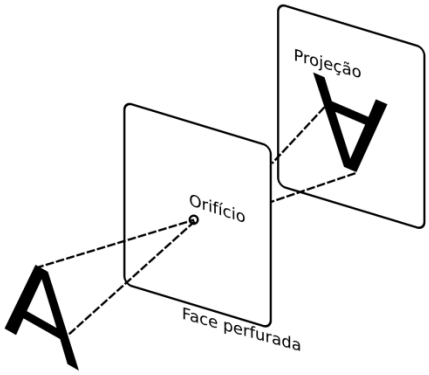
Neste capítulo são apresentados os conceitos matemáticos básicos necessários à compreensão e desenvolvimento de técnicas de rastreamento *top-down* baseados em algoritmos de otimização. Inicialmente, na seção 2.1 são explicados os conceitos geométricos do modelo de câmera de orifício, sistemas de coordenadas do mundo e da câmera, transformação e projeção de pontos de uma cena; na seção 2.2 são mostradas as características de imagens RGB-D e apresentado como é realizado o cálculo de nuvens de pontos 3D a partir desse tipo de imagem; por fim, o rastreamento baseado em abordagens *top-down*, bem como as técnicas de otimização utilizadas para implementar esse tipo de rastreamento, tais como PF e PSO, são discutidos em detalhes na seção 2.3.

2.1. Representação da Câmera

Ao fazer o registro de uma imagem em uma fotografia, o que uma câmera fotográfica comum basicamente faz é uma amostragem e mapeamento geométrico dos pontos 3D da cena do mundo para um plano de imagem 2D. Esse mapeamento dos pontos amostrados é chamado de *projeção* [12]. Existem diversos modelos de câmeras que podem ser usadas para projetar imagens de cenas 3D, dentre as quais existem aquelas que usam lentes e aquelas que não as usam, sendo os modelos sem lentes os mais simples.

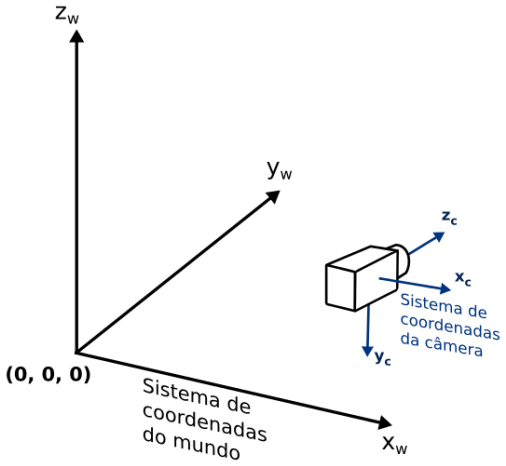
O modelo de câmera considerado no escopo deste trabalho será o da câmera estenopeica, ou simplesmente *câmera de orifício* (ou ainda *câmera pinhole*). O princípio de uma câmera de orifício é bastante simples, pois não utiliza lentes de distorção em sua composição e consiste de uma câmera escura com um pequeno orifício por onde a luz pode passar e ser projetada na face oposta à face perfurada. Uma descrição desse princípio pode ser vista na Figura 2.1. Em modelos de câmeras industrializadas, é na região de projeção que se encontra o filme em câmeras analógicas ou o sensor eletrônico como a matriz de dispositivo de carga acoplado (*charge-coupled device* – CCD) em câmeras digitais [13], ambos usados para registrar a imagem projetada.

Figura 2.1 – Princípio físico básico da projeção de uma imagem em uma câmera de orifício.



A despeito da importância dos aspectos físicos necessários para criar a projeção de imagens em uma câmera de orifício, o mais relevante para este trabalho são as características geométricas do funcionamento desse aparato. A princípio, para calcular a projeção da imagem de objetos reais de um mundo 3D em um plano 2D é necessário definir a posição dos pontos 3D dos objetos presentes na cena em relação a um sistema de coordenadas comum. Esse sistema de coordenadas genérico é chamado de *sistema de coordenadas do mundo*, sua origem é em $(0, 0, 0)$ e a localização de cada objeto da cena, inclusive da câmera, pode ser expressa por três coordenadas (x_w, y_w, z_w), conforme ilustrado na Figura 2.2.

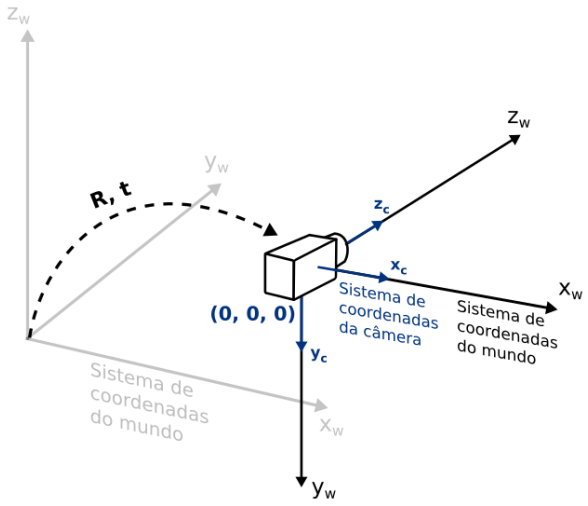
Figura 2.2 – Sistema de coordenadas do mundo.



Como a projeção da imagem de um objeto não depende apenas de sua localização no mundo, mas também da posição da câmera que o projeta em relação ao próprio objeto, é comum estabelecer um sistema de coordenadas alternativo considerando como origem as coordenadas do centro da câmera em relação ao mundo. Esse sistema é conhecido como *sistema de coordenadas da câmera*, cujos valores em relação a cada eixo são dados por (x_c, y_c, z_c). Esse novo sistema corresponde ao sistema de coordenadas do mundo após sofrer

uma determinada *transformação* (rotação e translação) que deixa seus eixos coincidentes com os respectivos eixos do sistema de coordenadas da câmera, como mostrado na Figura 2.3.

Figura 2.3 – Sistema de coordenadas da câmera obtido a partir de uma transformação do sistema de coordenadas do mundo.



A rotação e a translação que alinham os dois sistemas de coordenadas são representadas respectivamente por uma matriz de rotação $\boldsymbol{R}_{3\times3}$ e um vetor de translação $\boldsymbol{t}_{3\times3}$. Com o intuito de simplificar a notação é comum usar a concatenação dessas matrizes e representá-las como uma só matriz $[\boldsymbol{R}|\boldsymbol{t}]_{3\times4}$. A matriz $[\boldsymbol{R}|\boldsymbol{t}]$ é conhecida como *matriz de parâmetros extrínsecos* da câmera (*matriz de pose* ou simplesmente *pose*), pois com ela é possível conhecer os parâmetros externos do dispositivo, tais como localização e inclinação. A matriz de pose da câmera corresponde à transformação necessária para levar os pontos de um sistema de coordenadas para outro. Matematicamente, transformar um ponto 3D qualquer que está representado no sistema de coordenadas do mundo para o sistema de coordenadas da câmera usando $[\boldsymbol{R}|\boldsymbol{t}]$ significa calcular o produto dessa matriz com a matriz 4×1 formada pelas coordenadas homogêneas desse ponto:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_1 \\ r_{21} & r_{22} & r_{32} & t_2 \\ r_{31} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \tag{2.1}$$

Na equação (2.1), $[x_w, y_w, z_w, 1]^T$ é a matriz com as coordenadas homogêneas de um ponto no mundo e $[x_c, y_c, z_c]^T$ a matriz com as coordenadas do ponto transformado, isto é, no sistema de coordenadas da câmera.

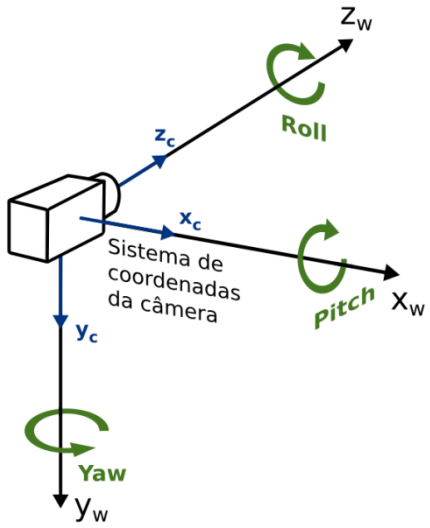
A matriz de rotação $\boldsymbol{R}_{3\times3}$ é obtida através do produto matricial das rotações nos eixos do sistema de coordenadas do mundo durante a transformação, ou seja, se essa rotação ocorre

com ângulos α, β e γ respectivamente nos eixos $\boldsymbol{X_w}$, $\boldsymbol{Y_w}$ e $\boldsymbol{Z_w}$, nessa mesma ordem, a matriz $\boldsymbol{R}$ pode ser calculada da seguinte forma:

$$\boldsymbol{R} = \begin{bmatrix} cos\,\gamma & -sen\,\gamma & 0 \\ sen\,\gamma & cos\,\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\,\beta & 0 & sen\,\beta \\ 0 & 1 & 0 \\ -sen\,\beta & 0 & cos\,\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\alpha & -sen\,\alpha \\ 0 & sen\,\alpha & cos\,\alpha \end{bmatrix}. \quad (2.2)$$

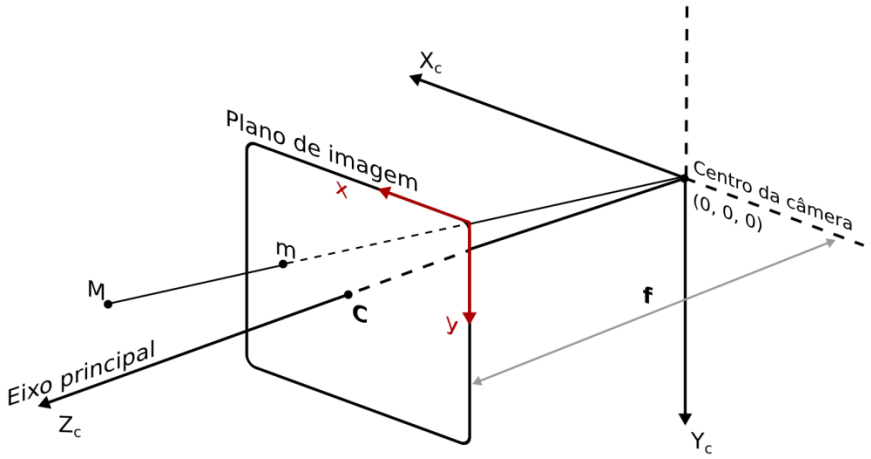
Essa rotação ainda pode ser representada por um vetor de apenas três dimensões, cujas componentes correspondem aos ângulos α, β e γ usados no cálculo de $\boldsymbol{R}$. Esses ângulos são chamados respectivamente de *pitch*, *yaw* e *roll* (inclinação guinada e rolamento) e compõem um caso particular muito usado de rotação em ângulos de Euler, que pode ser visto na Figura 2.4.

Figura 2.4 – Representação dos ângulos de Euler.



Contudo, apenas a matriz de pose $[\boldsymbol{R}|\boldsymbol{t}]$ não é suficiente para calcular corretamente a projeção dos pontos da cena, pois as câmeras possuem características físicas internas particulares e o resultado final da projeção de cada ponto da imagem vai depender de tais características. A Figura 2.5 ilustra o modelo matemático de uma câmera de orifício com algumas de suas principais características geométricas internas. Os eixos $\boldsymbol{X_c}$, $\boldsymbol{Y_c}$ e $\boldsymbol{Z_c}$ representam o sistema de coordenadas da câmera e sua origem $(0, 0, 0)$ é conhecida como *centro de projeção* ou *centro da câmera*. O *plano de imagem* é onde a imagem 2D da cena se forma após a projeção dos pontos capturados pela câmera. Nesse exemplo $\boldsymbol{M}$ é um ponto 3D qualquer no sistema de coordenadas da câmera e $\boldsymbol{m}$ um ponto que corresponde à projeção de $\boldsymbol{M}$ nas coordenadas do plano de imagem (em vermelho na Figura 2.5). O eixo $\boldsymbol{Z_c}$ é chamado de *eixo principal* ou *eixo óptico* e corta o plano de imagem no ponto $\boldsymbol{C}$ chamado de *ponto principal* onde $c_z = f$, em que f é a *distância focal* da câmera.

Figura 2.5 – Modelo matemático da câmera de orifício com algumas de suas principais características internas.



Para expressar as características internas da câmera usa-se uma matriz $\boldsymbol{K}$ conhecida como *matriz de parâmetros intrínsecos* da câmera. Essa matriz contém as informações sobre a câmera necessárias para possibilitar a realização de projeções corretas e pode ser descrita da seguinte forma:

$$\boldsymbol{K} = \begin{bmatrix} n_x f & 0 & c_x \\ 0 & n_y f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

em que c_x e c_y são as coordenadas do ponto principal $\boldsymbol{C}$ do plano da imagem, f a distância focal e n_x e n_y são fatores de escala que representam a razão entre os números de pixels por unidade de distância respectivamente nas direções do eixo x e do eixo y do plano de imagem.

De posse das matrizes de parâmetros intrínsecos $\boldsymbol{K}$ e de parâmetros extrínsecos da câmera $[\boldsymbol{R}|\boldsymbol{t}]$, é possível calcular as projeções dos pontos da cena segundo a equação:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} n_x f & 0 & p_x \\ 0 & n_y f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_1 \\ r_{21} & r_{22} & r_{32} & t_2 \\ r_{31} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{2.4}$$

em que s é um fator de escala e $[su, sv, s]^T$ corresponde às coordenadas homogêneas da projeção do ponto $[x_w, y_w, z_w]^T$. O produto entre as matrizes que representam os parâmetros intrínsecos e extrínsecos dá origem à matriz conhecida como *matriz de projeção* $\boldsymbol{P}$, matematicamente definida como:

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]. \tag{2.5}$$

Dessa forma, adotando ainda o fator de escala s, as coordenadas homogêneas $\tilde{\boldsymbol{m}}$ da projeção de um ponto 3D, representado pelas coordenadas homogêneas $\tilde{\boldsymbol{M}}$, podem ser definidas através de:

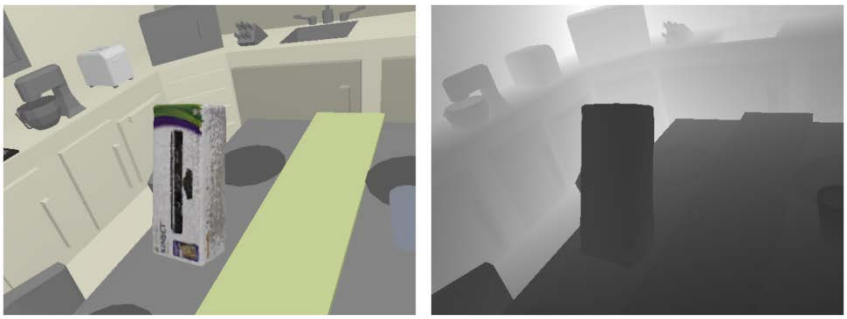
$$s\tilde{\boldsymbol{m}} = \boldsymbol{P}\tilde{\boldsymbol{M}}, \tag{2.6}$$

equivalente à equação (2.4), porém de uma forma simplificada.

2.2. Construção de Nuvens de Pontos 3D a partir de Imagens RGB-D

Alguns dispositivos comerciais usados para captura de imagens, tais como o Microsoft Kinect, já possuem sensores capazes de capturar não apenas imagens comuns em RGB da cena como também imagens conhecidas como *mapas de profundidade*. Esse tipo de imagem mapeia a profundidade de cada pixel da imagem em RGB em relação à câmera em tempo real. Como exemplo, na Figura 2.6 é possível observar uma imagem RGB sintética (à esquerda) e seu respectivo mapa de profundidade (à direita). Para possibilitar a visualização do mapa de profundidade nesse exemplo cada um de seus pontos foi representado por um pixel em escala de cinza com intensidade proporcional à distância em relação à câmera, sendo os pixels mais escuros aqueles mais próximos.

Figura 2.6 – Imagem RGB (esquerda) e seu mapa de profundidade (direita) representado em escala de cinza.



Para gerar o mapa de profundidade o Kinect utiliza um *projetor* e um *sensor* de luz infravermelha (*infrared* – IR) presentes na parte frontal do dispositivo, em detalhes na Figura 2.7. O projetor IR lança sobre a cena um padrão de luz IR conhecido e armazenado na memória do dispositivo e o compara com a imagem captada pelo sensor IR, assim através de uma correspondência estéreo é possível determinar a profundidade (canal D) dos pixels e formar uma imagem RGB-D da cena.

Figura 2.7 – Dispositivo Microsoft Kinect usado para captura de imagens RGB-D. Em destaque o projetor IR, o sensor IR e a câmera RGB.

A partir de uma imagem RGB-D e conhecendo-se os parâmetros intrínsecos $\boldsymbol{K}$ da câmera usada na captura é possível gerar uma *nuvem de pontos 3D* da cena e com isso ter uma representação da posição 3D de cada um dos pontos amostrados e visíveis nessa cena em relação à direção de vista da câmera no momento da captura. Para isso, seja $\boldsymbol{m}$ um ponto pertencente a uma imagem RGB-D com coordenadas (m_x, m_y) e profundidade igual a d, se a câmera usada para obter essa imagem possui foco f e ponto principal $\boldsymbol{C} = (c_x, c_y, f)$, o ponto $\boldsymbol{M}$ da nuvem de pontos 3D com coordenadas no sistema da câmera correspondente ao ponto $\boldsymbol{m}$ pode ser obtido da seguinte forma:

$$\boldsymbol{M} = \begin{bmatrix} (m_x - c_x)\frac{d}{f} \\ (m_y - c_y)\frac{d}{f} \\ d \end{bmatrix}. \tag{2.7}$$

Fazendo isso para o restante dos pontos da imagem em questão obtêm-se todos os pontos da nuvem 3D equivalente.

2.3. Rastreamento *Top-Down*

Rastrear objetos 3D com 6-DOF a partir da sequência de imagens capturadas por uma câmera digital implica em determinar a trajetória desse objeto no espaço ao longo do tempo em relação a essa câmera, ou o equivalente: significa determinar a trajetória da câmera em relação ao objeto que se deseja rastrear. Uma vez que a câmera faz apenas uma amostragem do movimento na cena, isto é, o número de imagens capturadas por uma câmera em um intervalo de tempo é finito, o rastreamento corresponde a determinar a pose do objeto rastreado em cada um dos quadros capturados de forma recursiva. Dessa forma, se $[\boldsymbol{R}|\boldsymbol{t}]_i$ correspondem à pose do objeto rastreado no i-ésimo quadro capturado pela câmera e N é o número total de quadros do vídeo, a trajetória de um objeto pode ser definida pelo conjunto $T = \{[\boldsymbol{R}|\boldsymbol{t}]_1, \dots, [\boldsymbol{R}|\boldsymbol{t}]_N)\}$ de poses e o rastreamento como o método que calcula os elementos de T a cada quadro capturado.

As técnicas de rastreamento podem ser identificadas segundo a natureza do método usado para calcular a pose do objeto a cada quadro, podendo ser classificadas como abordagens *bottom-up* ou *top-down*. Nas abordagens *bottom-up* primeiro são extraídas características da imagem capturada pela câmera para então, a partir dessas informações, calcular a pose do objeto naquele quadro [14]. Diferente das abordagens *bottom-up*, em abordagens *top-down* para determinar a pose do objeto rastreado em um determinado quadro

primeiro são avaliadas várias hipóteses de pose a partir da imagem capturada, tentando determinar a que mais se aproxima da pose real do objeto [4].

Uma possível direção de abordagens *top-down* é, para cada quadro q_i capturado pela câmera, inicialmente criar um conjunto de hipóteses de pose aleatórias em torno de uma pose central, geralmente a pose encontrada pelo algoritmo no quadro anterior q_{i-1}. Essas hipóteses de pose são então avaliadas usando as informações presentes na imagem em q_i segundo uma heurística predeterminada. Dessa forma, uma vez que é possível avaliar a qualidade de cada hipótese de pose, o problema de encontrar a melhor hipótese de pose em um determinado quadro pode ser interpretado como um *problema de otimização*, em que o conjunto de hipóteses de pose corresponde a um conjunto de soluções do problema e o objetivo é encontrar a *hipótese de pose ótima*, isto é, aquela que melhor se aproxima da pose real do objeto no instante da captura do quadro q_i atual. Duas técnicas de otimização que são comumente usadas para possibilitar o rastreamento *top-down* são o PF e o PSO.

2.3.1. Definição do PSO

Algoritmos de otimização são usados quando se pretende encontrar a melhor solução de um problema a partir de um conjunto de soluções possíveis, também chamado de *espaço de soluções* ou *espaço de busca*. Se for praticável avaliar as soluções pertencentes a esse espaço usando uma função matemática, o problema de otimização corresponde à tarefa de minimizar (ou maximizar) essa função. Frequentemente, a função em questão é chamada de *função objetivo* e a solução que a minimiza (ou a maximiza) é chamada de *solução ótima*, que por sua vez pode ser de natureza global ou local. Uma solução é chamada de *ótimo global* quando corresponde à melhor solução dentre todas as soluções pertencentes ao espaço de busca de um problema em questão. Já o *ótimo local* corresponde à melhor solução para um subconjunto desse espaço de busca.

Alguns problemas de otimização presentes no mundo real, por possuírem comportamentos complexos e imprevisíveis, não podem ser descritos por funções lineares [15], o que impossibilita a solução por meio de abordagens clássicas, tais como a diferenciação [16]. Problemas desse tipo são classificados como problemas de otimização não-lineares e podem ser investigados através de abordagens heurísticas.

Com o intuito de resolver classes de problemas de otimização não-lineares, foram criadas várias técnicas baseadas em otimização e inteligência artificial. Devido ao caráter genérico dessas técnicas, elas passaram a ser conhecidas como *meta-heurísticas* [17]. Alguns

exemplos amplamente conhecidos de meta-heurísticas são os Algoritmos Genéticos, Colônia de Formigas e PSO, dentre outros.

O PSO é uma técnica de otimização proposta por James Kennedy e Russel Eberhart em 1995 [18][19]. Esse método foi originalmente inspirado no comportamento coletivo de pássaros que vivem em bando, interagem entre si e possuem objetivos comuns. O algoritmo que o descreve é composto por uma população de indivíduos que se movimentam em um espaço de busca e trocam informações entre si a fim de encontrar boas soluções para o problema a ser resolvido [20]. No PSO, esse conjunto de indivíduos que interagem entre si é chamado de *enxame de partículas*. O movimento dessas partículas se dá por meio de atualizações na posição de cada uma delas e ocorre em função do histórico da partícula (conhecimento individual da partícula), do comportamento dos vizinhos mais próximos e da partícula melhor avaliada na população naquele instante (conhecimento coletivo). Uma *função de avaliação* (ou *função de aptidão*), que corresponde à função objetivo em problemas de otimização, é formulada de acordo com o problema e é usada para calcular a *aptidão* de cada partícula da população após atualização das posições das partículas de todo o enxame, ocorrendo a cada iteração do algoritmo. A aptidão de uma partícula está diretamente relacionada a quão boa é a solução que essa partícula representa.

Na prática, quando um PSO é instanciado para resolver um determinado problema, a princípio um conjunto de soluções candidatas é criado dentro de um espaço de soluções válido. Cada uma dessas soluções é representada por uma partícula e todas elas formam um enxame. A posição de cada uma das partículas dentro do espaço de soluções pode ser representada por um vetor, cujo número de dimensões depende do número de parâmetros que a solução precisa ter. A cada iteração do algoritmo, as partículas são avaliadas segundo uma heurística definida dentro do domínio do problema e se movem no espaço de busca seguindo as equações:

$$\boldsymbol{V_i}(t+1) = \omega \boldsymbol{V_i}(t) + c_1 r_1 \left[\boldsymbol{X_i^{best}}(t) - \boldsymbol{X_i}(t)\right] + c_2 r_2 \left[\boldsymbol{X_i^{gbest}}(t) - \boldsymbol{X_i}(t)\right], \quad (2.8)$$

$$\boldsymbol{X_i}(t+1) = \boldsymbol{X_i}(t) + \boldsymbol{V_i}(t+1), \quad (2.9)$$

onde:

- t indica em qual iteração se encontra o algoritmo;
- $\boldsymbol{X_i}$ é o vetor que representa a posição da partícula i dentro do espaço de busca;
- $\boldsymbol{X_i^{best}}$ é o vetor que guarda as componentes da melhor solução obtida pela partícula i até então;

- $\boldsymbol{X_i^{gbest}}$ é a melhor posição obtida por uma vizinhança da partícula i até então;
- $\boldsymbol{V_i}$ é o vetor que representa a velocidade da partícula i;
- ω é chamado de *peso de inércia* e regula a influência da velocidade das partículas no processo de otimização;
- c_1 e c_2 são constantes positivas que representam nesta ordem os parâmetros cognitivos e sociais do enxame;
- r_1 e r_2 são números gerados aleatoriamente dentro do intervalo $[0, 1]$.

A posição de cada partícula é atualizada usando adição de vetores, como expresso na equação (2.9). A velocidade da partícula, usada na equação (2.9), é atualizada a cada iteração segundo a equação (2.8), que é composta pela soma de três termos e cada um deles contribui de forma diferente para o cálculo da nova velocidade da partícula. No primeiro termo, $\boldsymbol{V_i}$ representa a velocidade da partícula i na iteração t anterior. Esse termo é controlado pelo peso de inércia ω que vai definir o quanto a velocidade anterior vai interferir na velocidade da próxima iteração, isto é, em $t + 1$. Valores altos para ω colaboram para uma exploração global no espaço de busca enquanto valores baixos favorecem uma busca local, ou seja, no espaço de busca próximo à posição em que se encontra cada partícula. Autores sugerem o uso do peso de inércia com valor de ω dinâmico limitado por um intervalo $[\omega_{max}, \omega_{min}]$ e que decresce em função do número de iterações já processadas [20]. Nesse modelo, o valor de ω pode ser calculado a cada iteração t em função do número máximo de iterações t_{max} do PSO, como descrito nas equações:

$$s = \frac{3 \cdot t_{max}}{4}, \tag{2.10}$$

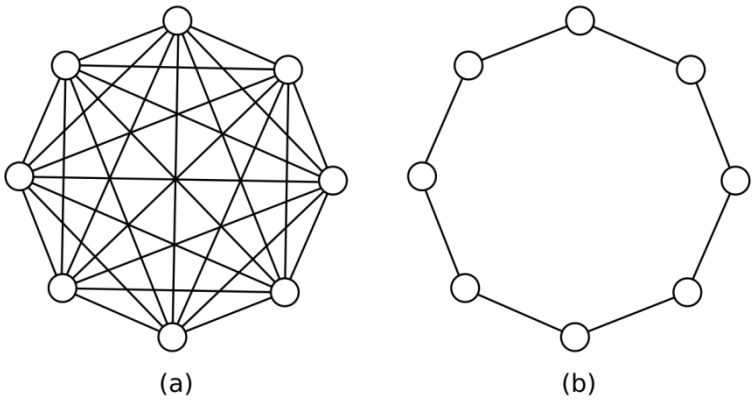
$$\omega = \begin{cases} \omega_{min} + (\omega_{max} - \omega_{min})\frac{s-t}{s} & se\ t \leq s \\ \omega_{min} & se\ t > s \end{cases}. \tag{2.11}$$

O segundo e terceiro termos da soma do lado direito da equação (2.8) representam respectivamente a influência do conhecimento individual e coletivo do enxame. O primeiro é calculado através da diferença entre o vetor que representa a posição atual da partícula e o vetor que guarda a melhor posição da partícula. Esse cálculo ainda é ponderado pelo parâmetro cognitivo c_1. O segundo, semelhante ao primeiro, é obtido através da diferença entre a posição atual e a posição da melhor partícula do enxame e ponderado pelo parâmetro social c_2.

Os parâmetros c_1 e c_2 podem ser alterados de forma que o movimento das partículas seja mais fortemente influenciado pelo caráter social ou cooperativo do enxame do que pelo caráter cognitivo individual, ou vice-versa, porém é comum usar valores iguais para c_1 e c_2 [21]. Entretanto, desta forma não existem garantias de que as influências cognitiva e social tenham o mesmo peso a cada atualização da partícula devido aos números aleatórios r_1 e r_2. Esses números, ainda no segundo e terceiros termos da equação (2.8), asseguram que as partículas apresentem trajetórias semi-aleatórias ao longo do movimento, uma vez que as componentes sociais e cognitivas passam a ter também uma influência estocástica [21].

Para um bom desempenho do PSO, é essencial que haja comunicação entre as partículas. Essa comunicação ocorre devido ao uso do vetor $\boldsymbol{X_i^{gbest}}$ durante a atualização das velocidades na equação (2.9). O vetor $\boldsymbol{X_i^{gbest}}$ guarda a melhor partícula considerando sua vizinhança. A maneira como é definida a vizinhança de uma partícula é o que determina a *topologia do PSO*. Existem diversas topologias, sendo as mais conhecidas a *topologia global*, chamada de *topologia gbest*, e a *topologia local*, também chamada de *topologia lbest*, ilustradas na Figura 2.8.

Figura 2.8 – Topologias do PSO: (a) global ou *gbest*; (b) local ou *lbest*.



Na topologia *gbest* a vizinhança de uma partícula qualquer é composta por todas as outras partículas pertencentes ao enxame. Dessa forma, como ilustrado na Figura 2.8a, todas as partículas estão conectadas entre si. Já na topologia *lbest* cada partícula se comunica apenas com dois vizinhos. Essa topologia deixa o enxame organizado em formato de anel, conforme a Figura 2.8b.

A *convergência* de um PSO ocorre quando o algoritmo encontra um ponto do espaço de busca que representa uma boa solução em relação às outras soluções já encontradas até então e, a partir de então, as partículas do enxame passam a se movimentar em direção a esse ponto. A convergência em um PSO é desejada quando esse ponto do espaço de busca

corresponde à solução ótima global, porém quando esse ponto representa um ótimo local a convergência é entendida como prematura e indesejada [20].

Na topologia *gbest*, uma vez que todas as partículas se comunicam entre si, a informação da melhor partícula do enxame é distribuída rapidamente e por isso essa topologia apresenta uma convergência mais rápida. Isso pode fazer com que as soluções sejam encontradas de forma eficiente, porém não garante que elas sejam boas. Assim, ao usar a topologia *gbest* a exploração do espaço de busca pode ser ágil, porém existe o risco de se diminuir a qualidade das soluções encontradas, uma vez que a chance do algoritmo não explorar adequadamente esse espaço e convergir para um mínimo local é grande. Na topologia *lbest*, como a comunicação de cada partícula ocorre apenas com uma vizinhança, a taxa de convergência é mais lenta, contudo pode retornar melhores resultados, uma vez que as chances de evitar uma convergência prematura são maiores em relação à topologia *gbest*.

2.3.2. Definição do PF

O PF é um método estatístico bayesiano usado para resolver problemas não-lineares ou não-gaussianos por meio de previsões dos estados de um sistema ao longo do tempo. Através de informações sobre o estado anterior de um sistema e de medições a respeito do seu estado atual, o PF consegue prever o estado imediatamente posterior desse sistema. Isso é possível através da representação de densidade de probabilidades de um conjunto de partículas que descrevem pontos contidos no espaço de estados válidos no sistema. Essa técnica pode ser usada na orientação de robôs [22], durante a estimação de poses [23] e rastreamento de objetos [6][8][24][25][26] ou na predição de séries temporais [27].

Sejam $\boldsymbol{X_t}$ e $\boldsymbol{Z_t}$ vetores que representam respectivamente a *variável de estado* e a *medição* do sistema em um instante t, e assumindo-se que $\boldsymbol{X_t}$ obedece a um processo de Markov, para possibilitar boas estimativas ao longo do tempo o PF representa a função de *densidade de probabilidade a posteriori* $p(\boldsymbol{X_t}|\boldsymbol{Z_{1:t}})$ por meio de um conjunto de N partículas definido por $S_t = \{(\boldsymbol{X_t^1}, \pi_t^1), \dots, (\boldsymbol{X_t^N}, \pi_t^N)\}$. Nesse conjunto, cada partícula é escrita como um par ordenado em que a primeira componente representa uma variável de estado válido $\boldsymbol{X_t^i}$ e a segunda o seu respectivo peso π_t^i. O valor do peso π_t^i de uma determinada partícula $\boldsymbol{X_t^i}$ define a importância da mesma no instante t e é calculado a partir da função de *densidade de importância* $p(\boldsymbol{Z_t}|\boldsymbol{X_t^i})$. Os pesos ainda são normalizados de modo que $\sum_{i=1}^{N} \pi_t^i = 1$.

Ao iniciar o algoritmo PF, isto é, na primeira iteração $t = 1$, as N partículas do conjunto S_1 são criadas aleatoriamente com valores válidos dentro do espaço de estados do

sistema. Nesse primeiro momento não existem ainda informações sobre a qualidade de cada uma das partículas, assim todas elas são instanciadas com o mesmo peso de importância normalizado, isto é, $\pi_1^0 = \pi_1^1 = \cdots = \pi_1^N = \frac{1}{N}$. A partir da criação do conjunto de partículas, cada iteração do algoritmo repete os seguintes passos básicos:

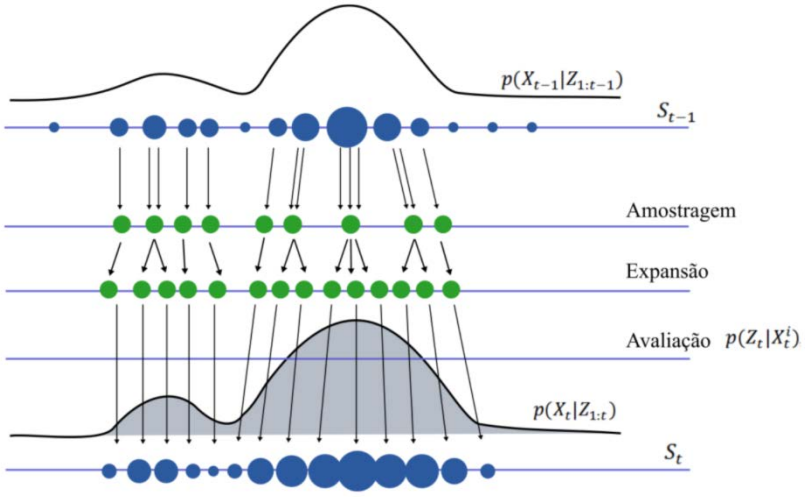
1. **Reamostragem:** é realizada uma reamostragem das partículas de acordo com o peso de importância π_t de cada uma delas, isto é, as partículas com os maiores π_t possuem mais chance de serem replicadas e sobreviverem. Já partículas com menores π_t possuem maior probabilidade de desaparecerem;
2. **Expansão:** as partículas se movem segundo um ruído aleatório a partir de suas respectivas posições iniciais (podendo ser um ruído gaussiano);
3. **Avaliação:** nesta etapa todas as partículas são avaliadas segundo a função de densidade de importância $p(\boldsymbol{Z_t}|\boldsymbol{X_t^i})$. É nessa etapa que os pesos de importância das partículas são redefinidos e normalizados;
4. **Estimação de estado:** o estado corrente $\boldsymbol{X_t}$ do sistema é estimado usando a média ponderada pelo peso π_t^i de cada partícula $\boldsymbol{X_t^i}$ presente no conjunto S_t:

$$X_t = \sum_{i=1}^{N} \pi_t^i \boldsymbol{X_t^i}. \tag{2.12}$$

Após estimar o estado no instante t, inicia-se a próxima iteração do algoritmo repetindo-se assim os quatro passos básicos descritos, com o intuito de determinar o estado na iteração $t + 1$. A Figura **2.9** apresenta graficamente cada um dos passos em uma iteração t qualquer. Nela inicialmente tem-se a distribuição de probabilidade do sistema no instante $t - 1$ amostrada pelo conjunto de partículas S_{t-1} e descrita pela função $p(\boldsymbol{X_{t-1}}|\boldsymbol{Z_{1:t-1}})$. Cada partícula $\boldsymbol{X_{t-1}^i}$ do conjunto é representada por um círculo azul, sendo a área do círculo proporcional ao seu peso de importância π_{t-1}^i. A segunda linha azul no desenho (de cima para baixo) diz respeito à etapa de reamostragem. Essa etapa é realizada sorteando-se, com reposição, N partículas do conjunto S_{t-1}. A chance de uma partícula ser escolhida em um sorteio é proporcional ao seu peso de importância. Ainda na Figura **2.9**, o número de vezes que cada partícula foi escolhida está representado pelo número de setas que saem da mesma. As partículas sorteadas são representadas por círculos verdes e possuem mesmo tamanho, pois nessa etapa ainda não foram reavaliadas. Na etapa de expansão (terceira linha azul), as partículas se deslocam de acordo com um pequeno ruído gaussiano em cada uma de suas

componentes, e na etapa de avaliação todas são reavaliadas segundo uma heurística associada ao domínio do problema ao qual pertence o sistema.

Figura 2.9 **– Passos básicos para o cálculo da densidade de probabilidade em um PF.**



Após a avaliação das partículas é possível redefinir os pesos de cada uma delas, dando assim origem a um novo conjunto S_t (quinta linha azul), que amostra a distribuição de probabilidade *a posteriori* $p(\boldsymbol{X_t}|\boldsymbol{Z_{1:t}})$. Apesar de algumas partículas não serem escolhidas na etapa de reamostragem, o número de partículas ao final de cada etapa precisa ser mantido. Isso pode ser controlado pelo número de sorteios nessa etapa.

3. Rastreamento de Objetos 3D

Existe uma grande variedade de métodos usados para rastrear objetos a partir de imagens capturadas em uma cena. Uma das formas de classificar esses métodos é através da técnica usada para calcular a pose da câmera a cada quadro da filmagem. Para facilitar essa tarefa, as primeiras abordagens usaram figuras inseridas artificialmente nos objetos pertencentes à cena. Essas figuras ficaram conhecidas como *marcadores*, pois ajudam a destacar alguns pontos da cena em relação aos demais. Com a evolução dessas técnicas e dos algoritmos de processamento de imagens, os marcadores artificiais passaram a ser substituídos por *marcadores naturais*. Isso possibilitou o desenvolvimento de métodos que não necessitam da pré-manipulação da ambiente. Dentre eles destacam-se nesta pesquisa aqueles métodos que se baseiam no uso do conhecimento prévio do modelo do objeto a ser rastreado.

Neste capítulo, a Seção 3.1 apresenta o conceito e as vantagens do rastreamento de objetos a partir de marcadores artificiais. Na Seção 3.2 são discutidos as definições e benefícios do rastreamento a partir de marcadores naturais e baseado em modelos. Já na Seção 3.3 é apresentada uma breve explicação de como pode ser realizado um rastreamento baseado em caracteríticas naturais a partir de imagens RGB-D.

3.1. Rastreamento a partir de Marcadores

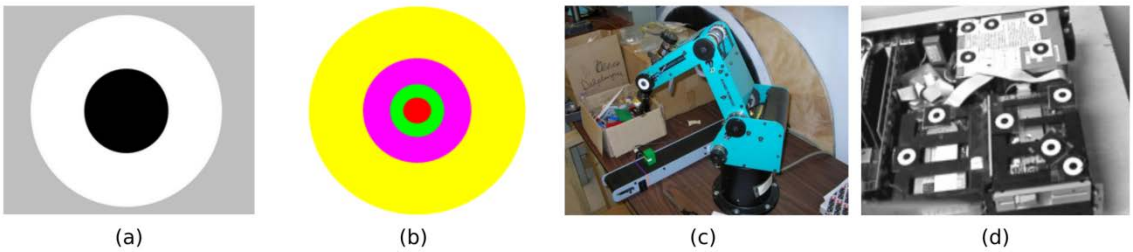
Para determinar a pose de um objeto em cada um dos quadros durante o rastreamento é preciso extrair informações da imagem usando alguma técnica de processamento de imagens [1]. Existem diversos métodos para possibilitar essa etapa de processamento. Alguns consideram a intervenção manual na cena a fim de inserir informações adicionais e com isso facilitar a identificação de pontos importantes na mesma. A adição de informações na cena é feita a partir de *marcadores*, que são figuras de fácil reconhecimento e localização em relação aos componentes presentes no restante da cena.

Os primeiros métodos de rastreamento a partir de marcadores usaram *marcadores pontuais*. Eles eram dispostos em locais estratégicos na cena para facilitar a correspondência entre os pontos reais do objeto e suas respectivas coordenadas 2D nas projeções. Devido à precisão e facilidade de rastreamento que esse tipo de marcador proporciona, eles ainda são utilizados em diversas aplicações, tais como no rastreamento do movimento de corpos ou para identificar expressões faciais humanas, como apresentado na Figura 3.1.

Figura 3.1 – Utilização de marcadores pontuais para o rastreamento de corpos e expressões faciais humanas. Imagens retiradas respectivamente de [28] **e** [29]**.**

Os marcadores pontuais também podem ser representados por círculos contrastantes e concêntricos (*concentric contrasting circle* – CCC). Cada CCC é formado por dois círculos concêntricos de cores e raios distintos, geralmente um preto e outro branco, como mostrado na Figura 3.2a, ou por vários círculos coloridos, como apresentado em [30] e exposto na Figura 3.2b. Eles podem ser usados separadamente, como por exemplo, no rastreamento de um braço robótico (Figura 3.2c) ou em conjunto quando são dispostos em um padrão geométrico previamente conhecido para determinar a pose de superfícies ou objetos [31] (Figura 3.2d).

Figura 3.2 – Marcador pontual em (a) e (b). Utilização de marcadores pontuais para o rastreamento de objetos em (b) e (c). A imagem (c) foi retirada de [32]**.**



Em [33], com rastreamento baseado no filtro de Kalman, foi introduzida a ideia de *marcadores planares*. Diferente dos marcadores pontuais, em que apenas as coordenadas do centro do marcador são levadas em consideração no cálculo da pose, estes novos marcadores eram formados por retângulos pretos em que a localização de cada um de seus vértices fornecia informações suficientes para calcular a pose da câmera através de uma correspondência entre os pontos da projeção e os pontos da cena. Esse tipo de marcador ainda possuía retângulos vermelhos menores em seu interior para identificação de cada marcador.

A partir de [34] os marcadores planares passaram a conter uma quantidade maior de informações em sua região interior disposta em uma espécie de código de barras em forma de matriz bidimensional (Figura 3.3). Além disso, com esse tipo de técnica foi possível estimar a pose 3D da câmera com um único marcador, viabilizando um rastreamento 3D preciso de baixo custo e em tempo real [1][35].

Figura 3.3 – Exemplos de marcadores planares.

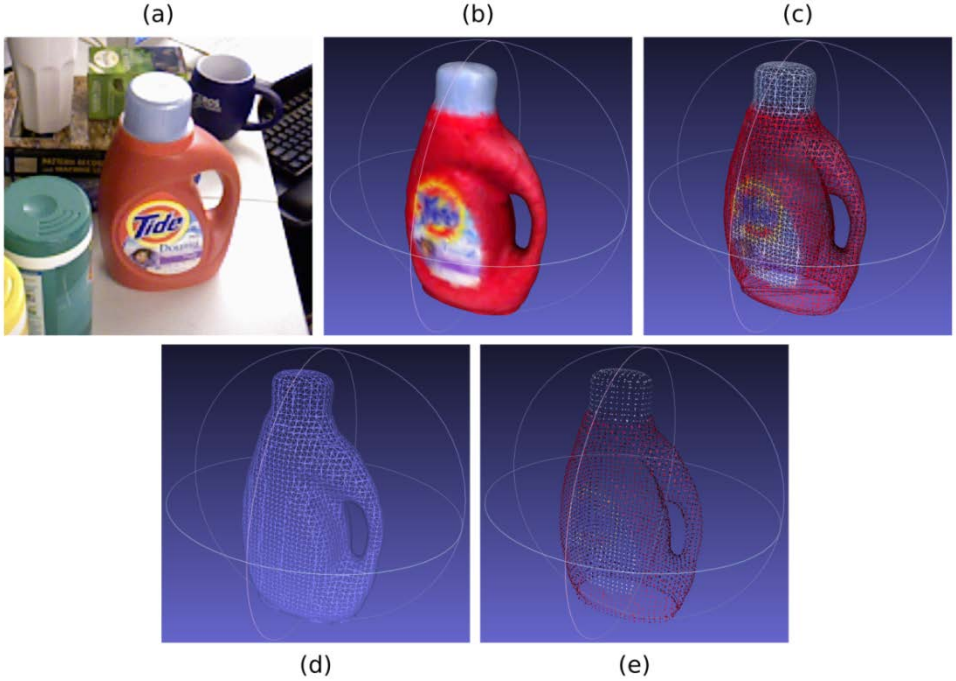
3.2. Rastreamento 3D Baseado em Modelos

Apesar da baixa complexidade e da eficiência dos métodos de rastreamento a partir de marcadores artificiais, um dos maiores problemas apresentados por esse tipo de técnica é a necessidade da manipulação da cena para inserção de figuras artificiais. Existem situações em que essa manipulação antes do rastreamento não é desejável ou simplesmente não é possível. Problemas assim podem ser contornados usando técnicas de rastreamento baseadas em *marcadores naturais*, isto é, aqueles que estão espontaneamente presentes na cena ou no objeto rastreado. Os marcadores naturais em uma cena podem ser características físicas inerentes ao objeto rastreado, tais como arestas, textura, pontos chave, cor, normais, dentre outros.

Os métodos de rastreamento que usam características naturais também são conhecidos como métodos de rastreamento *sem marcadores*. Em geral essas abordagens podem ser classificadas em dois grupos: as *baseadas em modelos* [1] e aquelas *baseadas em reconstrução 3D* [36]. São exemplos de técnicas de rastreamento 3D sem marcadores e baseadas em modelos: detecção de arestas, fluxo óptico, correspondência de modelos e pontos de interesse [1].

Como o próprio nome sugere, as técnicas baseadas em modelos necessitam do conhecimento prévio das características físicas do objeto que vai ser rastreado. Isso é feito através da obtenção de um modelo virtual desse objeto, que pode ser representado, por exemplo, por uma nuvem de pontos discreta obtida a partir da amostragem dos pontos do objeto real antes de iniciar o rastreamento. Na Figura 3.4 é possível observar um recipiente de sabão em close em um dos quadros da base de dados de [6] e o respectivo modelo virtual desse recipiente em diferentes perspectivas.

Figura 3.4 – Imagem do recipiente de sabão em (a) e seu respectivo modelo digital sob quatro diferentes representações, ou modos de visualização: superfícies suavizadas e sombreadas em (b); malha da triângulos com arestas coloridas e sombreadas em (c); malha de triângulos monocromática em (d); nuvem de pontos coloridos e sombreados em (e).



As técnicas que fazem uso de reconstrução 3D não utilizam modelos e o rastreamento da câmera ao longo do movimento é obtido a partir da própria estrutura 3D do ambiente sem considerar um pré-processamento *off-line* [36]. Porém, técnicas baseadas em modelos geralmente são mais robustas e mais tolerantes a falhas [1].

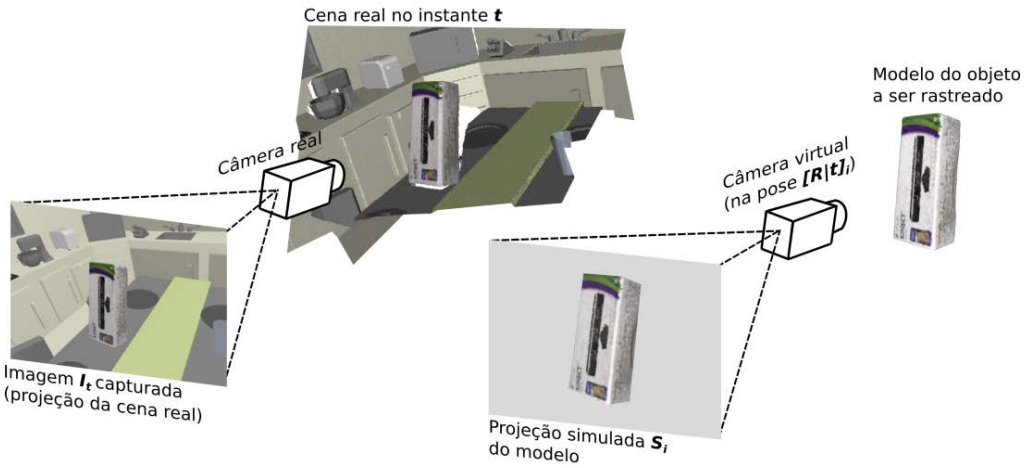
Uma das vantagens do uso de modelos no rastreamento sem marcadores é que, de posse do conhecimento prévio do objeto a ser rastreado, é possível criar uma câmera virtual e reproduzir projeções desse modelo segundo qualquer pose $[\boldsymbol{R}|\boldsymbol{t}]$ válida dessa câmera. Para isso, basta criar a matriz de projeção $\boldsymbol{P_i}$ com os parâmetros intrínsecos da câmera e a pose $[\boldsymbol{R}|\boldsymbol{t}]_i$ que se deseja avaliar e, a partir dela, calcular a projeção $\boldsymbol{m_j}$ de cada ponto $\boldsymbol{M_j}$ visível do modelo segundo a direção de vista da câmera simulada.

Essas projeções podem ser usadas para estimar ou avaliar a pose da câmera em um determinado quadro durante o rastreamento. Isso pode ser realizado da seguinte forma: seja I_t a imagem (projeção) da cena real capturada em um determinado instante t e S_i uma simulação da projeção do modelo obtido por uma câmera virtual de parâmetros intrínsecos iguais aos da câmera real segundo uma pose $[\boldsymbol{R}|\boldsymbol{t}]_i$ qualquer, conforme exemplo na Figura 3.5 (em que o objeto a ser rastreado é uma caixa de Kinect). De posse dessas informações, é possível medir o quão próximo a projeção do objeto real a ser rastreado, contida na imagem I_t, está da

projeção da parte visível do modelo contido em S_i e, dessa forma, uma vez que o resultado dessa avaliação depende diretamente da pose $[\boldsymbol{R}|\boldsymbol{t}]_i$ usada, calcular a qualidade dessa pose.

No exemplo da Figura 3.5 é possível notar que a projeção S_i do modelo ficou ligeiramente diferente da projeção da caixa de Kinect dentro da imagem I_t capturada pela câmera real (principalmente quanto à inclinação e ao tamanho da projeção). Observe ainda que ajustes na pose da câmera virtual em relação ao modelo poderiam levar a projeções mais semelhantes entre si, diminuindo o erro de projeção e consequentemente a diferença entre a pose $[\boldsymbol{R}|\boldsymbol{t}]_i$ e a pose real (desconhecida) da câmera.

Figura 3.5 – Projeção da cena segundo a pose da câmera real e projeção do modelo segundo a pose de uma câmera virtual.



A forma como é realizado o cálculo do erro entre a projeção do modelo em S_i e a projeção do objeto rastreado em I_t, bem como a técnica que será usada para manipular os resultados dessa comparação e fazer os ajustes necessários, são o que define as diferentes abordagens de rastreamento baseado em modelos. Considerando esses aspectos do rastreamento baseado em modelos, é nesse ponto que uma nova abordagem será proposta e avaliada no decorrer desta dissertação.

3.3. Rastreamento 3D a partir de Imagens RGB-D

O rastreamento baseado em modelos 3D pode ainda ser realizado a partir de imagens RGB-D [4][6][7][37]. O uso de imagens RGB-D durante esse tipo de rastreamento possibilita a criação de nuvens de pontos 3D da parte visível da cena a cada quadro capturado (Seção 2.2). A representação de cada quadro em uma nuvem de pontos 3D possibilita conservar as características geométricas tridimensionais da cena, diferente do que acontece em uma

representação bidimensional em que algumas informações espaciais são perdidas durante a projeção.

Dessa forma a relação entre os pontos da imagem I_t da cena e os pontos da projeção do modelo S_i pode ser usada para criar uma correspondência direta entre os pontos 3D da nuvem de pontos da cena, obtida a partir da imagem RGB-D, e os pontos visíveis do modelo segundo a direção de vista da câmera na pose avaliada naquele instante. Isso viabiliza a criação de diferentes formas de rastrear um objeto 3D, uma vez que permite uma avaliação de poses não apenas a partir de características presentes na projeção do modelo, mas também a partir de características geométricas extraídas diretamente do conjunto de pontos 3D do modelo e da nuvem de pontos da cena, tais como: coordenadas 3D dos pontos, curvatura, normais dos pontos, dentre outros.

4. Rastreamento de Objetos Usando PSO

Neste capítulo são apresentados trabalhos relacionados ao rastreamento usando PSO como método de otimização de múltiplas hipóteses de pose, além disso, é exposta ainda uma visão geral das etapas do método de rastreamento proposto nesta pesquisa. A Seção 4.1 apresenta trabalhos que utilizam o PSO como método para estimar a pose ou rastrear objetos 3D com diferentes graus de complexidade, bem como discorre sobre a principal diferença entre essas técnicas de rastreamento e o método sugerido nesta dissertação. A seção 4.2 expõe uma visão geral desse método. A Seção 4.3 explica a representação matemática da hipótese de pose como uma partícula do PSO. A Seção 4.4 mostra as características usadas para definir os critérios de avaliação das partículas do PSO. A Seção 4.5 por sua vez explica os detalhes dessa avaliação e ainda é dividida em duas subseções: a Subseção 4.5.1, que trata de como é realizada a correspondência entre os pontos do modelo e da cena, e a Subseção 4.5.2, que relata como é feita a comparação entre os pares de pontos em cada uma dessas correspondências. Por fim, a Seção 4.6 apresenta informações sobre o processamento da função de aptidão em GPU como primeiros passos para melhorar o desempenho da técnica.

4.1. Trabalhos Relacionados

O PSO tem sido aplicado com êxito como método de otimização de múltiplas hipóteses de pose tanto para estimar a pose quanto para rastrear objetos 3D em diversos trabalhos relacionados ao processamento de imagens, por exemplo: em [38] o PSO foi usada para estimar pose de objetos 3D obtendo resultados mais precisos e mais eficientes quando comparados com o método *Iterative Closest Point* (ICP); em [39] o PSO proporcionou o desenvolvimento de uma técnica para determinar a pose de cabeças humanas de forma precisa e tolerante à oclusão.

Além de estimar poses com precisão, a otimização de múltiplas hipóteses de pose usando PSO também se mostrou bastante eficiente em resolver problemas de rastreamento com objetos articulados, como em [11] onde PSO é combinado com o ICP durante o rastreamento de uma mão humana com 26-DOF. Em [10] foi proposta uma solução para um problema de rastreamento com um nível maior de complexidade, nesse trabalho o PSO é usado para rastrear pares de mãos humanas que interagem entre si, a técnica apresentou bons resultados mesmo contendo trechos com oclusões e considerando um espaço de busca de 54

dimensões, usadas para representar as configurações consideradas durante a interação entre as duas mãos. O PSO também pode ser empregado para desenvolver técnicas de captura do movimento de partes articuladas do corpo humano, como aquelas propostas em [9][40][41]. Nesse último trabalho, devido à convergência prematura do PSO clássico, foi usada uma abordagem com múltiplos enxames de partículas cooperativos na resolução do problema.

Ainda que os métodos de rastreamento baseados em PSO propostos pelos trabalhos citados nesta seção se mostrem aptos a resolver problemas de rastreamento complexos, todos eles atuam apenas em problemas de rastreamento de objetos que pertencem a classes específicas, previamente conhecidas e determinadas pelo domínio de cada problema. Abordagens desse tipo possuem a vantagem de poder usar como critério de avaliação de hipóteses de pose características naturais que são comuns a todos os objetos pertencentes à mesma classe em questão, mas que pode não estar presentes em objetos pertencentes a classes distintas. A desvantagem óbvia desse tipo de abordagem é que ela não pode ser usada para rastrear objetos muito diferentes daqueles para quais o algoritmo foi originalmente desenvolvido.

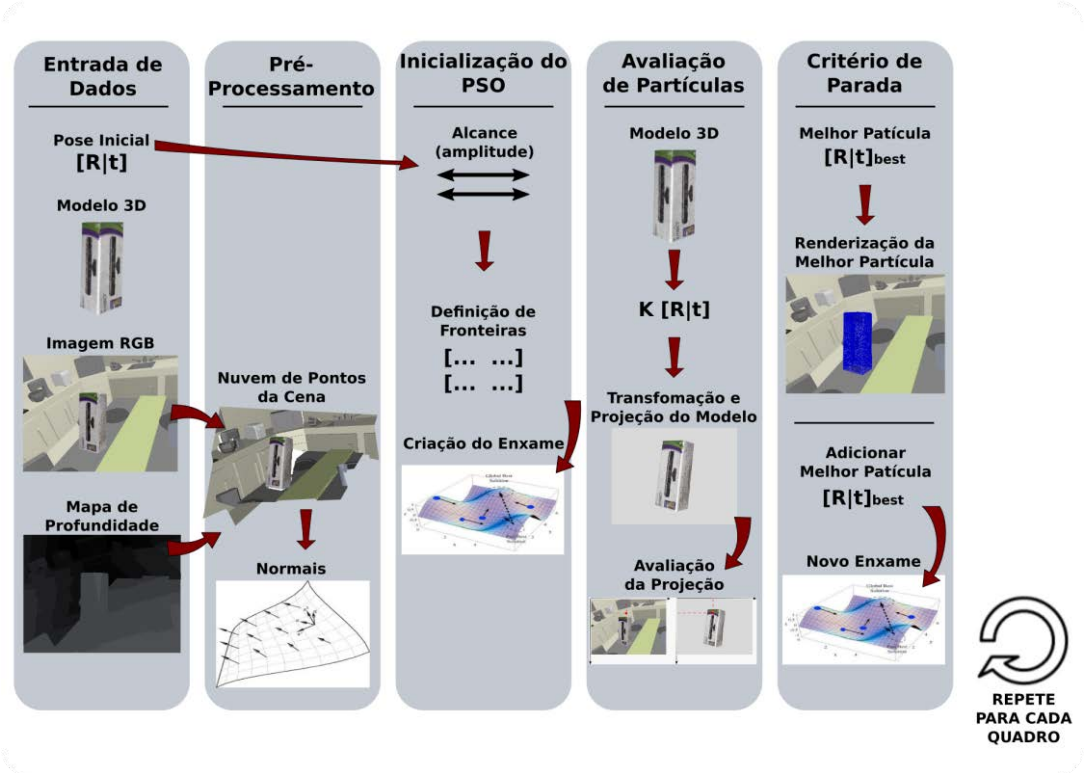
Contudo, existem situações em que é preciso rastrear objetos 3D *genéricos*, ou seja, aqueles que não pertencem necessariamente a uma classe específica de objetos, para resolver problemas dessa natureza alguns trabalhos apresentam técnicas de rastreamento baseado em PF como em [6][8][24][25]. Dado o cenário exposto, o presente trabalho propõe a criação de uma técnica de rastreamento sem marcadores de objetos 3D genéricos em que o PSO é usado como método de otimização de múltiplas hipóteses de pose com 6-DOF, essa abordagem considera ainda um rastreamento baseado em modelos e a partir de imagens RGB-D.

4.2. Visão Geral do Método Proposto

O método de rastreamento de objetos 3D sugerido por este trabalho consiste em uma abordagem *top-down*, baseada em modelos e a partir de características extraídas de imagens RGB-D. Múltiplas hipóteses de pose da câmera são geradas e avaliadas durante o rastreamento fazendo uso do PSO como algoritmo de otimização. Com o intuito de facilitar a compreensão da técnica sugerida, o cálculo da pose de um objeto em um quadro qualquer foi didaticamente dividido em cinco etapas, como mostra a Figura 4.1. Essa divisão é intuitiva e representa apenas uma abstração de como o sistema funciona, não contempla todos os detalhes do algoritmo real e não é a única possível.

Na primeira etapa, chamada aqui de “entrada de dados”, são apresentados os dados de entrada necessários ao início do processo de rastreamento a cada novo quadro q_i capturado. A matriz $[\boldsymbol{R}|\boldsymbol{t}]$ e o modelo 3D representam o conhecimento do ambiente obtido antes de dar início ao rastreamento, sendo $[\boldsymbol{R}|\boldsymbol{t}]$ a pose inicial do objeto obtida durante o rastreamento no quadro q_{i-1} anterior. O método proposto é recursivo, ou seja, o rastreamento em determinado quadro sempre considera como pose inicial a pose encontrada no quadro anterior. No entanto, para dar início ao algoritmo de rastreamento no primeiro quadro, assume-se conhecer a pose do objeto no instante inicial. O modelo 3D é uma nuvem de pontos do objeto contendo informações de coordenadas 3D, cores e normais dos pontos. O modelo, diferentemente da pose inicial, não sofre modificação ao longo do rastreamento. As imagens RGB e de profundidade são informações da cena capturadas pelo sensor RGB-D pertencentes ao quadro q_i. É a partir dessas informações que as hipóteses de pose poderão ser avaliadas.

Figura 4.1 – *Pipeline* com as principais etapas do algoritmo do método sugerido por este trabalho.



Na segunda etapa, nomeada de “pré-processamento”, essas imagens capturadas pelo sensor são usadas para compor uma nuvem de pontos da cena no instante da captura. As coordenadas 3D dos pontos da nuvem são usadas para calcular o vetor normal de cada um desses pontos. Uma nova nuvem da cena é criada e, assim como o modelo, passa a conter, além de informações de coordenadas 3D e cor, as normais de cada um de seus pontos. Ainda na segunda etapa, o sistema de cores dos pontos da nuvem é convertido de RGB para HSV e o

valor de cada componente da nova cor é normalizado para apresentar um valor real pertencente ao intervalo $[0, 1]$.

A terceira etapa, “inicialização do PSO”, diz respeito aos primeiros passos do processo de busca da pose do objeto em relação à câmera. Isso é realizado através da inicialização do PSO. Primeiro as fronteiras do espaço de busca do PSO são definidas a partir da pose inicial e de amplitudes preestabelecidas para as componentes de rotação e de translação dessa pose. É a partir dessas fronteiras que todas as partículas que compõem o PSO são criadas. Cada partícula corresponde a uma hipótese de pose válida para o objeto rastreado. As partículas se movem dentro do espaço de busca segundo regras bem definidas do PSO e são avaliadas conforme apresentado na próxima etapa (mais detalhes sobre o PSO na Subseção 2.3.1). Além da definição das fronteiras, é também nesta etapa que são definidas as configurações do PSO. Fazem parte dessa configuração: topologia, critério de parada, número de iterações, tamanho do enxame, valores das constantes dos parâmetros cognitivos e sociais, peso de inércia, dentre outros.

Na quarta etapa, “avaliação das partículas”, como o nome sugere, ocorre a avaliação das hipóteses de pose pela função de aptidão do PSO. Nela uma cópia do modelo 3D é transformada e projetada segundo a matriz de projeção obtida a partir da hipótese de pose contida em cada partícula. Os índices dos pontos contidos nessa projeção são usados para fazer uma correspondência entre os pontos do modelo transformado e os pontos da nuvem da cena, possibilitando assim a avaliação da transformação sofrida pelo modelo e consequentemente a qualidade da hipótese de pose que a gerou. A cada avaliação, são selecionados apenas os pontos visíveis do modelo segundo o vetor de direção de vista de cada ponto em relação à câmera. Esse procedimento é repetido para todas as partículas do PSO e a cada iteração do mesmo.

A quinta etapa, chamada de “critério de parada”, descreve o encerramento do PSO e a definição da pose do objeto. Essa etapa se inicia no momento em que o critério de parada predefinido do PSO é atingido. Nesse momento a melhor partícula do enxame passa a representar a pose do objeto encontrada para o quadro q_i. Com o intuito de viabilizar uma avaliação qualitativa do resultado, o algoritmo salva em um arquivo a pose encontrada e ainda renderiza, sobre a imagem RGB do quadro em questão, uma malha de pontos azuis que representa os pontos visíveis do modelo transformado segundo a pose encontrada. Na Figura 4.1 é possível ver um exemplo dessa renderização na coluna correspondente a essa etapa. A melhor partícula encontrada é conservada para compor o conjunto de partículas do próximo enxame e suas componentes são usadas como pose inicial para definir as fronteiras do espaço

de busca no rastreamento do próximo quadro q_{i+1}. Ao concluir o processo de rastreamento para o quadro q_i, a primeira etapa é reiniciada, desta vez com as imagens da cena em q_{i+1}.

4.3. Representação da Partícula

Em problemas de rastreamento de objetos 3D rígidos é possível definir corretamente a pose de um objeto na cena de acordo com os parâmetros intrínsecos e extrínsecos da câmera [1]. Quando esse rastreamento é baseado em uma abordagem *top-down*, cada hipótese de pose é uma suposição válida da pose desse objeto em relação à câmera em um determinado quadro, geralmente considerando um movimento com 6-DOF. Assumindo-se que durante o rastreamento os parâmetros intrínsecos da câmera são fixos e conhecidos, uma hipótese de pose corresponde apenas aos parâmetros extrínsecos $[\boldsymbol{R}|\boldsymbol{t}]$ da matriz de projeção $\boldsymbol{P}$ descrita na equação (2.5).

Em um problema de otimização, a dimensionalidade do espaço de busca depende da quantidade de componentes do vetor que representa uma solução válida desse problema. Para resolver o problema de rastreamento, a abordagem sugerida por este trabalho usa PSO como método de otimização, assim cada partícula do enxame corresponde a uma hipótese de pose e o número de componentes dessa hipótese é o que vai definir o número de dimensões do espaço de busca do PSO. Como uma pose é formada por uma matriz de rotação $\boldsymbol{R}_{3\times3}$ com 9 elementos e um vetor de translação $\boldsymbol{t}_{1\times3}$ com 3, o vetor de características que representa uma partícula do enxame precisaria ter pelo menos 12 componentes, o que resultaria em um PSO com um espaço de busca de 12 dimensões.

Uma forma de reduzir a dimensionalidade da pose é através da conversão da matriz de rotação $\boldsymbol{R}$ em *mapa exponencial*. Usando essa representação, a rotação em um objeto pode ser escrita na forma de um vetor com apenas três componentes: $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$. Essa notação, além de ser bastante compacta, evita que ocorram eventuais perdas de um grau de liberdade no movimento do objeto caso dois eixos de rotação quaisquer fiquem alinhados paralelamente durante um giro, fenômeno também conhecido *gimbal lock*. No mapa exponencial, o vetor $\boldsymbol{\omega}$ especifica o eixo a partir do qual o objeto será girado e sua norma $\theta = \|\boldsymbol{\omega}\|$ o ângulo que esse giro terá [42]. A relação entre a matriz de rotação e a representação em mapa exponencial pode ser expressa matematicamente através da fórmula de Rodrigues [43]:

$$\boldsymbol{R} = \cos\theta\ \boldsymbol{I} + (1 - \cos\theta)\boldsymbol{\omega}\boldsymbol{\omega}^T + \sin\theta\,\boldsymbol{\Omega}, \tag{4.1}$$

em que $\boldsymbol{I}$ é uma matriz identidade e

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (4.2)$$

Para encontrar a rotação em mapa exponencial a partir da matriz de rotação $\boldsymbol{R}$ basta utilizar:

$$\sin\theta\, \boldsymbol{\Omega} = \frac{\boldsymbol{R}-\boldsymbol{R}^T}{2}. \quad (4.3)$$

Com a conversão da matriz de rotação em mapa exponencial, uma hipótese de pose passa a ser representada por uma partícula no PSO composta por um vetor $\boldsymbol{p}$ com apenas seis elementos, isto é, $\boldsymbol{p} = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)$, em que os três primeiros correspondem à rotação e os três últimos à translação da transformação sofrida pelo objeto em relação ao sistema de eixos da câmera. Uma vez reduzido o número de componentes da hipótese de pose, o espaço de busca do PSO passa a ter apenas seis dimensões.

Ao instanciar um novo PSO, cada partícula $\boldsymbol{p}$ pertencente ao enxame é criada aleatoriamente dentro de um subconjunto contido no espaço de busca e limitado por fronteiras definidas para cada uma das componentes. Estas fronteiras são ajustadas a partir das componentes da pose inicial e os intervalos permitidos para cada uma delas são definidos por valores de amplitudes de rotação b_r e de translação b_t preestabelecidas. Dada uma pose inicial $\boldsymbol{p_1}$, o alcance das componentes de rotação e translação são definidos respectivamente pelos intervalos $[\omega_k - b_r, \omega_k + b_r]$ e $[t_k - b_t, t_k + b_t]$, com $k = \{1, 2, 3\}$.

4.4. Características

A avaliação de cada partícula do PSO é realizada a partir das imagens RGB-D capturadas a cada quadro no decorrer da filmagem da cena. O uso desse tipo de imagem possibilita a criação de uma nuvem de pontos a cada quadro (mais detalhes da criação de uma nuvem de pontos na Seção 2.2). Esse tipo de estrutura, diferente das imagens RGB convencionais, conserva características 3D da cena original que podem ser usadas posteriormente na avaliação das hipóteses de pose.

No método proposto, as características utilizadas na função de aptidão são obtidas a partir dos pontos dessa nuvem, sendo elas: coordenadas 3D, cor e vetor normal. As coordenadas 3D são representadas pelo vetor $\boldsymbol{M_i}$ de cada ponto com suas componentes representadas em milímetros. As componentes das cores originalmente no sistema de cor RGB são convertidas para HSV normalizado e expressas por um vetor $\boldsymbol{c_i}$. Essa conversão é

realizada devido ao sistema de cor HSV possuir menor sensibilidade a mudanças de iluminação na cena em relação ao sistema RGB.

Diferente das duas primeiras características citadas, os vetores normais não estão presentes na nuvem de pontos gerada a partir da imagem RGB-D, mas podem ser calculados a partir das coordenadas 3D dessa nuvem. O vetor normal em um ponto é um vetor unitário perpendicular à superfície definida por uma vizinhança desse ponto. É possível determinar um vetor normal $\boldsymbol{n_i}$ para cada ponto $\boldsymbol{M_i}$ encontrando o vetor médio $\bar{\boldsymbol{M}}$ dos n pontos vizinhos de $\boldsymbol{M_i}$ e com isso calcular a matriz de covariância $\boldsymbol{C}$ desses pontos usando:

$$\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{M_i} - \bar{\boldsymbol{M}})(\boldsymbol{M_i} - \bar{\boldsymbol{M}})^{\boldsymbol{T}}. \tag{4.4}$$

A vizinhança de $\boldsymbol{M_i}$ pode ser definida por um raio ou pelo número n de vizinhos mais próximos de $\boldsymbol{M_i}$. De posse da matriz $\boldsymbol{C}$, é possível encontrar seus autovalores λ_j e os respectivos autovetores $\boldsymbol{v_j}$ definidos como $\boldsymbol{C}.\boldsymbol{v_j} = \lambda_j.\boldsymbol{v_j}$, com $j \in \{0,1,2\}$. Assim, uma vez que a direção do eixo que minimiza a variância entre os pontos é o autovetor $\boldsymbol{v_k}$ cujo autovalor λ_k correspondente é o menor dentre os três autovalores calculados, o vetor normal $\boldsymbol{n_i}$ corresponde a esse autovetor $\boldsymbol{v_k}$ [44].

A partir daqui considera-se que cada ponto da nuvem de pontos da cena e do modelo contém as informações de suas coordenadas 3D, cor e normal como definidos nesta seção.

4.5. Função de Aptidão

A função de aptidão em um PSO é a parte do algoritmo usada para avaliar cada partícula atribuindo-lhes uma espécie de “nota”, também conhecida como *aptidão* da partícula. A aptidão é o que especifica quão boa é a solução que a partícula em questão representa. A forma como uma partícula é avaliada em um PSO depende do domínio do problema que está se tentando resolver. No método proposto, uma partícula é considerada bem avaliada quando representa uma hipótese de pose com valores próximos da pose real do objeto a ser rastreado naquele momento.

Para realizar a tarefa de avaliar cada partícula, a função de aptidão proposta precisa inicialmente receber como entrada uma nuvem de pontos 3D da cena obtida a partir das imagens capturadas pelo sensor RGB-D durante o rastreamento, o modelo 3D do objeto que será rastreado e o vetor que representa a hipótese de pose da partícula que será avaliada. De posse desses dados de entrada, o cálculo da aptidão é realizado através da comparação entre o

modelo transformado pela hipótese de pose contida na partícula e a representação 3D do objeto real contida na nuvem de pontos da cena.

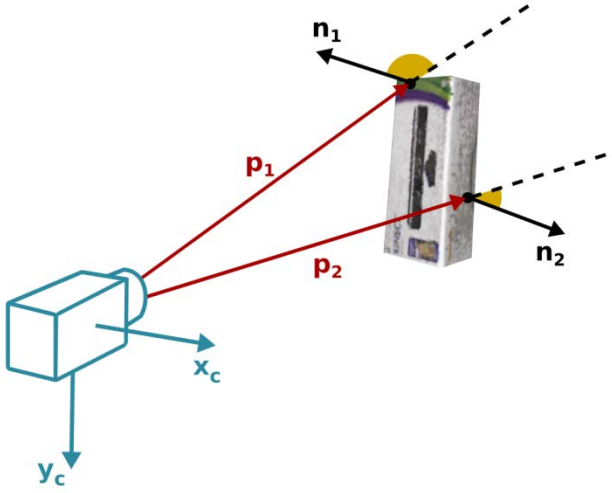
Seguindo esse método, o problema da avaliação de uma hipótese de pose se resume então em duas etapas: a primeira seria a criação de uma correspondência correta entre os pontos visíveis do modelo em relação à câmera (segundo a direção de vista da hipótese de pose avaliada) e os pontos da nuvem de pontos da cena; a segunda, uma vez que essa correspondência seja realizada, seria comparar estes pares de pontos e calcular o erro de reprojeção, que pode ser definido como a diferença entre os pontos do modelo transformado e seus correspondentes na nuvem de pontos da cena. A função de aptidão usada pelo método proposto trata justamente da realização dessas duas etapas para cada um dos pontos analisados. Elas são explicadas em detalhes nas duas subseções a seguir.

4.5.1. Correspondência entre os Pontos

Antes de realizar a correspondência entre os pontos do modelo e da cena, é preciso definir qual o conjunto de pontos visíveis do modelo segundo a hipótese de pose avaliada. Em modelos de objetos 3D, dado o ponto de vista de uma hipótese de pose, pode haver um conjunto de pontos oclusos, isto é, aqueles pontos que fazem parte de faces ou regiões não visíveis segundo esse ponto de vista. Para avaliação de hipóteses de pose próximas à pose real do objeto esses pontos oclusos não possuem pontos correspondentes na cena, dessa forma devem ser evitados, pois a projeção destes pontos junto aos pontos visíveis causa a adição de um ruído indesejado durante o cálculo da aptidão.

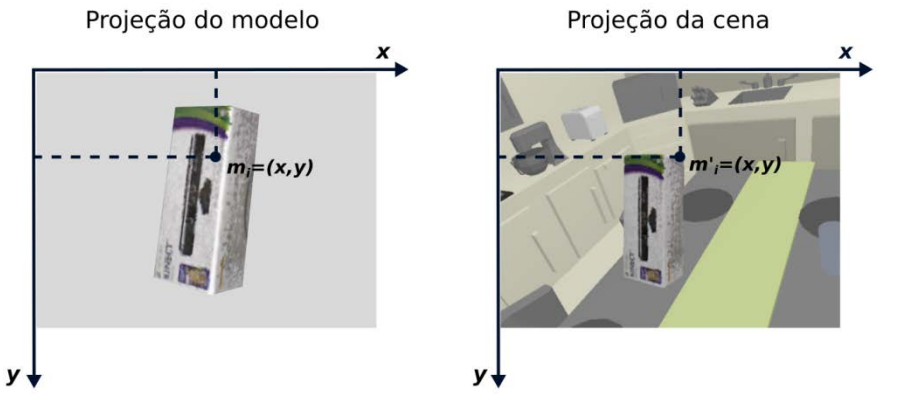
Para determinar se um ponto $\boldsymbol{M_i}$ do modelo transformado pela hipótese de pose analisada pertence ao conjunto dos pontos visíveis, é calculado o ângulo entre o vetor $\boldsymbol{p_i}$ composto pelas coordenadas desse ponto e o seu vetor normal $\boldsymbol{n_i}$. Se o valor desse ângulo for maior que 90°, o ponto é considerado visível. Isso é possível, pois ao transformar o modelo as coordenadas do centro de projeção da câmera passam a ser a origem $(0, 0, 0)$ e o vetor $\boldsymbol{p_i}$ passa a descrever também o vetor da direção de vista nesse ponto. Dois exemplos podem ser vistos na Figura 4.2, em que o ponto representado pelo vetor $\boldsymbol{p_1}$ é visível, uma vez que possui um ângulo maior que 90° em relação a seu vetor normal $\boldsymbol{n_1}$, e outro não visível representado por $\boldsymbol{p_2}$, pois possui um ângulo menor que 90° em relação ao vetor normal $\boldsymbol{n_2}$. Apesar de esse método de visibilidade não ser completamente válido em objetos côncavos, ele se mostrou suficiente para os fins do rastreamento proposto neste trabalho.

Figura 4.2 – Ângulo entre o vetor da direção de vista p_i de um ponto M_i e sua normal n_i.



No rastreamento a partir de imagens RGB-D, apesar do sensor fornecer informações sobre a profundidade de cada ponto $\boldsymbol{M_i'}$ da cena, todos estes pontos estarão representados pela projeção $\boldsymbol{m_i'}$ sobre um plano que retrata a parte visível da cena naquele instante. Dessa forma, a localização de cada ponto da cena pode ser expressa por um par ordenado $\boldsymbol{m_i'} = (x, y)$ que representa as coordenadas desse ponto no plano de projeção. O mesmo ocorre com a projeção do modelo do objeto rastreado, onde cada um de seus n pontos $\boldsymbol{M_i}$ visíveis pode ser projetado como $\boldsymbol{m_i} = (x, y)$ sobre um plano segundo a pose e características de uma câmera virtual, conforme visto na Figura 4.3.

Figura 4.3 – Projeção dos pontos visíveis da nuvem de pontos da cena e do modelo.



Dessa forma, usando as coordenadas x e y de cada ponto projetado $\boldsymbol{m_i}$ do modelo, é possível descobrir seu correspondente $\boldsymbol{m_i'}$ na projeção da cena. Essa correspondência também é válida para os pontos 3D $\boldsymbol{M_i}$ do modelo e $\boldsymbol{M_i'}$ da cena. Observe que a semelhança entre $\boldsymbol{m_i}$ e $\boldsymbol{m_i'}$ vai depender da qualidade da projeção do modelo, ou seja, quanto mais próximo for a hipótese de pose da pose real do objeto mais semelhante serão estas projeções.

4.5.2. Comparação entre os Pontos

Realizada a correspondência entre os pontos da cena e do modelo, é preciso ainda medir a diferença entre cada par $\boldsymbol{M_i}$ e $\boldsymbol{M_i'}$. Desse modo a aptidão de cada partícula analisada é definida como a média das diferenças entre os pares de pontos correspondentes, considerando todas as k correspondências realizadas. No método sugerido, uma vez que o rastreamento é baseado em imagens RGB-D, o cálculo dessa diferença foi realizado a partir de três características de cada um dos pontos: coordenadas 3D, coordenadas HSV normalizadas e coordenadas do vetor normal. Assim, a função de aptidão pode ser matematicamente definida da seguinte forma:

$$aptidão = \frac{1}{k}\sum_{i=1}^{k}[\lambda_p . d_p(\boldsymbol{M_i}, \boldsymbol{M_i'}) + \lambda_c . d_c(\boldsymbol{c_i}, \boldsymbol{c_i'}) + \lambda_n . d_n(\boldsymbol{n_i}, \boldsymbol{n_i'})], \quad (4.5)$$

em que, dado um ponto do modelo $\boldsymbol{M_i}$ transformado segundo uma hipótese de pose a ser avaliada com cor $\boldsymbol{c_i}$ e normal $\boldsymbol{n_i}$ e seu ponto correspondente $\boldsymbol{M_i'}$ na nuvem de pontos capturada a partir da cena com cor $\boldsymbol{c_i'}$ e normal $\boldsymbol{n_i'}$, as funções d_p, d_c e d_n calculam respectivamente a distância euclidiana entre as coordenadas 3D dos pontos, a distância euclidiana entre as coordenadas das cores e o ângulo entre as normais dos pontos. Essas funções podem ser definidas pelas equações:

$$d_p(\boldsymbol{M_1}, \boldsymbol{M_2}) = \begin{cases} \|\boldsymbol{M_1} - \boldsymbol{M_2}\| & se\ \|\boldsymbol{M_1} - \boldsymbol{M_2}\| \leq l \\ 1 & se\ \|\boldsymbol{M_1} - \boldsymbol{M_2}\| > l,' \end{cases} \quad (4.6)$$

$$d_c(\boldsymbol{c_1}, \boldsymbol{c_2}) = \|\boldsymbol{c_1} - \boldsymbol{c_2}\|, \quad (4.7)$$

$$d_n(\boldsymbol{n_1}, \boldsymbol{n_2}) = \frac{\cos^{-1}(\boldsymbol{n_1^T n_2})}{\pi}. \quad (4.8)$$

Em (4.5), k é o número de pontos visíveis do modelo em relação ao ponto de vista da câmera e de acordo com a pose analisada e as constantes λ_p, λ_c e λ_n são usadas como pesos para ponderar a influência de cada uma das características usadas sobre o valor da aptidão final. Na função descrita em (4.6) ainda é usado um limiar l que a torna constante a partir de um determinado valor. Isso evita que oclusões parciais do objeto rastreado façam com que a função d_p retorne valores muito altos em relação às outras funções, impedindo que hipóteses de pose boas fossem avaliadas como ruins devido a essas oclusões.

4.6. Processamento da Função de Aptidão em GPU

As GPUs atuais possuem uma arquitetura que possibilita um alto poder de processamento paralelo e são popularmente conhecidas por melhorar o desempenho de

aplicações gráficas como jogos digitais ou softwares de edição e processamento de imagens. Esse poder de processamento também tem sido usado por pesquisadores e desenvolvedores para resolver problemas que não possuem necessariamente computação gráfica, mas que, assim como aqueles que possuem, exigem a realização de um grande número de cálculos matemáticos. Essa prática é conhecida como computação de propósito geral em GPU (*General-Purpose Computing on GPU* – GPGPU) [45].

O poder de processamento em GPU está associado ao fato de que, diferentemente do processamento em CPU, o seu ambiente de desenvolvimento e sua arquitetura proporciona a execução de milhares de *threads* em paralelo. Assim, uma situação propícia ao uso de GPGPU ocorre quando os problemas abordados são passíveis de serem divididos em subproblemas menores que podem ser solucionados de forma independente uns dos outros e ao mesmo tempo [46]. Essa característica da GPU de executar uma grande quantidade de *threads* de forma eficiente pode ser usada para paralelizar o processamento da função de aptidão do PSO.

Uma vez que no rastreamento proposto por esse trabalho a quantidade de pontos usados para compor a nuvem de pontos do modelo é grande e a função de aptidão é chamada diversas vezes para cada partícula ao longo do processamento do PSO, o tempo necessário para fazer todas as avaliações corresponde à maior parte do tempo de processamento total do algoritmo proposto. A avaliação de uma hipótese de pose é um problema que pode ser facilmente paralelizável, pois o cálculo do erro de reprojeção de cada par de pontos correspondentes ocorre de forma independente da avaliação dos demais pares de pontos. Dessa forma, com o intuito de melhorar o desempenho da técnica sugerida, a função de aptidão foi implementada em GPU.

Ao iniciar o rastreamento no primeiro quadro capturado, a nuvem de pontos desse quadro e o modelo 3D do objeto são alocados na memória da placa gráfica. A cada novo quadro, uma vez que o modelo do objeto rastreado não muda, apenas a nuvem de pontos da cena é atualizada. Assim, para calcular a aptidão de uma partícula, suas componentes são copiadas para a memória do dispositivo e um *kernel* com um número de *threads* igual ao número de pontos do modelo é executado. Cada *thread* na GPU é responsável pelo processamento da comparação de apenas um ponto do modelo e seu correspondente na nuvem de pontos da cena. A execução dessa tarefa segue os seguintes passos:

- Transformar as coordenadas 3D e as normais do ponto do modelo segundo a hipótese de pose da partícula avaliada;

- Verificar se o ponto transformado é visível usando suas normais, como descrito na Subseção 4.5.1 deste capítulo. Em caso afirmativo, calcular as coordenadas da projeção desse ponto e determinar qual o seu ponto correspondente na nuvem de pontos. Caso o ponto não seja visível, a *thread* é encerrada;
- Caso a *thread* não seja encerrada no passo anterior, calcular a soma das distâncias segundo as funções em (4.6), (4.7) e (4.8) referentes a cada uma das características usadas considerando seus respectivos pesos;
- Por fim, antes de ser encerrada, a *thread* salva esse valor em um vetor global em que cada uma de suas posições guarda o “erro” associado à dessemelhança de cada par de pontos comparados.

Ao final do processamento de todas as *threads*, é calculado a média dos erros. Esse valor, que corresponde à aptidão da partícula avaliada, é então copiado para a CPU como resultado do *kernel* executado.

5. Experimentos e Resultados

Este capítulo apresenta os detalhes e os resultados dos experimentos realizados usando a técnica de rastreamento baseado em PSO proposta. Esses resultados ainda são comparados com técnicas de rastreamento que apresentam características semelhantes e que estão presentes no estado da arte. A Seção 5.1 apresenta os detalhes dos experimentos realizados e é dividida em três subseções: a Subseção 5.1.1, que mostra a base de dados e as métricas usadas para avaliar e comparar a precisão das técnicas; a Subseção 5.1.2, que faz uma breve descrição do ambiente usado para desenvolver e realizar os experimentos; e a Subseção 5.1.3, que apresenta a metodologia usada nos experimentos. A Seção 5.2, por sua vez, expõe os resultados dos experimentos, bem como oferece uma análise do significado desses resultados. Essa seção é organizada em três subseções: a Subseção 5.2.1, que apresenta os resultados obtidos a partir de uma base de dados sintética; a Subseção 5.2.2, que discute os resultados em uma base de dados real; e a Subseção 5.2.3, que mostra as melhorias no desempenho da técnica a partir do uso de GPU.

5.1. Metodologia Experimental

Os experimentos foram realizados visando alcançar dois objetivos: o primeiro propõe verificar a precisão e o desempenho do método proposto, determinando assim sua viabilidade em rastrear objetos 3D com 6-DOF a partir de imagens RGB-D; e o segundo objetivo é, uma vez conhecidos a precisão e o desempenho do método, compará-los com os de técnicas de rastreamento que usam PF presentes no estado da arte, como aquelas propostas em [6] e [8].

5.1.1. Base de Dados e Métricas

Em aplicações de RA, é desejável realizar o correto alinhamento de objetos virtuais em cenas reais, também conhecido como *registro*. Esse tipo de tarefa depende da qualidade do rastreamento das poses da câmera em relação à cena ao longo da filmagem. Uma forma de melhorar a qualidade do registro de objetos em uma cena é através do aumento da precisão do rastreamento. Diversas técnicas de rastreamento têm sido estudadas e desenvolvidas com o intuito de conseguir alcançar um rastreamento preciso. É possível verificar a precisão de técnicas de rastreamento através da análise do conjunto de poses encontradas por elas. Essa análise pode ser qualitativa, quando a parte visível do modelo é projetada sobre a imagem

capturada segundo a respectiva pose encontrada, ou quantitativa, quando as poses encontradas são numericamente comparadas com um *ground truth* preciso do rastreamento. Apesar da análise quantitativa ser mais confiável que a qualitativa, é comum fazer uso delas em conjunto com o objetivo de analisar os resultados de uma técnica de rastreamento isoladamente, bem como fazer comparações entres diferentes técnicas.

Para possibilitar a avaliação do método proposto, foram usados os casos de teste do *RGB-D Object Pose Tracking Dataset* [6]. A escolha dessa base de dados se deu por três principais motivos: ela possui diferentes casos de teste com diferentes níveis de dificuldade de rastreamento, dentre eles aqueles obtidos a partir de cenas sintéticas e cenas reais; oferece um *ground truth* preciso dos casos de testes sintéticos, o que possibilita uma análise quantitativa dos resultados; e é uma base de dados aberta. O uso dessa base proporciona ainda uma comparação dos resultados entre diferentes métodos de rastreamento, uma vez que tem sido usada por outros pesquisadores em trabalhos recentes [4][24][47][48].

O *RGB-D Object Pose Tracking Dataset* é composto por um conjunto de seis casos de testes, sendo quatro deles construídos a partir de um cenário sintético e dois obtidos a partir de imagens de ambientes reais. Cada um dos casos de teste contém uma sequência de 1.000 quadros com informações em RGB-D. Essas sequências de imagens oferecem diferentes situações que possibilitam testar o método de rastreamento sobre aspectos e níveis de dificuldade variados. A base usada ainda dispõe dos modelos 3D de quatro objetos rígidos e sem articulações (*Tide, Milk, Orange Juice* e *Kinect Box*,) e de um cenário (*Kitchen*) que podem ser usados durante o rastreamento em seus respectivos casos de teste. A Figura 5.1 apresenta esses modelos na mesma ordem em que foram citados.

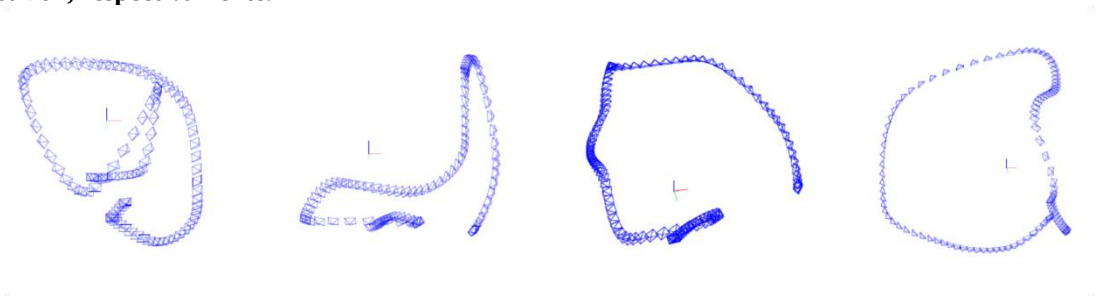
Figura 5.1 – Modelos disponíveis no *RGB-D Object Pose Tracking Dataset*.

Na avaliação quantitativa da precisão do rastreamento, a métrica utilizada foi a raiz do valor quadrático médio (*Root Mean Square* – RMS) dos erros. O RMS dos erros é calculado para cada componente da pose encontrada pelo algoritmo ao longo do rastreamento da trajetória em relação à trajetória descrita no *ground truth* segundo a equação:

$$RMS(k) = \sqrt{\frac{\sum_{i=1}^{n}(\boldsymbol{P_i}(k)-\boldsymbol{G_i}(k))^2}{n}}, \tag{5.1}$$

em que $\boldsymbol{P_i}$ representa a pose encontrada pelo método avaliado no i-ésimo quadro da sequência, $\boldsymbol{G_i}$ a i-ésima pose do *ground truth*, k corresponde à componente dentro do vetor de cada pose e n o número total de quadros usados durante o rastreamento. A Figura 5.2 apresenta o *ground truth* da trajetória da câmera ao redor do objeto em cada uma das quatro sequências sintéticas.

Figura 5.2 – Trajetórias da câmera nas sequências sintéticas do *RGB-D Object Pose Tracking Dataset*. Da esquerda para a direita, estão representadas as trajetórias nas sequências do *Tide*, *Milk*, *Orange Juice* e *Kinect Box*, respectivamente.



Para proporcionar a comparação entre o rastreamento baseado em PSO e outras técnicas de rastreamento do estado da arte, a unidade de medida do RMS dos erros referente às componentes do vetor de rotação foi convertida em ângulos de Euler, sendo sua unidade de medida o grau. Nas componentes do vetor de translação a unidade de medida usada foi o milímetro. Para avaliar o desempenho do método, foi usado o tempo de médio gasto em segundos para encontrar a pose do objeto em um quadro para cada caso de teste.

5.1.2. Ambiente de Desenvolvimento e Testes

Os experimentos foram desenvolvidos e realizados em um notebook com processador Intel Core i7-720QM @ 1.60 GHz, 16 GB RAM e uma placa gráfica NVIDIA GeForce GT 330M GPU. O algoritmo foi implementado em C/C++ usando as bibliotecas: *Point Cloud Library* (PCL)[1], para carregar e manipular as nuvens de pontos da base de dados; *Open Source Computer Vision Library* (OpenCV)[2], na representação e manipulação de vetores e matrizes; *GNU Scientific Library* (GSL)[3], para facilitar os cálculos matemáticos e para gerar números aleatórios; CUDA SDK[4], para implementar e processar os algoritmos em GPU; e

[1] http://pointclouds.org
[2] http://opencv.org
[3] http://www.gnu.org/software/gsl
[4] https://developer.nvidia.com/cuda-toolkit

Particle Swarm Optimization in C (PSO in C)[5], que foi modificada e usada como método de otimização do conjunto de hipóteses de pose em cada quadro do rastreamento.

5.1.3. Metodologia de Avaliação

Para avaliação do método proposto, foi definida uma sequência de experimentos em que todos os casos de testes sintéticos e reais da *RGB-D Object Pose Tracking Dataset* foram usados. Nessa sequência, os experimentos com cada um dos quatro casos de testes sintéticos foram repetidos cinco vezes, mantendo-se todas as configurações principais do PSO fixas e variando-se apenas o número de partículas. Isto é, com o critério de parada do PSO fixo em 100 iterações foram realizados experimentos com 8, 16, 32, 64 e 128 partículas. No PSO, o número total de chamadas da função de avaliação (ou simplesmente *número de avaliações*) pode ser calculado como o produto entre o número partículas do enxame e a quantidade de iterações usadas como critério de parada. Dessa forma, pode-se afirmar que foram realizados experimentos usando 800, 1.600, 3.200, 6.400 e 12.800 avaliações para cada caso de teste sintético. Os experimentos com diferentes tamanhos de enxames ajudam a entender a relação entre a precisão do rastreamento e a quantidade de partículas ou o número de avaliações usadas.

Todos os resultados dos experimentos realizados com os casos de teste sintéticos foram comparados com duas outras técnicas de rastreamento de objetos 3D com 6-DOF baseadas em PF, descritas em [6] e [8]. Essas abordagens são *top-down* e usam o PF como método de otimização de conjuntos de hipóteses de pose. A métrica usada para comparação foi o RMS dos erros das componentes das poses obtidas por cada técnica ao longo da sequência de quadros. Uma vez que os casos de teste foram repetidos para conjuntos de hipóteses de pose de diferentes tamanhos, os resultados podem ser agrupados não apenas por caso de teste, mas também pelo número de avaliações efetuadas por cada uma das técnicas.

Quanto aos casos de testes em ambientes reais, foram realizados experimentos com um PSO de 128 partículas e critério de parada de 100 iterações. Uma vez que esses casos de teste não possuem valores de *ground truth*, foi possível apenas uma análise qualitativa da precisão dos resultados. A avaliação de desempenho segue os mesmos critérios dos casos de teste sintéticos.

[5] https://github.com/kkentzo/pso

5.2. Resultados dos Experimentos

Em todos os experimentos o PSO foi instanciado usando os parâmetros descritos a seguir: peso de inércia w dinâmico, limitado por $w_{max} = 0{,}7298$ e $w_{min} = 0{,}3$; topologia *lbest*; parâmetros cognitivos e sociais $c_1 = c_2 = 1{,}496$; amplitudes das fronteiras do espaço de busca $b_r = 0{,}75$ e $y_t = 0{,}04$ m; limiar da distância euclidiana das coordenadas 3D na função de aptidão $l = 0{,}01$ m; critério de parada de 100 iterações. O número de partículas usadas variou de acordo com o objetivo de cada experimento, como explicado na Subseção 5.1.3. As constantes λ_p, λ_c e λ_n foram definidas para cada um dos modelos de modo que o retorno das funções (4.6), (4.7) e (4.8) tenham aproximadamente o mesmo peso na composição do resultado final da função de aptidão (4.2) a cada avaliação.

5.2.1. Base de Dados Sintética

Como apresentado em [6], as sequências de imagens e os objetos a serem rastreados nos casos de testes da base de dados apresentam diferentes níveis de dificuldade. Isso pode ser notado nos resultados, uma vez que o RMS dos erros em cada experimento apresentou valores bastante distintos, variando com o caso de teste ou o método de rastreamento em questão. As tabelas e gráficos nesta subseção exibem os dados referentes à precisão da técnica de rastreamento baseado em PSO proposta em cada um dos casos de teste sintéticos. Esses dados ainda foram comparados com os resultados de outras duas técnicas de rastreamento baseadas em PF presentes no estado da arte.

O primeiro conjunto de experimentos foi realizado com o caso de teste do modelo "*Orange Juice*". Esse modelo representa uma caixa de suco de laranja e é composto por uma nuvem de 19.121 pontos. Sua sequência de imagens não possui trechos com oclusão e, uma vez que se trata de uma caixa semelhante a um paralelepípedo, não possui auto-oclusão. As constantes usadas na função de aptidão foram $\lambda_p = 1{,}23$, $\lambda_n = 2{,}84$ e $\lambda_c = 1{,}0$. Os resultados quantitativos do rastreamento foram resumidos na Tabela 5.1. Nela pode ser observado o RMS dos erros das componentes das poses encontradas por cada técnica de rastreamento em relação ao *ground truth* correspondente. Os resultados são também apresentados de acordo com o número de avaliações usadas por cada técnica. Para facilitar a comparação entre as técnicas, os melhores resultados de cada experimento foram escritos em negrito.

Tabela 5.1 – RMS dos erros do rastreamento realizado no caso de teste sintético "*Orange Juice*" usando PSO, PF de [6] **e o PF da PCL** [8] **(melhores resultados em negrito).**

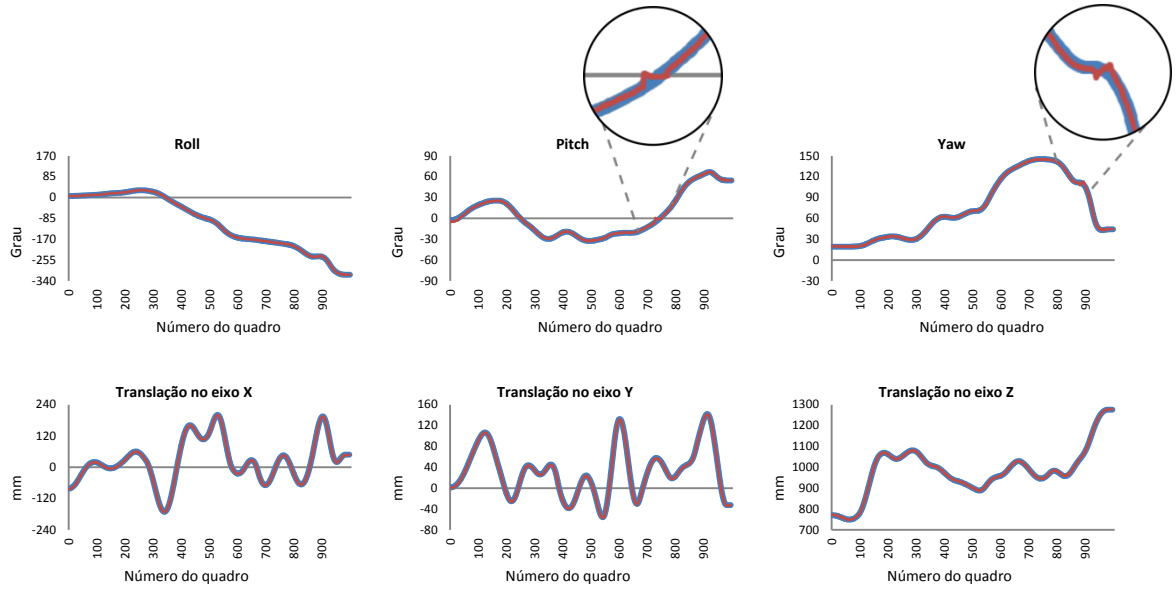
Objeto	Rastreador	Número de Avaliações	RMS dos Erros					
			Yaw (graus)	Pitch (graus)	Roll (graus)	X (mm)	Y (mm)	Z (mm)
Orange Juice	PSO	800	**0,80**	**1,12**	**0,67**	**1,20**	**1,02**	**1,50**
		1600	**0,61**	**1,04**	**0,59**	**1,16**	**0,92**	**1,20**
		3200	**0,61**	**0,83**	**0,57**	**1,09**	**0,90**	**1,15**
		6400	**0,59**	**0,80**	**0,57**	**1,09**	**0,90**	**1,07**
		12800	**0,58**	**0,74**	**0,55**	1,06	**0,90**	**1,12**
	PF de [6]	800	2,80	1,44	2,55	2,18	2,53	1,94
		1600	2,76	1,29	2,36	1,86	2,39	1,79
		3200	2,09	1,09	1,54	1,56	2,17	1,50
		6400	1,61	0,84	1,54	1,12	1,61	1,45
		12800	1,39	0,75	1,32	**0,96**	1,44	1,17
	PF da PCL [8]	800	45,87	42,36	84,76	3,06	2,54	2,44
		1600	46,37	41,65	84,42	2,61	2,39	2,11
		3200	50,67	51,81	97,34	26,76	5,18	10,83
		6400	65,90	53,43	92,67	26,86	5,30	11,19
		12800	46,37	42,12	85,81	2,53	2,20	1,91

A partir desses dados é possível inferir que o método de rastreamento baseado em PSO sugerido neste trabalho obteve de forma geral resultados mais precisos que as abordagens baseadas em PF de [6] e da PCL [8] usados como comparação em todos os casos de testes. Se os resultados forem observados no nível das componentes do vetor de pose, é possível verificar apenas uma exceção na componente X da translação no experimento com 12.800 avaliações, em que o PF de [6] obteve um RMS dos erros mais baixo do que aquele obtido pelo PSO.

Adicionalmente às informações da Tabela 5.1, a precisão do rastreamento do "*Orange Juice*" no experimento com 12.800 avaliações pode ser observada através dos gráficos da Figura 5.3. Esses gráficos apresentam os valores das componentes das poses encontradas durante o rastreamento ao longo da sequência de quadros (em vermelho) e seus respectivos valores de *ground truth* (em segundo plano e de cor azul). Uma vez que os gráficos referentes ao *ground truth* estão expostos em segundo plano, com o intuito de proporcionar uma melhor visualização, os mesmos foram renderizados propositadamente com uma espessura ligeiramente maior.

Alguns detalhes dos gráficos com as imprecisões mais acentuadas foram ampliados para melhor visualização. Uma vez que esses não são os únicos trechos com imprecisão, todos os gráficos são apresentados em tamanho ampliado no apêndice desta dissertação. Nos detalhes ampliados dos gráficos da Figura 5.3 é possível visualizar dois pequenos trechos em que ocorreram imprecisões mais acentuadas no rastreamento das componentes $pitch$ e yaw do vetor de rotação, o primeiro entre os quadros q_{700} e q_{800} e o segundo entre q_{800} e q_{900}.

Figura 5.3 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Orange Juice*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).



No segundo conjunto de experimentos foi usado o caso de teste do modelo "*Tide*". Esse é o menor dos modelos pertencentes à base de dados, ele representa um recipiente de sabão e é composto por uma nuvem com apenas 4.755 pontos. A sequência de quadros possui apenas um trecho com uma pequena oclusão do objeto que vai aproximadamente do quadro q_{384} até o q_{434}. Devido à alça do recipiente, esse modelo sofre auto-oclusão em vários trechos. Os pesos usados na função de aptidão foram $\lambda_p = 9{,}0$, $\lambda_n = 10{,}20$ e $\lambda_c = 1{,}0$. A precisão do rastreamento nessa sequência é exibida na Tabela 5.2.

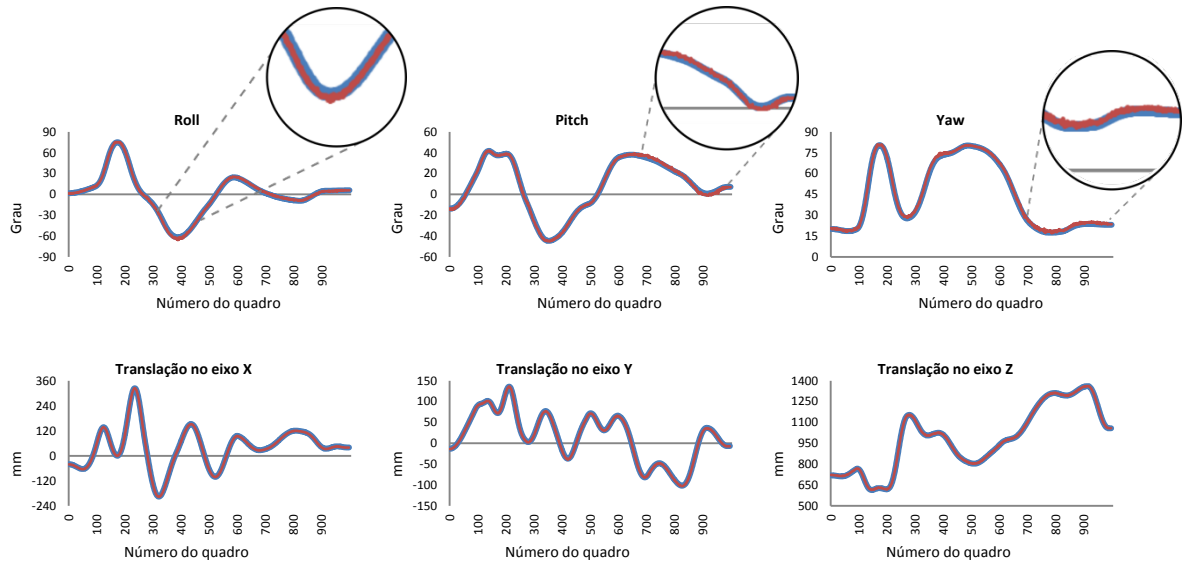
Tabela 5.2 – RMS dos erros do rastreamento realizado no caso de teste sintético "*Tide*" usando PSO, PF de [6] **e PF da PCL** [8] **(melhores resultados em negrito).**

Objeto	Rastreador	Número de Avaliações	RMS dos Erros					
			Yaw (graus)	Pitch (graus)	Roll (graus)	X (mm)	Y (mm)	Z (mm)
Tide	PSO	800	**1,34**	**1,05**	**1,58**	**1,04**	**0,85**	1,74
		1600	**1,04**	**0,90**	**1,04**	**0,94**	**0,72**	**1,46**
		3200	**0,99**	**0,88**	**0,97**	**0,93**	**0,70**	1,36
		6400	**0,97**	**0,85**	**0,93**	**0,93**	**0,71**	1,36
		12800	**0,94**	**0,85**	**0,91**	0,92	**0,70**	1,36
	PF de [6]	800	2,32	1,93	3,79	1,88	2,96	2,24
		1600	2,21	1,71	3,48	1,66	2,42	1,91
		3200	1,58	1,36	2,43	1,27	1,87	1,54
		6400	1,39	1,13	2,25	1,14	1,54	1,42
		12800	1,13	1,09	1,78	**0,83**	1,37	1,20
	PF da PCL [8]	800	3,20	2,35	5,43	1,74	2,79	**1,59**
		1600	3,39	2,23	5,58	1,69	2,61	1,51
		3200	3,00	2,17	5,26	1,49	2,50	**1,11**
		6400	2,93	2,15	5,02	1,34	2,11	**0,95**
		12800	2,98	2,13	5,15	1,46	2,25	**0,92**

Observando os dados no nível do vetor de pose, percebe-se que o método baseado em PSO obteve os melhores resultados em todos os experimentos nas componentes de rotação e na componente Y da translação. O PF da PCL obteve os melhores resultados em quase todos os experimentos em relação à translação na direção do eixo principal do sistema de coordenadas da câmera, isto é, em relação à componente Z no vetor de translação da pose. Ainda no nível das componentes, o PF de [6] obteve o resultado mais preciso na componente X com o experimento usando 12.800 avaliações.

A precisão do rastreamento de cada componente usando PSO com 12.800 avaliações pode ser visualizada ainda nos gráficos da Figura 5.4. Nela é possível verificar trechos com imprecisões nas componentes do vetor de rotação das poses encontrado pelo algoritmo. Nas componentes do vetor de translação não foram observadas grandes imprecisões a partir dos gráficos.

Figura 5.4 – Gráficos com as componentes das poses encontradas durante o rastreamento do *"Tide"* usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).



Os experimentos pertencentes ao terceiro conjunto usaram o modelo de uma garrafa de leite, chamado de "*Milk*". Esse modelo é formado por uma nuvem de 17.325 pontos. A sequência de quadros desse caso de teste apresenta dois trechos de oclusão, o primeiro no intervalo $[q_{254}, q_{361}]$ e o segundo no intervalo $[q_{647}, q_{743}]$. Assim como o modelo do "*Tide*", devido a sua alça o modelo desse caso de teste sofre auto-oclusão em diversos trechos. Para atribuir diferentes pesos às funções no cálculo da aptidão, foram usados os seguintes valores: $\lambda_p = 1{,}6$, $\lambda_n = 2{,}8$ e $\lambda_c = 1{,}0$. A Tabela 5.3 apresenta a precisão de cada uma das técnicas obtidas nesses experimentos.

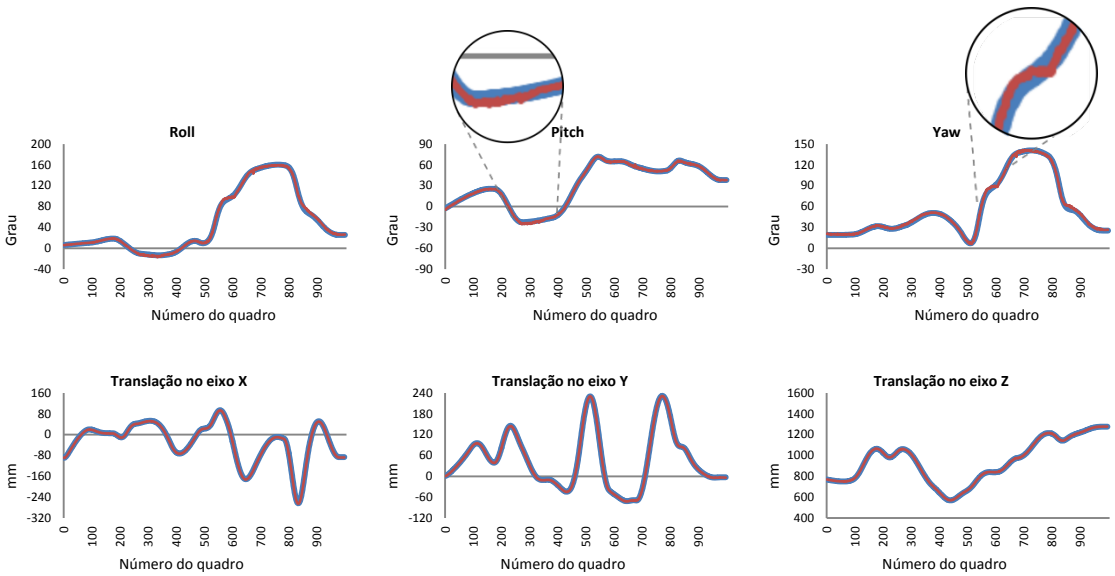
Tabela 5.3 – RMS dos erros do rastreamento realizado no caso de teste sintético "*Milk*" usando PSO, PF de [6] **e PF da PCL** [8] **(melhores resultados em negrito).**

Objeto	Rastreador	Número de Avaliações	RMS dos Erros					
			Yaw (graus)	Pitch (graus)	Roll (graus)	X (mm)	Y (mm)	Z (mm)
Milk	**PSO**	800	**1,81**	**0,81**	**1,77**	**1,01**	**0,67**	**1,54**
		1600	**1,55**	**0,72**	**1,55**	**1,00**	**0,65**	**1,39**
		3200	**1,50**	**0,71**	**1,51**	**1,00**	**0,65**	1,43
		6400	**1,45**	**0,71**	**1,46**	**1,01**	**0,66**	1,44
		12800	**1,47**	**0,72**	**1,49**	1,00	**0,66**	1,49
	PF de [6]	800	6,96	2,03	7,77	1,79	3,16	1,97
		1600	3,90	1,68	4,35	1,28	2,55	1,74
		3200	5,49	1,74	6,22	1,20	2,37	**1,41**
		6400	3,47	1,44	4,00	1,05	2,07	**1,21**
		12800	3,26	1,41	3,83	**0,93**	1,94	**1,09**
	PF da PCL [8]	800	42,89	33,78	52,23	2,36	4,81	2,03
		1600	40,22	33,65	49,27	1,78	3,99	1,69
		3200	74,78	21,52	64,45	13,68	43,72	24,92
		6400	45,67	34,51	55,55	2,03	4,24	1,57
		12800	75,03	19,58	59,37	13,38	31,45	26,09

Mais uma vez o método proposto neste trabalho obteve os melhores resultados globais. Analisando no nível das componentes, observam-se resultados semelhantes aos dos experimentos com o "*Tide*", em que PSO obteve os menores RMS dos erros em quase todas as componentes com exceção das componentes X e Z da translação em alguns experimentos. O PSO foi menos preciso em pelo menos uma componente em quatro experimentos. Em três deles o PF de [6] obteve os melhores resultados considerando a componente Z da translação e no quarto ele foi mais preciso em relação à componente X desse mesmo vetor. O PF da PCL não obteve o melhor resultado em nenhum dos experimentos.

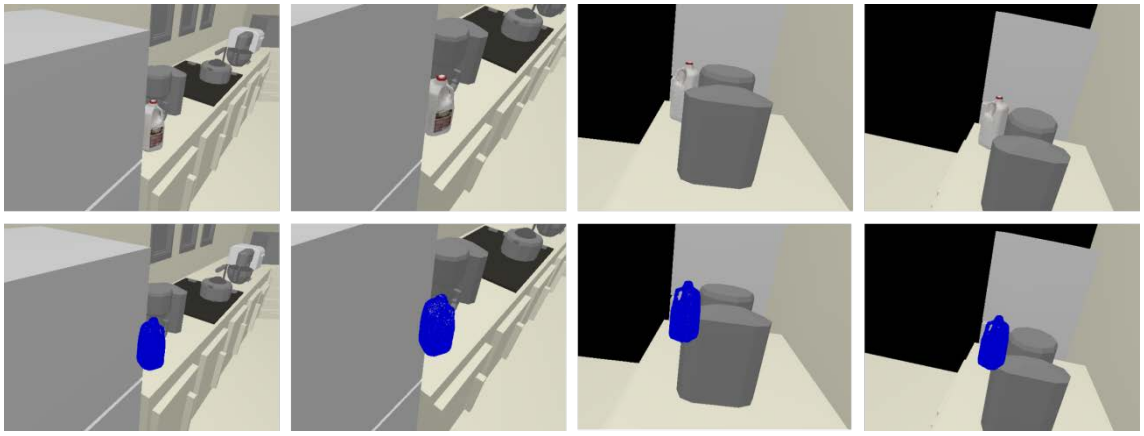
Os gráficos da Figura 5.5 apresentam visualmente a precisão da técnica proposta em relação ao *ground truth* no experimento com 12.800 avaliações. Nos detalhes ampliados são mostrados dois trechos em que houve maiores imprecisões no rastreamento das componentes $pitch$ e yaw do vetor de rotação. Segundo esses gráficos, não houve grandes imprecisões em relação ao vetor de translação das poses rastreadas.

Figura 5.5 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Milk*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).



Além da precisão do método baseado em PSO, os gráficos da Figura 5.5 apontam que o mesmo possui certa robustez à oclusão parcial durante o rastreamento. Apesar da oclusão sofrida pelo objeto rastreado nos trechos mencionados, os gráficos indicam que o rastreamento usando PSO se mostrou preciso nesses trechos, uma vez que durante as oclusões não se observa nenhuma grande diferença entre os valores das componentes rastreadas pelo método (em vermelho) e os valores das componentes do *ground truth* (em azul). A Figura 5.6 apresenta o resultado do rastreamento em quatro quadros pertencentes aos trechos com oclusão.

Figura 5.6 – Rastreamento baseado em PSO em trechos com oclusão na sequência do "*Milk*". Da esquerda para a direita, são apresentados os quadros em RGB q_{280}, q_{345}, q_{680} e q_{725} (acima) e os respectivos resultados do rastreamento (abaixo).



O último conjunto de experimentos foi realizado com a sequência de imagens do "*Kinect Box*". Esse caso de teste usa um modelo de uma caixa de um Kinect representada por

uma nuvem com 15.363 pontos. Dos quatro casos de teste sintéticos usados, esse apresentou o maior grau de dificuldade em ser rastreado de forma precisa. A sequência de quadros apresenta uma oclusão que se estende aproximadamente do quadro q_{355} até o quadro q_{735}. Esse caso de teste não possui trechos com auto-oclusão. As constantes usadas como pesos na função de aptidão para esses experimentos foram $\lambda_p = 9{,}0$, $\lambda_n = 10{,}2$ e $\lambda_c = 1{,}0$. Os dados exibidos na Tabela 5.4 apresentam a precisão de cada uma das técnicas analisadas em relação a esse caso de teste.

Tabela 5.4 – RMS dos erros do rastreamento realizado no caso de teste sintético "*Kinect Box*" usando PSO, PF de [6] **e PF da PCL** [8] **(melhores resultados em negrito).**

Objeto	Rastreador	Número de Avaliações	RMS dos Erros					
			Yaw (graus)	Pitch (graus)	Roll (graus)	X (mm)	Y (mm)	Z (mm)
Kinect Box	**PSO**	800	**5,34**	1,63	**5,36**	**1,75**	**1,78**	**2,54**
		1600	**5,28**	1,57	**5,34**	**1,50**	**1,38**	2,60
		3200	**3,54**	1,55	**3,58**	**1,35**	**1,24**	**2,73**
		6400	**3,29**	1,50	**3,31**	**1,32**	**1,21**	2,79
		12800	**2,92**	1,50	**2,94**	**1,33**	**1,21**	2,81
	PF de [6]	800	8,43	**1,58**	8,40	3,67	5,28	2,60
		1600	7,86	**1,17**	7,86	3,37	4,13	**2,15**
		3200	7,51	**1,03**	7,63	6,11	9,16	7,51
		6400	7,52	**1,42**	8,14	2,38	3,28	**1,52**
		12800	6,32	**0,76**	6,41	1,84	2,23	**1,36**
	PF da PCL [8]	800	10,63	2,31	10,23	44,83	43,50	56,51
		1600	10,58	2,33	9,80	43,93	42,70	55,70
		3200	11,33	1,94	11,82	44,59	42,93	55,78
		6400	7,74	2,08	7,21	43,43	41,92	55,78
		12800	8,31	1,87	7,62	43,99	42,51	55,89

Apesar de esse conjunto de experimentos ter sido aquele em que o rastreamento baseado em PSO obteve os resultados menos precisos em média, no nível dos vetores de pose a técnica proposta ainda obteve os melhores resultados em relação às outras técnicas, uma vez que foi melhor que os outros métodos na maioria das componentes do vetor de pose. Porém, observando no nível dessas componentes, o rastreamento usando o PF de [6] se mostrou mais preciso em relação ao $pitch$ do vetor de rotação em todos os experimentos e em relação à componente Z do vetor de translação em três deles. O PF da PCL não obteve resultados mais precisos em relação às outras duas técnicas em nenhum dos experimentos.

Apesar desse caso de teste possuir um grande trecho com oclusão, de acordo com os gráficos da Figura 5.7 e a partir de uma avaliação qualitativa das projeções das poses rastreadas (como na Figura 5.8), não é possível afirmar que a diminuição da precisão do rastreamento nesses casos de teste tenha sido causada pela oclusão. Contudo, observando-se os gráficos na Figura 5.7 é possível verificar uma diminuição da precisão do rastreamento em relação ao *ground truth* em três momentos específicos: o primeiro nas proximidades do quadro q_{100}, mais precisamente no intervalo $[q_{72},\ q_{122}]$; o segundo no intervalo $[q_{321},\ q_{358}]$;

e o terceiro em $[q_{485},\ q_{515}]$. Esses intervalos com imprecisões são observados principalmente em relação à componente $pitch$ do vetor de rotação, justamente aquela componente em que o PSO se mostrou menos preciso em todos os experimentos, como foi apresentado na Tabela 5.4. As ampliações no gráfico dessa componente facilitam a confirmação dessa afirmação.

Figura 5.7 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Kinect Box*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).

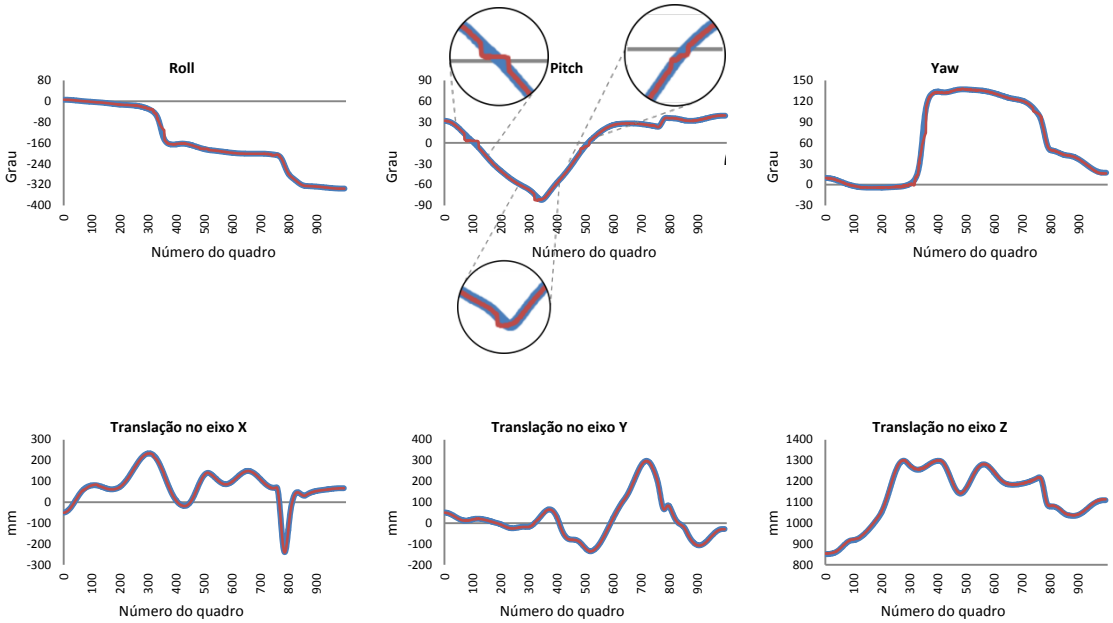
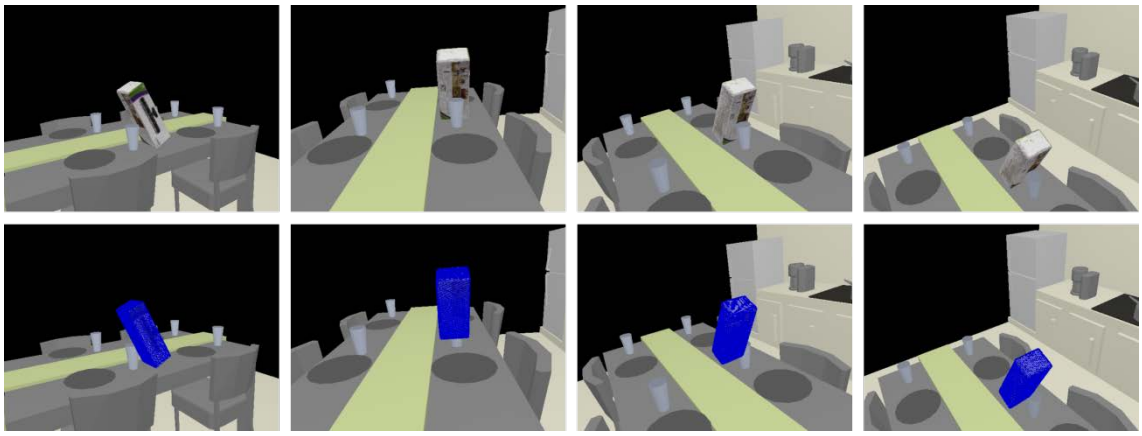


Figura 5.8 – Rastreamento baseado em PSO em trechos com oclusão na sequência do "*Kinect Box*". Da esquerda para a direita, são apresentados os quadros em RGB q_{400}, q_{500}, q_{600} e q_{700} (acima) e os respectivos resultados do rastreamento (abaixo).

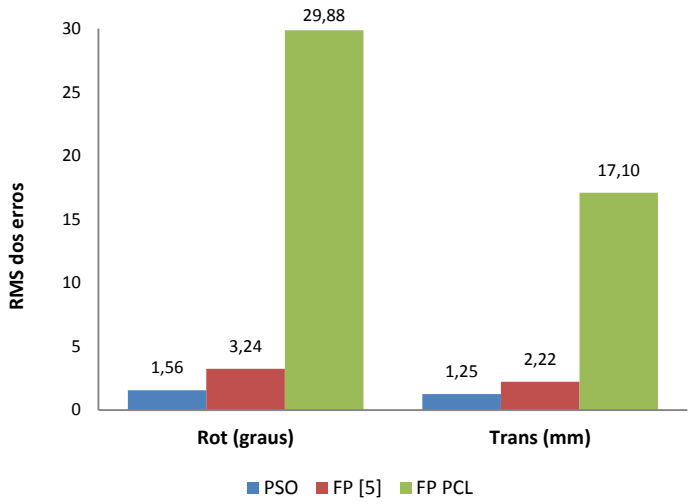


Os trechos de imprecisão no rastreamento mencionados anteriormente correspondem aos quadros próximos aos instantes em que a câmera gira em torno da caixa de Kinect e uma de suas faces passa a não ser mais visível. Esse problema se mostrou mais evidente nos experimentos com um menor número de avaliações, porém como observado nos gráficos apresentados, mesmo nos experimentos com quantidades maiores de partículas a perda da

visibilidade de uma das faces aparentemente continuou causando a diminuição da qualidade do rastreamento usando PSO.

A partir dos resultados dos experimentos expostos com os casos de testes sintéticos, é possível observar que o rastreamento baseado em PSO, apesar de ter demostrado alguns resultados menos precisos no nível das componentes do vetor de pose, apresentou os melhores resultados globais em todos os experimentos realizados. Isso pode ser reafirmado quando observada a precisão do rastreamento no nível dos vetores de rotação e translação através da média geral do RMS dos erros das componentes desses vetores em relação aos valores de *ground truth* obtidos em todos os experimentos. O gráfico da Figura 5.9 colabora com essa análise, uma vez que apresenta os valores dessas médias em todos os casos de testes sintéticos usados.

Figura 5.9 – Média geral do RMS dos erros das rotações e translações em relação ao *ground truth* da base de dados obtida por cada uma das técnicas analisadas.



A partir dos resultados expostos, percebe-se que em todos os experimentos com PSO e PF a precisão do rastreamento tende a aumentar à medida que é usada uma quantidade maior de partículas, ou nesse caso o equivalente: um número maior de avaliações. É importante observar ainda que o PSO apresentou resultados precisos mesmo quando instanciado com uma quantidade muito pequena de partículas. Uma vez que nos primeiros experimentos para cada caso de testes foram realizadas 800 avaliações e o critério de parada usado no PSO foi fixado em 100 iterações, isso significa que os enxames de partículas nesses experimentos foram compostos por apenas 8 partículas. Nesse sentido, é possível afirmar que na comparação dos experimentos com PSO de menor enxame com os resultados obtidos pelas

outras técnicas usando o maior conjunto de partículas, isto é, aqueles com 12.800 partículas, o rastreamento baseado em PSO ainda apresenta os melhores resultados em vários deles.

Apesar de o rastreamento baseado em PSO se mostrar preciso mesmo quando são usados pequenos enxames de partículas, esses resultados só começaram a ser registrados a partir do momento em que os experimentos passaram a ser executados usando como critério de parada do PSO um maior número de iterações e com a mudança no algoritmo do PSO relativa à implementação do reaproveitamento da melhor partícula do enxame anterior a cada novo enxame. As configurações do PSO usado antes das mudanças mencionadas são aquelas apresentadas em [4].

5.2.2. Base de Dados Reais

A técnica de rastreamento baseado em PSO também foi usada para rastrear objetos em imagens obtidas a partir de ambientes reais. A base de dados *RGB-D Object Pose Tracking Dataset* possui dois casos de teste reais que utilizam respectivamente os modelos "*Tide*" e "*Milk*" para possibilitar o rastreamento. Ambos os casos de teste possuem trechos com auto-oclusão, uma vez que, como já foi mencionado na Subseção 5.2.2, esses modelos possuem alças. Como pode ser observado na Figura 5.10 e na Figura 5.11, as duas sequências também possuem trechos com forte oclusão parcial causada por outros objetos pertencentes à cena, tais como potes plásticos ou a mão da pessoa que manipula os objetos rastreados.

Durante os experimentos foram usados enxames de 128 partículas e critério de parada de 100 iterações. O restante dos parâmetros fixos são os mesmos das instâncias usadas nos experimentos com os casos de teste sintéticos descritos na Subseção 5.2.1. Uma vez que essas sequências não possuem valores de *ground truth*, é possível realizar apenas uma análise qualitativa dos resultados. Na Figura 5.10 podemos observar uma pequena amostra do resultado do rastreamento. Nela são exibidos os quadros q_{200}, q_{400}, q_{600} e q_{800} de ambas as sequências e logo abaixo de cada um desses quadros seus respectivos resultados.

Figura 5.10 – Rastreamento baseado em PSO nas sequências de imagens reais dos objetos "*Milk*" e "*Tide*". Para cada sequência, são exibidas a entrada RGB (acima) e o resultado do rastreamento (abaixo).

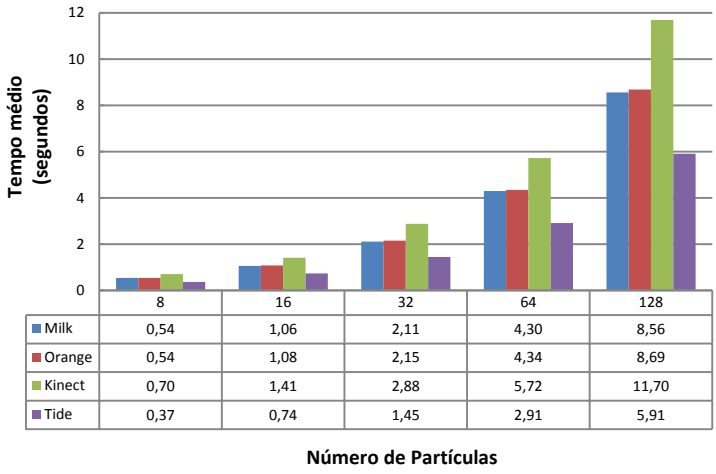
Fazendo uma análise qualitativa dos resultados, apesar do modelo 3D do "*Tide*" ter algumas dimensões diferentes do recipiente "*Tide*" real, é possível afirmar que o método baseado em PSO foi capaz de rastrear os objetos ao longo das sequências com poses visualmente muito próximas da pose real na maioria dos quadros. Sequências de poses visualmente imprecisas foram registradas em alguns quadros, aparentemente causadas por dois motivos: a oclusão parcial produzida pela mão da pessoa que manipula o objeto e posições do objeto em que o número de pontos visíveis diminui consideravelmente. Porém, não é possível fazer uma relação de causalidade definitiva e previsível entre esses motivos citados e a imprecisão das poses, uma vez que houve momentos em que ambos ocorreram e a qualidade da pose aparentemente não foi afetada. Na Figura 5.11 é possível visualizar alguns quadros em que o rastreamento baseado em PSO gerou poses imprecisas. Em outros trechos do rastreamento, em que os objetos foram parcialmente oclusos pelos recipientes plásticos cilíndricos presentes na cena, não foi apresentada visualmente perda na qualidade das poses encontradas.

Figura 5.11 – Poses visualmente imprecisas obtidas durante o rastreamento baseado em PSO. Acima os quadros q_{384}, q_{524}, q_{618} e q_{661} da sequência real "*Milk*" e abaixo os quadros q_{421}, q_{465}, q_{619} e q_{675} da sequência real "*Tide*".

5.2.3. Uso da GPU

O processamento da função de aptidão do PSO em GPU foi usado nas sequências reais e sintéticas e se mostrou bastante promissor, uma vez que, com o uso desse tipo de processamento, o desempenho da abordagem proposta melhorou satisfatoriamente em relação à primeira versão apresentada em [4] no qual, durante os experimentos com a mesma base de dados, o PSO foi processado integralmente em CPU e levou em média cinco minutos para completar o rastreamento de apenas um quadro da sequência usando uma configuração com 4.000 avaliações (100 partículas e critério de parada de 40 iterações). O tempo médio de processamento de cada quadro da sequência por caso de teste em cada um dos experimentos após a implementação da função de aptidão em GPU é exposto no gráfico da Figura 5.12.

Figura 5.12 – Tempo médio para rastrear o objeto 3D em um quadro de cada caso de teste e de acordo com a quantidade de partículas usadas no PSO.



	8	16	32	64	128
Milk	0,54	1,06	2,11	4,30	8,56
Orange	0,54	1,08	2,15	4,34	8,69
Kinect	0,70	1,41	2,88	5,72	11,70
Tide	0,37	0,74	1,45	2,91	5,91

Nesse gráfico verifica-se um crescimento do tempo de processamento aproximadamente linear em relação ao número de partículas usado. O processamento da sequência “*Tide*” foi o mais rápido dos quatro casos de teste. Isso ocorreu porque o modelo 3D do “*Tide*” possui aproximadamente um terço da quantidade de pontos dos outros modelos.

Quando implementado em GPU o rastreamento usando técnicas baseadas em PF apresentou resultados com tempos de processamento mais baixos que aqueles mostrados aqui usando PSO. Em alguns casos o PF em GPU possibilitou a execução do rastreamento em tempo real, como pode ser observado em [6]. Porém, apesar desses resultados apontarem que o PF possui um desempenho melhor em relação ao tempo de processamento se comparado à versão da técnica proposta nesta pesquisa, não foi possível realizar uma comparação direta uma vez que o hardware usado nos experimentos em [6] é diferente daquele que foi usado nos experimentos realizados neste trabalho.

6. Conclusão

Este capítulo apresenta as conclusões finais desta dissertação em três seções. A Seção 6.1 mostra uma síntese da avaliação dos resultados obtidos durante os experimentos, ressaltando tanto as qualidades quanto as dificuldades apresentadas pelo método proposto. A Seção 6.2 apresenta as principais contribuições deste trabalho. Já a Seção 6.3 traz sugestões de trabalhos futuros.

6.1. Considerações Finais

O método de rastreamento de objetos 3D baseado em PSO proposto por este trabalho foi desenvolvido, testado em uma base de imagens RGB-D e comparado com outras técnicas presentes no estado da arte. Os resultados dos experimentos, além de mostrarem que o PSO pode ser usado como método de otimização de múltiplas hipóteses de pose no rastreamento de objetos 3D, revelaram ainda que o mesmo proporcionou um rastreamento mais preciso quando comparado com técnicas de rastreamento *top-down* recentes baseadas em PF. O método proposto ainda proporcionou uma boa acurácia mesmo em situações em que o rastreamento de objetos 3D se torna mais difícil, como nos trechos com oclusões parciais e auto-oclusão do objeto rastreado. A extração de características 3D através de imagens RGB-D contribuiu significativamente para o desenvolvimento da técnica proposta, uma vez que essas características foram usadas na composição da função de aptidão do PSO e como método para determinar os pontos visíveis do modelo a cada pose. O uso de processamento paralelo em GPU possibilitou melhorias consideráveis no tempo de processamento em relação ao mesmo algoritmo processado em CPU, apontando assim um caminho que possibilite aumentar ainda mais a taxa de quadros do método.

Entretanto, alguns problemas ainda precisam ser superados. O primeiro, e talvez aquele que mais tenha causado imprecisão no rastreamento, ocorreu em trechos do rastreamento em que há uma iminente perda na visualização dos pontos de uma das faces do objeto devido à movimentação da câmera ao redor do mesmo. O segundo, porém não menos importante, diz respeito ao processamento em GPU, que apesar de ter diminuído bastante o tempo de processamento ainda não proporcionou uma taxa de quadros elevada.

É importante ressaltar que os métodos propostos em [24][47][48] apresentaram resultados muito precisos usando a mesma base de dados apresentada nesta dissertação,

porém uma vez que esses trabalhos propõem métodos que necessitam de aprendizagem de máquina antes de iniciar o rastreamento, os mesmos não foram levados em conta na comparação, que focou em técnicas baseadas em otimização.

6.2. Contribuições

As principais contribuições desta dissertação foram as seguintes:

- O desenvolvimento de um método de rastreamento sem marcadores de objetos 3D genéricos, *top-down* e baseado em PSO;
- A publicação de um artigo completo no evento *19th Symposium on Virtual and Augmented Reality* [4];
- A submissão de um artigo completo para o periódico *Neurocomputing*, que se encontra em processo de revisão;
- Uma avaliação do método proposto.

6.3. Trabalhos Futuros

Como trabalhos futuros, sugere-se um estudo com o intuito de resolver o problema da imprecisão do rastreamento em trechos em que uma das faces do objeto está próxima a ficar paralela em relação à direção do eixo principal da câmera, ou seja, no momento em que há uma iminente perda de visibilidade destes pontos. Quanto ao desempenho do algoritmo, uma vez que o processamento em GPU foi usado apenas na função de aptidão no nível dos pontos do modelo, acredita-se que uma implementação de um paralelismo no nível das partículas do PSO possa diminuir o tempo do processamento usado para rastrear o objeto a cada quadro.

A técnica proposta ainda necessita de uma avaliação através de um conjunto de experimentos mais completo. Uma sugestão seria testá-la em outros casos de teste, pois algumas situações que são encontradas em ambientes reais, tais como mudanças na iluminação, diferentes porcentagens de oclusão como em [5] ou movimentações bruscas de câmera, não foram contempladas nos experimentos realizados nesta dissertação. Propõe-se ainda a utilização da abordagem de otimização proposta juntamente com técnicas de aprendizagem de máquina, seguindo a tendência sugerida por [24].

REFERÊNCIAS BIBLIOGRÁFICAS

[1] V. Lepetit and P. Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey,” *Found. Trends Comput. Graph. Vis.*, vol. 1, no. 1, pp. 1–89, 2005.

[2] F. Umpierre, “Aldeia,” 2018. [Online]. Available: http://aldeia.biz/blog/inovacao/realidade-aumentada-como-ela-mudara-a-maneira-de-lidarmos-com-aplicativos-em-celulares/.

[3] J. Lima, F. P. M. Simões, L. S. Figueiredo, V. Teichrieb, J. Kelner, and I. H. F. Santos, “Model Based 3d Tracking Techniques for Markerless Augmented Reality,” *Symp. Virtual Augment. Real.*, pp. 37–47, 2009.

[4] J. dos Santos Júnior and J. Lima, “3D Object Tracking in RGB-D Images Using Particle Swarm Optimization,” *Symp. Virtual Augment. Real.*, pp. 107–115, 2017.

[5] M. Garon and J. F. Lalonde, “Deep 6-DOF Tracking,” *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 11, pp. 2410–2418, 2017.

[6] C. Choi and H. I. Christensen, “RGB-D Object Tracking: A Particle Filter Approach on GPU,” *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1084–1091, 2013.

[7] O. H. Jafari, D. Mitzel, and B. Leibe, “Real-time RGB-D based people detection and Tracking for mobile robots and head-worn cameras,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5636–5643, 2014.

[8] R. Ueda, “pcl::tracking,” 2012. [Online]. Available: https://goo.gl/o5rmzU.

[9] L. Mussi, S. Ivekovic, and S. Cagnoni, “Markerless articulated human body tracking from multi-view video with GPU-PSO,” *Proc. 9th Int. Conf. Evolvable Syst. from Biol. to Hardw.*, pp. 97–108, 2010.

[10] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Tracking the articulated motion of two strongly interacting hands,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1862–1869, 2012.

[11] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, “Realtime and Robust Hand Tracking from Depth,” *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1106–1113, 2014.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2ª edition. New York, 2004.

[13] R. C. Gonzalez, R. C.; Woods, *Processamento Digital de Imagens*, 3ª edição. São Paulo, 2010.

[14] L. Vacchetti, V. Lepetit, and P. Fua, “Stable Real-Time 3D Tracking Using Online and Offline Information,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp.

1385–1391, 2004.

[15] A. Rezaee Jordehi, “Particle swarm optimisation for dynamic optimisation problems: a review,” *Neural Comput. Appl.*, vol. 25, no. 7–8, pp. 1507–1516, 2014.

[16] A. R. Jordehi and J. Jasni, “Approaches for FACTS optimization problem in power systems,” *Power Eng. Optim. Conf.*, pp. 355–360, 2012.

[17] M. Souza, “Inteligência Computacional para Otimização,” 2011. [Online]. Available: http://www.decom.ufop.br/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf.

[18] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. 1995 IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.

[19] R. Eberhart and J. Kennedy, “A New Optimizer using Particle Swarm Theory,” *Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.

[20] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” *IEEE Swarm Intell. Symp.*, pp. 120–127, 2007.

[21] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial,” *Chemom. Intell. Lab. Syst.*, vol. 149, pp. 153–165, 2015.

[22] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2, pp. 1665–1670, 2001.

[23] J. Deutscher, A. Blake, and I. Reid, “Articulated body motion capture by annealed particle filtering,” *Comput. Vis. Pattern Recognit.*, vol. 2, pp. 126–133, 2000.

[24] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother, “6-DOF Model Based Tracking via Object Coordinate Regression,” *Proc. ACCV*, pp. 384–399, 2014.

[25] P. Azad, D. Munch, T. Asfour, and R. Dillmann, “6-DoF model-based tracking of arbitrarily shaped 3D objects,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 5204–5209.

[26] C. Teulière, E. Marchand, and L. Eck, “Using multiple hypothesis in model-based tracking,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4559–4565, 2010.

[27] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[28] “Fokya,” 2018. [Online]. Available: https://fokya.wordpress.com.

[29] “Metrópoles,” 2018. [Online]. Available: https://www.metropoles.com/postpatrocinado

/20-cenas-do-cinema-que-nao-sao-tao-impressionantes-sem-efeitos-visuais.

[30] Y. Cho, J. Lee, and U. Neumann, “A Multi-ring Color Fiducial System and An Intensity-invariant Detection Method for Scalable Fiducial-Tracking Augmented Reality,” *Int. Work. Augment. Real.*, pp. 147–165, 1998.

[31] W. A. Hoff, K. Nguyen, and T. Lyon, “Computer vision-based registration techniques for augmented reality,” *Proc. Intell. Robot. Comput. Vis. XV*, vol. 2904, pp. 538–548, 1996.

[32] “Lectures,” 2015. [Online]. Available: http://inside.mines.edu/~whoff/courses/EENG510/lectures.

[33] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan, “Real-time Vision-Based Camera Tracking for Augmented Reality Applications,” *Proc. ACM Symp. Virtual Real. Softw. Technol.*, pp. 87–94, 1997.

[34] J. Rekimoto, “Matrix: a realtime object identification and registration method for augmented reality,” *Asia Pacific Comput. Hum. Interact.*, pp. 63–68, 1998.

[35] H. Kato and M. Billinghurst, “Marker tracking and HMD calibration for a video-based augmented reality conferencing system,” *Augment. Reality, 1999. (IWAR ’99) Proceedings. 2nd IEEE ACM Int. Work.*, pp. 85–94, 1999.

[36] V. Teichrieb, J. Lima, E. Apolinário, M. Bueno, J. Kelner, and I. H. F. Santos, “A Survey of Online Monocular Markerless Augmented Reality,” *Int. J. Model. Simul. Pet. Ind.*, vol. 1, no. 1, pp. 1–7, 2007.

[37] H. Liang, J. Yuan, D. Thalmann, and N. Magnenat-thalmann, “AR in Hand: Egocentric Palm Pose Tracking and Gesture Recognition for Augmented Reality Applications,” *Proc. 23rd ACM Int. Conf. Multimed.*, pp. 743–744, 2015.

[38] X. Zabulis, M. Lourakis, and P. Koutlemanis, “3D Object Pose Refinement in Range Images,” *Lect. Notes Comput. Sci.*, vol. 9163, pp. 263–274, 2015.

[39] P. Padeleris, X. Zabulis, and A. A. Argyros, “Head pose estimation on depth data based on Particle Swarm Optimization,” *2012 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 42–49, 2012.

[40] V. John, E. Trucco, and S. Ivekovic, “Markerless human articulated tracking using hierarchical particle swarm optimisation,” *Image Vis. Comput.*, vol. 28, no. 11, pp. 1530–1547, 2010.

[41] S. Saini, N. Zakaria, D. R. A. Rambli, and S. Sulaiman, “Markerless Human Motion Tracking Using Hierarchical Multi-Swarm Cooperative Particle Swarm Optimization,” *PLoS One*, pp. 1–22, 2015.

[42] J. Lima, “Object Detection and Pose Estimation from Rectification of Natural Features Using Consumer RGB-D Sensors,” Tese de Doutorado, Universidade Federal de Pernambuco, Recife, 2014.

[43] R. W. Brockett, “Robotic manipulators and the product of exponentials formula,” *Math. Theory Networks Syst.*, vol. 58, pp. 120–129, 1984.

[44] J. Berkmann and T. Caelli, “Computation of Surface Geometry and Segmentation Using Covariance Techniques,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 11, pp. 1114–1116, 1994.

[45] J. D. Owens *et al.*, “A Survey of General-Purpose Computation on Graphics Hardware,” *Comput. Graph. Forum*, pp. 80–113, 2007.

[46] Y. S. G. Nashed, R. Ugolotti, P. Mesejo, and S. Cagnoni, “libCudaOptimize : an Open Source Library of GPU-based Metaheuristics,” *Proc. 14th Annu. Conf. companion Genet. Evol. Comput.*, pp. 117–124, 2012.

[47] S. C. Akkaladevi, M. Ankerl, G. Fritz, and A. Pichler, “Real-time tracking of rigid objects using depth data,” *Comput. Vis. Robot.*, 2016.

[48] D. J. Tan, F. Tombari, S. Ilic, and N. Navab, “A versatile learning-based 3d temporal tracker: Scalable, robust, online,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 11–18–Dece, no. 2, pp. 693–701, 2016.

APÊNDICE A

Neste apêndice são apresentados os gráficos (em tamanho maior) referentes aos resultados do rastreamento baseado em PSO usando 128 partículas e critério de parada de 100 (ou 12.800 avaliações) usando os casos de teste sintéticos da base de dados *RGB-D Object Pose Tracking Dataset* [6]. Neles os valores das componentes das poses encontradas pelo método de rastreamento sugerido ao longo da sequência de quadros aparecem em vermelho e seus respectivos valores de *ground truth* em azul. Para proporcionar uma melhor visualização, os gráficos em azul foram desenhados com uma espessura ligeiramente maior.

Figura A.1 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Orange Juice*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).

Figura A.2 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Tide*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).

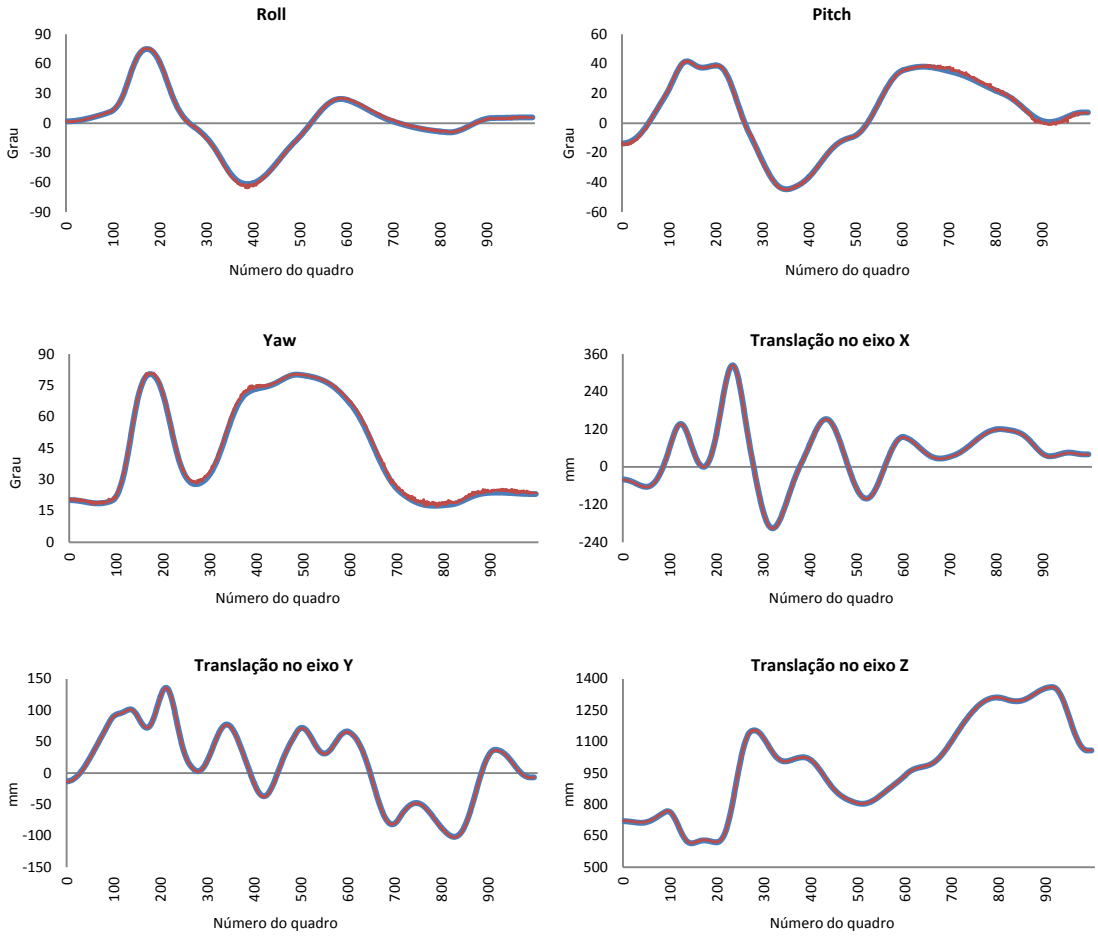


Figura A.3 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Milk*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).

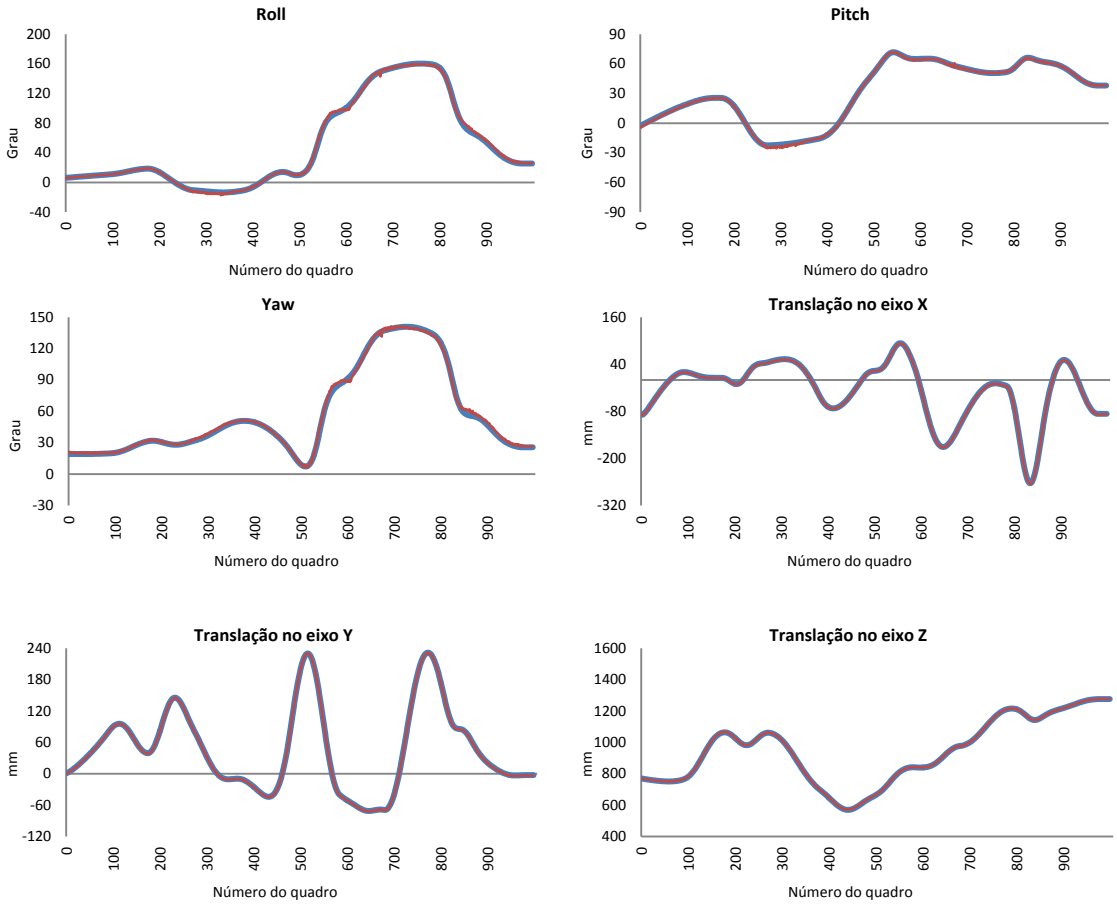


Figura A.4 – Gráficos com as componentes das poses encontradas durante o rastreamento do "*Kinect Box*" usando o PSO (em vermelho) e seus respectivos valores de *ground truth* (em azul).